DOCTOR OF SCIENCES

Interpretability and Explainability in Machine Learning and their Application to Nonlinear Dimensionality Reduction

Bibal, Adrien

Award date:
2020

Awarding institution:
University of Namur

Interpretability and Explainability in Machine Learning

with Application to Nonlinear Dimensionality Reduction

Adrien Bibal

Jury

Prof. Anthony Cleve
University of Namur, Belgium

Prof. Bruno Dumas
University of Namur, Belgium

Prof. Benoît Frénay
University of Namur, Belgium

Dr. Luis Galárraga
INRIA/IRISA Rennes, France

Prof. John A. Lee
Université catholique de Louvain, Belgium

Prof. Wim Vanhoof
University of Namur, Belgium

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the subject of Computer Science

Supervised by Prof. Benoît Frénay

University of Namur

PReCISE Research Center

ABSTRACT

Machine learning (ML) techniques are more and more frequently used today because of their high performance in many contexts. However, the rise in performance comes at the cost of a lack of control over the model that is learned. Indeed, while modelling was mainly done by experts in the past, the surge of data makes it possible to automatically derive models. Unfortunately, this automatization can result in the production of non-understandable models. This concept of model understandability is referred to as interpretability in the literature. Furthermore, when models are not interpretable, it is their ability to be explained (their explainability) that is exploited.

This thesis explores interpretability and explainability in ML. Several aspects of these concepts are studied. First, the problem of defining interpretability and explainability, as well as the vocabulary used in the literature, is presented. Second, the requirements of the law for these concept are studied. Then, the way interpretability and explainability involve users in their evaluation is discussed and guidelines from the human-computer interaction community are presented.

This thesis also applies the concepts of interpretability and explainability to the problem of nonlinear dimensionality reduction (NLDR). While the subjects of interpretability and explainability in NLDR have barely been touched in the literature, this thesis provides a conceptualization of interpretability and explainability in the context of NLDR, as well as new techniques to deal with them. In particular, two questions are central in this thesis "how can interpretability can be measured in NLDR?" and "how can non-interpretable NLDR mappings be explained?".

For measuring interpretability in NLDR, we analyze how existing metrics from different communities can be combined to predict user understanding of NLDR embeddings. In particular, ML quality metrics are used to assess how low-dimensional (LD) embeddings are faithful to the high-dimensional (HD) data, and information visualization quality metrics are used to assess how understandable visualizations are. In the context of NLDR mappings that are considered to be non-interpretable, IXVC was developed to explain the mapping between visual clusters in a NLDR embedding and HD data through an interactive pipeline. Another approach for explaining NLDR mappings through the embedding dimensions was developed in our two techniques BIR and BIOT. Even though previous work has tried to develop more explicit, parametric, mappings, to the best of our knowledge, our works in this thesis are the first to elaborate on the term "interpretability" in the field of NLDR.

Keywords: machine learning, interpretability, explainability, nonlinear dimensionality reduction

RÉSUMÉ

Les techniques de *machine learning* (ML) sont de plus en plus fréquemment utilisées aujourd'hui grâce à leur haute performance dans beaucoup de situations. Cependant, cette montée en performance a pour résultat une perte de contrôle sur le modèle qui est appris. En effet, alors que la modélisation était principalement réalisée par des expert par le passé, l'augmentation de la quantité de données a rendu possible l'automatisation de la production de modèles. Malheureusement, cette automatisation peut produire des modèles qui sont incompréhensibles. Ce concept de compréhension des modèles est appelé interprétabilité dans la littérature. De plus, lorsque les modèles ne sont pas interprétables, c'est leur capacité à être expliqués (explicabilité) qui est travaillée.

Cette thèse explore l'interprétabilité et l'explicabilité dans le ML. Plusieurs aspects relatifs à ces concepts sont étudiés. Premièrement, le problème de la définition de ces concepts et du vocabulaire utilisé dans la littérature est présenté. Deuxièmement, les exigences de la loi par rapport à ces concepts sont étudiées. En troisième lieu, le besoin d'utilisateurs dans l'évaluation de nos concepts d'intérêt est discuté et des lignes de conduite provenant de la communauté de l'interaction homme-machine sont présentées.

Cette thèse applique également les concepts d'interprétabilité et d'explicabilité à la réduction de dimensions non linéaire (RDNL). Alors que l'interprétabilité et l'explicabilité sont deux sujets qui n'ont quasiment pas été abordés dans la littérature en RDNL, cette thèse fournit une conceptualisation de ces sujets, en plus de techniques pour travailler ces sujets. En particulier, deux questions sont centrales dans cette thèse : "de quelle manière l'interprétabilité peut-elle être mesurée dans la RDNL?" et "de quelle manière peut-on expliquer les mappages non interprétable provenant de RDNL?".

Afin de mesurer l'interprétabilité dans la RDNL, nous analysons la manière dont des mesures existantes provenant de différentes communautés peuvent être combinées pour prédire la compréhension des utilisateurs d'*embeddings* provenant de RDNL. En particulier, les mesures de qualité provenant du ML sont utilisées pour mesurer la correspondance de l'embedding de basse dimension par rapport à la haute dimension, et les mesures de qualité provenant de la communauté de visualisation de l'information sont utilisées pour mesurer la compréhensibilité des visualisations. Dans le contexte des mappages de RDNL qui sont non interprétables, IXVC est un pipeline interactif qui a été développé pour expliquer le mappage entre des groupes dans une visualisation et les données qui ont servis à produire la

visualisation. Une autre approche pour expliquer les mappages de RDNL grâce aux dimensions de l'embedding a été développée dans deux de nos techniques: BIR et BIOT. Bien que des travaux précédents ont tentés de développer des mappages plus explicites et paramétriques, à notre connaissance, nos travaux dans cette thèse sont les premiers à élaborer sur le terme d'interprétabilité dans le domaine de la RDNL.

Mots clés : machine learning, interprétabilité, explicabilité, réduction de dimensions non linéaire

ACKNOWLEDGEMENTS/REMERCIEMENTS

En premier lieu, je tiens à remercier le Prof. Benoît Frénay, mon promoteur de thèse. Il est parfois facile de penser que les connaissances qu'on acquiert viennent de notre propre travail, alors qu'elles viennent souvent des autres. Benoît m'a transmis une très grande partie des connaissances que j'ai aujourd'hui en machine learning, comme celles sur la science et le processus de publication. Toujours disposé à expliquer les choses si nécessaire et à sortir son bic rouge pour bien indiquer les modifications à faire dans les articles, il est difficile d'imaginer ce que pourraient être mes connaissances, mes articles ou ma thèse sans lui.

Une autre personne ayant eu un impact difficile à imaginer sur moi et sur ma thèse : mon épouse Becca. En plus du soutient émotionnel non-négligeable, de ses conseils vis-à-vis de mes articles et surtout de cette thèse, ainsi que des cafés servis en fin de thèse, c'est une personne sur qui j'ai pu compter lors de tous les moments difficiles. En plus de tout ça, c'est une grande chercheuse en statistiques, avec qui j'ai pu avoir de magnifiques conversations et avec qui j'ai pu écrire des articles dont on peut être très fiers. Merci pour tout FE !

En parlant de soutient émotionnel, mais aussi de grande amitié, je me dois de mentionner et de remercier Cédric Libert. C'est une belle amitié que j'ai la chance d'avoir, et qui s'est très certainement renforcée lors de notre travail à la fac.

Mon comité d'accompagnement m'a été d'une grande aide et de bon conseil pendant ces années de thèse. Ainsi, je remercie Prof. Anthony Cleve, Prof. Bruno Dumas et Prof. John Lee pour avoir répondu à mes questions et pour leur positivité. Je remercie également Dr. Luis Galárraga et Prof. Tassadit Bouadi, avec qui j'ai eu l'immense chance d'organiser des workshops. C'était super chouette de passer ce temps avec vous !

Je remercie également ma famille, en particulier ma maman, mon frère et ma belle famille, mais aussi mes amis (que je ne citerai pas pour ne pas faire de jaloux !) qui ont toujours été présents et sur qui je sais que je pourrai toujours compter.

Enfin, c'est aux collègues que je dois également un grand remerciement. Si ces dernières années se sont bien passées, c'est aussi grâce à eux. Qu'ils soient doctorants/chercheurs autour d'une table de jeux de société, en train de courir avec moi ou simplement en train de discuter ; ou encore professeurs avec qui les échanges ont toujours été plus que plaisants, ils ont rendu très belle la vie à la fac.

Merci à tous !!!

CONTENTS

PREFACE

Context of the Thesis

Machine learning has become increasingly used in our society. From advertising and spam filters, to its overwhelming use in well-known software and websites, machine learning has made itself indispensable for many companies. Because of this surge in use, law makers have begun to think about the potential danger of such tools and to design laws for taking care of privacy and non-discrimination. Likewise, machine learning is starting to become more and more used by social science researchers, along with statistics, to gain knowledge about their data. Indeed, because of this interest in knowledge extraction from models in research and in some companies, the understandability of such models has been balanced with their accuracy. It is in this context that the notions of *interpretability* and *explainability* emerge.

These notions, which have vaguely existed in the literature for two or three decades [42, 68], were brought to center stage thanks to several workshops dedicated to them. Among these workshops, one can cite the workshop on interpretable machine learning for complex systems (NIPS 2016), the workshop on interpreting, explaining and visualizing deep learning (NIPS 2017), the symposium and workshop on interpretable and Bayesian machine learning (NIPS 2017), the workshop on human interpretability in machine learning (ICML 2016, 2017, 2018 and 2020), the workshop on explainable AI (CVPR 2019) and the workshop on advances in interpretable machine learning and artificial intelligence (EGC 2018, ECML/PKDD 2019 and CIKM 2020), for which I was co-organizer for the 2019 and 2020 editions.

This recent interest can be explained by the rise in use of high-performing machine learning models, for which the good performance is mainly due, nowadays, to deep learning models. However, one of the characteristics of deep learning, despite its high performance in some fields, is its opaqueness. Indeed, it is hard to understand the process that makes it possible for deep learning models to make their accurate decisions.

This thesis is built around 12 articles that I wrote in the field of interpretability and explainability, most of them applied to nonlinear dimensionality reduction. The necessary background for understanding these articles is presented in this thesis, as well as the main contributions. Furthermore, particular attention is paid to the explanation of how the contributions participate in the global subject of interpretability and explainability, with application to nonlinear dimensionality reduction.

List of Contributions

As mentioned in the previous section, 12 articles (and 2 additional extended abstract papers) compose the backbone of this thesis. Throughout this thesis, we refer to our contributions using the citation style [contribX], where X is a number identifying the article. Other articles from the literature are cited with the standard numerical style.

- [contrib1] An introduction to the interpretability of models and their representations, as well as a study of the vocabulary used in the literature, is presented in
  - *Adrien Bibal and Benoît Frénay. Interpretability of machine learning models and representations: an introduction. In **Proceedings of ESANN**, pages 77–82, 2016*

  [contrib2] An extended abstract of this article was also published:
  - *Adrien Bibal and Benoît Frénay. Introduction to interpretability in machine learning. In **BENELEARN**, 2016*
- [contrib3] A study on how the law constrains machine learning models regarding explainability is presented in
  - *Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. **Artificial Intelligence and Law**, 2020*

  In order to perform this task, a link between legal vocabulary and machine learning vocabulary is made. [contrib4] An extended abstract was also presented in
  - *Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Impact of legal requirements on explainability in machine learning. In **ICML Workshop on Law and Machine Learning**, 2020*
- [contrib5] Guidelines from the human-computer interaction literature on how to run user-based experiments for interpretability are given in
  - *Adrien Bibal, Bruno Dumas, and Benoît Frénay. User-based experiment guidelines for measuring interpretability in machine learning. In **EGC Workshop on Advances in Interpretable Machine Learning and Artificial Intelligence**, 2019*
- [contrib6, contrib7, contrib8] A study on how interpretability can be measured in nonlinear dimensionality reduction and how a new measure can be extracted from existing measures is presented in
  - *Adrien Bibal and Benoît Frénay. Learning interpretability for visualizations using adapted cox models through a user experiment. **NIPS Workshop on Interpretable Machine Learning in Complex Systems**, 2016*
  - *Adrien Bibal and Benoît Frénay. Measuring quality and interpretability of dimensionality reduction visualizations. In **SafeML ICLR Workshop**, 2019*
  - *Cristina Morariu, Adrien Bibal, Rene Cutura, Michael Sedlmair, and Benoît Frénay. Combining quality measures for predicting user assessment of dimensionality reduction visualization quality. **To be submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)***

- [contrib9] A study on how constraints can be used to extract interpretable visualizations was completed in
  - *Viet Minh Vu, Adrien Bibal, and Benoît Frénay. Constraint preserving score for automatic hyperparameter tuning of dimensionality reduction methods for visualization.* ***To be submitted to IEEE Transactions on Artificial Intelligence (TAI)***
- [contrib10] A study on the use of decision trees to explain the projection of clusters in t-SNE embeddings was completed in
  - *Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay. An interactive technique for explaining visual clusters in dimensionality reduction visualizations with decision trees.* ***Submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)***
- [contrib11, contrib12, contrib13] Two methods, BIR and BIOT, were developed to tackle the problem of explaining MDS embedding dimensions using external features:
  - *Adrien Bibal, Rebecca Marion, and Benoît Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In* ***Proceedings of ESANN****, pages 537–542, 2018*
  - *Rebecca Marion, Adrien Bibal, and Benoît Frénay. BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables.* ***Neurocomputing****, 342:83–96, 2019*
  - *Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay. BIOT: Explaining multidimensional MDS embeddings using the best interpretable orthogonal transformation.* ***Submitted to Neurocomputing***
- [contrib14] While BIR and BIOT explain embedding dimensions globally, a technique based on LIME was developed to explain t-SNE dimensions locally:
  - *Adrien Bibal, Viet Minh Vu, Géraldin Nanfack, and Benoît Frénay. Explaining t-SNE embeddings locally by adapting LIME. In* ***Proceedings of ESANN****, pages 393–398, 2020*

In addition to the 12 articles and the 2 extended abstracts presented above, two other scientific works were proposed during this thesis:

- A non-peer reviewed article presented at the Center on Regulation in Europe (CERRE) on explainability in machine learning and in the law:
  - *Alexandre de Streel, Adrien Bibal, Benoît Frénay, and Michael Lognoul. Explaining the black box: when law controls AI.* ***CERRE****, 2020*
- A work on how machine learning is used in the formal verification literature was also proposed:
  - *Moussa Amrani, Levi Lúcio, and Adrien Bibal. ML+FV=♡? A survey on the application of machine learning to formal verification.* ***arXiv preprint arxiv:1806.03600****, 2018*

Structure of the Thesis

In order to present our contributions, this thesis is structured in three parts. First, Part I presents interpretability and explainability. In that part, the concepts of interpretability and explainability that are at the very heart of this thesis are first introduced in Chapter 1. Then, the way the law actually constrains interpretability and explainability in machine learning is explained in Chapter 2. Finally, the way users can be included in experiments evaluating interpretability is presented in Chapter 3.

Second, the measure of interpretability in the context of nonlinear dimensionality reduction is discussed in Part II. In order to do that, a background on nonlinear dimensionality reduction is first proposed in Chapter 4. Chapter 5 then answers the questions "what is interpretability in nonlinear dimensionality reduction?" and "how is it measured?". In the last chapter of this part, a way to extract interpretable nonlinear dimensionality reduction results without the burden of defining a measure is presented. Indeed, Chapter 6 introduces user interaction for the selection of NLDR visualizations.

Third, the last part of this thesis focuses on explaining NLDR mappings through their embeddings. After first introducing explainability in NLDR, a technique called IXVC for cluster explanation is proposed in Chapter 7. Then, Chapter 8 presents BIR and BIOT, two techniques that were developed for dimensional explanation of NLDR mappings.

Finally, Part IV closes this thesis with a conclusion in Chapter 9 and ideas for further work in Chapter 10.

Part I

Interpretability and Explainability

CHAPTER 1

INTRODUCTION TO INTERPRETABILITY AND EXPLAINABILITY

This thesis focuses on interpretability and explainability in machine learning. The research on interpretability is becoming more and more important for several reasons. First, in many cases, predictive models are interesting for the description of the data they provide and not only for their accuracy. For instance, psychologists may want a model for learning something in new their field of research, and not solely for its predictive performance. Second, as it is explained in greater depth in Chapter 2 of this thesis, the law may require learning algorithms to produce models that are understandable.

In this chapter, the main notions of this thesis are introduced: interpretability and explainability. First, Section 1.1 provides a general background on machine learning and models. Then, Section 1.2 explains and develops the concept of interpretability, a particular property of models that we presented in *Adrien Bibal and Benoît Frénay. Interpretability of machine learning models and representations: an introduction. In* ***Proceedings of ESANN****, pages 77–82, 2016* [contrib1]. If a model is not interpretable, one may need explanations to understand it. Section 1.3 presents the problem of the explainability of black-box models.

1.1 Machine Learning and Models

Machine learning can be defined as a set of techniques that are used to solve a problem, based on data and an objective function to optimize. The two most common problems to solve are supervised and unsupervised learning problems. In the case of supervised learning problems, a common task is to find a mapping between a target vector $\mathbf{t}$ ($n \times 1$), contained in the data, and the rest of the data, which corresponds to

d features of n instances (making an $n \times d$ matrix $\mathbf{X}$). $\mathbf{t}$ can be, for instance, a list of n cancer statuses (having cancer or not) corresponding to n patients characterized by d medical features (such as their blood pressure, age, etc.).

A model can be seen as a function $\tilde{f}$ that approximates the true mapping f from the input matrix $\mathbf{X}$ ($n \times d$) to the target vector $\mathbf{t}$ ($n \times 1$):

$$f : \mathbf{X} \rightarrow \mathbf{t}.$$

Several families of models can be used to approximate f: linear models, decision trees, neural networks, etc. Choosing a particular family means approximating f in a particular way. For instance, a linear model such as $\tilde{f} = w_0 + w_1\mathbf{x}_1 + w_2\mathbf{x}_2$, with the weights w_0, w_1 and w_2 on the features $\mathbf{x}_1$ and $\mathbf{x}_2$, can be used to approximate f. Most of the time, the best way to approximate f is not known in advance, and the model family is chosen empirically.

In unsupervised learning, no target $\mathbf{t}$ is provided. The goal is then to find patterns in the data without being directed, or supervised. An example of an unsupervised learning problem is clustering, where groups (or clusters) of instances in the data are to be found. In this particular example, the mapping to learn is between the features in the data and the clusters. The fact that the target $\mathbf{t}$ is not provided means that the result of the model must be assessed by the user. In clustering, the goal is to find an appropriate mapping between K clusters and the data:

$$f : \mathbf{X} \rightarrow \mathbf{c},$$

where $\mathbf{c}$ ($n \times 1$) is a binary vector containing the membership of each of the n instances to one of the K clusters. As $\mathbf{c}$ is not provided in advance, as opposed to $\mathbf{t}$ in supervised learning, it must be inspected by users in order to assess if it makes sense. Again, the clustering technique that is used to approximate the mapping f conditions the type of results that can be obtained.

Both in supervised and unsupervised learning, the mapping that is learned between the features and the desired result (a target $\mathbf{t}$ or a set of clusters $\mathbf{c}$) is called a model. The key difference between having a target in advance or not can be observed in the loss function used during the training of the model. In the case of supervised learning, the loss function is clearly defined, as the output of the model must match the target as best as possible. A simple loss function would be the squared distance between the predicted target value $\tilde{t}$ and the true target value t: $||\tilde{t} - t||_2^2$. However, in unsupervised learning, no ground truth in the form of a target is provided. Because of that, a vaguer loss function based on what would be best to avoid is used. In the case of dimensionality reduction, for instance, a classical example is the stress, where dissimilarities between pairwise instances (i, j) in the high-dimensional space (d_{ij}^{HD}) and in the low-dimensional space (d_{ij}^{LD}) must be minimized: $\sqrt{\frac{\sum_{ij}(d_{ij}^{\text{HD}} - d_{ij}^{\text{LD}})^2}{\sum_{ij} d_{ij}^{\text{HD}^2}}}$.

When learned, the model is used to make predictions or to acquire knowledge from data. As the model is the entity at the core of the analysis, it is the main entity required and constrained when social discrimination or other legal and ethical issues are to be detected. The transparency of the model, which can make such detection possible, is called interpretability and is presented in the next section.

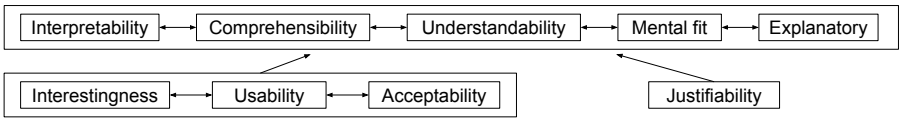


Figure 1.1: Figure reproduced from [contrib1]. Terms that are used as synonyms in the machine learning literature. An arrow from a box A to another box B means that the concept behind the term in A depends on the concept behind the terms in B.

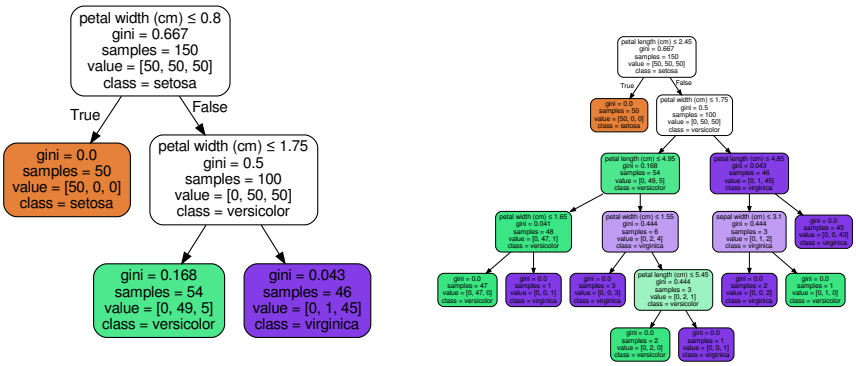
1.2 Interpretability as a Property of Models

Interpretability in machine learning is an interest that goes back to at least the 80s, and was revived with the "Comprehensibility Manifesto" of Kodratoff [42]. The idea is simple: one may need to understand the models that are produced by machine learning techniques, meaning that those very techniques should enforce the comprehensibility of the models they produce.

However, for various reasons, enforcing such a constraint in models is not an easy task. First, the machine learning literature took some time before converging towards a common vocabulary and a definition of interpretability. This convergence was accelerated in 2015 by the MIT thesis of Been Kim [40] and, in the following years, by various workshops organized in well-known ML conferences such as NIPS, ICML and ICLR. Before that moment, *interpretability*, *comprehensibility* and *understandability*, among other words such as *mental fit*, were used to refer to this idea that models should be understandable by users (see Figure 1.1 for an overview of synonyms in the literature) [contrib1]. While the machine learning literature converged on the use of the term "interpretability", a more refined decomposition of the process of "interpretation" could be made based on the use of "interpretability", "understandability" and "comprehensibility", among other terms, in the human-computer interaction, visualization and psychology literature.

Another difficulty related to interpretability is the fact that the precise entity that should be interpreted is not clear either. Indeed, several layers can be mentioned when interpretability is considered. First, the model refers to the mathematical abstraction that represents the mapping to be learned. Most of the time, in the literature, the complexity of the abstraction is quantified in order to measure the interpretability. For instance, a higher number of nodes in a decision tree corresponds to a more complex model, and therefore, a model that is more difficult to understand (see Figure 1.2 for an example).

The second layer that can be considered is the representation of the model. Indeed, the decision boundary of a decision tree can be a really complex mathematical expression that cannot rival the simplicity of a linear model (see Figure 1.3 for an example). The decision tree boundary is more complex in the sense that more parameters are needed in the mathematical expression in order to define it than for a linear classifier. However, what makes the decision tree understandable is that this potentially complex mathematical expression can be represented in the form of a tree, which is easier to process for humans. Figure 1.4(b) is the tree representation



(a) Small, trivial, decision tree.

(b) Larger, more complex, decision tree.

Figure 1.2: Comparison of decision tree complexity built on the Iris dataset [31]. Larger decision trees, with more nodes, seem intuitively less interpretable.

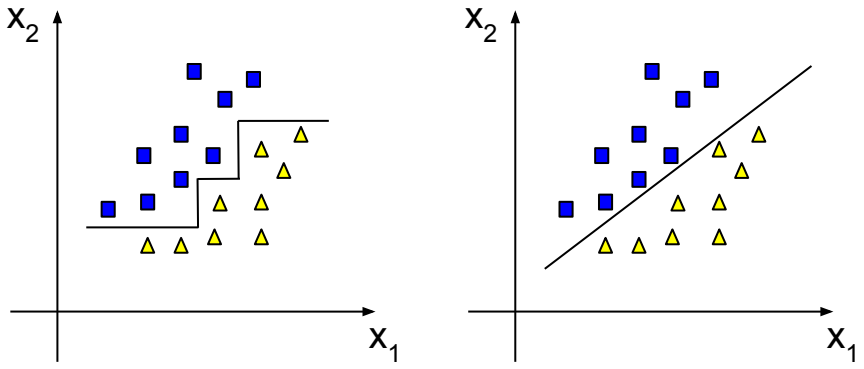
of the decision boundary in Figure 1.4(a). This tree is the representation of the large staircase function

$$\forall x_1, x_2 \in \mathbb{R}, \tilde{f}(x_1, x_2) = \begin{cases} \triangle & \text{if } x_2 < 3 \\ \square & \text{if } x_1 < 5 \text{ and } x_2 > 3 \\ \triangle & \text{if } x_1 > 5 \text{ and } x_2 < 4.8 \\ \square & \text{if } x_1 < 6.3 \text{ and } x_2 > 4.8 \\ \triangle & \text{if } x_1 > 6.3 \text{ and } x_2 < 6.8 \\ \square & \text{if } x_1 > 6.3 \text{ and } x_2 > 6.8 \end{cases}$$

The argument that the representation of the model plays a large role in its interpretation is amplified by the fact that decision trees, for instance, can be represented under many forms, not only under the form of a tree, but also under a set of logical rules or even under a textual representation.

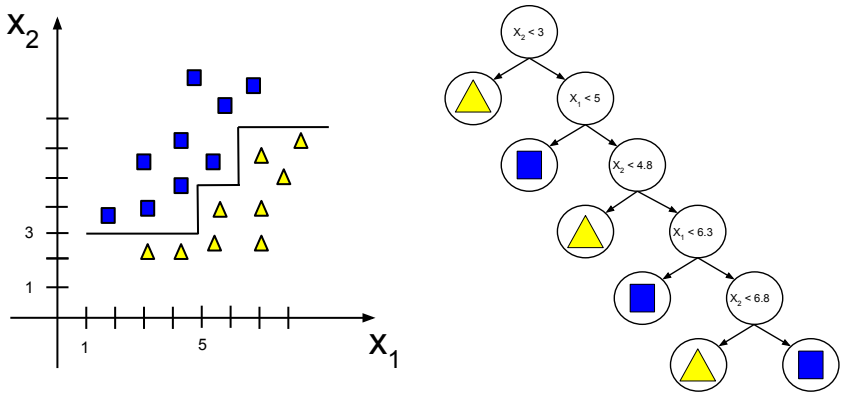
The third and last layer is the visualization of the representation. Indeed, the representation, by itself, is only an abstraction of what will be shown on the screen in practice. The size of the nodes of the tree, the information presented in the nodes, the presence of colors, etc., are all elements that also influence the model's ease of understanding. A more thorough analysis of the concept of interpretability and the difficulty of defining and measuring it can be found in our introduction [contrib1]. One potential avenue for addressing this difficulty is to build and choose the representation and the visualization such that the user's mental model matches the machine learning model to be interpreted (see Figure 1.5 for a pipeline on this idea).

Recently, a survey proposed by Guidotti et al. [36] suggests to view interpretability as a property that can be measured on a continuous scale and that can be influenced by several factors. First, as explained above, the model itself can be more or less easy to understand for humans. Second, in addition to that, the time needed for users to comprehend the model influences its interpretability. For instance, a decision



(a) Decision tree type of decision boundary. (b) Linear classifier type of decision boundary.

Figure 1.3: Comparison of decision boundaries between a decision tree and a linear classifier.



(a) Complex decision tree boundary. (b) Decision tree corresponding to the boundary.

Figure 1.4: Decision tree representation (on the right) corresponding to a complex boundary (on the left).

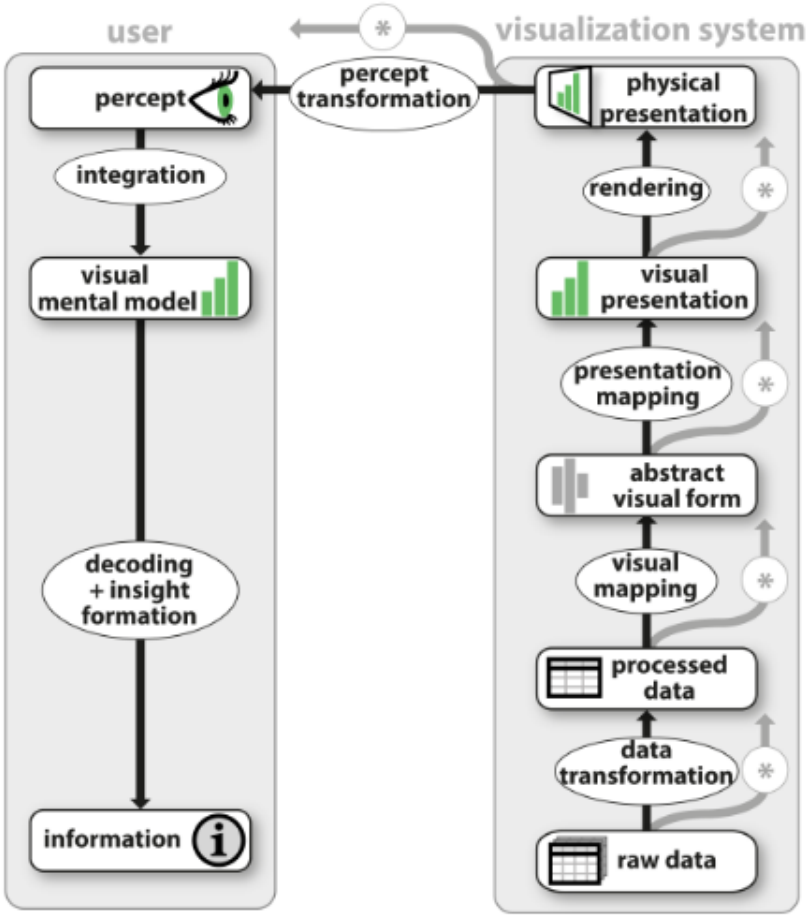


Figure 1.5: Figure from Jansen et al. [38] on the relationship between the visualization that can be created for a machine learning model (processed data) and the visual mental model of the user.

tree with a depth of 10 can still be considered interpretable, as the different paths in the tree can be followed, studied and understood by a user, unless a very limited amount of time is provided to the user. Furthermore, and this is the third factor, the level of expertise of the user also influences his ability to grasp the model. All three of these factors (the complexity of the model, the time to analyze it and the knowledge of the user), when grouped together, compose the interpretability of the model in a particular context. Therefore, we propose to define interpretability as

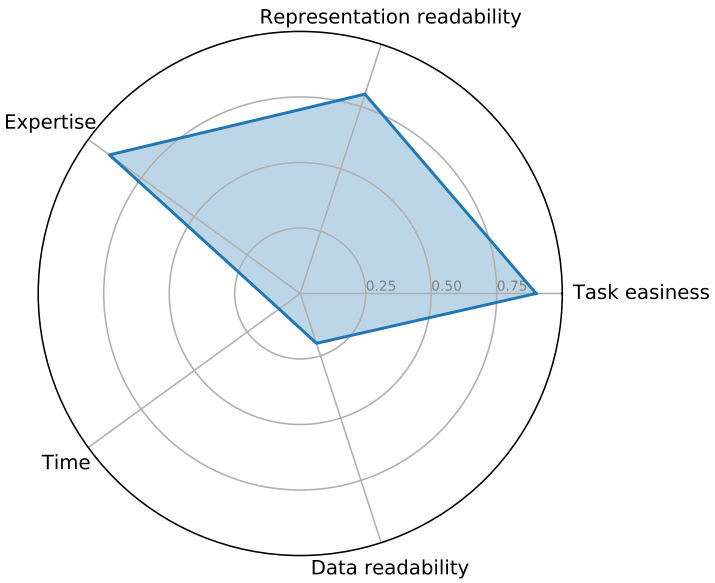


Figure 1.6: Example of a radar chart that can characterize interpretability. This example represents a medical scenario where a medical doctor should examine a decision tree for a classification task with complex data in a short amount of time.

Definition 1. a model can be said to be *interpretable* if, *within a given time limit*, the *level of expertise of the user* allows him to *understand* the model through its *representation*.

Of course, the ability of users to grasp the model does not only depend on the user, but also on (i) the complexity of the model and (ii) of the data. For instance, (i) a very complex neural network and (ii) features with a complex semantic (e.g., medical features or pixels), even in a simple decision tree, make it harder for users to grasp the model. All of the factors influencing interpretability can be summed up in a radar chart, as for example in Figure 1.6.

Contextual factors make it difficult to define the interpretability of a model, as well as to define a measure of interpretability. In most works in the literature, and in this manuscript, when a measure of interpretability is considered, it is through the lens of the first factor: the intrinsic ability of the model to be interpretable. Focusing on the first factor has the benefit of making it possible to analyze the interpretability as an objective property. Indeed, many will agree with the idea that, for instance, decision trees are more interpretable than neural networks. The question is: how can this be measured objectively?

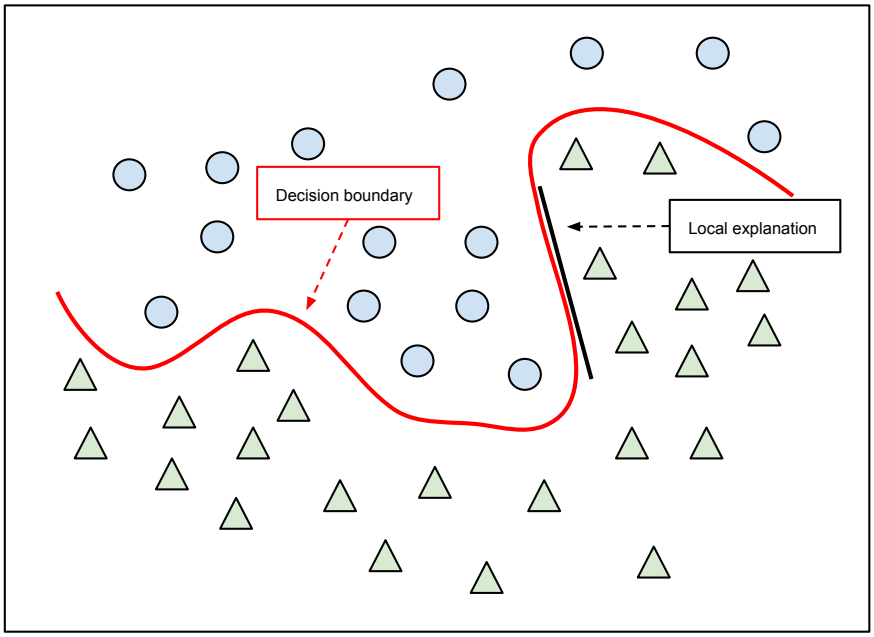


Figure 1.7: Figure reproduced from [contrib3] representing a local explanation of a decision boundary with a linear model.

While some families of models are considered to be interpretable (linear models, tree-based models, etc.), others are considered to be black boxes (neural networks, support vector machines, etc.) [36, 67]. The use of these latter models is often justified by their performance. If one wants to open the black box, instead of directly using interpretable models, some explanatory techniques are needed. In the next section, the problem of opening black boxes, and a popular technique to do so, is introduced.

1.3 Using Explainability to Open the Black Box

If an interpretable model cannot be chosen, e.g., because high performance is required and can only be obtained with a black-box model, then the main way to get insights about the model is through external means. The explainability of a model refers to its capacity to be explained by external tools or techniques (also called post-hoc explanations [52]).

The most well-known technique for explaining black boxes today is the local interpretable model-agnostic explanations (LIME) [66] technique. The particularity of LIME is its algorithm for explaining locally (see Algorithm 1). In order to explain why a certain black-box model (BM) decision is made for an instance $\mathbf{x}_i$, new instances $\mathbf{s}_j$ similar to $\mathbf{x}_i$ are generated (line 4). Then, the BM is queried in order to

learn the decision $\mathbf{z}_j$ that is made for each $\mathbf{s}_j$ (line 6). When the set $\mathbf{z}$ of predictions made by the BM for the sampled instances $\mathbf{S}$ are collected, the input-output couples $(\mathbf{s}_j, \mathbf{z}_j)$ are used to train an interpretable model (IM), e.g., a linear model (line 9). This interpretable model is then returned by the algorithm, and can be used to understand how the BM makes its decisions for instances similar to $\mathbf{x}_i$. Because this explanation is provided without any indication of what is inside the BM, the explanation is model-agnostic. Figure 1.7 shows an example of a local explanation (with a linear model) of a complex decision boundary from a black-box model.

Algorithm 1: LIME algorithm rewritten from [66].

Data: $\mathbf{X}$: a dataset
Data: i: the index of the instance in $\mathbf{X}$ for the local explanation
Data: p: the number of samples to generate for explaining the decision
Data: BM: a black-box model to explain
Result: An interpretable model explaining the prediction of $\mathbf{x}_i$

1 $\mathbf{S} = \emptyset$; `/* S is a matrix of new instances */`
2 $\mathbf{z} = \emptyset$; `/* z is a vector of new predictions */`
3 **for each** j **in** $1...p$ **do**
4 create a new instance $\mathbf{s}_j$ that is neighbor of $\mathbf{x}_i$;
5 $\mathbf{S} = \mathbf{S} \cup \mathbf{s}_j$;
6 z_j = prediction of $\mathbf{s}_j$ by BM;
7 $\mathbf{z} = \mathbf{z} \cup z_j$;
8 IM = train_interpretable_model($\mathbf{S}$, $\mathbf{z}$) ; `/* IM = interpretable model */`
9 **return** IM;

The type of explanation provided by LIME is model agnostic because it explains black boxes without using any of their internal properties. This is done by only considering the black-box model outputs for particular inputs. Therefore, no assumptions are made regarding what is inside the black box. In the case of supervised learning, looking at the input-output pairs involves considering only particular predictions. As particular predictions are considered, only local insights are provided. In other words, the explanation is local because it focuses on particular predictions, instead of the whole model (i.e., all possible predictions), which would otherwise result in a global explanation. Recently, some works explore the possibility of combining local explanations to provide a more global explanation of the model (e.g. [70]).

Other techniques exist to get insights about the behavior of black boxes, such as the feature importance provided by SHAP (a method based on Shapley values) [53], the out-of-bag error in random forests [21] and feature perturbation [30]. The important distinction between explainability and interpretability is that interpretability is a property of a model and its representation, while explainability is the capacity of a model to be explained by external resources such as LIME and SHAP. Note that it could also be interesting for an interpretable model to have some degree of explainability, as it would then be possible to use some tools to increase the understanding of the model. As such, it cannot be said that explainable models are strictly uninterpretable, and vice-versa.

Note also that the burden of improvement regarding explainability is rarely put on the black-box models (i.e. changing how they work such that explaining their behavior is easier), but rather on the techniques used to explain them. This is even more prominent for model-agnostic explanation methods, as they do not even consider the internal behavior of black boxes.

1.4 Conclusion

Interpretable models are used in many research fields and companies, and some prominent scientists urge using them, instead of black-box models, when possible [67]. However, black-box models, like deep learning models, are still used in many applications and websites today because of their high performance. In some cases, these black boxes must be explained, because the company that trains them needs to learn something from them, or it needs to verify that they make sense, or because guarantees about their behavior must be provided to another party. For instance, in some countries, banks cannot deny credits without being able to provide an explanation for this denial to the client [56].

Among all fields that may require them, the concepts of interpretability and explainability are of utmost importance in the legal field. Considering the legal literature when building interpretable models or explanation techniques is also important because of the lack of clear vocabulary or definitions for these machine learning concepts. The next chapter introduces how the law constrains machine learning models w.r.t. their interpretability and explainability, and presents what is already present or is lacking in the ML literature w.r.t. these legal constraints.

CHAPTER 2

INTERPRETABILITY AND EXPLAINABILITY REQUIREMENTS IN THE LAW

One of the main motivations for interpretability and explainability lies in the requirements of the law. Indeed, while interpretability can be needed for trust issues or to gain knowledge from the model, jurists from around the world are starting to design legal requirements for automated decisions. The two sections of this chapter present the requirements that are considered by the law (Section 2.1) and the way the machine learning literature handles these constraints (Section 2.2). Additional attention is paid, in this chapter, to the fact that the vocabulary used in the two fields is not always aligned. In order to clarify the vocabulary and to analyze the exact needs of jurists, legal researchers helped us read legal texts. On our side, we, as machine learning researchers, translated these needs in technical terms and searched the literature for existing techniques that address these needs. This chapter is based on *Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Legal requirements on explainability in machine learning.* ***Artificial Intelligence and Law****, 2020* [contrib3] and *Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Impact of legal requirements on explainability in machine learning. In* ***ICML Workshop on Law and Machine Learning****, 2020* [contrib4].

2.1 Interpretability/Explainability Requirements in the Law

The term "explainability" is used more and more often in legal texts to refer to requirements with different levels of strength. The largest difference of strength is between requirements in the private sector (business-to-client, or B2C) and the public sector (government-to-citizen, or G2C).

Requirements for explainability in B2C are weaker than those in G2C. These weaker requirements can have two facets: either they are vertical (applied to a specific sector) or horizontal (applied across different sectors). The well-known general data protection regulation (GDPR) is a case of horizontal rules, as it applies to all private firms across all sectors. In the GDPR, it is for instance asked that "meaningful information about the logic involved, as well as [...] the envisaged consequences of such processing for the data subject" (art. 13(2f) and 14(2g) of the GDPR) must be provided to the data subject. Other examples of horizontal rules are the consumer protection law that requires providing "the main parameters determining ranking [...] of offers presented to the consumer as result of the search query and the relative importance of those parameters as opposed to other parameters" (new art. 6(a) of Directive 2011/83 on Consumers Rights) and the requirement for online intermediation services and search engines by the European Union, which states that these services must "set out in their terms and conditions the main parameters determining ranking and the reasons for the relative importance of those main parameters as opposed to other parameters" (art. 5 of Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services).

Requirements for explainability also exist specifically for certain sectors (i.e. vertical rules). For instance, for the financial sector, the authorities "may require the investment firm to provide, on a regular or ad-hoc basis, a description of the nature of its algorithmic trading strategies, details of the trading parameters or limits to which the system is subject, the key compliance and risk controls that it has in place [...] and details of the testing of its systems. The competent authority [...] may, at any time, request further information from an investment firm about its algorithmic trading and the systems used for that trading" (art. 17(2) of the Directive 2014/65 on Markets in financial Instruments). Another sectoral example is for insurance in Belgian law, where insurance companies must be able to explain on what basis their tariffs are proposed (art. 46 of the Belgian law of 4 April 2014 on insurances).

Stronger requirements for explainability are required in public, G2C, decisions. This kind of decision can be decomposed into two types that also differ in their strength. First, administrative decisions have the weakest requirements among G2C decisions. For administrative decisions, what is called the "motivation" of the decision is required. This motivation includes the legal basis that is used to make the decision, alongside the facts that have been used in the decision. This means that, for instance, a decision tree would have to output the set of laws that supports each of its decisions, alongside the set of facts that were used for the decisions.

Judicial decisions are subject to stronger explainability requirements than administrative decisions. The requirements for judicial decisions are similar to the ones of administrative decisions, but, in addition, answers to the arguments of the parties must also be provided. This means that in addition to the explanation of how the facts are used to make a decision and to the legal grounds that serve as basis to make this decision, the model must also consider textual arguments of the parties as input, and must output answers to these arguments.

Main features • Directive 2011/83 on Consumer Rights, art. 6(a): obligation to provide "the main parameters" and "the relative importance of those parameters" • Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services, art. 5: obligation to provide "the main parameters" and "the relative importance of those parameters"
All features • Guidelines on Automated individual decision-making and Profiling: obligation to provide "the criteria relied on in reaching the decision" • Belgian law of 4 April 2014 on insurances, art. 46: obligation to provide "the segmentation criteria"
Combination of features Guidelines on Automated individual decision-making and Profiling: obligation to provide "the rationale behind the decision"
Whole model Directive 2014/65 on Markets in Financial Instruments, art. 17: obligation to provide "information [...] about its algorithmic trading and the systems used for that trading"

Table 2.1: Examples of legal texts supporting the four levels of requirements for explainability in B2C. This table is reproduced from [contrib3].

In [contrib3], these different levels of requirements are studied and the current technical solutions in machine learning that can meet these requirements are presented. In addition, a link between the vocabulary of the legal and the machine learning literature is made. In the next section, we present how these legal requirements translate into machine learning terms and solutions.

2.2 Impact of Legal Requirements in Machine Learning

The legal literature on explainability requirements for B2C is the most extensive. However, the vocabulary is not fixed and does not always coincide with the vocabulary from the machine learning literature. Based on an analysis of legal texts with legal experts, four levels of machine learning requirements were extracted from legal texts: the requirement of providing the main features used in a model, providing all features used in a model, providing an idea of how features are combined in a model and providing the whole model. These four levels, as well as examples of legal texts that support them, are reported in Table 2.1.

As opposed to the requirements for G2C, well-known machine learning solutions can already be used for the legal requirements for explainability for B2C. The first level of requirements, e.g., the ones that mention that the "main parameters" must be provided, according to the Directive 2011/83 on Consumer Rights and the Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services, refers to the main features used by a model. In-

terpretable models such as linear models and decision trees already provide such insights. For linear models, these insights are obtained based on the absolute value of the coefficients, and for decision trees, based on the features that are close to the root node. It is also possible to go further by defining weakly and strongly relevant features in linear models [32,39,43]. Solutions also exist to provide the main features used for black-box models, like the out-of-bag error of random forests [21]. Furthermore, perturbing input features to observe the impact on the output is one external solution that can provide the importance of input features in a model for all kinds of models [30].

The second level of B2C requirements concerns all features used in a model. This requirement appears, for instance, in the Guidelines on Automated individual decision-making and Profiling, where "the criteria relied on in reaching the decision" need to be provided. As explained in our work [contrib3], providing all features used in a model is not a challenge, from a machine learning perspective. The challenge is rather to make the model use as few features as possible, such that the number of provided features is reasonable enough to be grasped by a human. For example, the Lasso is a common solution to penalize linear models in order to make them use as few input features as possible [73].

The third level of strength in B2C is the requirement to provide the combination of features used to make a decision. While interpretable models provide such a combination by design, some techniques exist to approximate how features are combined in black-box models. For instance, LIME (presented in Section 1.3) can be used to generate a local approximation of a black-box model, through the use of an interpretable one.

Finally, the requirement to provide the whole model (fourth level) essentially makes interpretable models mandatory. This strongest requirement in B2C can be found, for instance, in the Directive 2014/65 on Markets in Financial Instruments (art. 17), which states about a model that all "information [...] about its algorithmic trading" must be provided.

The G2C requirements, for their part, are stronger than the B2C requirements in the sense that they add new requirements on top of the ones already existing for B2C. These new requirements open the gates to new challenges in machine learning. G2C requirements can be decomposed into two branches: requirements for administrative decisions and requirements for judicial decisions. While the requirements for administrative decisions add the need to provide a legal basis supporting the decisions on top of the explainability of B2C , the requirements for judicial decisions also add the need to address the arguments of the parties.

Some works in the machine learning literature already try to tackle part of these stronger legal explainability requirements. For instance, based on facts extracted from texts and a domain knowledge database, Ashley et al. explain how the facts can be combined to correspond to a legal issue, such as "trade secret misappropriation", that is predicted by a machine learning model [3].

For administrative decisions, not only the way facts are combined should be provided, but also the legal articles that support the decision. One way to deal with this problem in the machine learning literature is to consider it like a multi-

task learning problem. In this kind of problem, each task is solved separately, and then the solutions are combined to compose the final solution. For instance, from a description of the facts, Luo et al. use neural networks to solve different subtasks, such as predicting the criminal charges and the legal articles supporting the prediction [54]. Based on the same idea, Zhong et al. decompose the problem into three tasks: learning (i) the applicable legal articles, (ii) the charges and (iii) the terms of penalty of the legal judgment [85].

Concerning judicial decisions, a reaction to the arguments provided by parties must also be provided. Therefore, the model must provide, in addition to its decision, the facts that have been used, the legal articles that are used as a basis for the decision and how the arguments of the parties are addressed. In practice, some work from the natural language processing (NLP) literature try to tackle this challenge. The solution stems from the NLP literature due to the need to analyze texts and extract information from them. For instance, Ye et al. use sequence-to-sequence (seq2seq) learning to learn from all textual information and the charges in order to generate a court view on the case [84].

2.3 Conclusion

This chapter showed that requirements for explainability and interpretability exist in legal texts. However, most of the time, the legal vocabulary is vague and the strength required in these legal texts is not always evaluated. Based on our work [contrib3], we presented in this chapter a hierarchy of requirements by strength, based on a mapping of the vocabulary from the legal community to the machine learning community. Furthermore, the current state-of-the-art literature was reviewed to assess if the legal requirements pose challenges for machine learning researchers or if the solutions are well-known.

It appears that the stronger the requirements are, the more they require processing textual information. This may indicate that, in order to comply with the legal requirements for explainability, the field of explainability in machine learning should be more in touch with the literature in NLP. Indeed, legal requirements both impose that, at some point, textual input must be processed (e.g., arguments) and textual output generated (e.g., reactions to the arguments).

In [contrib3, contrib4], we conclude by noting that two views on explainability can be extracted. The machine learning, technical, point-of-view considers explainability as related to the mathematical abstraction of the model. A model is therefore explainable if techniques can somehow be used to get insights about the internal model behavior. However, this may not be the legal point-of-view on explainability, for which it is only necessary to provide a rationale on how a particular decision is made. If this is the case, machine learning techniques focusing on satisfying legal requirements for explainability may target the generation of rationale, instead of providing representations and post-hoc explanations of the mathematics behind the models.

One element that was not considered in [contrib3, contrib4], but rather left as a future work, is how having clear explanations can increase the number of actors responsible for the decision making. Indeed, in the context of credit denial, the responsibility of the decision was on the bank and the responsibility of communication was on the agent communicating the reasons for the denial. Now that machine learning models are used and that explanations are automatically generated, more actors come into play in terms of responsibility: the one who designed the algorithm that extracted the model, the one who designed the explanation extraction and the one who analyzes the model and the explanation to provide to the client. Further actors can even be found, such as the bank-expert who designed the specifications for the model and the explanations.

CHAPTER 3

USER-BASED EXPERIMENTS FOR ASSESSING INTERPRETABILITY

Interpretability and explainability are concepts that are, in the end, evaluated by users. Indeed, unlike accuracy, which can be evaluated objectively, users must be involved in the evaluation of interpretability and explainability. Few papers in the machine learning literature evaluate their techniques by means of user-based evaluations. Most of them use heuristics, such as the reduction in model complexity, to show that their techniques are more interpretable [contrib1]. This chapter is based on *Adrien Bibal, Bruno Dumas, and Benoît Frénay. User-based experiment guidelines for measuring interpretability in machine learning. In **EGC Workshop on Advances in Interpretable Machine Learning and Artificial Intelligence**, 2019* [contrib5], which proposes guidelines from the human-computer interaction (HCI) community to evaluate the interpretability and explainability of machine learning models.

In order to present user-based experiments and the guidelines from the HCI community, Section 3.1 first presents examples of user-based experiments for evaluating interpretability in the machine learning literature. Section 3.2 will then present some guidelines from the HCI community for evaluating interpretability.

3.1 Examples of User-Based Experiments for Assessing Interpretability

Even though user-based experiments are not well developed in the machine learning literature, some papers use such kind of experiments for evaluating interpretability [contrib1, contrib5]. This section reviews some of these papers.

Allahyari and Lavesson [1] evaluate the interpretability of classification models (rule-based and tree-based models) by presenting pairs of models to participants. The participants, who are students with knowledge of models and their representations (rules, trees and graphs), are asked to rate how one element of the pair is more understandable than the other.

Huysmans et al. [37] consider rule-based models of different complexity and under different representations, namely decision tables, decision trees and decision rules. The authors gathered students for their experiments because of (i) the homogeneity of the group and because (ii) they can filter students based on their curriculum. The evaluation was based on three criteria: the accuracy of user answers, the time needed to answer and the reported confidence in the answers. The authors defined three tasks for evaluating interpretability. In the first task, participants were asked to classify instances by following a particular model. In the second task, comprehension questions were asked about models. In the last task, questions were asked about the correspondence between a model and its discriminative boundary, as represented in a 2D scatter plot.

Piltaver et al. propose a way to evaluate decision tree interpretability in [64], with a design presented in [63]. The authors propose proxy tasks to approximate interpretability, such as classifying (checking how well users can use the tree), explaining the tree, discovering interesting properties in the tree and comparing different trees. The validation of their survey is based on 18 students and one dataset. The purpose of their study is to discover the properties of decision trees (such as the depth of the tree and the number of leaves) that make them interpretable for users. In [65], the same authors used their tasks with 52 participants (divided into 3 groups of expertise) to answer the question "what is in decisions trees that influences their interpretability for users." They found that the number of leaves, and therefore the number of paths in the tree, played a major role in the interpretability of trees.

Narayanan et al. [60] similarly propose to evaluate the interpretability of decision rules by varying three factors: the complexity of the rule sets, the number of different cognitive chunks in the rules and the number of repeated elements. Six experiments containing 100 participants recruited via Amazon Mechanical Turk (AMT) were run to assess the importance of each of these factors for interpretability. The same authors conducted similar experiments with the same three factors in [45], but the number of participants from AMT was 150 and the interpretability was evaluated through the accuracy of users when using the model, their response time and their level of satisfaction with the explanations.

While some authors, like the ones cited here, evaluate interpretability and explainability through user-based experiments, this kind of evaluation is not widespread in the literature. Indeed, most of the time, heuristics like the complexity of the model are used [contrib1, 52]. Furthermore, no precise guidelines are provided to machine learning researchers for conducting such experiments. In the next section, guidelines from the HCI community are presented.

3.2 Guidelines for User-Based Experiments for Assessing Interpretability

The guidelines presented in our work [contrib5] can be summed up in three important questions that must be answered before any experiment involving users. While these questions represent the backbone of any user-based experiment in fields like HCI, they are not well-known in machine learning. Indeed, these questions must even be reframed for interpretability and explainability researchers so that they can be applied in practice.

The first question is "what do you want to measure?" and, therefore, refers to the task to analyze. The first step is to understand and describe precisely the real task that will be performed. Indeed, users do not interpret models in a vacuum, but for a particular purpose. This purpose guides the way models must be interpretable and makes it possible to set up tasks to evaluate interpretability. When the task is identified, it should be noted whether the real task can be set up in the experiment or if a proxy needs to be used instead (such as the "classify", "explain", "explore" and "compare" tasks of Piltaver et al. [63, 64]) [29].

The second question focuses on users: "who are your users?". Indeed, the profile of model users indicates the kind of interpretability or explanations that needs to be provided. For instance, users with a high level of expertise in machine learning will interpret models more easily than novice users. This is in line with Guidotti et al. [36] (see Section 1.2), who state that user expertise is a dimension of interpretability, in addition to being a model property.

In practice, this means analyzing who are the real users that will perform the real task for which the interpretable model is developed. In the case of banks, for instance, the user of the interpretable model could be the banker (who may be asked to provide explanations himself), a machine learning consultant (who would provide the necessary knowledge based on the model to the banker) or even the person who developed the model. All of these people probably have different knowledge and different levels of expertise, which require an adapted level of interpretability. Nowadays, the need for interpretable and explainable models is rising, as the complexity of machine learning models is on the rise and these models are extensively used by laymen.

The third question is "which type of metric can you use?". This question refers to the fact that, in all cases, a measure must be used to quantify the interpretability during the evaluation. From the examples of the previous section, we saw that the accuracy of users when using the model, the response time to questions and subjective evaluation of interpretability were such metrics.

These guidelines are the first questions that machine learning researchers should ask themselves before setting up a user-based experiment. These three questions stress the fact that the interpretability of models depends on the users and will be biased by the way it is measured. Therefore, asking (i) what is the real task and if it can be reproduced as such in the experiment, (ii) who are the users and (iii) how will interpretability be measured is paramount in user-based experiments.

3.3 Conclusion

This chapter focused on an aspect that is often missing in machine learning evaluation: the users. The first section of this chapter provided examples of user-based evaluations in the context of interpretability and explainability. However, this kind of experiment is not widespread, and no precise guidelines were previously available to researchers that want to undertake user-based experiments. In order to overcome this issue, preliminary guidelines in the form of questions were proposed in [contrib5].

It is worth noting that this thesis contains classical machine learning experiments, as well as user-based experiments. While interpretability and explainability are indeed tied to users, two issues remain that can make classical machine learning experiment relevant. First, some elements regarding interpretability are intuitive. For instance, a tree of 5 nodes will be more interpretable than a tree of 1 billion nodes, and a linear model with 5 coefficients will also be more interpretable than a linear model with 1 billion coefficients. Therefore, finding methods that produce very sparse models (i.e. models with very few coefficients) is still relevant. This also makes it possible to develop generic machine learning methods and algorithms that are not tied to solving a very specific task.

One can also note that, for the moment, the machine learning community and the HCI/information visualization (VIS) community are still well separated. This means that, in order to enter the machine learning community, a more classical machine learning evaluation must be provided, while entering the HCI/VIS community requires limiting the contribution of technical details. The solution would therefore be to create a new community that would integrate both standards, and that would also include other important areas of expertise like psychology.

Part II

Measuring Interpretability in Dimensionality Reduction

CHAPTER 4

BACKGROUND ON NONLINEAR DIMENSIONALITY REDUCTION

Measuring the interpretability of machine learning models is a difficult task that involves users, as discussed in the previous chapter. This thesis focuses on a particular machine learning problem that involves users by definition, and requires them to understand the result. This particular problem is called dimensionality reduction visualization (mentioned as *visualization* in the remainder of this thesis).

The first section of this chapter introduces visualization through nonlinear dimensionality reduction (NLDR). This thesis does not focus on linear dimensionality reduction techniques, like principal component analysis (PCA), as they are often considered interpretable already. Section 4.2 then explains the need for interpretability and explainability in NLDR and proposes a formalization of these concepts for this particular problem. Finally, Section 4.3 concludes this background section by discussing the problem of interpretability and explainability in the context of non-parametric mappings.

4.1 Introduction to NLDR

Dimensionality reduction (DR) is the process of mapping n instances lying in a m-dimensional space to a p-dimensional space, such that $p \ll m$. The goal of such a process is to avoid issues like the curse of dimensionality [6, 48]. The curse of dimensionality refers to the fact that the number of instances needed to estimate a function grows exponentially with the number of dimensions, or features, that define the space in which this function is defined [48]. As the number of instances is often scarce, because it may be difficult to find new patients, for instance, having a large number of features often leads to issues when learning a machine learning

model. In particular, the empty space phenomenon represents the fact that with a number of features that is too large w.r.t. the number of instances, a large portion of the space between the instances is empty [48]. The goal of dimensionality reduction is therefore to embed the instances that are provided in a high-dimensional space into a space of lower dimension. Dimensionality reduction is often based on the hypothesis that the instances lie in a subspace of the high-dimensional space, called a *manifold*. This is why the field is sometimes called *manifold learning*. In the specific case where the number of reduced dimensions $p = 2$, the two reduced dimensions can be plotted and visualized in a scatter plot. The machine learning problem of finding the best two dimensions representing the initial m dimensions is called *visualization*.

One classical example of DR is principal component analysis (PCA). The goal of PCA is to find the p dimensions, also called principal components, that retain the maximum of variance in the data. PCA is a linear dimensionality reduction because the new p dimensions are linear combinations of the original m dimensions. This makes the mapping of PCA parametric, and it is often considered to be interpretable because one can inspect the new dimensions and understand them in terms of a linear combination of the original ones (see Figure 4.1 for an example).

The main drawback of PCA, and linear mappings in general, is that all HD patterns cannot be faithfully projected to a lower dimensional space because of the simplicity of the mapping. In order to address this issue, nonlinear dimensionality reduction (NLDR) techniques were introduced [48]. As their name indicates, these techniques project instances to the LD space through a nonlinear mapping. Two popular and well-known NLDR techniques are explained in this section, because of their use in the remaining of this thesis: multidimensional scaling (MDS) [44] and t-distributed stochastic neighborhood embedding (t-SNE) [76].

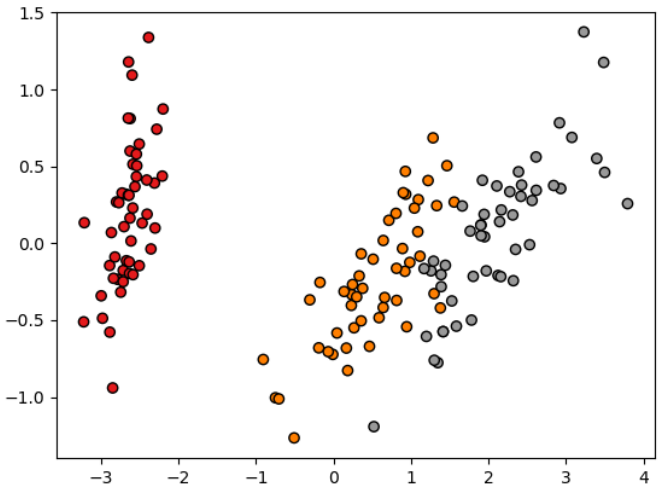
MDS is a technique that attempts to preserve pairwise distances between all instances. In order to do that, its objective function, called the stress, is used to compute the difference between the pairwise distances in HD (often considered, more generally, as dissimilarities) and the corresponding distances in LD. The Kruskal's stress is defined as

$$\text{Stress} = \sqrt{\frac{\sum_{ij} \left(d_{ij}^{\text{HD}} - d_{ij}^{\text{LD}}\right)^2}{\sum_{ij} d_{ij}^{\text{HD}^2}}},$$

where d_{ij}^{HD} (d_{ij}^{LD}) are the pairwise distances/dissimilarities (res. distances) between the instances $\mathbf{x}_i$ and $\mathbf{x}_j$ in HD (resp. LD) [44]. MDS is widely used in fields like psychology. For an example of MDS visualization, see Figure 4.2a.

More recently, t-SNE [76] and UMAP [57] were designed to accentuate the preservation of HD neighborhoods, instead of all pairwise distances. For instance, t-SNE's objective function considers the pairwise distances in HD and LD as probabilities. In HD, the probability $p_{j|i}$ that $\mathbf{x}_j$ is a neighbor of $\mathbf{x}_i$ in HD is defined as

$$p_{j|i} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||\mathbf{x}_k - \mathbf{x}_i||^2/2\sigma_i^2)}.$$



(a) Two first components of a PCA applied to the Iris dataset [31]. The colors represent the different types of flowers.

	sepal length	sepal width	petal length	petal width
x axis	0.36	-0.08	0.85	0.36
y axis	0.67	0.73	-0.17	-0.08

(b) Contribution of each feature in the Iris dataset [31] to the x and y axes of the PCA plot.

Figure 4.1: Scatter plot of the first two components of a PCA applied to the Iris dataset [31], with a table showing the contribution of each feature to the components.

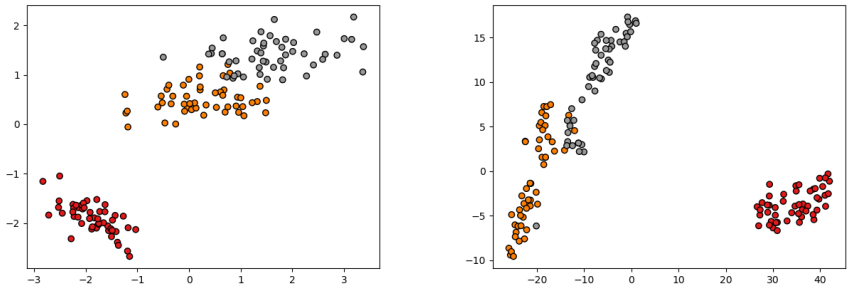
where σ_i roughly represents the size of the neighborhood to consider around $\mathbf{x}_i$ in HD and is found thanks to the *perplexity*, a hyper-parameter of t-SNE that must be provided before running the algorithm. Because $p_{j|i}$ and $p_{i|j}$ can be different, which contradicts intuition, van der Maaten et al. introduce the probability p_{ij} that $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors to each other:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n},$$

where n is the number of instances.

While all p_{ij} are calculated using a Gaussian distribution, focus is given to small neighborhoods in LD through the use of a Student t-distribution (hence the t in t-SNE). Indeed, the probability q_{ij} that $\mathbf{y}_i$ and $\mathbf{y}_j$, the projections of $\mathbf{x}_i$ and $\mathbf{x}_j$ in LD, are neighbors is defined as

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j)||^2)^{-1}}{\sum_{k \neq l}(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}.$$



(a) Result of MDS run on the Iris dataset.

(b) Result of t-SNE run on the Iris dataset.

Figure 4.2: Resulting embeddings of two NLDR techniques applied to the Iris dataset. The colors represent the different types of flowers.

The goal of t-SNE is to find the distribution Q (containing the q_{ij}) in LD that best matches the distribution P (containing the p_{ij}) in HD, computed based on a certain size of neighborhood provided by the perplexity. In order to compare P and Q, the Kullback-Leibler (KL) divergence

$$D_{\mathrm{KL}}(P||Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

is used, where P is considered to be the base distribution on which the comparison is done. The KL divergence makes it possible to measure how different two distributions are, giving a value of 0 if they are equal ($D_{\mathrm{KL}}(P||Q) = \sum_{i \neq j} p_{ij} \log(1) = 0$ if $p_{ij} = q_{ij} \forall i, j$) and an arbitrarily large magnitude otherwise. For an example of t-SNE visualization, see Figure 4.2b.

When the HD instances are projected into an LD space with methods like PCA, MDS or t-SNE, this LD space can be visually analyzed to get insights about the HD data. Hopefully, the patterns that can be observed in the LD space are similar to those present in the HD space. However, it is not always clear how HD instances are really mapped to the LD space. As discussed earlier, PCA can provide an interpretable mapping, but this is not always the case for other methods, such as MDS and t-SNE. While interpreting the mapping of these techniques is difficult, or even impossible, it may be important to understand them nevertheless. In order to better understand why, the next section explains the importance of the interpretability and explainability of NLDR mappings.

4.2 The Need for Interpretability/Explainability in NLDR

Interpretability and explainability are not developed for NLDR methods in the literature. Indeed, influential authors and papers, such as “The Mythos of Model Interpretability” by Lipton [52], mostly focus on supervised learning.

In the mean time, PCA is often used because of its interpretability. Indeed, as mentioned earlier, one of the main advantages of PCA is that it provides new dimensions (i.e. the principal components) that are understandable in terms of the original features. However, the lack of interpretability of NLDR visualizations leads to the need for explanations by experts (e.g., [46]). However, this solution requires human resources and is prone to error. Indeed, one key error that experts can make when explaining a DR visualization is to inject knowledge that was not used by the method that produced the visualization.

In addition to the flaws of such subjective explanations, for some fields, objective explanations are their main focus. For instance, in psychology, two experiments for collecting different data can be run to analyze if the dataset from the first experiment can explain a visualization built from data generated in the second. In [41], a first experiment asks participants to compare how similar social groups (e.g., scientists and old people) are, and, in a second experiment, participants are asked to score the same social groups w.r.t. stereotypes such as intelligent, conservative, wealthy, etc. The authors then look for an "implicit mapping", which means, in this particular case, explaining how participants compare social groups w.r.t. stereotypes, without asking the participants explicitly. This is done by using the dataset of rated social groups w.r.t. stereotypes to predict how stereotypes were implicitly used, in the first experiment, to compare the social groups.

More generally, in a simpler setup, NLDR visualizations are often used to explore the data and gain some insights. Therefore, providing explanations about the mapping is an important addition to the analysis.

For many reasons, interpretability and explanations can be required for NLDR mappings. However, the most powerful techniques such as t-SNE and UMAP, and other popular ones such as MDS, do not provide any clues about how the new dimensions were produced. This issue is amplified by the fact that some mappings are not even provided through parameters (e.g., coefficients). This critical issue is developed in the next section.

4.3 The Problem of Non-parametric Mappings in NLDR

In most supervised learning settings that are studied in the field of interpretability and explainability, a model is under scrutiny. The model is often seen as a set of parameters that are optimized by the learning algorithm. For instance, weights in linear regressions are the parameters of the model to optimize. The model is then interpreted through the value of those parameters.

While parametric mappings have the advantage of being analyzable, they have the drawback of strongly restraining the form that the mappings can take [19]. For instance, a linear mapping of the form $w_0 + w_1\mathbf{x}_1 + w_2\mathbf{x}_2 + w_3\mathbf{x}_3$ has w_0, w_1, w_2 and w_3 as parameters, which can be analyzed as the importance given to the features $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. However, only the mappings that can take this linear form can be represented, which is a strong limitation in most NLDR problems. In order to lift this limitation, the position of each of the instances in the LD space can be learned directly (see t-SNE explained in Section 4.1 for an example). As these methods

are not based on parameters that define the form of the mapping, they are called *non-parametric*.

Another way to see this is that the instances themselves are the parameters of the mapping, as they are the elements used to define it. In this case, the problem is that the number of "parameters" is very large and makes the mapping very hard to interpret. Indeed, interpreting the mapping means, in this context, understanding how each instance is projected in LD, which is impossible in most cases.

Given this issue, the only elements that are left to explain the mapping are the original dataset and the embedding produced (i.e., in practice, the 2D scatter plot). Therefore, explanation methods have to, somehow, approximate the link between HD and LD by the means of these two elements only. Our work on measures of how users can subjectively grasp the non-parametric mapping is studied in the next chapter (Chapter 5), while our work on explaining the mapping is the subject of Part III.

4.4 Conclusion

This chapter introduced NLDR, which is at the basis of the remainder of this thesis. First, Section 4.1 explained what dimensionality reduction (DR), and more specifically non-linear dimensionality reduction (NLDR), is. Section 4.2 then presented why interpretability and explainability are important in NLDR. Finally, Section 4.3 explained the problem of explaining DR mappings when no parameters are provided.

This last issue is the core difficulty when explaining NLDR mappings. Part III of this thesis focuses on methods to deal with this issue, by using the embedding and the original dataset to approximate the non-parametric mapping.

CHAPTER 5

MEASURING "ACCURACY" AND "INTERPRETABILITY" IN NLDR

One major challenge for interpretability in machine learning is to find a way to measure it. Solving this challenge would make it possible to automatically tune models w.r.t. their interpretability, and balance this quality with accuracy-like scores.

Despite recent works, interpretability is still a fuzzy concept in supervised learning, and is not even defined for dimensionality reduction. While PCA is often considered interpretable, no definition really exists to state why t-SNE is not. This chapter aims to unite the machine learning community and the information visualization (VIS) community to find a measure that would assess the interpretability of NLDR visualizations. The final goal of such a measure is to make it possible to highlight visualizations that accurately represent the high-dimensional space, while having a mapping that is interpretable for users. This can be a basis for choosing the NLDR technique that is the most adapted to a particular dataset, or even for choosing the right hyperparameter values for techniques such as t-SNE.

The first section of this chapter presents measures related to NLDR techniques to evaluate how a particular NLDR embedding accurately represents a HD space. In the second section, we develop, for this thesis, the concept of *interpretability*, in opposition to the notion of *accuracy*, for NLDR methods. The last section presents our contributions on a way to measure this notion of interpretability. This chapter is based on *Adrien Bibal and Benoît Frénay. Learning interpretability for visualizations using adapted cox models through a user experiment.* ***NIPS Workshop on Interpretable Machine Learning in Complex Systems****, 2016* [contrib6], *Adrien Bibal and Benoît Frénay. Measuring quality and interpretability of dimensionality reduction visualizations. In* ***SafeML ICLR Workshop****, 2019* [contrib7] and *Cristina Morariu, Adrien Bibal, Rene Cutura, Michael Sedlmair, and Benoît Frénay. Com-*

bining quality measures for predicting user assessment of dimensionality reduction visualization quality. ***To be submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)*** [contrib8].

5.1 Measuring "Accuracy" in NLDR

The machine learning literature is mainly focused on measures of the "accuracy", or the faithfulness, of NLDR techniques. By accuracy, we mean NLDR results that accurately represent the high-dimensional patterns or manifolds. These metrics are either designed for the training process as objective functions or for the sole purpose of measuring the quality of embeddings.

Among the metrics of the first kind, MDS's stress and t-SNE's objective function, which were presented in Chapter 4, are good examples. Indeed, through the stress, the quality of an embedding is defined as the conservation of all pairwise distances, while t-SNE and related methods put a focus on neighborhood preservation. While these objective functions are valid quality measures, they are not the most well adapted ones for measuring information preservation outside of the learning process. Indeed, these measures are designed to be optimized, which adds constraints to their definitions (e.g., being differentiable) [49]. Furthermore, some of these measures require the tuning of a hyper-parameter (e.g., the perplexity of t-SNE's objective function), and it may be considered biased to use these objection functions to measure the quality of the DR techniques that optimize them. Having said that, we nonetheless use them in [contrib8] when combined with many other quality metrics to evaluate a large variety of DR methods.

Some other metrics, however, have been designed for the sole purpose of measuring NLDR "accuracy." Some of these measures, such as the *local continuity meta-criterion* (LCMC) [25] and the measure of *trustworthiness and continuity* (T&C) [77], focus on local patterns. LCMC is a measure that compares the overlap of neighborhoods of size k between HD and LD:

$$LCMC(k) = \frac{1}{n}\sum_{i=1}^{n} |\nu_i^k \cap \rho_i^k|, \tag{5.1}$$

where n is the number of instances, ν_i^k is the set of k nearest neighbors of $\mathbf{x}_i$ in HD and ρ_i^k is the set of k nearest neighbors of $\mathbf{y}_i$ (the projection of $\mathbf{x}_i$) in LD.

Like LCMC, T&C is a measure of the correspondence between the HD and the LD neighborhoods of size k. For Venna et al., a projection is *trustworthy* if "the k closest neighbors of a point on the display are also neighbors in the original space" [77]. In other words, one can trust a projection if the visual patterns really exist in HD. The trustworthiness $T(k)$ for a particular neighbor size k is defined as

$$T(k) = 1 - \frac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in U_k(i)} (r^{HD}(i,j) - k), \tag{5.2}$$

where n is the number of instances, $U_k(i)$ is the set of the k nearest neighbors of the instance i in LD that are not neighbors of the corresponding instance i in HD, and

$r^{HD}(i,j)$ is the rank of the instance j w.r.t. to its closeness to the instance i in HD. The other part of T&C, the continuity, measures if the neighbors of an instance in HD are also neighbors of its corresponding projection in LD. The continuity $C(k)$ is defined in a similar way to the trustworthiness:

$$C(k) = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^{n} \sum_{j \in V_k(i)} (r^{LD}(i,j) - k), \tag{5.3}$$

where $V_k(i)$ is the set of the k nearest neighbors of the instance i in HD that are not neighbors of the corresponding instance i in LD, and $r^{LD}(i,j)$ is the rank of the instance j w.r.t. to its closeness to the instance i in LD. The trustworthiness and the continuity can then be combined, for instance with a simple mean, to obtain the final measure T&C.

Some other measures conciliate the focus on local patterns with more global patterns. For instance, one can consider all local and global patterns by computing the correlation between the vector of pairwise distances in HD and the vector of pairwise distances in LD [33]. In a more complex fashion, Q_Y [58] is a measure that combines measures focused on the preservation of local patterns (e.g., LCMC and T&C) with a global measure, in order to take all kinds of patterns into account. The global measure used for Q_Y is called Q_{GB} and is defined by

$$Q_{GB} = 1 - \frac{6\sum_{i=1}^{k} d_i^2}{F}, \tag{5.4}$$

where d_i is the difference between a HD and a LD ranking (constructed based on a spanning tree, see [58]) that characterizes HD and LD global structures, and F is a normalization term. Q_{GB} is then combined with any measure focused on local patterns, referred to as Q_{LC} in [58], to provide Q_Y:

$$Q_Y = \alpha Q_{GB} + (1-\alpha) Q_{LC}, \tag{5.5}$$

where $\alpha \in [0,1]$ strikes the balance between the importance given to the global measure and the local measure.

AUC_{log}RNX [47] is another metric that conciliates local with global patterns, but by considering all neighborhood sizes. This measure is defined based on two other measures that depend on a particular neighborhood size k, namely $Q_{NX}(k)$ and $R_{NX}(k)$. First, $Q_{NX}(k)$ is defined as

$$Q_{NX}(k) = \frac{1}{nk} \sum_{i=1}^{n} |\nu_i^k \cap \rho_i^k|, \tag{5.6}$$

where n is the number of instances, ν_i^k is the set of k neighbors of $\mathbf{x}_i$ in HD and ρ_i^k is the set of k neighbors of $\mathbf{y}_i$ (the projection of $\mathbf{x}_i$) in LD. $Q_{NX}(k)$ roughly measures, for a particular neighborhood size k, the size of the common neighborhood of the instance $\mathbf{x}_i$ in HD (ν_i^k) and the neighborhood of its projection $\mathbf{y}_i$ in LD (ρ_i^k). $Q_{NX}(k)$ is then rescaled to define $R_{NX}(k)$:

$$R_{NX}(k) = \frac{(n-1)Q_{NX}(k) - k}{n-1-k}. \tag{5.7}$$

Based on $R_{NX}(k)$, AUC$_{log}$RNX can then be computed. In order to do so, the area under the $R_{NX}(k)$ curve is taken in the log-scale of k:

$$\text{AUC}_{log}\text{RNX} = \left(\sum_{k=1}^{n-2} \frac{R_{NX}(k)}{k}\right) / \left(\sum_{k=1}^{n-2} \frac{1}{k}\right). \tag{5.8}$$

AUC$_{log}$RNX is defined to consider the conservation of neighborhoods, for all neighborhood sizes, with a focus on smaller neighborhood sizes (hence the log-scale).

Drawing parallel with supervised learning, these quality measures are called "accuracy measure" because their role is to measure how accurate the projection is. Furthermore, similarly to supervised learning, it may be interesting to reduce the accuracy of the projection if it significantly increases the interpretability. The next section focuses on defining what interpretability is in the context of NLDR.

5.2 The Meaning of "Interpretability" in NLDR

NLDR visualizations are usually produced to explore data and gain some insights about the HD patterns. This means that a visualization of which users cannot make sense is not useful. Making sense of a projection corresponds to understanding how the instances in HD were projected into LD. Therefore, in the context of NLDR, increasing interpretability at the expense of accuracy means distorting projected HD patterns to make them more salient and comprehensible for users. This explains the success of t-SNE, which accentuates the presence of clusters at the expense of the accurate projection of large pairwise distances [80]. Indeed, thanks to this sacrifice, t-SNE is able to expose HD neighborhoods more clearly in LD, which makes the visualization, most of the time, easier to grasp for users.

Definition 2. The *interpretability of dimensionality reduction mappings* corresponds to the possibility for users to understand how the high-dimensional patterns are projected into the low-dimensional space.

Following Kruskal and Myron [44], this understanding can either be provided by a mathematical expression of the mapping (called the dimensional explanation) or by an explanation of the projected patterns (called the cluster explanation).

The exaggeration of visual patterns is an important aspect of interpretability in NLDR, mostly in the case where the mapping is not provided (see Section 4.3 on non-parametric mappings). Indeed, in this case, the only way to provide clues about the mapping is to analyze the projected patterns in the visualization. This is why checking for the presence of patterns that are readable for humans in the visualization (e.g., clusters, outliers, etc.) is important in the quest for measuring NLDR mapping interpretability. The next section presents ideas and results from our work [contrib6, contrib7, contrib8] on measuring the balance of accuracy with interpretability in the context of NLDR.

5.3 Measuring "Interpretability" in NLDR

Measures of the presence of human-readable patterns in visualizations are developed in the information visualization (VIS) community (see, e.g., [7]). This community focuses on presenting information in an intelligible way and centers its evaluations on users. In this context, the VIS community develops quality measures of 2D scatter plots, where quality means that the visualization contains patterns that are understandable for users.

Many of the VIS measures, which are called interpretability measures here, revolve around the detection of clusters in visualizations [4, 69]. One of the best-performing measures [69], in terms of finding what users prefer to see in visualizations, is the distance consistency (DSC) measure [71]:

$$\text{DSC} = \frac{|\mathbf{y}_i \in \mathbf{Y} : \text{CD}(\mathbf{y}_i, centr(c_{clabel(\mathbf{x}_i)})) \neq true|}{n}, \tag{5.9}$$

which computes the number of instances $\mathbf{y}_i$ in the visualization $\mathbf{Y}$ of $\mathbf{X}$ that are closest (calculated with $\text{CD}(\cdot,\cdot)$) to a prototype of another class than their own (written as $c_{clabel(\mathbf{x}_i)}$). The prototype of a class is the virtual instance that is the most representative of that class and is, in practice, often considered to be the mean of the m instances it represents ($\frac{1}{m}\sum_{k=1}^{m} y_k$). Other supervised measures (as they require class labels) of the presence of visual clusters include the average between-within clusters (ABW) [50], the hypothesis margin (HM) [34], the Caliński-Harabasz index (CAL) [23] and the neighborhood hit (NH) [62].

ABW measures the average distances inside clusters, versus the average distances between different clusters [50]. Formally, it is defined as

$$\text{ABW} = \frac{avg_{\mathbf{y}_i \overset{C}{\not\sim} \mathbf{y}_j} dist(\mathbf{y}_i, \mathbf{y}_j)}{avg_{\mathbf{y}_i \overset{C}{\sim} \mathbf{y}_j} dist(\mathbf{y}_i, \mathbf{y}_j)} \quad \forall \mathbf{y}_i, \mathbf{y}_j \in \mathbf{Y}, \tag{5.10}$$

where $\mathbf{Y}$ is the set of projected instances, $\mathbf{y}_i \overset{C}{\not\sim} \mathbf{y}_j$ means that $\mathbf{y}_i$ and $\mathbf{y}_j$ are in different clusters, and $\mathbf{y}_i \overset{C}{\sim} \mathbf{y}_j$ means that $\mathbf{y}_i$ is in the same cluster as $\mathbf{y}_j$. The clusters are defined by labels that were not used during the dimensionality reduction process.

HM also uses the idea of inter- and intra-cluster distances, except that it considers closest instances as references for the clusters [34]. More formally,

$$\text{HM} = \sum_{\mathbf{y}_i \in \mathbf{Y}} \frac{1}{2}(\text{dist}(\mathbf{y}_i, \text{nearmiss}(\mathbf{y}_i)) - \text{dist}(\mathbf{y}_i, \text{nearhit}(\mathbf{y}_i))), \tag{5.11}$$

where $\text{nearmiss}(\mathbf{y}_i) = \mathbf{y}_j$, the closest neighbor of $\mathbf{y}_i$, s.t. $\mathbf{y}_i \overset{C}{\not\sim} \mathbf{y}_j$, and $\text{nearhit}(\mathbf{y}_i) = \mathbf{y}_j$, the closest neighbor of $\mathbf{y}_i$, s.t. $\mathbf{y}_i \overset{C}{\sim} \mathbf{y}_j$.

CAL is also based on the idea of measuring the contrast between the distances between clusters and the distances within clusters. In order to measure this, two intermediary measures are defined: within groups (WG) and between groups (BG). WG is defined as

$$\text{WG} = \frac{1}{2}\sum_{C_k}(n_{C_k}-1)\bar{d}^2_{C_k}, \tag{5.12}$$

where n_{C_k} is the number of instances in the cluster C_k and $\bar{d}^2_{C_k}$ is the squared average distance between instances in C_k. The second intermediary measure, BG, is defined as

$$\text{BG} = \frac{1}{2}((k-1)\bar{d}^2 + (n-k)A_k), \tag{5.13}$$

where $\bar{d}^2$ is the squared average distance between all instances and A_k is defined as

$$A_k = \frac{1}{(n-k)}\sum_{C_k}((n_{C_k}-1)(\bar{d}^2 - \bar{d}^2_{C_k})). \tag{5.14}$$

As Caliński and Harabasz explain in their article, A_k is nothing more than "a weighted mean of the differences between the general and the within-group mean squared distances" [23]. CAL is then defined by combining BG and WG [23]:

$$\text{CAL} = \frac{\text{BG}}{(k-1)} / \frac{\text{WG}}{(n-k)} = (\bar{d}^2 + \frac{(n-k)}{(k-1)}A_k)/(\bar{d}^2 - A_k). \tag{5.15}$$

Finally, NH is a measure that applies the k-nearest neighbors algorithm by checking if the majority of the k-nearest neighbors of each instance $\mathbf{y}_i$ have the same label as $\mathbf{y}_i$. This is done given, as before, labels that were not used during the dimensionality reduction process [62].

One of the interesting things about these metrics is that they basically measure the same thing (i.e. how salient are clusters in the visualization), but by defining the objects they measure differently. Other measures of this kind can be found, such as the 2,002 separability measures found by Aupetit and Sedlmair [4].

Beyond cluster separability and supervised measures (i.e. measures that make use of labels), other measures are defined to capture other types of patterns. For instance, Scagnostics measures use a graph definition of visualizations in order to find particular shapes in the visualization, outliers or zones of high or low density, etc. [81, 82].

5.4 Combining "interpretability" with "accuracy"

The issue with the use of the measures presented in the previous section is that they do not take into account how accurate the mapping is. This means that a visualization that exhibits clear clusters that do not exist at all in HD will still have a very good score. Moreover, we show in [contrib6] that $AUC_{log}RNX$, an "accuracy" metric presented in Section 5.1, can be a good predictor of user preferences between visualizations (see Table 5.1). Furthermore, we show that combining $AUC_{log}RNX$ with VIS measures can further improve the agreement with user preferences. The problem is, therefore, to understand how to combine the notions of interpretability and accuracy of NLDR in a way that would be useful for users.

# classes	ABW	HM	DSC	$AUC_{log}RNX$	Cox_{pref}
63.6% ± 0.1	65.6% ± 0.1	67% ± 0.2	68.5% ± 0.2	71.5% ± 0.1	76.4% ± 0.2

Table 5.1: Table borrowed from [contrib6]. Average percentage of agreement with user preferences and their 95% confidence intervals. ABW, HM and DSC are measures of cluster separation in the visualization. The table confirms research in the literature suggesting that DSC seems superior to the other cluster separability measures. It can also be observed that $AUC_{log}RNX$, an "accuracy" measure, scores better at predicting user preferences than DSC. Finally, Cox_{pref} is a learned linear combination of all measures in the table.

In order to solve this problem, we propose in [contrib6, contrib7, contrib8] to combine "accuracy measures" from the machine learning community with "interpretability measures" from information visualization community. This solution, which was first presented in [contrib6] and further developed in [contrib7], is to combine (e.g., linearly) the two types of measures:

$$\begin{aligned}\text{overall quality measure} = (\alpha_1 * AM_1) + ... + (\alpha_i * AM_i) + ... + (\alpha_m * AM_m) \\ + (\beta_1 * IM_1) + ... + (\beta_j * IM_j) + ... + (\beta_u * IM_u),\end{aligned} \tag{5.16}$$

where $AM_1, ..., AM_i, ..., AM_m$ are m accuracy measures from machine learning and $IM_1, ..., IM_j, ..., IM_u$ are u interpretability measures from VIS.

Different metric combinations were assessed in [contrib8] with a user-based experiment. The experiment consisted of 54 participants knowledgeable in machine learning who were asked to rate, several times, 8 visualizations per dataset for 11 datasets. The possible rating was "dislike" (crossed heart) or a score from 1 to 4 (selection of 1 to 4 hearts). In total, 15 hearts could be distributed among the 8 visualizations, in order to force the participants to choose a ranking among the good visualizations. The experiment was designed in order to be comparable with the one of Lewis et al. [51]. Figure 5.1 shows the interface of the user-based experiment. Furthermore, a background questionnaire was proposed before the tasks, a final questionnaire was proposed after the experiment, and experimental variables were controlled, such as the time taken by each participant in each task.

The data from the experiment was analyzed from different points of view in order to assess the fact that the conclusions do not derive from the way the problem is defined. First, a simple classification was applied to the data. In order to do that, the data was binarized such that a crossed heart given to a visualization was described as -1 and any positive number of hearts as 1. The problem was, therefore, to classify good and bad visualizations, given accuracy and interpretability metrics as features.

After trying several types of models, XGBoost [26] was the model with the best test accuracy on the dataset. This model, based on an ensemble of trees, allowed us to observe that the classification problem was easily solved with two main metrics: skewness $((q_{90} - q_{50})/(q_{90} - q_{10})$, where the q are quantiles on edge lengths of a spanning tree) and sparsity (if instances are scattered in the visualization) from the Scagnostics measures [81, 82]. Note that these two metrics only measure the presence of patterns in the low-dimensional embedding and do not consider the

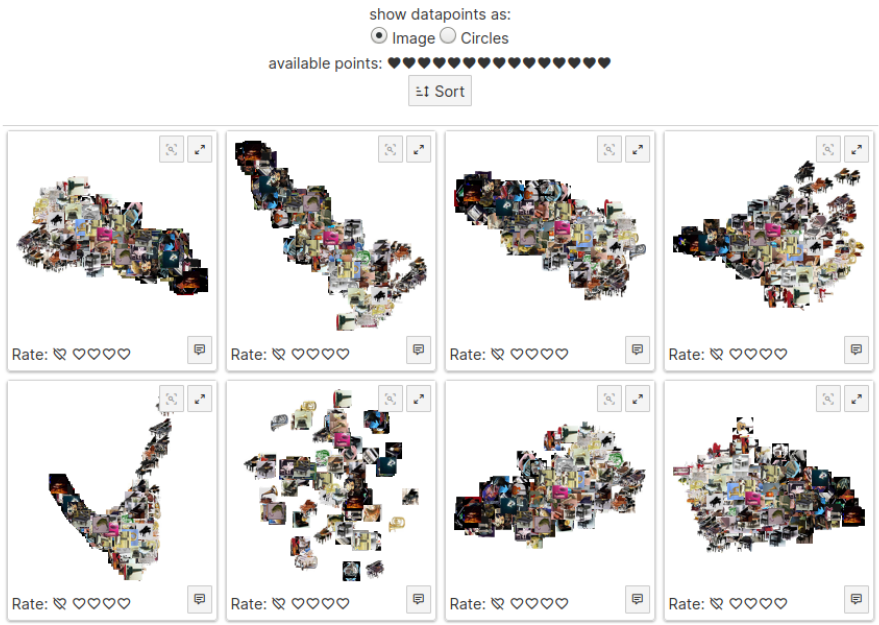


Figure 5.1: Interface of the user-based experiment in [contrib8] for gathering user preferences among visualizations. 15 hearts could be distributed among 8 visualizations. Each visualization could be enlarged, each particular instance could be seen by hovering over it and a version of each visualization with points colored by labels could be selected. Comments could be submitted for each visualization, as well as for the task itself (e.g., comments on the difficulty of the task). The task was defined for 12 datasets, but participants could stop the experiment when they wanted.

accuracy of the DR result. The conclusion of this analysis is therefore that it is easy to separate good and bad visualizations, as participants seem to discard visualizations that are not skewed and/or not sparse.

The second formulation of the problem went further and checked if the participant ranking in each trial could be reconstructed. Indeed, in each trial, participants gave 4s, 3s, 2s, 1s and crossed hearts to visualizations. The hearts were not considered as visualization scores, but rather as way of saying, in each trial, that a visualization with 4 hearts is preferred to a visualization with 3 hearts, for instance. Given this ranking by trial, a new dataset was created with pairwise comparison of visualizations, such that $v_i > v_j$ means that, in a particular trial for a particular participant, the visualization v_i gathered more hearts than the visualization v_j. When all trials of all participants are considered, percentages were associated to all pairs (v_i, v_j) in order to represent the percentage of time v_i was preferred over v_j.

In order to solve the problem "how can metrics be combined to reconstruct participant preferences", well-known preference learning models were used: Bradley-Terry models (BTm) [20]. BTm define the probability of v_i being preferred to v_j as

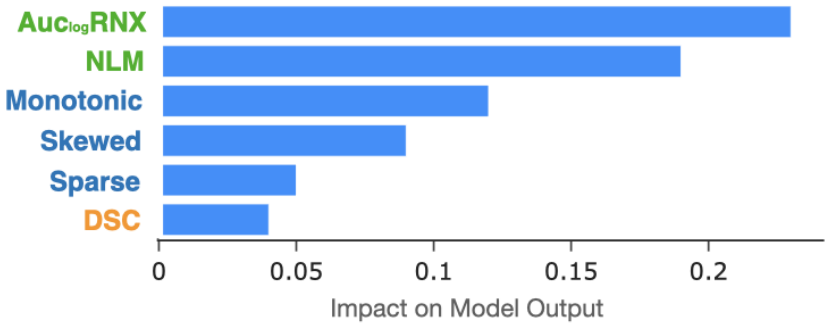


Figure 5.2: Absolute value of weights obtained in [contrib8] when reconstructing user preferences with a sparse BTm. The accuracy of the model is 62.3%, with the 95% confidence interval being [58.39%, 66.22%].

$$\begin{aligned} P(v_i > v_j) &= \frac{e^{w_0+w_1 m_{1,i}+\ldots+w_q m_{q,i}}}{e^{w_0+w_1 m_{1,i}+\ldots+w_q m_{q,i}} + e^{w_0+w_1 m_{1,j}+\ldots+w_q m_{q,j}}} \\ &= \frac{1}{1+e^{(w_0+w_1 m_{1,i}+\ldots+w_q m_{q,i})-(w_0+w_1 m_{1,j}+\ldots+w_q m_{q,j})}}, \end{aligned} \tag{5.17}$$

where $m_{k,i}$ (resp. $m_{k,j}$) is the the k^{th} metric evaluated on v_i (resp. v_j) and $w_0, \ldots, w_d = \mathbf{w}$ is a vector of weights to estimate from the dataset of preferences. Note that, for estimating the weights, the Lasso penalty [73] was used to avoid effects related to the high correlation between certain quality metrics, as well as to induce a sparse and more interpretable vector of weights $\mathbf{w}$. Non-intuitively, weights with a positive (resp. negative) value mean that the associated measure has a negative (resp. positive) impact on the model. Indeed, let us consider a simple model with only one quality measure m_1, where the greater the value of the measure, the better visualization is. In this case, the model to evaluate is

$$P(v_i > v_j) = \frac{1}{1+e^{w_1(m_{1,i}-m_{1,j})}}. \tag{5.18}$$

If $m_{1,i} > m_{1,j}$, which suggests that v_i is better than v_j, then $m_{1,i} - m_{1,j}$ is positive. Now if w_1 is positive and tends towards infinity, then Eq. 5.18 tends towards zero. With the opposite reasoning, if w_1 is negative and tends towards negative infinity, then Eq. 5.18 tends towards one. To sum up, positive (resp. negative) weights tend to make $P(v_i > v_j)$ smaller (resp. larger), meaning that they characterize measures of disliking (resp. liking) visualizations. The absolute value of the resulting weights from our experiment can be found in Figure 5.2.

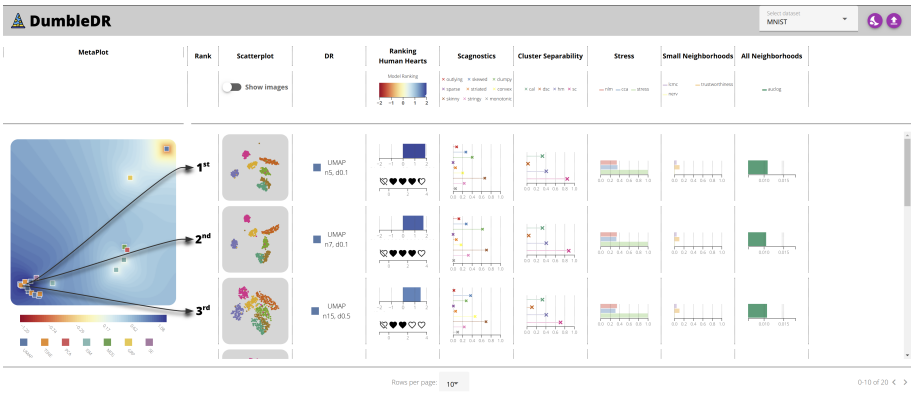


Figure 5.3: Figure from [contrib8] presenting the tool developed for ranking projections. Boosted trees are used to combine quality metrics and rank projections.

The important insight from the analysis is that a variety of metrics needs to be used to reconstruct user preferences. While sparsity and skewness from Scagnostics were enough to throw away bad visualizations, it can be seen in Figure 5.2 that assessing the DR accuracy is also important (AUC_{log}RNX and NLM), as well as cluster separability measures (e.g., DSC). This research shows that machine learning researchers, who largely focus on DR accuracy measures, should consider users' need to see LD patterns in the visualization. Furthermore, as seen in the binary classification problem, the accuracy of the DR is of no interest if, for instance, the visualization is a compact cloud of points (the opposite of sparsity). From the VIS point-of-view, the common viewpoint that separability metrics, and in particular DSC, are the metrics that best represent user perception is questioned, since DR accuracy also counts. More fundamentally, the need for sparsity in the visualization may encompass the need to see well separated clusters.

In a third formulation of the problem is a learning-to-rank problem using a nonlinear model, as opposed to the linear model of the second formulation. This formulation is a bit different than the second one in the sense that a ranking of all visualizations is learned instead of preferences between pairwise visualizations. Boosted trees were used as a nonlinear model for this task. The accuracy of our model is 70%, with a confidence interval of [65.6%, 74.4%].

5.5 Use Case

In order to present our method in [contrib8], a tool has been developed. Figure 5.3 shows a screenshot of the tool. Suppose that a user want to automatically find the best projection method, and the best parametrization if relevant, for his dataset. The user uploads, for example, the pets dataset [61] in the tool. The dataset contains a set of images of cats and dogs. After computing the projections and the quality metrics, the tool provide a ranking, based on the boosted tree presented in the previous section, along a metamap of all computed projections (see Figure 5.4). Thanks to

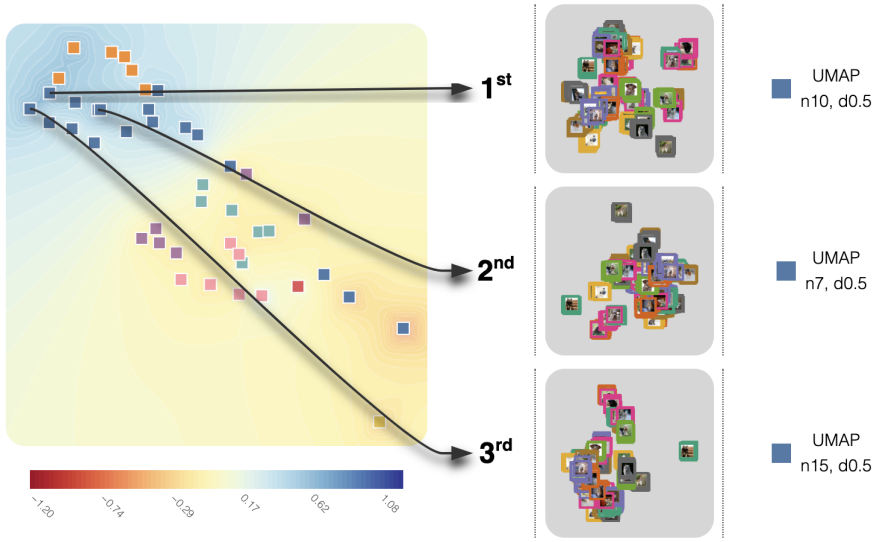


Figure 5.4: Figure from [contrib8] presenting the result of our method on the pets dataset. The ranking shows that UMAP with some particular hyper-parametrizations offers the best visualizations. In the metamap on the left, the heatmap color corresponds to the quality of visualizations and the point colors correspond to projection techniques (e.g., blue for UMAP).

the tool, the user has now a clear view of what technique work best for the dataset (here UMAP), as well as the particular parametrization needed to obtain a good visualization. Please note that this last point is important, as the worst visualizations for this dataset are also generated by UMAP, albeit with a different parametrization.

5.6 Conclusion

This chapter explored different ways to measure quality in DR visualizations. The machine learning community tends to study metrics on the quality of the DR process. In this thesis, we call these metrics "accuracy metrics", in reference to supervised learning. The VIS community rather considers metrics of patterns that are appealing for users, such as well-separated clusters. We call such measures "interpretability metrics", as they assess how readable, or understandable, the visualization is.

After presenting how accuracy and interpretability could be framed in NLDR, metrics from the literature were presented. Our work on combining metrics to (i) identify the important metrics to predict user preferences and (ii) find a way to measure visualizations under different aspects of quality was then presented.

In addition to evidence that accuracy and interpretability should both be considered when designing DR quality metrics, we showed that different aspects should be considered and combined. The developed combination of metrics can be used to select DR techniques and hyper-parameter values, such as the perplexity of t-SNE.

CHAPTER 6

INDUCING INTERPRETABILITY THROUGH USER INTERACTION

The measures discussed in the previous chapter can be used for selecting the visualizations, among a large set, that have the highest potential of interest for users. For instance, among thousands of visualizations generated by t-SNE with different perplexity values, the most interesting visualizations for users can be selected using these measures.

Another solution to select visualizations that are interpretable for users is to ask them what they expect to see in the visualization, in order to select the visualizations that best meet these expectations. The first section in this chapter details the idea of selecting interpretable results by drawing a parallel with supervised learning. The second section proposes a way to solve the problem using constraints. This section is based on *Viet Minh Vu, Adrien Bibal, and Benoît Frénay. Constraint preserving score for automatic hyperparameter tuning of dimensionality reduction methods for visualization.* ***To be submitted to IEEE Transactions on Artificial Intelligence (TAI)*** [contrib9].

6.1 Selecting Interpretable Results

In this chapter, we present the idea that, instead of looking for interpretability explicitly, for instance by measuring it, one can also describe what one would expect to see (and would therefore be understood), in order to find what best aligns with this description.

In supervised learning, the burden of striking a balance between accuracy and interpretability is on the shoulders of the user. Indeed, when a regularization term is added to the objective function of a linear regression model, the right balance

between accuracy and interpretability is not chosen automatically, but by the user. The user has to look at the different solutions, and choose the one that has the best accuracy, given the understanding that he has of the chosen solution.

The main issue with this manual way of selecting interpretable solutions is that it does not scale with the number of possible solutions. Indeed, if the learning algorithm can produce thousands of solutions, the user cannot go through all of them to select the best balance between accuracy and interpretability. As we have seen in the previous chapter, one solution for tackling this problem is to define a measure of interpretability. By doing this, a set of interpretable solutions, given by measure scores, can be extracted, which lightens the burden for the user.

Another way to see the problem is to say that if the model is aligned with existing user knowledge, then the chance that the user understands the model is higher. Let us consider random forests (i.e. a black box model) applied on medical data, and the feature importance that can be extracted from the model (e.g., using its out-of-bag error). If the most important features are aligned with existing medical rules, then the model has a higher chance of being understood by users than one for which the important features relate to no existing knowledge.

In the next section, we present how to define such kind of solution for finding interpretable NLDR visualizations. Our work in [contrib9], which is presented in the next section, is not based on a particular knowledge base to know what can be interpretable. Instead, the position that each user knowledge and interest are unique is taken, leading to an interactive solution.

6.2 Implicit Interpretability through Constraints

In [contrib9], user knowledge is defined through constraints that are used to select the NLDR visualization. Most precisely, similar and dissimilar links are defined for instances that are expected to be close together or far away in the visualization. In order to obtain these constraints, users can either link instances in a given visualization, or labels can be provided for all instances. Through labels, it can be stated that instances with the same label (e.g., shoes) are expected to be close together, while instances of different labels (e.g., shoes and t-shirts) are expected to be farther apart from each other than for instances of their own label. Figure 6.1 shows examples of instances linked as similar or dissimilar by a user.

When linked instances are provided by users, similar links are defined in a way similar to the t-SNE objective function as

$$f_{score}(\mathscr{S}) = \frac{1}{|\mathscr{S}|} \log \prod_{(\mathbf{y}_i,\mathbf{y}_j)\in\mathscr{S}} q_{ij} = \frac{1}{|\mathscr{S}|} \sum_{(\mathbf{y}_i,\mathbf{y}_j)\in\mathscr{S}} \log q_{ij}, \tag{6.1}$$

where $\mathscr{S}$ is the set of pairs $(\mathbf{y}_i,\mathbf{y}_j)$ that are considered similar by the user, and with

$$q_{ij} = \frac{(1+||\mathbf{y}_i-\mathbf{y}_j||^2)^{-1}}{\sum_{k\neq l}(1+||\mathbf{y}_k-\mathbf{y}_l||^2)^{-1}}. \tag{6.2}$$

(a) Examples of similar links. (b) Examples of dissimilar links.

Figure 6.1: Figure from [contrib9] representing examples of instances from the dataset FASHION_1K [83] that are linked as similar (a) and dissimilar (b).

Similarly, the dissimilar links are defined as

$$f_{score}(\mathscr{D}) = -\frac{1}{|\mathscr{D}|}\log \prod_{(\mathbf{y}_i,\mathbf{y}_j)\in\mathscr{D}} q_{ij} = -\frac{1}{|\mathscr{D}|}\sum_{(\mathbf{y}_i,\mathbf{y}_j)\in\mathscr{D}} \log q_{ij}, \tag{6.3}$$

with $\mathscr{D}$ being the set of pairs of instances considered to be dissimilar by the user.

When the scores for the similarity and dissimilarity links are computed, they are combined to compose the final score

$$f_{score}(\mathscr{S},\mathscr{D}) = \frac{1}{2} f_{score}(\mathscr{S}) + \frac{1}{2} f_{score}(\mathscr{D}). \tag{6.4}$$

In our experiments, we found that the number of constraints in $\mathscr{S}$ and in $\mathscr{D}$ should be roughly the same, as they compensate each other in f_{score}.

In addition to the selection of visualizations that match user expectations, we found that f_{score} makes it possible to explore interesting solutions that are not discovered by other quality scores. The top row of Figure 6.2 represents a metamap,

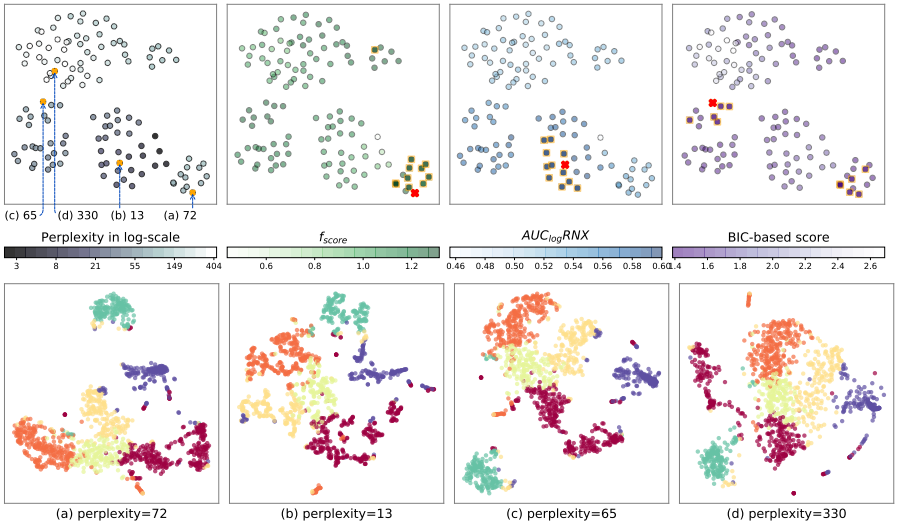


Figure 6.2: Figure from [contrib9] containing metamaps and t-SNE visualizations generated from the NEURON_1K dataset [72]. Visualizations with the highest scores according to different metrics are highlighted in the metamaps of the top row. On the bottom row, the first visualization (a) is chosen using f_{score}, the second (b) is chosen using $AUC_{log}RNX$, the third (c) is chosen using a BIC-based score studied in [contrib9] and the last one (d) is not considered good by any of the four scores.

which is a projection of several visualizations. Two points in the metamap, corresponding to two visualizations, are close to each other if the two visualizations are similar to each other. The second, third and fourth metamap in this row contain points that are highlighted w.r.t. the best scores obtained by, respectively, f_{score}, AUC$_{log}$RNX and a BIC-based score studied in the paper. It can be seen that the visualizations with the highest scores, for the three scores, cover different regions of the metamap. The second row of Figure 6.2 presents examples of visualizations from these regions. These visualization are selected from the regions indicated in the metamaps in the first row.

We also show that these constraints allow users to discover, using f_{score} only, different aspects of the data [contrib9]. Indeed, while the other quality measures from the literature provide a fixed score value for a particular visualization, the value provided by f_{score} depends on what the user expects to see. This means that different regions of the metamap presented in 6.2 can be explored, depending on how instances are linked together. For instance, if known labels are used to automatically build $\mathscr{S}$ and $\mathscr{D}$, some interesting visualizations can be highlighted with certain labels and other visualizations with other labels. This makes $f_{score}(\mathscr{S},\mathscr{D})$ a flexible measure, in the sense that it can highlight different aspects of data, given the knowledge encoded in the constraints. Figure 6.3 shows how changing the constraints can change the visualizations that are provided by f_{score}.

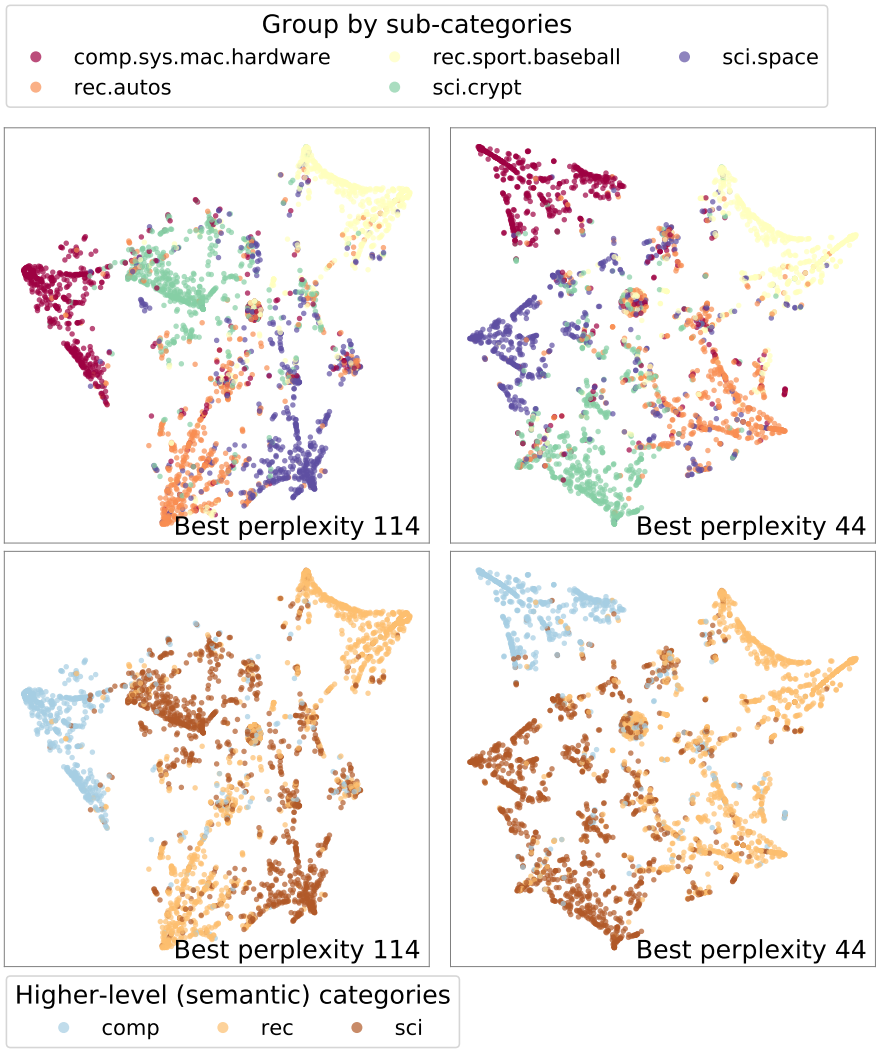


Figure 6.3: Figure from [contrib9] containing different visualizations generated by t-SNE applied to the 20NewsGroups dataset. The first column represents a visualization that has the highest f_{score} with constraints made from 5 categories of text: hardware, baseball, space, automobiles and cryptography. In the first row, colors are assigned to the texts corresponding to these categories. In the first visualization of the second row, the same visualization as before is now colored by other, higher-level, categories: computer topics, recreational topics and science topics. It can be seen, through this second coloring, that another visualization (second one in the second row) better corresponds to what is expected to be seen. For instance, the recreational instances (rec, orange in the visualizations of the second row) were separated in two groups (first visualization of the second row), while being reunited when used in the constraints (second visualization of the second row).

6.3 Conclusion

This chapter presented an implicit way of defining interpretability for NLDR. Indeed, instead of defining what interpretability is explicitly, e.g., through a metric, users can provide clues about what they would interpret more easily. This way of selecting interpretable NLDR visualizations has several advantages.

First, it assumes that what is interpretable changes from one user to another. Indeed, what is expected to be seen and understood depend on user knowledge and expertise. This echoes the definition of interpretability that we formulated in Section 1.2. Second, even for users with the same background, they may want to visualize different aspects of HD patterns, which is made possible by the different ways the instances can be linked.

While the technique proposed in this chapter increases the chances that a more comprehensible visualization is extracted, the main drawback of such a method is that it begs the question: if the user wants to see something, even if it is wrong w.r.t. the HD data, everything will be done to show it. The danger would be that the method proposes a very bad visualization, in terms of DR accuracy, while however meeting the constraints. Echoing the previous chapter, an avenue for future work would be to combine the approach presented in our work [contrib9] with a DR accuracy metric that would penalize the selection of inaccurate visualizations.

Part III

Explaining Nonlinear Dimensionality Reduction Mappings through Embeddings

CHAPTER 7

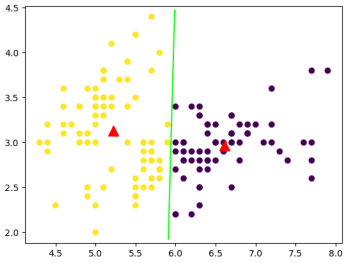
CLUSTER EXPLANATION OF EMBEDDINGS

As with supervised learning, another way to deal with the problem of understanding mappings is to explain black-box mappings, instead of selecting interpretable ones. Kruskal proposes two ways for explaining NLDR mappings: by explaining the embedding's dimensions or by explaining the clusters in the embedding [44]. Explaining the clusters means explaining how instances are grouped together in the visualization. Indeed, the way the instances are grouped defines how the visualization should and will be interpreted. In this chapter, the question of how to explain NLDR mappings through clusters is investigated in more detail. The next chapter will then focus on how it is possible to explain the mapping through the embedding's dimensions.

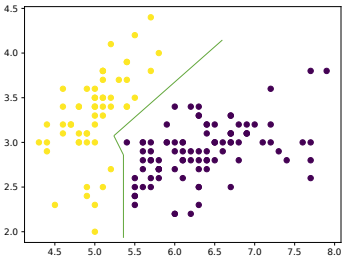
In *Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay. An interactive technique for explaining visual clusters in dimensionality reduction visualizations with decision trees.* ***Submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)*** [contrib10], we opted for using an interpretable model in order to explain the mappings of NLDR techniques such as t-SNE. The goal of the proposed technique is to explain how the patterns are mapped from HD to LD, while not changing anything in the result of t-SNE. Furthermore, we do not want to make any hypothesis in advance about the forms that the clusters may have. Our solution is described in Section 7.1 and the user-based experiment validating the method is presented in Section 7.2.

7.1 Explaining NLDR Clusters with Decisions Trees

Several issues can arise when analyzing clusters in an NLDR visualization. First, a fundamental problem of clustering is the definition of clusters. Indeed, in order to find clusters, the forms that these clusters can take must first be defined. In



(a) k-Means clustering of the Iris dataset. The two red triangles are the prototypes (most representative instances for each cluster) for k-Means. All instances on the left (resp. right) of the green line are closer to the left (resp. right) triangle than the right (resp. left) one.



(b) Bottom-up hierarchical clustering of the Iris dataset. The green line separates the two clusters.

Figure 7.1: Different clustering results from k-Means (left column) and a bottom-up hierarchical clustering (right column) of the Iris dataset [31]. Due to the different objectives of the two techniques, the clusters formed have very different shapes.

all clustering techniques, a hypothesis is made about what clusters can be when defining the objective function. For instance, in k-Means, a prototype for each of the k clusters is created and all instances are assigned to the cluster corresponding to their closest prototype. By doing that, one makes hypotheses about the shape the clusters can have. Another example is bottom-up hierarchical clustering, where two instances (or sub-clusters) are considered to be in the same cluster if they are closer together than to any other instance (or sub-cluster). This means that clusters can be formed differently than when k-Means is run on the same data (see Figure 7.1 for an example).

The second issue related to the explanation of clusters in visualizations is that it is often the role of experts to explain them. This may indeed be an issue because experts can add extra knowledge that is not present in the data, or can provide explanations through combinations of features that are not possible given the type of mapping. These two issues come back to the problem that users cannot fully understand black-box and non-parametric mappings.

In order to explain clusters and solve these two issues, we proposed a solution called Interactive eXplanation of Visual Clusters (IXVC) [contrib10]. The first step of IXVC is its interactive part. Indeed, given the multiple shapes that the clusters can take (first issue presented above), the user must choose, in the visualization (e.g., (1) in Figure 7.2), the clusters for which he wants an explanation by circling each of them (see (2) in Figure 7.2). Thanks to this interactive aspect, no assumption is made about what the clusters are and what form they should take. The idea is that if a user visually identifies a cluster, then it can be considered to be a cluster.

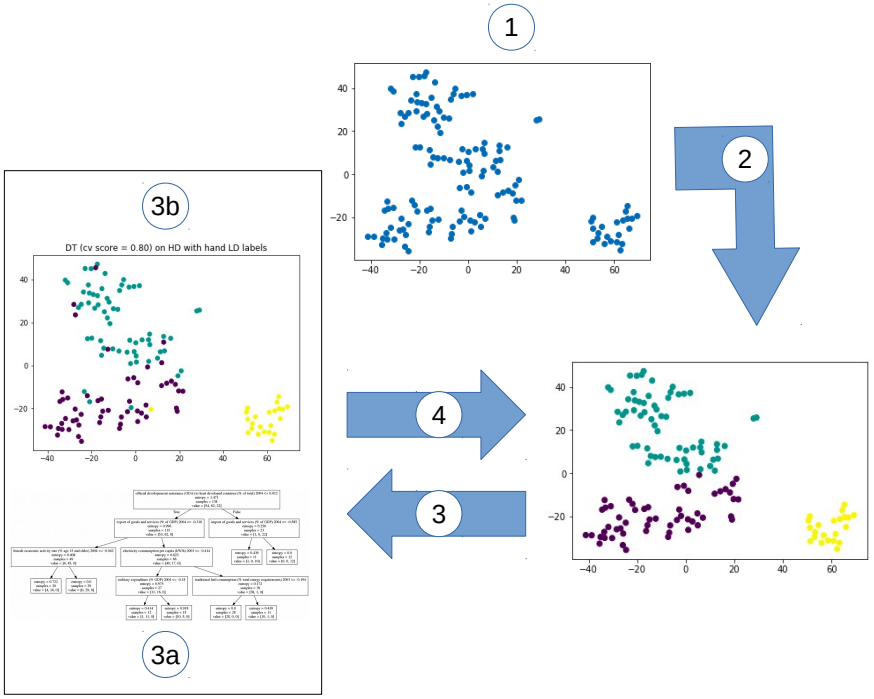


Figure 7.2: Figure reproduced from [contrib10]. For a given DR visualization (1), the user selects clusters (2). The three clusters in color correspond to the clusters selected by the user. Then, a visualization of the errors made by a decision tree using the initial features (3a) for explaining the manual clustering is provided (3b). The user can then decide to select the clusters differently and re-run the algorithm (4).

The second step is to use the clusters formed by the user as labels for a decision tree trained on the initial dataset. By doing this, the decision tree (e.g., (3a) in Figure 7.2) approximates the mapping between the HD space and the clusters selected in the LD space (see (3b) in Figure 7.2). The decision tree therefore links the HD data to the LD clusters. Given the explanation provided by the decision tree, the user can decide to select other clusters (or select the clusters differently) and re-run the algorithm (see (4) in Figure 7.2). The user then stops when he has enough explanations for the visualization.

Thanks to this procedure, users can have an understanding of how the clusters were mapped from the initial dataset to the embedding. The clusters that are explained are selected by the user, which allows us to make no assumptions about how clusters are defined. In the next section, a user-based experiment for validating the approach is presented. This experiment both evaluates the quality of the implemented interface and the pipeline presented in this section.

Figure 7.3: Interface of IXVC used for the evaluation.

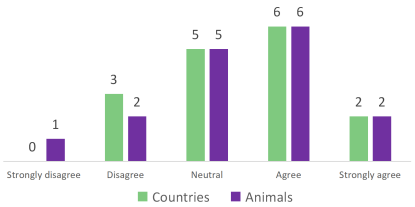
7.2 Validation with a User-Based Experiment

IXVC was validated through a user-based experiment. Before running the experiment, a preliminary evaluation was performed with two computer science researchers in order to find bugs in the interface and improve it. Figure 7.3 shows the interface that was used for the evaluation.

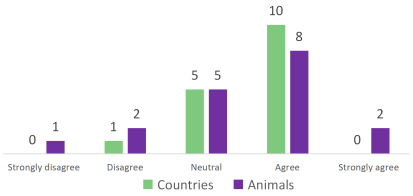
Then, 16 computer science students participated in a 45-minute session where they were asked to use the pipeline to gain insights about two datasets: a dataset with socio-economic information about countries [74] and a dataset about animals [28]. All participants had previously studied the country dataset with t-SNE, but without IXVC, in a machine learning course.

After explaining the basics of the task, the participants were left free to use the tool to gain insights about the datasets. During the experiment, an observation phase was conducted where information on the use of the pipeline was collected. At the end of the experiment, a questionnaire was provided to the users in order to assess the usability of the interface and the usefulness of IXVC.

The observation phase allowed us to see that most participants intuitively used the pipeline to gain insights about the datasets. Some unexpected behaviors could also be detected, such as the use of IXVC to check if a particular HD feature was used to cluster instances, instead of using it to understand how instances are clustered.



(a) Extent to which IXVC helped users to conduct a more objective analysis of the visualizations.



(b) Extent to which IXVC helped users to better understand each dataset.

Figure 7.4: Example of results from the evaluation questionnaire on IXVC.

The usability score (SUS) [22], well-known in the HCI literature, is assessed by asking 10 questions using a 5-point Likert scale on the usability of the interface. Some examples of questions from the SUS questionnaire are “I think that I would like to use this system frequently” and “I felt very confident using the system”. For the interface developed for the experiment, the SUS was 77 (95% confidence interval was [72, 82]), meaning that the interface was not an obstacle when using the pipeline. Indeed, in the literature, a SUS of 68 is the threshold to surpass in order for the system to be considered above average in usability, and 71.4 is the threshold for “good usability” [5].

The results of the questionnaire on the usefulness of IXVC were encouraging, with 75% of the participants stating that using the tool was better than not, and 81% stating that they would likely use it again for interpreting t-SNE. Furthermore, participants found the interface useful for gaining a better understanding of the datasets (median of 4 on the 5-point Likert scale). Figure 7.4 shows examples of results from the questionnaire conducted to evaluate IXVC.

7.3 Conclusion

This chapter presented a pipeline called IXVC for explaining how clusters are mapped from HD data to NLDR visualizations. In order to do that, a decision tree is trained to learn how a user-selected set of clusters can be explained using the original features in the dataset. The user can interact with the visualization by selecting different sets of visual clusters until he has enough explanations to understand the mapping.

As IXVC relies on user selection of visual clusters, a user-based experiment was set up to evaluate the pipeline. The well-known SUS score was used to assess the usability of the interface and a questionnaire was provided in order to evaluate the usefulness of the pipeline. The results of the experiment showed that IXVC helped users understand how the clusters were mapped.

CHAPTER 8

DIMENSIONAL EXPLANATION OF EMBEDDINGS

One first way of explaining an NLDR mapping is through an analysis of the visual clusters in the visualization, as presented in the previous chapter. The other way to explain the mapping is by finding a meaning in the LD dimensions by using the HD features. Relating the embedding dimensions to the original dimensions is a natural way to interpret DR mappings, e.g., with PCA. However, some elements can make this kind of interpretation difficult.

First, if the DR techniques are non-parametric (meaning that no model parameters are provided to understand the mapping, see Section 4.3), the mapping is implicit and, therefore, cannot be extracted and analyzed. In this case, a way to approximate it must be found. Second, the original features that can be used for the explanation may not ease the user understanding. This is the case when users cannot understand the values of the features, and therefore, even a simple linear combination of these features is not interpretable. Third, in some cases like the one in psychology presented in Chapter 4, using the original features is not even desired. Indeed, in the psychology example, all instances are compared to all other instances, which means that the features of each instance are the values of (dis)similarity w.r.t. the other instances. These n features characterizing the n instances are difficult to use and to analyze.

For all these reasons, a solution could be to use external features, meaning features on the same instances, but coming from a different source. This problem can be stated as a multi-view problem, where the different views are different sets of features collected from different sources. For instance, a medical doctor can describe the symptoms of a patient by questioning him (first source) and through a blood test (second source). The two sources describe roughly the same things for the same patient, but differently.

This chapter is based on four contributions on dimensional explanations (i) Adrien Bibal, Rebecca Marion, and Benoît Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In **Proceedings of ESANN**, pages 537–542, 2018 [contrib11], (ii) Rebecca Marion, Adrien Bibal, and Benoît Frénay. BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables. **Neurocomputing**, 342:83–96, 2019 [contrib12], (iii) Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay. BIOT: Explaining multidimensional MDS embeddings using the best interpretable orthogonal transformation. **Submitted to Neurocomputing** [contrib13] and (iv) Adrien Bibal, Viet Minh Vu, Géraldin Nanfack, and Benoît Frénay. Explaining t-SNE embeddings locally by adapting LIME. In **Proceedings of ESANN**, pages 393–398, 2020 [contrib14]. First, Section 8.1 presents property fitting (PROFIT), a technique that is widely used to explain MDS dimensions with external features. The limits of such an approach are discussed, and our techniques to solve the problem, BIR and BIOT, are presented in Section 8.2. In the case where a dimensional explanation cannot be provided globally, our solution to use BIR with LIME in order to explain NLDR mappings locally is presented in Section 8.4.

8.1 Dimensional Explanation of MDS with PROFIT

One way to have dimensional explanations of embeddings is to use linear models. Linear models are used in social sciences to get understandable insights about visualizations. Social sciences have empirically shown that linear models can provide a reasonable approximation of nonlinear dimensionality reduction embeddings, at least when pairwise distances are preserved such as with Kruskal's Stress. Furthermore, it is possible to linearly combine external features (i.e. features that were not used for the dimensionality reduction process) to explain the nonlinear combination of the original features.

Let $\mathbf{Q}$ be a matrix of (dis)similarities between instances, $\mathbf{X}$ be an embedding generated from $\mathbf{Q}$ and $\mathbf{F}$ be a matrix of external features for explaining $\mathbf{X}$. One popular way of obtaining an explanation is to link $\mathbf{X}$ and $\mathbf{F}$ with a linear model. PROperty FITting (PROFIT) is a method, based on a linear model, that allows users to see trends in a visual embedding. The problem, which dates back to Kruskal's book on MDS [44], is to find the matrix of weights $\mathbf{W}$ in

$$\mathbf{F} = \mathbf{X}\mathbf{W} + \mathbf{E}, \tag{8.1}$$

with $\mathbf{E}$ being an error term. When $\mathbf{W}$ is found, the vector of weights $\mathbf{w}_j$ corresponding to each feature $\mathbf{f}_j$ can be plotted onto the embedding. In practice, the unit vector $\hat{\mathbf{w}}_j$ is used:

$$\hat{w}_{jk} = \frac{w_{jk}}{\sqrt{w_{j1}^2 + w_{j2}^2}}, \tag{8.2}$$

where $\hat{w}_{jk}$ is the k^{th} element of $\hat{\mathbf{w}}_j$. w_{j1} and w_{j2} correspond to how important is the feature $\mathbf{f}_j$ for the two dimensions of the embedding. When w_{j1} and w_{j2} are found for each j, a vector representing the j^{th} feature can be plotted onto the embedding.

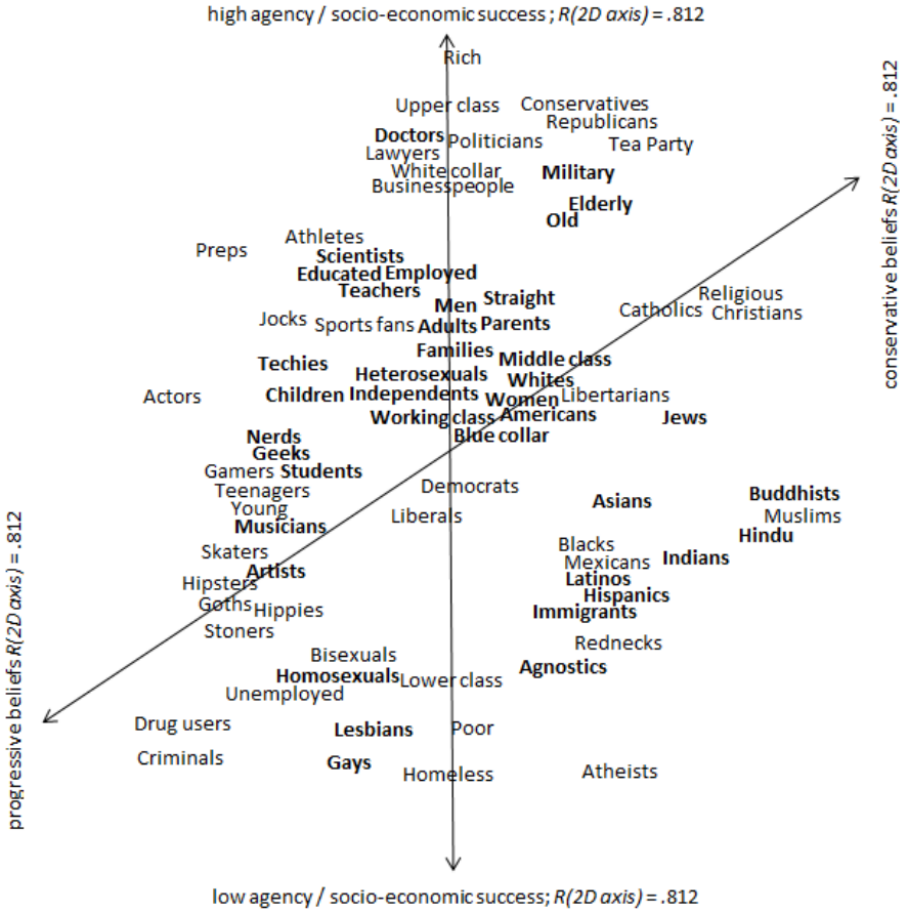


Figure 8.1: Figure reproduced from Koch et al. [41] presenting two stereotype trends in an MDS embedding of social groups: socio-economic success (vertical line) and progressive/conservative beliefs (oblique line).

An example of such trends is presented in Figure 8.1, where the two features progressive/conservative beliefs and socio-economic status from a matrix $\mathbf{F}$ are used to describe an embedding $\mathbf{X}$ of social groups obtained from a dissimilarity matrix $\mathbf{Q}$. For instance, while the embedding shows that members of the military and the elderly are similar in the USA (north-east of the visualization), PROFIT explains this similarity with their similar conservative beliefs and their socio-economic success. This last stereotype can be explained by the fact that the financial situation of members of the military is good and stable in the USA, which is similar to the elderly, who have generally accumulated wealth during their life.

One issue with this solution is that the trends can only be visualized and, therefore, it does not work with embeddings of more than two dimensions. This can be a problem when the 2D embedding preserves much less information than, e.g., the 3D or 4D embeddings. Another issue is that the features are projected one by one on the embedding, and no information is provided about how their combinations can help understand the visualization. Indeed, by treating features individually, PROFIT may provide no satisfactory results, whereas combining 2 or 3 features would explain some trends. BIR and BIOT are two methods that were developed to solve these issues. They are presented in the next sections.

8.2 Global Dimensional Explanations with BIR

In order to solve the issues presented in the previous section, one intuitive solution would be to reverse the problem. That is to say, instead of finding $\mathbf{W}$ in Equation 8.1, the problem to solve becomes to find $\mathbf{W}$ in

$$\mathbf{X} = \mathbf{F}\mathbf{W} + \mathbf{E}. \tag{8.3}$$

By looking at the problem this way, finding $\mathbf{W}$ means finding the linear combinations of external features $\mathbf{F}$ that best explain the dimensions of the embedding $\mathbf{X}$.

While being more intuitive, this problem is in fact harder to solve. This is due to the fact that, because of the objective of MDS (i.e. the stress), the result $\mathbf{X}$ of MDS is invariant to rotation. Indeed, through the stress, MDS tries to minimize the difference between the pairwise distances in HD and LD. However, if the embedding $\mathbf{X}$ is rotated (i.e. the orthogonal axes revolve around the instances in LD), the pairwise distances in LD are still the same, and so is the stress. This makes MDS results *invariant to rotation*, i.e. rotating the embedding does not change the score of the objective function. Because of this, the weights $\mathbf{W}$ are arbitrary, as a different $\mathbf{W}$ will be found for each orientation of $\mathbf{X}$. The new problem is therefore to find the orientation $\mathbf{R}$ of $\mathbf{X}$ that makes it possible to find the best interpretable solution $\mathbf{W}$, hence the name of our method: Best Interpretable Rotation (BIR) [contrib11, contrib12].

After analyzing the effect of rotating $\mathbf{X}$ on $\mathbf{W}$ given different regularization settings in [contrib11, contrib12], BIR was introduced as the problem of finding the best angle θ^* such that

$$\theta^* = \arg\min_{\theta} \frac{1}{2n} ||\mathbf{X}\mathbf{R}^{\theta} - \mathbf{F}\mathbf{W}^{\theta}||_F^2 + \lambda \sum_{k=1}^{2} ||\mathbf{w}_k^{\theta}||_0, \tag{8.4}$$

where $\mathbf{W}^{\theta}$ are Lasso weights found given a particular angle θ.

Figure 8.2 shows some results for BIR compared to competitive methods. In this figure, it can be seen that BIR's curve (in blue) is almost always under the other curves, which means that for a fixed mean squared error (MSE), BIR can explain the embedding with fewer features.

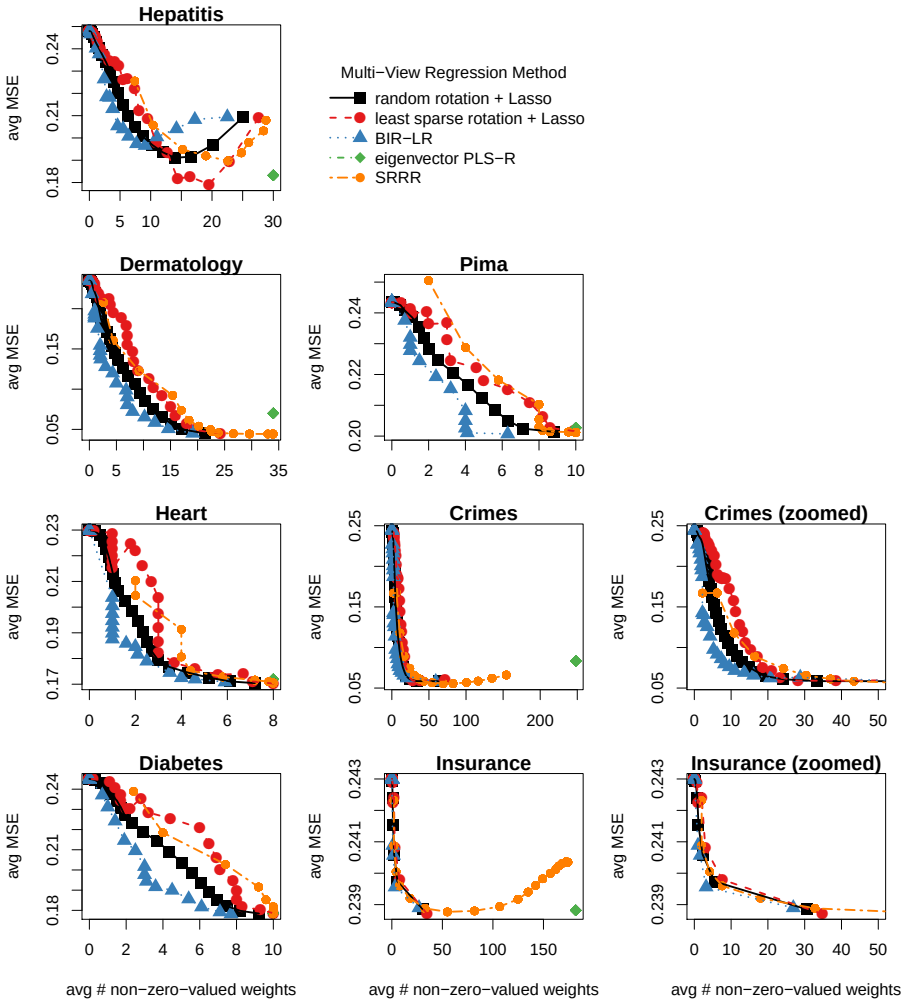


Figure 8.2: Figure reproduced from [contrib12] presenting the mean squared error (MSE) of competitive techniques for different levels of sparsity on various datasets. Each point represents an average value over 10 folds for a particular hyperparameter setting, e.g., a λ for Lasso models. For sparse solutions (i.e. explanations with few variables), the blue curve representing BIR is often below the other curves. This means that for a particular level of sparsity, the error of BIR is lower.

While our solution solves the problem of combining features to explain the visualization, one may want to go further. In particular, BIR's objective function is non-convex and is optimized with simulated annealing. Furthermore, one may want to drop the need to visualize embeddings in order to explain the mapping, and try to explain embeddings with any number of dimensions. In the next section, we present BIOT, a solution that extends BIR in line with these ideas.

8.3 From BIR to BIOT

Best interpretable orthogonal transformation (BIOT) is a technique introduced in [contrib13] that extends BIR for explaining embeddings of any number of dimensions. BIOT contains a new optimization procedure and the optimization of an orthogonal transformation matrix, rather than just a rotation matrix. The BIOT algorithm is described in Algorithm 2.

Algorithm 2: BIOT algorithm

Data: MDS embedding $\mathbf{X}$ and feature matrix $\mathbf{F}$
Result: Explanation of $\mathbf{X}$ with sparse weights $\mathbf{W}$

$\mathbf{R} = \mathbf{I}$;
$\mathbf{X} = \mathbf{XR}$;
$\mathbf{W}$ is obtained by solving Eq. (8.5) for each k of $\mathbf{X}$;
while $\mathbf{W}$ changes **do**
    `// Optimizing R`
    $\mathbf{R}$ is obtained by solving Eq. (8.6);
    $\mathbf{X} = \mathbf{XR}$;
    `// Optimizing W`
    for each dimension k **of** $\mathbf{X}$ **do**
        $\mathbf{W}_{\bullet,k}$ is obtained by solving Eq. (8.5);
return $\mathbf{W}$ and $\mathbf{R}$

The BIOT algorithm is iterative, in the sense that, iteratively, $\mathbf{R}$ is optimized with $\mathbf{W}$ fixed, then $\mathbf{W}$ with $\mathbf{R}$ fixed and then $\mathbf{R}$ again, until convergence. $\mathbf{W}$ is optimized, for each embedding dimension k, as follows

$$\mathbf{w}^*_{\bullet,k} = \arg\min_{\mathbf{w}_{\bullet,k}} \frac{1}{2n} ||\mathbf{X}\mathbf{r}_{\bullet,k} - \mathbf{F}\mathbf{w}_{\bullet,k}||_2^2 + \lambda ||\mathbf{w}_{\bullet,k}||_1, \tag{8.5}$$

which is a Lasso problem when $\mathbf{R}$ is fixed. Then, when $\mathbf{W}$ is fixed, the orthogonal transformation matrix $\mathbf{R}$ is optimized as follows

$$\arg\min_{\mathbf{R}} \frac{1}{2n} ||\mathbf{X} - \mathbf{F}\mathbf{W}\mathbf{R}^\top||_F^2 + \lambda \sum_{k=1}^{m} ||\mathbf{w}_{\bullet,k}||_1 \quad \text{s.t. } \mathbf{R} \text{ is an orthogonal matrix.} \tag{8.6}$$

This is an orthogonal Procrustes problem that can be rewritten as

$$\arg\min_{\mathbf{T}} ||\mathbf{A} - \mathbf{B}\mathbf{T}||_F^2 \text{ s.t. } \mathbf{T}\mathbf{T}^\top = \mathbf{T}^\top\mathbf{T} = \mathbf{I}, \tag{8.7}$$

where $\mathbf{A} = \mathbf{X}/\sqrt{2n}$, $\mathbf{B} = \mathbf{FW}/\sqrt{2n}$ and $\mathbf{T} = \mathbf{R}^\top$. For all details on how the Lasso problem and the Procrustes problem are dealt with in BIOT, see [contrib13].

Table 8.1 from the evaluation of BIOT shows that for solutions that are equally sparse between competitors, BIOT solutions have a lower error. For each method, the left column is the average number of non-zero weights (i.e. the number of features used in the model) and the right column is the average error of the model. The method with the lowest number of non-zero weights has its number highlighted in bold, and its lowest error highlighted in italic. More than one method can have bold or italic numbers, if their number of non-zero weights or error level is not significantly different. Note that results for BIR were only given for 2D embeddings, as BIR can only be applied to 2D embeddings. The column "stress" indicates the error of the dimensionality reduction process.

It can be seen in Table 8.1 that, while it is popular in the literature to work with 2D visualizations because they can be analyzed by experts, BIOT can provide explanations of MDS embeddings with more dimensions. Therefore, the analyzed embeddings can have a lower stress, less features are used on average for explaining each of the embedding dimensions and the explanations have a lower error.

Thanks to BIOT, it is now possible to explain embeddings of more than two dimensions, which means embeddings with a lower information preservation error. For instance, instead of explaining the 2D embedding of Figure 8.1, 3D, 4D or 5D embeddings of better quality can now be explained.

Table 8.2 presents an example of embeddings that can now be explained (i.e. embeddings with a number of dimensions higher than two). Instead of the two trends found by PROFIT in Figure 8.1, BIOT finds the trends wealthy, traditional-conventional and not smart in a 3D embedding of the same dataset. For a 4D embedding, the traditional-conventional is split into two dimensions, which provides the four trends wealthy, religious-traditional, conventional and not smart. The last example is for a 5D embedding, where an additional trend is found: conservative. The five dimensions are thus explained by the five trends wealthy, traditional-religious, conventional, not smart and conservative. As it can be observed, increasing the number of dimensions to analyze makes it possible to avoid erroneously putting orthogonal trends in the same basket.

BIR and BIOT can be used to explain dimensions of NLDR embeddings that are invariant to rotation. However, some NLDR techniques are invariant to rotation, but cannot be explained globally. Indeed, some techniques such as t-SNE do not preserve long pairwise distances well [80]. In this case, BIR and BIOT can be modified to explain embeddings locally. This work is presented in the next section.

8.4 Using BIR and BIOT for Local Explanations

t-SNE is a popular and efficient NLDR technique, but it has an important drawback: the distances between the instances cannot be trusted, at least for long distances [80]. This is due to the objective of t-SNE, which is to focus on the preservation of the neighborhood of instances. Because of this issue, the dimensions of t-SNE cannot be analyzed and techniques for global explanations like BIR and BIOT cannot be

	m	stress	BIR	BIOT	ePLS	SRRR
Do	2	0.070	4.8, *30*	**3.9**, *29*	5.0, *30*	6.9, *29*
	3	0.038		**3.1**, *25*	4.0, *25*	7.8, *24*
	4	0.026		**2.7**, *21*	5.7, *21*	10.1, *20*
	5	0.018		**2.2**, *19*	2.9, *19*	9.5, *18*
	6	0.013		**1.7**, *17*	2.5, *17*	9.8, 17
	7	0.012		**1.5**, *14*	2.1, *15*	9.7, 16
	8	0.008		**1.3**, *13*	1.9, *14*	10.1, 15
	9	0.006		**1.1**, *12*	1.7, *12*	10.1, 14
	10	0.005		**1.0**, *11*	1.5, *11*	10.1, 13
	11	0.004		**0.9**, *10*	1.4, *11*	10.1, 12
	12	0.003		**0.8**, *9*	1.3, *10*	10.1, 11
Mi	2	0.144	3.2, *77*	**2.8**, *78*	**3.1**, *78*	4.8, *77*
	3	0.112		**2.0**, *51*	4.6, *53*	5.1, *53*
	4	0.091		**1.2**, *41*	3.8, *44*	4.5, *43*
	5	0.077		**1.1**, *35*	3.0, *37*	5.2, *36*
	6	0.065		**2.0**, *30*	3.1, *31*	10.1, *31*
	7	0.057		**1.8**, *26*	2.6, *27*	10.2, *27*
	8	0.049		**1.6**, *23*	2.6, *24*	10.9, *24*
	9	0.044		**1.4**, *21*	2.5, *22*	11.1, *22*
	10	0.040		**1.3**, *19*	3.7, *20*	11.4, *20*
	11	0.036		**1.1**, *17*	3.3, *18*	8.0, *18*
	12	0.032		**1.0**, *16*	3.1, *17*	11.4, *17*
Sp	2	0.089	9.8, *39*	**7.5**, *40*	9.4, *37*	11.3, *37*
	3	0.055		**4.1**, *36*	7.3, *33*	10.9, *37*
	4	0.037		**3.4**, *31*	4.8, *30*	11.4, *30*
	5	0.025		**2.7**, *28*	3.9, *26*	12.3, *28*
	6	0.019		**2.2**, *25*	3.3, *24*	12.5, *24*
	7	0.016		**2.0**, *22*	2.9, *21*	12.6, *21*
	8	0.012		**1.6**, *20*	2.5, *19*	11.1, *20*
	9	0.007		**1.5**, *18*	2.2, *18*	11.1, *18*
	10	0.004		**1.4**, *17*	2.0, *16*	11.2, *16*
	11	0.001		**1.2**, *15*	1.8, *15*	11.2, *15*
St	2	0.291	12.8, *26*	**9.1**, *26*	13.1, *26*	14.9, *27*
	3	0.207		12.8, 17	**11.2**, *16*	20.3, *16*
	4	0.169		**12.6**, *20*	15.3, *19*	26.7, *19*
	5	0.146		**8.3**, 17	18.4, *16*	27.9, *16*
	6	0.134		**14.6**, *14*	**15.0**, *15*	30.3, *14*
	7	0.127		**9.0**, *13*	18.6, 14	30.2, *13*
	8	0.122		**8.5**, *12*	17.9, *12*	30.1, *12*
	9	0.120		**8.0**, *11*	17.5, *11*	30.1, *11*
	10	0.118		**7.4**, 11	17.1, 11	28.9, *10*
	11	0.116		**6.7**, *10*	11.8, *10*	29.2, *10*
	12	0.116		**6.2**, *9*	16.0, 9	29.2, *9*
	13	0.115		**5.6**, 9	15.2, 9	29.4, *8*

Table 8.1: Each result is a pair (average number of non-zero weights, average mean squared error (MSE) $\times 10^3$) corresponding to the λ with the smallest average test MSE. The datasets used are Doubs (Do), Mite (Mi), Spider (Sp) and Stereotypes (St).

$m = 3$	$m = 4$	$m = 5$
wealthy (0.28)	**wealthy** (0.26)	**wealthy** (0.22)
scientific (0.06)		power (0.05)
diversity (0.05)	diversity (0.06)	diversity (0.01)
traditional (0.17)	traditional (0.04)	**traditional** (0.08)
religious (0.01)	**religious** (0.15)	**religious** (0.09)
	comfort (0.04)	comfort (0.04)
	prevention (0.02)	prevention (0.04)
conventional (0.15)	**conventional** (0.22)	**conventional** (0.14)
loyalty (0.05)	loyalty (0.07)	loyalty (0.01)
familiarity (0.01)		individualistic (0.01)
not smart (0.16)	**not smart** (0.13)	**not smart** (0.16)
egoistic (0.07)	egoistic (0.05)	egoistic (0.01)
masculine (0.06)	masculine (0.09)	masculine (0.04)
competitive (0.06)		competitive (0.05)
typical (0.04)	typical (0.03)	typical (0.03)
	intolerant (0.02)	
	familiarity (0.01)	
		conservative (0.14)
		masculine (0.03)
		preservation (0.03)

Table 8.2: Figure from [contrib13] presenting explanations for 3D, 4D and 5D embeddings ($m = 3$, $m = 4$ and $m = 5$) of social groups (dimensions in rows, model weights in parentheses). The most important features for each dimension are in bold.

applied. In order to solve the explainability problem for 2D t-SNE embbedings, a technique introduced in [contrib14] was developed to apply BIR locally. This dimensional explanation of t-SNE makes the hypothesis that pairwise distances are sufficiently well preserved locally to be explained.

As we have seen in Chapter 1, LIME is a popular technique nowadays to locally explain supervised learning models. However, using it to explain t-SNE is hard because the black box cannot be "queried" like other models in supervised learning. Indeed, while it is possible to ask how a supervised learning black-box model would classify a new instance, a t-SNE embedding is fixed after being produced (this is due to its non-parametric nature). In other words, in order to know where a new instance would be placed in the visualization, one would have to re-run t-SNE completely with the new instance, which would have the effect of providing a different embedding than the one that needed to be explained.

In [contrib14], a solution to adapt LIME for t-SNE is proposed. The pipeline summing up the algorithm is presented in Figure 8.3. The first step is to select an instance $\mathbf{x}^{LD}$ in the embedding around which an explanation is desired (Figure 8.3 (a)). The zone around this instance in the visualization will be under scrutiny by the local explainer.

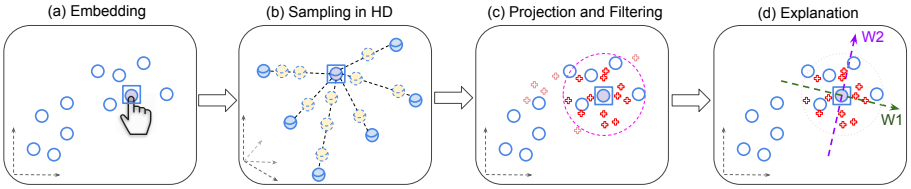


Figure 8.3: Figure reproduced from [contrib14] representing the pipeline of the adaptation of LIME for explaining t-SNE.

Then, new instances $\mathbf{z}_j$ are sampled, or generated, in HD such that their features take values in between the feature values of the instance under scrutiny and the feature values of one of its neighbor (Figure 8.3 (b)). The neighbors that are selected for sampling the $\mathbf{z}_j$ are the ones that were considered to be neighbors by t-SNE when constructing the embedding. In practice, the SMOTE algorithm [24] is used for generating the samples, such that $\mathbf{z}_j = \mathbf{x}^{HD} + \alpha * (\mathbf{x}_j^{HD} - \mathbf{x}^{HD})$, with $\alpha \in [0,1]$. The idea behind the generation of samples $\mathbf{z}_j$ that are between the query point $\mathbf{x}^{HD}$ and one of its neighbor $\mathbf{x}_j^{HD}$ is that the generated samples lie roughly on the HD manifold around $\mathbf{x}^{HD}$. Not doing this is a common issue when using LIME. Indeed, if, for instance, $\mathbf{x}^{HD}$ is an image of a face, then randomly perturbing the pixels of $\mathbf{x}^{HD}$ will probably give an image neighbor of $\mathbf{x}^{HD}$ in HD that is not a face, but a set of pixels with random colors. However, if an image is taken in between $\mathbf{x}^{HD}$ and another image of a face, which is a neighbor of $\mathbf{x}^{HD}$, then the chance to sample a new image of a face is higher.

When the set of sampled instances $\mathbf{Z}$ are generated in HD, an approximation of where they would be projected in the embedding must be found. This is an issue with t-SNE, because the mapping is not provided, which means that the way the instances were projected onto the embedding is not known. Furthermore, the objective of the presented work [contrib14] is to explain the classical t-SNE, and not to use a transformed version of it that would make it parametric (e.g., [35, 75]). In order to approximate where the samples $\mathbf{z}_j$ would have been projected if they were present in the dataset that was used to create the embedding, the t-SNE algorithm is partially re-run with a matrix composed of $\mathbf{X}$ and $\mathbf{Z}$. The algorithm is partially re-run because the original instance positions are fixed, and only the new sampled instances $\mathbf{z}_j$ are projected onto the embedding. After t-SNE is run again, the positions of all $\mathbf{z}_j$ in LD are approximated. However, because of errors due to t-SNE and to the approximation, some sampled instances $\mathbf{z}_j$ may be projected far away from the query instance in the embedding, i.e. far from the zone that needed to be explained. In order to keep the explanation local, the sampled instances that are projected too far away from the queried instance in the embedding are filtered out (Figure 8.3 (c)).

Finally, the last step is to provide a local explanation of the zone of interest in the embedding. In order to do that, BIR is applied on the local zone, providing linear combinations of the HD features that explain how the projection was performed in

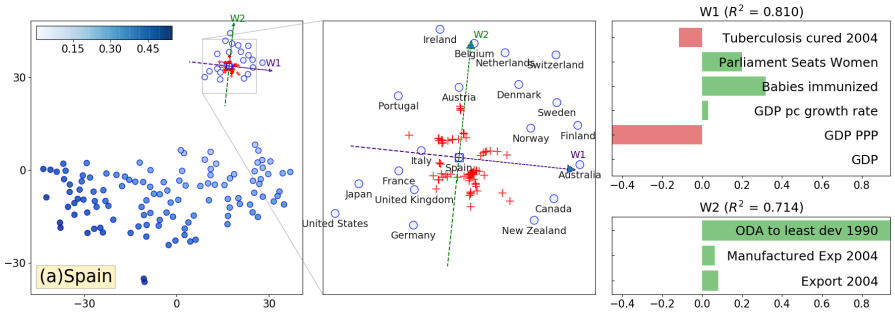


Figure 8.4: Figure from [contrib14] showing explanations of a region of an embedding generated by t-SNE. BIR is used to explain orthogonal trends in this region, based on a sparse combination of the features that were used to generate the embedding.

that zone. This means that the original problem of BIR, which is to find $\mathbf{W}$ and $\mathbf{R}$ in

$$\mathbf{X}^{LD}\mathbf{R} = \mathbf{FWR} + \mathbf{E}, \tag{8.8}$$

with $\mathbf{X}^{LD}$ being the embedding and $\mathbf{F}$ being external features, becomes

$$\mathbf{Z}^{LD}\mathbf{R} = \mathbf{X}^{HD}\mathbf{WR} + \mathbf{E}, \tag{8.9}$$

where $\mathbf{Z}^{LD}$ is the projection of the samples $\mathbf{Z}$ that have not been filtered out and $\mathbf{X}^{HD}$ is the original dataset. By solving this problem, BIR explains the local region in the embedding around the projection of $\mathbf{x}$, the originally selected instance.

Figure 8.4 presents a case study where the adaption of LIME that includes BIR was applied to a visualization generated by t-SNE from the country dataset. Some regions can be seen in the embedding, like a cluster of developed countries. Therefore, a question can be, for instance: what are the two orthogonal axes that best explain, using the HD features, the region of developed countries in the embeddings?

In the case study of Figure 8.4, the queried instance $\mathbf{x}_{LD}$ is the country Spain and the red points around it are the sampled instances $\mathbf{z}_j$. The quasi-horizontal axis (W1) around Spain in the middle subfigure of Figure 8.4 is explained by 6 of the original 45 features: percentage of tuberculosis cases cured in 2004 (highest on the left), percentage of seats in parliament held by women (highest on the right), percentage of one-year-old babies immunized against measles (highest on the right), growth rate of the GDP per capita (highest on the right), GDP according to the purchasing power parity (PPP) (highest on the left) and GDP (small impact in the model). What is interesting is that, while the axis is mainly explained by the economy, the healthcare of babies and the place of women in politics, the strong economies in 2006 can indeed be found on the left (e.g., USA, Japan, Germany). Furthermore, Iceland is at the far right (outside the picture), being a well-known country for having a very high number of women in parliament and for having had the world's first female president.

As for the quasi-vertical axis, it is explained by 3 of the 45 original features: the official development assistance (ODA) to the least developed countries in 1990 (most important feature in the explanation), percentage of manufactured exports in 2004 and the export of goods and services in percentage of GDP. The countries from the bottom to the top are therefore the ones that provided the most help to the developed countries in 1990.

8.5 Conclusion

In this chapter, three techniques that were developed to explain dimensions of embeddings were presented. Focus was given to the explanation of embeddings with external features, meaning features that were not used for learning the embedding. Best interpretable rotation (BIR) was first presented to solve the problem of explaining dimensions of 2D embeddings that are invariant to rotation. Because of this property, a rotation matrix $\mathbf{R}$ must be optimized at the same time as a sparse matrix of weights $\mathbf{W}$. Then, best interpretable orthogonal transformation (BIOT) was introduced as an extension of BIR for explaining embeddings of more than two dimensions. Furthermore, changes were made to BIR, in BIOT, to consider orthogonal transformations and to not rely on simulated annealing during optimization. Finally, a hybrid of LIME and BIR was presented for explaining embeddings that are only locally explainable (e.g., t-SNE).

Part IV

Postface

CHAPTER 9

CONCLUSION

Three elements were developed in this thesis, each of which have interpretability or explainability at their center. While the first part of this thesis focused on a broad study of interpretability and explainability, the second and third parts were focused on interpretability and explainability in the context of nonlinear dimensionality reduction (NLDR). Very few works in the NLDR literature develop these two concepts. Therefore, in order to introduce interpretability and explainability in the context of NLDR, as well as to propose some techniques, two directions were taken. The first was to focus solely on interpretability and to explore how it can be measured for NLDR visualizations. The second direction was to focus on explainability and to provide solutions in line with the two axes identified by Kruskal [44]: explaining NLDR mappings through their dimensions or through their clusters.

First, in Part I, interpretability and explainability were analyzed through an analysis of the literature, and its vocabulary in particular, in [contrib1, contrib2]. The understanding of explainability in the law was also studied in [contrib3]. These first two contributions have clarified the vocabulary and the meaning of interpretability and explainability in machine learning and in the law. This first part concluded with a section on the role of user-based experiments for assessing interpretability in machine learning, with guidelines proposed in [contrib5]. As user-based experiments are not often used in the machine learning literature, these guidelines were designed to translate important human-computer interaction questions for the field of interpretability.

Second, a focus was given on measuring interpretability in nonlinear dimensionality reduction (NLDR) in Part II. A first contribution in this part was the explanation of how to understand interpretability and explainability in the context of NLDR. Then, the use of quality measures in machine learning and information visualization were studied in [contrib6, contrib7], in order to find a way to define and measure

interpretability through user-based experiments. Based on these measures, a new measure was developed in [contrib8] as the combination of existing measures that best represent user understanding of visualizations. An implicit way to select interpretable visualizations was also proposed in [contrib9]. This thesis provided clues about which quality measures are important in the literature and about whether it is possible to use a combination of them to predict user preferences.

Third, techniques for explaining non-interpretable NLDR visualizations were proposed in Part III. BIR and BIOT, two techniques for explaining MDS dimensions with external features were proposed in [contrib11, contrib12, contrib13]. These techniques can also be applied locally to explain mappings that cannot be explained globally by combining them with LIME [contrib14]. Instead of explaining the embedding dimensions, clusters can also be used to understand the mapping. In [contrib10], IXVC, an interactive pipeline based on decision trees, was proposed for explaining how clusters are projected.

We conclude by stressing that while users play an important role in the study of interpretability and explainability, some reduction in model complexity can increase the interpretability for all users. Indeed, interpreting a linear model containing thousands of weights is certainly more difficult than interpreting a linear model with only three weights. This explains why some techniques developed in this thesis are based on user-based experiments (e.g. the interactive explanations of NLDR clusters) and others are based on the idea that sparse linear regression will always be more interpretable than an implicit, non-parametric mapping (e.g. BIR and BIOT). We hope we have helped define the notion of interpretability in the context of NLDR, and that the techniques proposed to measure interpretability or to explain NLDR mappings will open the way to further work in the field.

CHAPTER 10

GOING FURTHER

This chapter presents different perspectives that can be drawn from the three parts that were developed in this thesis.

As presented in the first chapter of thesis, the literature is not clear about what exactly should be interpretable: the abstract model, its representation or the visualization of the representation. While one future work could be to clarify this, one can also argue that all three components could be optimized in terms of interpretability. First, it may be important to optimize the interpretability of the model, especially when the model is complex and a readable representation is not possible or does not exist. Second, readable representations, such as those for decision trees, could also be the subject of research. Neural networks and support vector machines, for instance, would benefit from interpretable representations. Finally, the visualization could be subject to optimization, as is already studied in the field of the same name.

A limitation from the second chapter of this thesis is that it only considered the point-of-view of the law on interpretability and explainability. However, other fields could also constrain ML models to be interpretable or explainable. For instance, ethics and philosophy may use and reason about interpretability and explainability, just as the legal community does. A first future work could be to combine analyses from these other fields with what is done in ML. Do these fields use the same vocabulary as the field of machine learning, and are their views on how interpretability and explainability work similar to what is studied in the machine learning community? Creating a link between these communities, as was done in this thesis with the legal community, is an interesting perspective. Another field with which forces could be joined is psychology. By studying what interpretability really means for human beings, the machine learning community would be able to better align its methods and measures to the reality of users.

As presented in Chapter 2, natural language processing (NLP) will be indispensable for going further on the subject of explanations. While the legal literature on explainability already requires elements of language in explanations, such as answers to arguments, generating textual explanations remains an open question. One can argue that simple equations (e.g., sparse linear models) and readable model representations (e.g., decision trees) may not be enough for providing understandable explanations for laymen. For that purpose, textual explanations can be seen as a replacement of the actual equations and representation. In order to test these explanations, a Turing test could be run to see if automatically generated textual explanations are convincing.

Chapter 3 focused on some questions that need to be asked when preparing user-based experiments. What was not addressed is the question of how to choose the most relevant proxy task (e.g., user classification and explanation tasks) when the real task cannot be simulated. Guidelines on how to choose a proxy task, given the real task, is a future work.

Concerning the measures of interpretability in NLDR, found in the second part of this thesis, two positions can be taken. The first one, taken in this thesis, argues that a carefully designed measure of interpretability can be useful for all human beings. This is based on the idea that even if a particular decision tree can be understood by some users and not by others, the fact that increasing the complexity of the tree decreases its interpretability is probably true for a large part of the population. Indeed, it does not make sense to say that greatly increasing the complexity will, in general, increase the interpretability. Taking this position, the measure proposed in this thesis is a combination of existing measures. While this combination allowed us to identify the important and non-important measures in the combination, a future work could be to combine the expertise of machine learning and information visualization communities in order to create a new measure based on our proposition in [contrib8]. The second position states that each user is unique and that a single measure cannot sum up all human beings. From this point-of-view, our combination of measures is limited because it cannot be adapted to each user. Therefore, another future work would be to develop an interactive pipeline that would make it possible to relearn our combination of measures for each user. Our current combination of measures could be used, in this case, as a prior in a Bayesian learning framework.

Three ideas could also complement the techniques IXVC, BIR and BIOT presented in the third part of this thesis. First, for linear explanations like BIR and BIOT, it is assumed that the feature matrix $\mathbf{W}$ is sparse enough to be interpretable. However, this may not be always the case. One solution to address this issue would be to insert a penalty into the BIR and BIOT objective functions for grouping features in order to enhance the power of their explanations. This would make it possible to pursue two goals: (i) grouping correlated features together, instead of arbitrary dropping some of them with a regularization like Lasso and (ii) grouping features into meta-features for each dimension. Thanks to this second point, the 12 features that explain a given dimension could be interpreted using 3 meta-features, which would represent subsets of these 12 features.

A second limitation of the third part of this thesis is that the cluster and dimensional explanations are studied separately. One idea to further develop this part of the thesis would be to analyze whether cluster or dimensional explanations are more adapted to users in general. A joint study between machine learning and psychology researchers would make it possible to understand if NLDR mappings are understood more easily when the LD dimensions are explained, or when the mapping of the clusters in visualizations is explained.

A third limitation is that the machine learning literature only focuses on two-dimensional visualizations. Moreover, BIR and IXVC are technically restricted to explanations for two-dimensional embeddings. While BIOT explains higher-dimensional embeddings, this is done without visualization. In contrast, the visualization literature has developed ways to visualize the information from more than two dimensions by using visual channels [79]. Leveraging these visual channels to visualize higher-dimensional embedding information or to design adapted explanation techniques are two possible future works.

CONTRIBUTIONS

[contrib1] Adrien Bibal and Benoît Frénay. Interpretability of machine learning models and representations: an introduction. In **Proceedings of ESANN**, pages 77–82, 2016.

[contrib2] Adrien Bibal and Benoît Frénay. Introduction to interpretability in machine learning. In **BENELEARN**, 2016.

[contrib3] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. **Artificial Intelligence and Law**, 2020.

[contrib4] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Impact of legal requirements on explainability in machine learning. In **ICML Workshop on Law and Machine Learning**, 2020.

[contrib5] Adrien Bibal, Bruno Dumas, and Benoît Frénay. User-based experiment guidelines for measuring interpretability in machine learning. In **EGC Workshop on Advances in Interpretable Machine Learning and Artificial Intelligence**, 2019.

[contrib6] Adrien Bibal and Benoît Frénay. Learning interpretability for visualizations using adapted cox models through a user experiment. In **NIPS Workshop on Interpretable Machine Learning in Complex Systems**, 2016.

[contrib7] Adrien Bibal and Benoît Frénay. Measuring quality and interpretability of dimensionality reduction visualizations. In **SafeML ICLR Workshop**, 2019.

[contrib8] Cristina Morariu, Adrien Bibal, Rene Cutura, Michael Sedlmair, and Benoît Frénay. Combining quality measures for predicting user assessment of dimensionality reduction visualization quality. **To be submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)**.

[contrib9] Viet Minh Vu, Adrien Bibal, and Benoît Frénay. Constraint preserving score for automatic hyperparameter tuning of dimensionality reduction methods for visualization. **To be submitted to IEEE Transactions on Artificial Intelligence (TAI)**.

[contrib10] Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay. An interactive technique for explaining visual clusters in dimensionality reduction visualizations with decision trees. **Submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)**.

[contrib11] Adrien Bibal, Rebecca Marion, and Benoît Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In **Proceedings of ESANN**, pages 537–542, 2018.

[contrib12] Rebecca Marion, Adrien Bibal, and Benoît Frénay. BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables. **Neurocomputing**, 342:83–96, 2019.

[contrib13] Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay. BIOT: Explaining multidimensional MDS embeddings using the best interpretable orthogonal transformation. **Submitted to Neurocomputing**.

[contrib14] Adrien Bibal, Viet Minh Vu, Géraldin Nanfack, and Benoît Frénay. Explaining t-SNE embeddings locally by adapting LIME. In **Proceedings of ESANN**, pages 393–398, 2020.

BIBLIOGRAPHY

[1] Hiva Allahyari and Niklas Lavesson. User-oriented assessment of classification model understandability. In **Scandinavian Conference on Artificial Intelligence**, 2011.

[2] Moussa Amrani, Levi Lúcio, and Adrien Bibal. ML+FV=♡? A survey on the application of machine learning to formal verification. **arXiv preprint arxiv:1806.03600**, 2018.

[3] Kevin D Ashley and Stefanie Brüninghaus. Automatically classifying case texts and predicting outcomes. **Artificial Intelligence and Law**, 17(2):125–165, 2009.

[4] Michael Aupetit and Michael Sedlmair. Sepme: 2002 new visual separation measures. In **IEEE Pacific Visualization Symposium (PacificVis)**, pages 1–8, 2016.

[5] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. **Journal of Usability Studies**, 4(3):114–123, 2009.

[6] Richard E Bellman. **Adaptive control processes: a guided tour**. Princeton university press, 1961.

[7] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, 17(12):2203–2212, 2011.

[8] Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay. An interactive technique for explaining visual clusters in dimensionality reduction visualizations with decision trees. **Submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)**.

[9] Adrien Bibal, Bruno Dumas, and Benoît Frénay. User-based experiment guidelines for measuring interpretability in machine learning. In **EGC Workshop on Advances in Interpretable Machine Learning and Artificial Intelligence**, 2019.

[10] Adrien Bibal and Benoît Frénay. Interpretability of machine learning models and representations: an introduction. In **Proceedings of ESANN**, pages 77–82, 2016.

[11] Adrien Bibal and Benoît Frénay. Introduction to interpretability in machine learning. In **BENELEARN**, 2016.

[12] Adrien Bibal and Benoît Frénay. Learning interpretability for visualizations using adapted cox models through a user experiment. **NIPS Workshop on Interpretable Machine Learning in Complex Systems**, 2016.

[13] Adrien Bibal and Benoît Frénay. Measuring quality and interpretability of dimensionality reduction visualizations. In **SafeML ICLR Workshop**, 2019.

[14] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Impact of legal requirements on explainability in machine learning. In **ICML Workshop on Law and Machine Learning**, 2020.

[15] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. **Artificial Intelligence and Law**, 2020.

[16] Adrien Bibal, Rebecca Marion, and Benoît Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In **Proceedings of ESANN**, pages 537–542, 2018.

[17] Adrien Bibal, Rebecca Marion, Rainer von Sachs, and Benoît Frénay. BIOT: Explaining multidimensional MDS embeddings using the best interpretable orthogonal transformation. **Submitted to Neurocomputing**.

[18] Adrien Bibal, Viet Minh Vu, Géraldin Nanfack, and Benoît Frénay. Explaining t-SNE embeddings locally by adapting LIME. In **Proceedings of ESANN**, pages 393–398, 2020.

[19] Christopher M Bishop. **Pattern recognition and machine learning**. Springer, 2006.

[20] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. **Biometrika**, 39(3/4):324–345, 1952.

[21] Leo Breiman. Random forests. **Machine learning**, 45(1):5–32, 2001.

[22] John Brooke. SUS – A quick and dirty usability scale. **Usability Evaluation in Industry**, 189(194):4–7, 1996.

[23] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. **Communications in Statistics-theory and Methods**, 3(1):1–27, 1974.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research (JAIR)**, 16:321–357, 2002.

[25] Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. **Journal of the American Statistical Association (JASA)**, 104(485):209–219, 2009.

[26] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In **Proceedings of the ACM international conference on knowledge discovery and data mining (SIGKDD)**, pages 785–794, 2016.

[27] Alexandre de Streel, Adrien Bibal, Benoît Frénay, and Michael Lognoul. Explaining the black box: when law controls AI. **CERRE**, 2020.

[28] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[29] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. **arXiv preprint arXiv:1702.08608**, 2017.

[30] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. **arXiv preprint arXiv:1801.01489**, 2018.

[31] Ronald A Fisher. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, 7(2):179–188, 1936.

[32] Benoît Frénay, Daniela Hofmann, Alexander Schulz, Michael Biehl, and Barbara Hammer. Valid interpretation of feature relevance for linear data mappings. In **IEEE Symposium on Computational Intelligence and Data Mining (CIDM)**, pages 149–156, 2014.

[33] Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, 35(6):1098–1107, 2005.

[34] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Margin based feature selection-theory and algorithms. In **Proceedings of the international conference on Machine learning (ICML)**, page 43, 2004.

[35] A. Gisbrecht, A. Schulz, and B. Hammer. Parametric nonlinear dimensionality reduction using kernel t-SNE. **Neurocomputing**, 147:71–82, 2015.

[36] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. **ACM computing surveys**, 51(5):1–42, 2018.

[37] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. **Decision Support Systems**, 51(1):141–154, 2011.

[38] Yvonne Jansen and Pierre Dragicevic. An interaction model for visualizations beyond the desktop. **IEEE Transactions on Visualization and Computer Graphics**, 19(12):2396–2405, 2013.

[39] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In **International Conference on Machine Learning (ICML)**, pages 121–129, 1994.

[40] Been Kim. **Interactive and interpretable machine learning models for human machine collaboration**. PhD thesis, Massachusetts Institute of Technology, 2015.

[41] Alex Koch, Roland Imhoff, Ron Dotsch, Christian Unkelbach, and Hans Alves. The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion. **Journal of Personality and Social Psychology**, 110(5):675–709, 2016.

[42] Y. Kodratoff. The comprehensibility manifesto. **AI Communications**, 7(2):83–85, 1994.

[43] Ron Kohavi and George H John. Wrappers for feature subset selection. **Artificial Intelligence**, 97(1-2):273–324, 1997.

[44] Joseph B Kruskal and Myron Wish. **Multidimensional Scaling**. Sage, 1978.

[45] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. **arXiv preprint arXiv:1902.00006**, 2019.

[46] Alexandre Lebel, Michael Cantinotti, Robert Pampalon, Marius Thériault, Lindsay A Smith, and Anne-Marie Hamelin. Concept mapping of diet and physical activity: uncovering local stakeholders perception in the Quebec City region. **Social Science & Medicine**, 72(3):439–445, 2011.

[47] John A Lee, Diego H Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. **Neurocomputing**, 169:246–261, 2015.

[48] John A Lee and Michel Verleysen. **Nonlinear dimensionality reduction**. Springer Science & Business Media, 2007.

[49] John A Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. **Pattern Recognition Letters**, 31(14):2248–2257, 2010.

[50] Joshua Lewis, Margareta Ackerman, and Virginia de Sa. Human cluster evaluation and formal quality measures: A comparative study. In **Proceedings of the Annual Meeting of the Cognitive Science Society**, volume 34, 2012.

[51] Joshua Lewis, Laurens Van der Maaten, and Virginia de Sa. A behavioral investigation of dimensionality reduction. In **Proceedings of the Annual Meeting of the Cognitive Science Society**, volume 34, 2012.

[52] Zachary C. Lipton. The mythos of model interpretability. In **ICML Workshop on Human Interpretability of Machine Learning**, New York, USA, 2016.

[53] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In **NIPS**, pages 4765–4774, 2017.

[54] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. In **Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pages 2727–2736, 2017.

[55] Rebecca Marion, Adrien Bibal, and Benoît Frénay. BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables. **Neurocomputing**, 342:83–96, 2019.

[56] David Martens, Jan Vanthienen, Wouter Verbeke, and Bart Baesens. Performance of classification models from a user perspective. **Decision Support Systems**, 51(4):782–793, 2011.

[57] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. **arXiv preprint arXiv:1802.03426**, 2018.

[58] Deyu Meng, Yee Leung, and Zongben Xu. A new quality assessment criterion for nonlinear dimensionality reduction. **Neurocomputing**, 74(6):941–948, 2011.

[59] Cristina Morariu, Adrien Bibal, Rene Cutura, Michael Sedlmair, and Benoît Frénay. Combining quality measures for predicting user assessment of dimensionality reduction visualization quality. **To be submitted to IEEE Transactions on Visualization and Computer Graphics (TVCG)**.

[60] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. **arXiv preprint arXiv:1802.00682**, 2018.

[61] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats And Dogs. In **Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)**, pages 3498–3505, 2012.

[62] Fernando V. Paulovich, Luis G. Nonato, Rosane Minghim, and Haim Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, 14(3):564–575, 2008.

[63] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. Comprehensibility of classification trees – survey design. In **Proceedings of the International Multiconference Information Society**, pages 70–73, 2014.

[64] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. Comprehensibility of classification trees – survey design validation. **Proceedings of the International Conference on Information Technologies and Information Society**, pages 5–7, 2014.

[65] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. What makes classification trees comprehensible? **Expert Systems with Applications**, 62:333–346, 2016.

[66] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?" explaining the predictions of any classifier. In **ACM SIGKDD**, pages 1135–1144, 2016.

[67] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. **Nature Machine Intelligence**, 1(5):206–215, 2019.

[68] S. Rüping. **Learning interpretable models**. PhD thesis, Universität Dortmund, 2006.

[69] Michael Sedlmair and Michaël Aupetit. Data-driven evaluation of visual quality measures. In **Computer Graphics Forum**, volume 34, pages 201–210, 2015.

[70] Mattia Setzu, Riccardo Guidotti, Anna Monreale, and Franco Turini. Global explanations with local scoring. In **ECML/PKDD Joint International Workshop on AMLAI & XKDD**, 2019.

[71] Mike Sips, Boris Neubert, John P Lewis, and Pat Hanrahan. Selecting good views of high-dimensional data using class consistency. In **Computer Graphics Forum**, volume 28, pages 831–838, 2009.

[72] 10X Genomics. 1k brain cells from an e18 mouse, 2018.

[73] Robert Tibshirani. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, 58(1):267–288, 1996.

[74] United Nations Development Programme. Human development report, 2006.

[75] Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In **Proceedings of AISTATS**, pages 384–391, 2009.

[76] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. **Journal of Machine Learning Research (JMLR)**, 9(Nov):2579–2605, 2008.

[77] Jarkko Venna and Samuel Kaski. Local multidimensional scaling. **Neural Networks**, 19(6-7):889–899, 2006.

[78] Viet Minh Vu, Adrien Bibal, and Benoît Frénay. Constraint preserving score for automatic hyperparameter tuning of dimensionality reduction methods for visualization. **To be submitted to IEEE Transactions on Artificial Intelligence (TAI)**.

[79] Colin Ware. **Information visualization: perception for design**. 2013.

[80] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. **Distill**, 1(10):e2, 2016.

[81] Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In **IEEE Symposium on Information Visualization**, pages 157–164, 2005.

[82] Leland Wilkinson, Anushka Anand, and Robert Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, 12(6):1363–1372, 2006.

[83] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. **arXiv preprint arXiv:1708.07747**, 2017.

[84] Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In **Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)**, pages 1854–1864, 2018.

[85] Haoxi Zhong, Guo Zhipeng, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In **Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pages 3540–3549, 2018.

Part V

Publications

CHAPTER 11

INTERPRETABILITY OF MACHINE LEARNING MODELS: AN INTRODUCTION

The article presented in this chapter was published in the proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) in 2016.

Interpretability of Machine Learning Models and Representations: an Introduction

Adrien Bibal and Benoît Frénay

Université de Namur - Faculté d'informatique
Rue Grandgagnage 21, 5000 Namur - Belgium

Abstract.
Interpretability is often a major concern in machine learning. Although many authors agree with this statement, interpretability is often tackled with intuitive arguments, distinct (yet related) terms and heuristic quantifications. This short survey aims to clarify the concepts related to interpretability and emphasises the distinction between interpreting models and representations, as well as heuristic-based and user-based approaches.

1 Introduction

According to the literature, measuring the interpretability of machine learning models is often necessary [1], despite the subjective nature of interpretability making such measure difficult to define [2]. Several arguments have been made to highlight the need to consider interpretability alongside accuracy. Some authors note the importance to consider other metrics than accuracy when two models exhibit a similar accuracy [3, 4]. Other authors point out the link between interpretability and the usability of models [5–7]. Often, the medical domain is taken as example. To accept a predictive model, medical experts have to understand the intelligence behind the diagnostic [8], in particular when the decisions surprise them [9]. Furthermore, the detection by experts of anomalies in the model is only possible with interpretable models [10]. Moreover, in some countries, credit denial legally has to be supported by clear reasons, which means that the model supporting this denial has to be interpretable [8]. Finally, it can also be argued that the model itself is a source of knowledge [6, 11, 12].

This survey addresses two issues in the machine learning literature. First, many terms are associated to interpretability, sometimes implicitly referring to different issues. Second, the literature, scattered because of the difficulty to measure interpretability, is neither united nor structured. Although interpretability is often associated with the size of the model, Pazzani wrote in 2000 that "there has been no study that shows that people find smaller models more comprehensibible or that the size of a model is the only factor that affects its comprehensibility" [4]. In 2011, the situation has not changed, according to Huysmans et. al [12] who echo Freitas [11]. Therefore, this survey addresses the two above issues by proposing an unifying and structured view of interpretability focused on models and representations, and concludes by exposing gaps in the literature.

This survey tackles the questions "what is interpretability?" and "how to measure it?". We do not review techniques to make models more interpretable, because we consider the measure of interpretability as being anterior to this

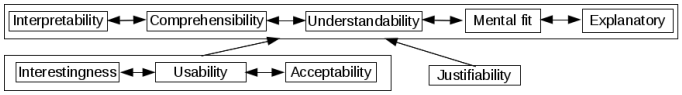


Figure 1: Structure of the main terms used in the literature. $A \rightarrow B$ means that the measure of B requires the measure of A. $A \longleftrightarrow B$ means that measuring A is equivalent to measuring B. Boxes highlight equivalence classes of problems.

problem. In order to answer "what is interpretability?", one needs to gather and unify terms dealing with the problem of interpretability. Section 2 presents several terms used to refer to "interpretability". The second question can be rephrased in terms of comparisons. Sections 3 and 4 review interpretability based on comparisons of models and representations, respectively. Section 5 concludes by highlighting gaps in the literature and corresponding research questions.

2 Different Terms for Different Problems?

This section proposes a unified and structured view of the main terms related to interpretability in the literature. To help researchers when reading papers with distinct terms actually referring to the same problems, a two-level structure is proposed in Fig. 1: the first level consists of synonyms of interpretability and the second level contains terms that rely on interpretability to be measured.

Because of the subjective nature of interpretability, there is no consensus around its definition, nor its measure. First of all, as noted by Rüping [2], the interpretability of a model is not linked to the understandability of the learning process generating this model. Rather, interpretability can be associated to three sub-problems: accuracy, understandability and efficiency [2]. Understandability is central to the problem: an interpretable model is a model that can be understood. Rüping adds accuracy as a necessary criterion in the evaluation of interpretability because "it is always possible to generate a trivial, easily understandable hypothesis without any connection to the data" [2]. Finally, efficiency concerns the time available to the user to grasp the model. Without this criterion, it could be argued that any model could be understood given an infinite amount of time. Other authors use the term interpretability as strict synonym of understandability [5, 10] or comprehensibility [3, 6, 8, 13].

Feng and Michie [14], as other authors after them [15, 16], add "mental fit" to the terms interpretability and comprehensibility. Whereas "data fit" corresponds to predictive accuracy [15], "mental fit" is linked to the ability for a human to grasp and evaluate the model [14]. These authors often link interpretability to explainability, e.g. in [15]. An explanatory model "relates attributes to outcomes in a clear, informative, and meaningful way" [17]. According to Ustun and Rudin, interpretability is intuitive for the expert and closely linked to transparency, sparsity, and explanatory [17].

Some other terms are used in combination with interpretability, but actually refer to other problems. Among them, we can consider usability, acceptability

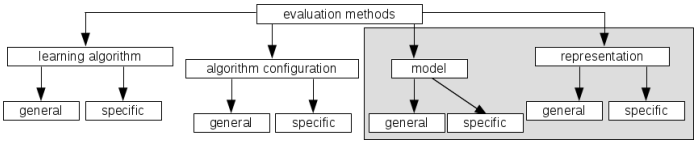


Figure 2: Taxonomy adapted from [20] augmented by representations. Interpretability can be measured for both models and representations (shaded area).

and interestingness. Freitas provides an example where simplicity, often closely linked to interpretability, does not correspond to acceptability for an expert [9]. According to him, following the medical example of [18], experts can be opposed to over-simplistic models. For instance, a three-node tree is probably interpretable, but could be rejected by experts because of its over-simplistic structure [9]. One should note that these two concepts, interpretability and acceptability, are strongly linked but not synonyms, as an acceptable model has to be interpretable, but not vice-versa. In the same way, a model can be considered as not interesting, although being interpretable.

Finally, the term "justifiability" can also be observed alongside interpretability, as it requires an expert to assess that the model "is in line with existing domain knowledge" [8, 19]. As for usability and interestingness cited above, justifiability depends on the interpretability of the model [8].

3 Comparing Models in Terms of Interpretability

Even if we agree on terms to discuss interpretability, one still needs an actual measure of interpretability. In general, measures can be applied on many components of learning systems. Lavesson and Davidsson propose a taxonomy for evaluation methods that classifies them according on whether they assess learning algorithms, algorithm configurations (meta-parameters) or models [20]. Those three elements can be either specific or general, like e.g. evaluation methods for a specific type of model or for distinct types. Similarly, we believe that it is necessary to make a clear distinction between interpretability measures depending on what specific component they target. In this view, we extend the taxonomy of Lavesson and Davidsson by also considering representations in Fig. 2 inspired by [20]: measures of interpretability can be applied to either models or representations through specific approaches. First, comparing mathematical entities such as models requires to define quantitative measurements. This approach is one of the two approaches highlighted by Freitas [9] and can be called the heuristic approach [2]. The second approach uses user-based surveys to assess the interpretability of models. However, unlike the first approach, the models are evaluated through their representations. This second approach is closely linked to information visualisation. This section considers interpretability of models and Section 4 deals with interpretability of their representations.

The heuristic approach can compare models from the same type, e.g. two

SVM models. The size of the model is one of the most used heuristic [2, 6]. For instance, two decision rule lists/sets can be compared in terms of their number of rules and terms [21, 22] and two decision trees can be compared in terms of their number of nodes [23]. Some authors base their heuristics on the psychological theory of Miller, stating that human beings can only deal with 7 ± 2 abstract entities at the same time [24]. For instance, Wheis and Sondhauss propose a maximum of 7 in the number of dimensions [15]. Another way to evaluate the complexity of models is the minimum description length (MDL) [20], but the result depends on the coding scheme for the model parameters, also making this technique specific to the model type [20].

Comparing models of distinct types is more challenging, as the characteristics related to the interpretability of a model from a certain type can be missing in the model from another type. For instance, one cannot minimise the number of nodes of a SVM model. To overcome this difficulty, Backhaus and Seiffert propose to consider three generic criteria: "the ability of the model to select features from the input pattern, the ability to provide class-typical data points and information about the decision boundary directly encoded in model parameters" [25]. For instance, SVM models are graded 1 out of 3, because they only satisfy the third criterion thanks to the "stored support vectors and kernel" [25]. SVM models compete with other models ranked 1 out of 3, but are less interpretable than others ranked 2 or 3 out of 3. Whereas this ranking is able to compare models of distinct types, it does not allow to compare the interpretability of models from the same type. Other limitations of heuristics exist, i.e. they deal with "syntactical interpretability" and do not consider semantic interpretability [9].

4 Comparing Representations in Terms of Interpretability

The limitations depicted in Section 3 are overcome by a measure based on users that evaluate models through their representations. Allahyari and Lavesson used a survey filled by users to evaluate the interpretability of models generated by 6 learning algorithms [26]. This user-based study compared models pairwise by asking questions like "is this model more understandable than the other one?" [26]. Such surveys allow comparing models of the same type, but also models of distinct types. Following the same idea, Piltaver et. al. designed a survey [27] validated [13] on decision trees. Huysmans et. al. checked the accuracy, answer time and answer confidence of users who were asked to grasp a certain model through its representation [12]. Other questions evaluate the understanding of the model. These authors compared the interpretability of three different representations: trees, decision tables and textual representation of rules. Thanks to this kind of evaluation, they could check the link between interpretability and the simplicity of the model, and could conclude by asking questions such as: "to what extent the representations discussed in this study continue to remain useful once they exceed a certain size" [12]. This new trend echoes Rüping when he wrote that "due to the informal nature of the concept of interpretability, a survey over human expert is the most promising measure" [2].

Evaluating representations before evaluating accuracy of models has a particular advantage. Following the idea of Wheis and Sondhauss [15], one could first choose the type of representation having the highest interpretability for a certain group of users, and only then select the type of model having the highest accuracy among those that can be represented by the selected representation.

As a final remark, one could argue that, in the context of interpretability, there is no such thing as a comparison of models, but only comparisons of representations. One can go further and only compare visualisations of model representations, since representations can be either uninterpretable or highly interpretable depending on the way they are shown to the user. Yet, it should be noted that the user-based approach (comparing representations and visualisations thereof) does not allow to quantify interpretability. In contrast, heuristics can be integrated in learning through multi-objective optimisation techniques.

5 Conclusion

This paper presents two major difficulties in the measure of interpretability. First, distinct terms are used in the literature. We separated them into the ones used as strict synonyms (e.g. understandability and comprehensibility) and the ones that depend on interpretability to be defined but related to distinct problems (e.g. justifiability and usability). Second, papers in the literature can be divided into comparisons of the interpretability of models and representations, that is comparisons based on mathematical heuristics or user-based surveys.

In the literature, there is no clear-cut distinction between the interpretability measure of models and representations. The two research questions "what is an interpretable model?" and "what is an interpretable representation?" need to be investigated independently. Furthermore, many papers rely on intuition in the use of interpretability, which leads to a focus on "white-boxes" (decision trees, decision rules, etc.) and a lack of consideration of "black-boxes" (SVM, neural networks, etc.). This distinction would benefit from a grey-scale approach. Finally, there is a lack of literature around user-based measures of interpretability, leaving the question "do heuristics accurately model the understanding of users?" with almost no answer. There is a need to link the results of user-based surveys with heuristics in order to translate the former into the latter and hopefully optimise mathematically the interpretability described by users.

References

[1] Y. Kodratoff. The comprehensibility manifesto. *AI Communications*, 7(2):83–85, 1994.

[2] S. Rüping. *Learning interpretable models.* PhD thesis, Universität Dortmund, 2006.

[3] C. Giraud-Carrier. Beyond predictive accuracy: what? In *Proc. ECML*, pages 78–85, Chemnitz, Germany.

[4] M. J. Pazzani. Knowledge discovery from data? *IEEE Intelligent Systems and their Applications*, 15(2):10–12, 2000.

[5] G. Nakhaeizadeh and A. Schnabl. Development of multi-criteria metrics for evaluation of data mining algorithms. In *Proc. KDD*, pages 37–42, Newport Beach, CA, USA, 1997.

[6] I. Askira-Gelman. Knowledge discovery: comprehensibility of the results. In *Proc. HICSS*, volume 5, pages 247–255, Maui, HI, USA, 1998.

[7] A. Vellido, J. D. Martin-Guerroro, and P Lisboa. Making machine learning models interpretable. In *Proc. ESANN*, pages 163–172, Bruges, Belgium, 2012.

[8] D. Martens, J. Vanthienen, W. Verbeke, and B. Baesens. Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793, 2011.

[9] A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

[10] A. Andrzejak, F. Langner, and S. Zabala. Interpretable models from distributed data via merging of decision trees. In *Proc. CIDM*, pages 1–9, Singapore, 2013.

[11] A. A Freitas. Are we really discovering interesting knowledge from data? *Expert Update*, 9(1):41–47, 2006.

[12] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

[13] R. Piltaver, M. Luštrek, M. Gams, and S. Martinčić-Ipšić. Comprehensibility of classification trees - survey design validation. In *Proc. ITIS*, pages 5–7, Šmarješke toplice, Slovenia, 2014.

[14] C. Feng and D. Michie. Machine learning of rules and trees. *Machine Learning, Neural and Statistical Classification. Ellis Horwood, Hemel Hempstead*, 1994.

[15] C. Weihs and U.M. Sondhauss. Combining mental fit and data fit for classification rule selection. In *Exploratory Data Analysis in Empirical Research*, pages 188–203. Springer, 2003.

[16] O. Maimon and L. Rokach. Decomposition methodology for knowledge discovery and data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 981–1003. Springer, 2005.

[17] B. Ustun and C. Rudin. Methods and models for interpretable linear classification. *arXiv preprint arXiv:1405.4047*, 2014.

[18] T. Elomaa. In defense of c4. 5: Notes on learning one-level decision trees. In *Proc. ICML*, pages 62–69, Beijing, China, 2014.

[19] D. Martens, M. De Backer, R. Haesen, B. Baesens, C. Mues, and J. Vanthienen. Ant-based approach to the knowledge fusion problem. In *Proc. ANTS*, pages 84–95, Brussels, Belgium, 2006.

[20] N. Lavesson and P. Davidsson. Evaluating learning algorithms and classifiers. *International Journal of Intelligent Information and Database Systems*, 1(1):37–52, 2007.

[21] M. Schwabacher and P. Langley. Discovering communicable scientific knowledge from spatio-temporal data. In *Proc. ICML*, pages 489–496, Williamstown, MA, USA, 2001.

[22] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. An interpretable stroke prediction model using rules and bayesian analysis. In *Proc. AAAI*, Bellevue, WA, USA, 2013.

[23] A. Van Assche and H. Blockeel. Seeing the forest through the trees: Learning a comprehensible model from an ensemble. In *Proc. ECML*, pages 418–429, Warsaw, Poland, 2007.

[24] G. A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81–97, 1956.

[25] A. Backhaus and U. Seiffert. Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. *Neurocomputing*, 131:15–22, 2014.

[26] H. Allahyari and N. Lavesson. User-oriented assessment of classification model understandability. In *Proc. SCAI*, pages 11–19, Trondheim, Norway, 2011.

[27] R. Piltaver, M. Luštrek, M. Gams, and S. Martinčić-Ipšić. Comprehensibility of classification trees–survey design. In *Proc. IS*, pages 70–73, Ljubljana, Slovenia, 2014.

CHAPTER 12

INTRODUCTION TO INTERPRETABILITY IN MACHINE LEARNING

The extended abstract presented in this chapter was published at BENELEARN in 2016.

Introduction to Interpretability in Machine Learning

Adrien Bibal ADRIEN.BIBAL@UNAMUR.BE
Benoît Frénay BENOIT.FRENAY@UNAMUR.BE
PReCISE, Faculty of Computer Science, University of Namur, rue Grandgagnage 21, 5000 Namur, Belgium

Keywords: interpretability, comprehensibility, measures, machine learning models

Interpretability is considered as important in the literature. Yet, measuring interpretability seems challenging due to its subjective nature. This abstract presents the survey of the literature by Bibal and Frénay (2016).

Several terms are used in the literature alongside interpretability. Usability is one of those terms. A model is not usable if it is rejected despite an acceptable accuracy. In the medical domain, Freitas (2014) noted that a simple, easy to read, decision tree can be refused because medical doctors may consider that a simple model cannot represent complex medical situations. Usability depends on the interpretability of the model to be measured. The same situation can be observed with justifiability (Martens et al., 2011) which bridges the gap between the description of the data made by the model and the knowledge of the application domain: "does the model justify (or correspond to) the existing knowledge of the domain?".

Interpretability is more fundamental and corresponds to the ability of a human to comprehend (Giraud-Carrier, 1998) or to understand (Rüping, 2006) the model. Two ways to handle the measure of interpretability are proposed in the literature.

On the one hand, heuristics correspond to approximations of the human understanding made by the machine learning researcher (Rüping, 2006), e.g. the model complexity. This approach is easy to formalise but can hardly compare models of different types. For instance, one cannot compare the number of nodes of a decision tree with those of a neural network.

On the other hand, users can be considered in the evaluation of human comprehensibility. Some authors work with user-based surveys to assess the interpretability of models (Allahyari & Lavesson, 2011). In this methodology, measuring the interpretability consists in asking questions such as "do you find this model understandable?" or "is this model more understandable than that one?". This goes beyond the limits of the heuristics approach as users can compare models of distinct types. However, the user-based approach is limited by the difficulty to quantify interpretability from the answers of surveys and the existence of different model representations.

Bibal and Frénay (2016) highlights gaps in the literature. First, there are almost no links between the two approaches in the literature. The heuristics approach does not use the user-based approach for validation and the user-based approach does not try to extract heuristics that could be used to quantify interpretability. This is mostly due to the lack of research in the direction of the user-based surveys approach. Second, models considered as black boxes are neglected. The measure of interpretability mostly deals with white-boxes (e.g. decision trees and rule lists). However, one could argue that a very simple SVM may be more interpretable than a very complex decision tree. The measure of interpretability needs more investigation and a grey-scale measure taking advantage of the two approaches presented here could be developed.

References

Allahyari, H., & Lavesson, N. (2011). User-oriented assessment of classification model understandability. *Proc. SCAI* (pp. 11–19).

Bibal, A., & Frénay, B. (2016). Interpretability of machine learning models and representations: an introduction. *Proc. ESANN 2016* (pp. 77–82).

Freitas, A. A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, *15*, 1–10.

Giraud-Carrier, C. (1998). Beyond predictive accuracy: what? *Proc. ECML* (pp. 78–85).

Martens, D., Vanthienen, J., Verbeke, W., & Baesens, B. (2011). Performance of classification models from a user perspective. *Dec. Support Syst.*, *51*, 782–793.

Rüping, S. (2006). *Learning interpretable models*. Doctoral dissertation, Universität Dortmund.

CHAPTER 13

LEGAL REQUIREMENTS ON EXPLAINABILITY IN MACHINE LEARNING

The article presented in this chapter was published in the journal Artificial Intelligence and Law in 2020. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Artificial Intelligence and Law, Legal Requirements on Explainability in Machine Learning, Adrien Bibal, Michael Lognoul, Alexandre de Streel and Benoît Frénay, 2020.

Artificial Intelligence and Law manuscript No.
(will be inserted by the editor)

Legal Requirements on Explainability in Machine Learning

Adrien Bibal · Michael Lognoul · Alexandre de Streel · Benoît Frénay

Published: 30 July 2020

Abstract Deep learning and other black-box models are becoming more and more popular today. Despite their high performance, they may not be accepted ethically or legally because of their lack of explainability. This paper presents the increasing number of legal requirements on machine learning model interpretability and explainability in the context of private and public decision making. It then explains how those legal requirements can be implemented into machine-learning models and concludes with a call for more inter-disciplinary research on explainability.

Keywords Interpretability · Explainability · Machine Learning · Law

1 Introduction

As deep learning and other highly accurate black-box models develop, the social demand or legal requirements for interpretability and explainability of machine learning models are becoming more significant (Pasquale, 2015; Doshi-Velez and Kortz, 2017). Interpretability can be defined as the ability for a model to be understood by its users (Kodratoff, 1994). For instance, decision trees with a small number of nodes can be considered interpretable, while support vector machines and neural networks are often considered as black boxes. However, despite these intuitions, interpretability has yet to be defined formally in the literature (Bibal and Frénay, 2016; Lipton, 2016). Such a definition is hard to provide as it may depend, among other things on semantics and the level of expertise of the model's users. Furthermore, in machine learning, interpretability and explainability have

A. Bibal · B. Frénay
PReCISE - Faculty of Computer Science - NADI
University of Namur
rue Grandgagnage 21, B-5000 Namur, Belgium
E-mail: {adrien.bibal,benoit.frenay}@unamur.be *Present address:* of F. Author

M. Lognoul · A. de Streel
CRIDS - Faculty of Law - NADI
University of Namur
Rempart de la Vierge 5, B-5000 Namur, Belgium
E-mail: {michael.lognoul,alexandre.destreel}@unamur.be

often been used as synonyms of each other (Bibal and Frénay, 2016). Nowadays, the two terms are beginning to have different meanings (Guidotti et al., 2018), with interpretability describing the fact that the model is understandable by its nature (e.g. decision trees) and explainability corresponding to the capacity of a black-box model to be explained using external resources (e.g. visualizations).

In law and ethics, the definitions are not precise either. The European Commission notes that: "explainability of the algorithmic decision-making process, adapted to the persons involved, should be provided to the extent possible [...] In addition, explanations of the degree to which an AI system influences and shapes the organizational decision-making process, design choices of the system, as well as the rationale for deploying it, should be available, hence ensuring not just data and system transparency, but also business model transparency" (Communication from the Commission of 8 April 2019, Building Trust in Human-Centric Artificial Intelligence, COM(2019) 168).

This review paper aims at clarifying the meaning of explainability in law and studies how the legal requirements on explainability could be interpreted and applied in machine learning. This means finding how the concept of explainability that is discussed in legal texts can be translated in machine learning solutions. This also means presenting how the machine learning literature implements the technical solutions derived from this translation. Section 2 reviews the main legal requirements on explainability of machine learning models and decisions. Section 3 presents the possible translation of explainability from the legal to the machine learning literature, as well as the machine learning challenges that emerge from the legal requirements. Finally, Section 4 concludes by discussing the conceptual difference between the legal requirements on explainability and their technical implementation and by proposing future directions in machine learning related to these challenges.

2 Legal Requirements on Explainability

Explainability obligations depend on who makes the decisions, and on the degree of automation of the decision-making process. Indeed, requirements are stronger for public authorities than for private firms. They are also stronger when the decision-making process is completely automated (i.e., when no humans are in the loop). As the desired technical outcomes of legal requirements are not always clearly understandable from the legal texts, they need to be clarified on the basis of their objectives. In that perspective, this section analyses in turn explainability obligations that exist in private decision-making (Section 2.1), in public decision-making (Section 2.2), and the reasons for such requirements (Section 2.3).

2.1 Weaker Explainability Requirements in B2C and B2B

While public decision-making, in administration and justice, always needs an explanation (see Section 2.2 below), private decision-making in Business-to-Consumers (B2C) and Business-to-Business (B2B) relationships only needs explanation when a specific law requires it. This section considers two different types of laws that can

impose explanation obligations on private companies, namely horizontal (transversal) and vertical (sectoral) rules. The first ones apply to all sectors of the economy, while the second ones only apply to specific sectors providing more detailed rules to better take into account their characteristics.

2.1.1 Horizontal Rules and Explainability Requirements

The main explainability obligations come from data protection law (in the European Union, the General Data Protection Regulation 2016/679, GDPR). They apply when the decisions (i) involve the processing of personal data, (ii) are based solely on an automated processing of data and (iii) produce legal or significant effects on the recipient of the decision, whatever the field of activity in which those decisions occur. For instance, an automatic refusal of an online credit application is subject to such obligations (art. 22(1) and recital 71 of the GDPR).

In this case, the processors of personal data have the obligation to give certain information to recipients of decisions. One type of information relates to explainability and is defined as "meaningful information about the logic involved, as well as [...] the envisaged consequences of such processing for the data subject" (art. 13(2f) and 14(2g) of the GDPR). This information must be given to the data subjects at the time of the collection of personal data, before any automated decision is made. The same information may also be required by data subjects at any time, before and/or after such a decision is made (art. 15(1h) of the GDPR). In addition, processors of personal data should implement suitable measures in order for recipients of automated decisions to be able to express their point of view and to contest the decision *ex post*, after the decision is made and communicated to its recipient (art. 22(3) of the GDPR).

Those articles of the GDPR do not explicitly require data processors to provide the explanation of decisions made, but the different obligations imposed on processors of personal data can be interpreted as imposing such explanation. This interpretation is confirmed by the recital 71 of the GDPR which provides for the right to obtain an explanation of fully-automated decision in order to be able to challenge the decision. The existence of an explanation requirement in the GDPR is still debated among legal scholars as explainability is only specifically mentioned in a non-binding recital and not in a binding article of law. The majority of scholars support this requirement of explanation (Goodman and Flaxman, 2016; Malgieri and Comandé, 2017; Edwards and Veale, 2018; Selbst and Powles, 2017). They argue that Recital 71 should be used to complement and explain the binding requirements of the Regulation, on the basis of a systemic interpretation of the text (i.e. a type of interpretation of legal texts that focuses on the law as a whole, given its context and objectives). However, a minority of scholars reject the existence of a right to explanation for the following reasons (Wachter et al., 2017). They argue that the European Parliament wanted the requirement for data controllers to explain their automated decisions inside the binding part of the text (Article 22), but this was finally not agreed during the political negotiations leading to the adoption of the GDPR. Hence, they argue, the term explanation was voluntarily placed within the non-binding Recital 71 by the European legislator. Thus, the final interpretation will have to be given by the Court of Justice of the European Union at some point in the future.

The type of explanation to be given by the processors of personal data is not clear either. In their interpretative guidance on the meaningful information to be given, the data protection authorities in Europe note that the processors "should find simple ways to tell the data subject about the rationale behind or the criteria relied on in reaching the decision," but not "a complex explanation of the algorithms used or the disclosure of the full algorithm. The information provided should, however, be sufficiently comprehensive [...] to understand the reasons for the decision" (Guidelines of the European Data Protection Board of 3 October 2017 on Automated individual decision-making and Profiling, p. 25). This interpretation leaves uncertainty on the type and content of explanations to be given by data processors, as the "rationale behind the decision" and the "criteria relied upon" are not the same and imply different technical solutions. In addition, the level of detail of the explanation that should be given to data subjects is not specified.

Next to data protection law, explainability obligations in B2C relationships may also derive from consumer protection law. As consumers are in a situation of weakness and lack bargaining power in their relations with businesses, several rules protect them from unfair practices. In the European Union, a reform of consumer protection law, adopted in 2019, imposes on online marketplaces an obligation to provide "the main parameters determining ranking [...] of offers presented to the consumer as result of the search query and the relative importance of those parameters as opposed to other parameters" (new art. 6(a) of Directive 2011/83 on Consumer Rights). The reform clarifies that "parameters determining the ranking mean any general criteria, processes, specific signals incorporated into algorithms or other adjustment or demotion mechanisms used in connection with the ranking" (recital 22 of Directive 2019/2161 on better enforcement and modernization of EU consumer protection rules).

In parallel, the European Union has adopted very similar obligations for online intermediation services and search engines to the benefit of their business users (B2B). Indeed, the business users of such services are in a situation of weakness that can be compared to the one of consumers, in their relations to this type of service providers. Providers of online intermediation services have to "set out in their terms and conditions the main parameters determining ranking and the reasons for the relative importance of those main parameters as opposed to other parameters". Similarly, the providers of online search engines have to "set out the main parameters, which individually or collectively are most significant in determining ranking and the relative importance of those main parameters, by providing an easily and publicly available description, drafted in plain and intelligible language, on the online search engines of those providers" (art. 5 of Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services). The Regulation clarifies that "the notion of main parameter should be understood to refer to any general criteria, processes, specific signals incorporated into algorithms or other adjustment or demotion mechanisms used in connection with the ranking" (recital 24 of Regulation 2019/1150).

2.1.2 Sectoral Rules and Explainability Requirements

Some legal rules are designed for particular sectors and contain more detailed norms tailored to the needs and characteristics of each sector. For instance, this

is the case for the financial and insurance sectors. Regarding the trading of financial instruments, the investment firm that engages in algorithmic trading should notify it the financial regulator so that the authority "may require the investment firm to provide, on a regular or ad-hoc basis, a description of the nature of its algorithmic trading strategies, details of the trading parameters or limits to which the system is subject, the key compliance and risk controls that it has in place [...] and details of the testing of its systems. The competent authority [...] may, at any time, request further information from an investment firm about its algorithmic trading and the systems used for that trading." Moreover, when an investment firm engages in a high-frequency algorithmic trading technique, it should "store in an approved form accurate and time sequenced records of all its placed orders, including cancellations of orders, executed orders and quotations on trading venues and make them available to the competent authority upon request" (art. 17(2) of the Directive 2014/65 on Markets in financial Instruments).

Regarding the provision of insurance services to consumers, the Belgian law states that insurance providers must inform their subscribers, in an individual and understandable way, of the segmentation criteria used to determine a tariff and the extent of the guarantee. The insurers also have to inform their customers of the criteria that might have an impact on the future of the insurance policy. Furthermore, in the case of a proposal for a modification of the tariff or of the extent of the guarantee, due to a modification of the risk that an insured person represents, the insurer has to motivate his proposal on the basis of the data and criteria used to assess the modification of the risk (art. 46 of the Belgian law of 4 April 2014 on insurances).

2.2 Stronger Explainability Requirements in G2C

When decisions are adopted by public authorities such as administrations and judges in Government-to-Citizens relationships (G2C), providing explanations on those decisions is always compulsory, and the legal obligations for explainability are stronger than in B2C. In law, this type of requirement is called 'motivation'. Among public authorities, the obligations are stronger for judges than for administrations. This subsection analyses the requirements of motivation for administrative decisions and, then, for judicial decisions.

2.2.1 Administrative Decisions and Explainability Requirements

Administrative decisions must comply with a principle of formal motivation (Wiener, 1969), requiring that all factual and legal grounds on which the decision is based should be mentioned and explained. The motivation has to be clear, precise and reflect the real motives behind a decision (e.g. the Belgian law of 29 July 1991 on the formal motivation of administrative decisions). This requirement is imposed at the European Union level by the Charter of Fundamental Rights, which states that "every person has the right to have his or her affairs handled impartially, fairly and within a reasonable time by the institutions, bodies, offices and agencies of the Union. This right includes: [...] the obligation of the administration to give reasons for its decisions"(art. 41 of the Charter of Fundamental Rights of the European Union).

The intensity of the motivation depends on the level of discretionary power enjoyed by the administrative authority (Autin, 2011). If an administrative decision is made on the basis of objective conditions, the required motivation is weaker, so that the administration only has to explain in its decision that the conditions required by the applicable legal text are fulfilled. An example could be the award of a university degree. If all the credits of the curriculum are passed by a student, the university can limit its motivation to that finding to give the degree. When administrative bodies have more discretionary power, they have to motivate more their choices and legal reasoning. For example, staff selection requires more precise and specific motivation. Another example of more extensive motivation requirements for administrative decisions could be the award of contract after a public tender. Among the various proposals submitted by applicants, the administrative authority has to choose one, and explain precisely why it chooses that one over another one. In this regard, European law provides that public contracting authorities should inform each candidate and tenderer of decisions reached concerning the conclusion of the public procurement including the grounds for any decision. On request from the candidate or tenderer concerned, the contracting authority should "inform: (a) any unsuccessful candidate of the reasons for the rejection of its request to participate, (b) any unsuccessful tenderer of the reasons for the rejection of its tender, [...] (c) any tenderer that has made an admissible tender of the characteristics and relative advantages of the tender selected as well as the name of the successful tenderer [...], (d) any tenderer that has made an admissible tender of the conduct and progress of negotiations and dialogue with tenderers" (art. 55 of the Directive 2014/24 on public procurement).

When the administrative decision-making process is automated, additional explainability requirements may apply. One of the most comprehensive set of rules is in the French law which provides that "the administration gives to the person subject to the individual decision adopted on the basis of an algorithmic process, upon request of such person, in a intelligible manner and without prejudice of any trade secret protected by law, the following information: (1) the degree and the manner to which the algorithmic process contributed to the decision-making, (2) the data processed and their sources, (3) the parameters used for the process and, where appropriate, their weighting, applied to the individual case, (4) the operations carried out by the processing" (art. R. 311-3-1-2 of the French Code on the relationships between the public and the administration).

An example of such an automated administrative decision-making process is the French software Parcoursup that determines which studies students should start, on the basis of their background, results in high school, available places in the chosen fields of studies, etc. When this software produces outputs for students, the French Code on the relationships between the public and the administration explained above should apply in principle. However, there is a specific derogation for Parcoursup, in order to protect the secrecy of the deliberations of the selecting teams. This derogation limits the information to be given to recipients of the decisions to the administrative documents used to make the decision, and forbids the disclosure of the weighting of parameters used to make the decisions, as well as the disclosure of the operations carried out by the processing (art. L. 612-3 of the French Code on education).

2.2.2 Judicial Decisions and Explainability Requirements

Judicial decisions must also comply with the principle of motivation. This obligation is imposed by several laws, in particular the European Convention on Human Rights. The European Court of Human Rights decided in various cases that: "in accordance with Article 6(1) of the Convention, judgments of courts and tribunals should adequately state the reasons on which they are based" (Cases Salov v. Ukraine, request n° 65518/01, 6 September 2005, 89; Boldea v. Romania, request n° 19997/02, 15 February 2007, 23; Gradinar v. Moldova, request n° 7170/02, 8 April 2008, 107). In addition, the European countries have similar obligations in their Constitutions (e.g. in Belgium, art. 149 of the Constitution, and art. 79 of the Code on judicial proceedings).

The judicial motivation requirement is more stringent than the one applicable to administrative decisions (Alonso, 2012). Judges have to explain all the factual and legal grounds on which their decisions are based, but they also have to answer all the arguments made by the parties during the trial. As judges need to interpret and apply the relevant laws to given cases, they need to strongly motivate how they make a specific legal decision, and why they retain the various arguments of the parties supporting their claims. However, the level of detail required for the answers of judges to the arguments of the parties is dependent on the circumstances of the case (European Court of Human Rights, Garcia Ruiz v. Spain, request n°30544/96, 21 January 1999, 26). If a judgment is produced by machine learning tools, the same rules apply in relation to the motivation of the judgment, as these rules do not focus on whom (i.e. a judge or a machine) makes the decision but only on the fact that a judgment is made.

2.3 Why Legal Requirements on Explainability?

Previous sections showed that European and national laws already contain several obligations on explainability and, given the ethical importance of the issue, those rules may be strengthened in the future (Commission White Paper on AI, COM(2020)65, p. 20). Some rules apply generally, to all types of decision-making, while other rules, often stricter, apply specifically to automated decision-making. Stricter rules apply to automated decision because, as precised in the Commission White Paper on AI (p. 11), errors and biases may have much larger effects in AI decision making than in human decision making. Moreover, it seems that many humans trust less AI systems than other humans. Both types of rules are often general and imprecise. This means that clarifications will have to be given by the enforcers of the rules, and ultimately by the judges, in case of conflict on the meaning and implications of a particular explainability obligation. To decide on the interpretation of an unspecified legal rule, enforcers and judges rely on legal texts, but also on the goals pursued by the rules. Legal obligations on explainability pursue in general two main objectives. The first one benefits the recipients of the decisions, while the second one benefits the public enforcers or the judges.

The first objective of explainability rules is to allow the recipients of a decision to understand its rationale and to act accordingly (Alonso, 2012). Indeed, it is very difficult, if not impossible, to react to a decision when the reasoning and process that led to the outcome are unknown. In B2C or in B2B relationships,

customers can act by changing providers and/or by contesting the decision before a Court when they think the decision is based on illegal grounds. For instance, if a customer seeks credit from her bank and such credit is denied, the applicant needs to receive meaningful explanation of the denial (e.g. the income is not sufficient). On that basis, the customer can decide to go to another bank relying on other (and more favourable) criteria or to contest the negative decision before courts if it was based on prohibited selection criteria (such as race or gender in some cases). In G2C relationships, the recipient of the decision cannot "vote with their feet" and change administration or judge when dissatisfied with the criteria used by the public authority, but can always contest the legality of the decision before a superior judge.

The second objective of explainability is to allow the public authority, before which a private or a public decision is contested, to exercise a meaningful effective control on the legality of the decision (Commission White Paper on AI, p. 14). Going back to the previous example of credit denial, the judge has to know the criteria on which the refusal was based to determine whether prohibited criteria were used to refuse the credit. In addition, even if a specific decision is not contested, more transparency and explainability increase the incentives of decision makers not to rely on illegal criteria as it would be more difficult (but not impossible) for them to hide the use of such illegal criteria, and hence easier to condemn them if they were using them. Reflecting the traditional view that "sunlight is the best disinfectant", transparency and explainability increase the effectiveness of the whole legal system by facilitating the identification of its violation.

3 Impact of the Legal Requirements on ML Explainability

The legal requirements on explainability explained in Section 2 raise several challenges in machine learning at varying degrees. This Section shows how the legal requirements on explainability can be expressed in machine learning terms. It also shows how difficult it may be to comply with these legal requirements.

In order to introduce the technical understanding of the requirements imposed by the law, Section 3.1 presents some background on machine learning as well as some technical vocabulary. Section 3.2 proposes a technical interpretation of legal requirements in B2C and B2B. Those requirements relate to the weaker requirements of Section 2.1. Finally, some technical solutions that can be provided to the stronger requirements on explainability that are encountered in administrative and judicial decisions (G2C) of Section 2.2 are presented in Section 3.3.

3.1 Background on Machine Learning

This section introduces some background (and associated terms) needed to understand the impact of legal explainability on machine learning. Fig. 1 presents a typical machine learning pipeline. As this paper is concerned with decisions, the pipeline is focused on supervised learning. It starts with data (Fig. 1(1)), also called a *dataset*) that are generally gathered by experts. These data contains two parts: (i) the targets to predict, which can be a continuous variable (e.g. the amount of a fine to be paid) or a categorical variable (e.g. guilty or not), and (ii) a

set of instances (e.g. persons) characterized by features. The data are provided to a training algorithm (Fig. 1(2)) that optimizes the mathematical parameters of a model (i.e. the mathematical expression that is learned to make decisions, see Fig. 1(3)) given the data at hand. When the model is trained, it can be used with instances that have not been used for the training phase (called *unseen* instances) to predict the unknown target value of these instances (Fig. 1(4)). When a set of predictions have been made, performance measures are run on the result to assess the quality of the model (Fig. 1(5)). In the context of category prediction (a task called *classification*), a typical performance measure is the accuracy, which corresponds to the amount of correct predictions over all predictions that have been made by the model.

Despite the fact that regulating any module of the model production process affects the learned model, the notion of explainability studied here relates to explanations that can be provided on the model and its decisions. More precisely, two kinds of models can be described: interpretable models and black-box models. Interpretable models are models that are understandable either because they have a simple mathematical expression (e.g. linear models) or because their representation allows users to understand their mathematical expression (e.g. decision trees). On the contrary, black-box models are models with a complex mathematical expression that, moreover, do not possess a representation that can ease their understanding (Bibal and Frénay, 2016). In the context of black-box models, which are not interpretable by definition, the way to improve understanding is through explanations. Explainability is therefore the capacity of a model to be explainable by using methods that are external to the black-box model (e.g. visualizations, approximating it with interpretable models, etc.) (Guidotti et al., 2018; Mittelstadt et al., 2019).

Furthermore, three hierarchical elements of the model can be focused by legal requirements. Fig. 2 shows these three views with a schematic decision tree as example. Note that the hierarchy presented in this paper follows the legal requirements. Indeed, other hierarchy of model explanation can be proposed (e.g. see Lepri et al. (2018)). The first view of the model that we present is the whole model, which is, in the example of Fig. 2, the complete decision tree (see Fig. 2(1)). When using this kind of model to reach a decision, a first question Q_1 is asked. If the answer is yes (or true), the question Q_2 is asked, and Q_3 otherwise. This process continues until the end of the tree (also called a *leaf*) is reached, where a decision D_i is taken. The second view of the model that can be targeted by requirements is a particular decision made by the model (see Fig. 2(2)). Finally, the features that are involved in a particular decision can also be the focus of legal requirements (see Fig. 2(3)). In that case, it is not asked how the features are combined to make the decision, but only to provide the list of features that are used to make a decision.

The different ways legal requirements on explainability in B2C and B2B decisions can be considered in machine learning are presented in Section 3.2. The technical solutions to the stronger legal requirements in G2C are discussed in Section 3.3.

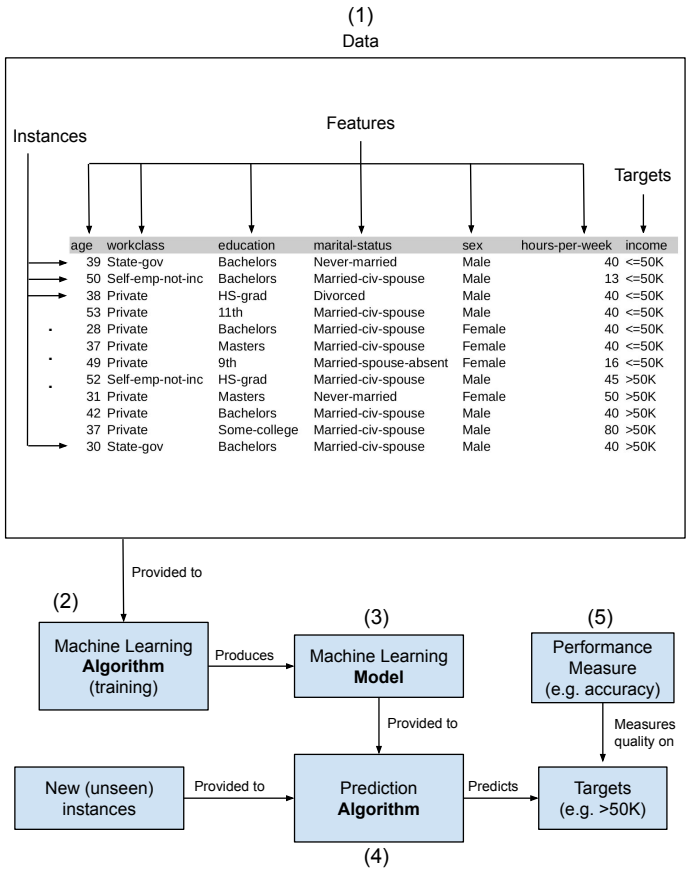


Fig. 1 The user classically provides structured data in a tabular format (1). The columns of the data table correspond to features of the instances in the rows. In this example based on the Adult dataset (Dua and Graff, 2017), the instances are people that are characterized by socio-demographic features. The target is a special column containing what should be predicted (whether a particular person earns more or less than $50K/year, in this example). The data is provided to a training algorithm (2) that will learn a model (3). When the model is learned, it can be used to make predictions on instances that have not been used for training (called unseen instances) (4). Performance measures (such as the accuracy of the predictions) are then computed to evaluate the performance of the model (5).

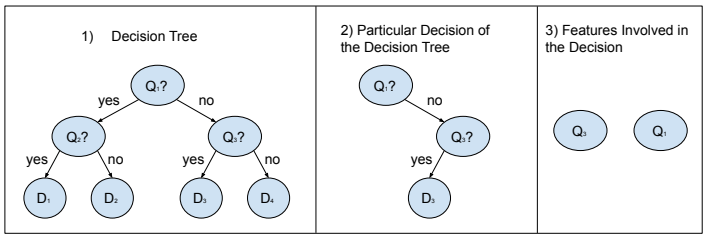


Fig. 2 Weaker requirements can focus on three views of a model: (1) the whole model, (2) a particular decision of the model (here when Q_1 = no and Q_3 = yes), or (3) the features involved in a particular decision (e.g. the age of person (X_1) and the salary (X_3) if they are used in questions such that "is his age lower than 18?" (Q_1) and "is his annual salary lower than 50k/year?" (Q_3)).

3.2 Weaker Requirements: Different Explainability Levels

As shown in Section 2, there is no unique definition of explainability in law. Some explainability requirements relate to the model while others relate to the decision (Wachter et al., 2017; Selbst and Barocas, 2018). In addition, some explainability requirements merely relate to the features used by the model to adopt the decision, while others go further and relate to the way the features are combined to make the decision. Furthermore, there can be technical ambiguities regarding legal texts and their interpretation. For instance, the interpretative guidelines on the GDPR by the data protection authorities refer to the "rationale behind" or the "criteria relied on in reaching the decision" (Guidelines on Automated individual decision-making and Profiling, p. 25), which correspond to two technically different requirements.

Therefore, this section analyzes in turn the technical understanding of four levels of legal requirements: providing the main features that are used in the model or in a decision (Section 3.2.1), providing all features that are used in a particular decision (Section 3.2.2), providing the feature combination that is used to make a decision (Section 3.2.3) and providing an interpretable model (Section 3.2.4). The inputs and outputs of the learning process constrained by the weaker requirement in B2C and B2B are presented in Fig. 3 and the way directive and regulation examples are technically interpreted is summed up in Table 1.

3.2.1 Requirements on the Main Features

From a machine learning point of view, the legal texts in B2C and B2B (see Section 2.1) refer to four levels of requirement. The first and weakest requirement asks to provide the "main parameters" of the model, or used by the model to take a particular decision. "Parameters" in those legal texts correspond to features of instances in machine learning (see Fig. 1 for a background on machine learning). The methodology used to make the distinction between the main features and the other features is not described in the legal texts. Many machine learning models make it possible to extract the main features they use, even black-box models. The new art. 6(a) of Directive 2011/83 on Consumer Rights states that the "main

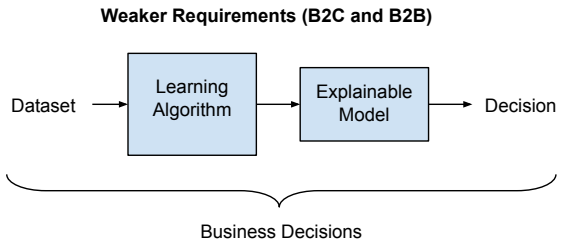


Fig. 3 Input and output of the learning process with legal requirements on explainability in B2C and B2B.

Main features • Directive 2011/83 on Consumer Rights, art. 6(a): obligation to provide "the main parameters" and "the relative importance of those parameters" • Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services, art. 5: obligation to provide "the main parameters" and "the relative importance of those parameters"
All features • Guidelines on Automated individual decision-making and Profiling: obligation to provide "the criteria relied on in reaching the decision" • Belgian law of 4 April 2014 on insurances, art. 46: obligation to provide "the segmentation criteria"
Combination of features Guidelines on Automated individual decision-making and Profiling: obligation to provide "the rationale behind the decision"
Whole model Directive 2014/65 on Markets in Financial Instruments, art. 17: obligation to provide "information [...] about its algorithmic trading and the systems used for that trading"

Table 1 Summary of the legal texts used as examples in Section 3.2.

default parameters" should be provided, without the obligation to instantiate for a particular decision. This means that the main features used by the entire model, not for a particular decision, should be provided.

Providing the main features used in a model is well-developed in the machine learning literature. For linear models, such as linear regression models, a kind of interpretable (or transparent) model, one only has to look at the weights that have been learned for determining the main features that are used. Indeed, given d features $\mathbf{f}_1$, $\mathbf{f}_2$, ..., $\mathbf{f}_d$ for predicting a target $\mathbf{t}$, the goal of the linear model training algorithm is to find the weights w_1, w_2, ..., w_d, such that the linear combination $(w_1 * \mathbf{f}_1) + (w_2 * \mathbf{f}_2) + ... + (w_d * \mathbf{f}_d)$ best predicts $\mathbf{t}$. If the d features are transformed in order to be in the same scale (a transformation called *scaling*), sorting the absolute value of the computed weights provide a ranking of the feature importance in the model. In particular, a weight w_j of zero means that the feature $\mathbf{f}_j$ is not used. For instance, if $\mathbf{t}$ correspond to house prices to predict, $|w_{number\ of\ rooms}| = 5$ and $|w_{house\ age}| = 2.5$ mean that the feature "number of rooms" is twice as important

as the feature "house age" when predicting house prices. Some works go further and try to determine the features with a non-zero weight that are particularly relevant in a given linear model (e.g. Yu and Liu (2004); Frénay et al. (2014)). In that context, a feature is considered strongly relevant if, by removing it, the performance of the model drops. Some features can also be characterized as weakly relevant if they bring new information, but only if other features are removed (John et al., 1994; Kohavi and John, 1997; Frénay et al., 2014). These techniques for studying feature relevance in models such as linear models are important because such simple models are widely used in academia, as well as in industry.

In the case of black-box models, features may also be sorted by importance. For instance, random forests (Breiman, 2001) use an out-of-bag error during the learning of the decision tree ensemble that can be used to rank features by importance. More precisely, when learning the decision trees in the forest, different sub-sets of training instances are used for learning each tree. The out-of-bag error is defined as the average error made in the prediction of each instance x_i for each decision tree that has not been trained using this x_i (Breiman, 2001). In order to know what are the important features, values of each feature $\mathbf{f}_j$ are perturbed (i.e. the values of the feature are changed randomly) and the out-of-bag error is computed. A feature $\mathbf{f}_j$ is considered important if the perturbation of its values increases the out-of-bag error, with respect to the out-of-bag error computed without the perturbation. This idea of perturbing feature values to assess the importance of each feature used in a model can in fact be extrapolated and applied to any machine learning algorithm (Fisher et al., 2018). In the case of computer vision, where images are used as input, it is less relevant to extract the main features (i.e pixels) used by a model. Instead, one is rather interested in the internal representation used by the model (the extracted features in hidden layers of neural networks). However, extracting the internal features of neural networks is a challenging problem in the machine learning literature. For deep convolutional neural networks, techniques such as saliency maps (Simonyan et al., 2013) and Grad-CAM (Selvaraju et al., 2017) can be used, yet they only extract the main features for specific decisions.

3.2.2 Requirements on all Features

The second requirement level is to provide all the features used to take a particular decision. For instance, this is the case of the obligations arising from the GDPR as interpreted by the data protection authorities, under the terms "the criteria relied on in reaching the decision" (Guidelines on Automated individual decision-making and Profiling, p. 25). This means providing all features with a non-zero coefficient in a linear model, or the features in a specific decision path of a decision tree, without necessarily providing the whole tree. Providing all features involved in a particular decision may be motivated by the need to verify the absence of features (or proxies) that are forbidden to use by law (e.g. those that illegally discriminate people).

While it is still possible for all models (using perturbation, for instance, in the case of black-box models) to produce such list of features used, the issue lies in the size of the list. Indeed, providing all features with a non-zero coefficient in a linear model is straightforward, but processing thousands of them as a human is difficult. In order to avoid this issue, some machine learning algorithms incorporate a trade-off between the model accuracy and its complexity. For instance, a technique called

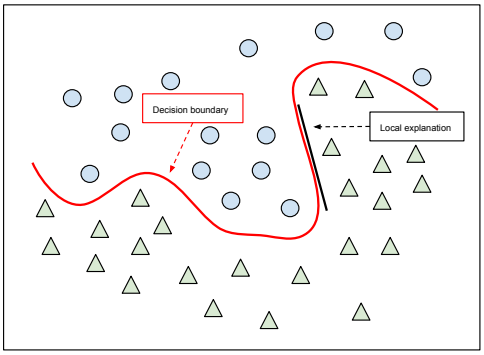


Fig. 4 Figure inspired by Ribeiro et al. (2016). Local explanation of a complex decision boundary (i.e. separating circles and triangles) by using a linear model. The linear model is easy to understand (using the relative value of the weights), but only provides an explanation on the complex decision boundary locally, where it is used.

Lasso makes it possible to set as many weights w_j as possible to zero when learning a linear model (effect called *sparsity*), while keeping a good enough predictive accuracy (Tibshirani, 1996). This makes the resultant linear models much easier to understand. In practice, the balance between accuracy and complexity of the model has to be tuned by the user, depending on his needs. If no means to control the model complexity are provided in the learning algorithm, the problem slides from the machine learning side to the information visualization side, where questions about how to efficiently present information to users is the core issue.

3.2.3 Requirements on the Combination of Features in a Decision

The third requirement level is about the complete explanation of a decision. For instance, this is the case of the obligations arising from the GDPR as interpreted by the data protection authorities, under the terms "the rationale behind the decision" (Guidelines on Automated individual decision-making and Profiling, p. 25). This means providing not only the features used in a decision, but also their combination used to make the particular prediction.

As developed in the literature on interpretability and explainability, this requires to use transparent models such as decision trees or linear models, to create new ones (e.g. supersparse linear integer models (SLIM) (Ustun et al., 2013a,b; Ustun and Rudin, 2016)) or to create ways to explain black-box models (e.g. local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016)). One typical example of interpretable models are the sparse linear models. Most of the time, sparsity in linear models is achieved using Lasso through the ℓ_1-norm (minimize the sum of all weight absolute values, $\sum_i |w_i|$), which is an approximation of the difficult-to-optimize ℓ_0-norm (minimize the number of non-zero weights). SLIM are models that optimize the ℓ_0-norm by transforming the problem. Indeed,

instead of optimizing weights w_j with real values, those weights can now take values among a finite set of integer values. Thanks to that transformation, SLIM are more interpretable than classical models by being sparser and by using only integers (instead of reals) as weights, while obtaining similar accuracy scores.

In the case of black-box models, model-agnostic ways to understand those models can be considered. LIME is a technique used to understand specific decisions of a black-box model through the use of an interpretable model. For instance, a specific decision on an instance i made by a black-box neural network can be understood by approximating the decision through local decisions (i.e. decisions that are made on instances similar to the instance i, see Fig. 4). This local model, explaining local decisions, should be interpretable. This local model, that can be a linear model, does not globally explain the black box, but instead provides clues on why a specific decision has been taken by the black-box model. This can be compared to the explanation of a particular path in a decision tree: the explanation of the path is local and does not globally explain the whole tree.

3.2.4 Requirements on the Whole Model

A global understanding of the model would be the maximal explainability requirement that can be asked for a model. Indeed, as a total understanding of the model is required, local explanations cannot suffice. In that case, the legal requirement would constrain the possible usable models to interpretable ones. This kind of requirement exists in the case of financial algorithms, as in addition to provide "a description of the nature of its algorithmic trading strategies, details of the trading parameters or limits to which the system is subject, the key compliance and risk controls that it has in place [...] and details of the testing of its systems," investment firms can be asked to provide information "about its algorithmic trading and the systems used for that trading" (art. 17(2) of the Directive 2014/65 on Markets in Financial Instruments).

Moreover, in machine learning, Rudin (2019) argues for the need to use interpretable models (such as linear or rule-based models (Guidotti et al., 2018)), instead of explaining black boxes, in the case of high-stake decisions. This is justified by the fact that the drop in accuracy caused by the choice of an interpretable model can be marginal, for the benefit of having a model that can be understood and trusted by its users.

Section 3.2 presented a translation in vocabulary from the legal literature to the machine literature through four requirement levels related to the weak (B2C and B2B) legal requirements. The four levels were (i) providing the main features used in a decision or the model, (ii) providing all features processed by the model, (iii) providing a comprehensive explanation of a specific decision taken by the model and (iv) providing an interpretable model. The next section focuses on the stronger (G2C) legal requirements and the new machine learning problems that emerge.

3.3 Stronger Requirements: New Machine Learning Problems

In addition to the different levels of explanation described in Section 3.2, legal motivations need to be given when administrative decisions are made by the model.

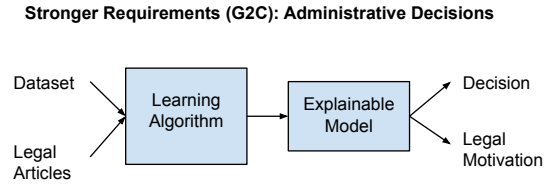


Fig. 5 Inputs/outputs of the learning process for administrative decisions

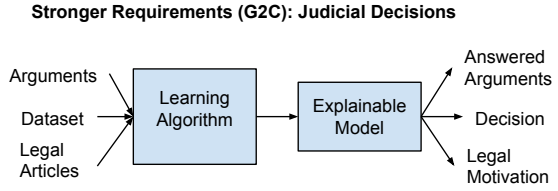


Fig. 6 Inputs/outputs of the learning process for judicial decisions

In the case of administrative decisions, models are now required to provide the law articles behind each decision. This may require to learn the link between decisions and legal rules supporting these decisions (see Fig. 5). For instance, decision trees would have to output the legal bases supporting the paths leading to decisions.

On top of this, according to requirements applicable to judicial decision making, judicial decisions need to respond to the arguments submitted by the parties. In this context, a situation described by facts is provided as input to the model, along with textual arguments from both opposing parties. The model has then to output the decision, supported by legal articles as for administrative decisions, while at the same time answering all arguments (see Fig. 6). This means that the provided arguments have to be considered by the model, such that the processed arguments logically support the decision.

Note that the interpretability/explainability problem is re-framed in the context of administrative and judicial decisions. Indeed, in addition to the need for interpretable/explainable models, the stronger requirements ask for the processing of heterogeneous data (i.e., different types of data) for producing not a single output (the decision) but two (decision and legal articles related to the legal motivation) or three (decision, articles and arguments supporting the decision) outputs. This section presents examples of how the machine learning literature tackles the automated judicial decision by only considering the factual description (Section 3.3.1), the facts and legal articles (Section 3.3.2) and all three possible data elements, i.e. the facts, legal articles and arguments (Section 3.3.3).

3.3.1 Explaining Judicial Decisions with Facts Only

In the AI and law literature, explainability has not always been linked to the necessity to provide legal motivation and to answer arguments. For instance, Ashley and Brüninghaus (2009) extract facts (called Factors) from case texts in order to predict the decision on the case. In order to do that, the authors derive issues related to the extracted facts by using a domain model. Such domain models are trees defining how facts must be combined to define an issue (such as "trade secret misappropriation"). Given the extracted issues and facts, an algorithm called IBP (i) predicts whether the plaintiff or the defendant is favored and (ii) only provides an explanation of how these facts and issues are used to make the prediction. This kind of explainability is similar to the one discussed in Section 3.2.

3.3.2 Explaining Judicial Decisions with Facts and Legal Articles

A more problem-oriented way to see the stronger requirements of explainability is through multi-task learning. Multi-task learning is a way to learn a global model by splitting the learning into smaller tasks to learn (Zhong et al., 2018). This results in a set of small tasks that is easier to learn than learning solely the global task. Luo et al. (2017) propose to use a neural network (with a mechanism called *attention*) to predict the charges in criminal cases while also providing legal articles supporting the decision. The neural network is defined in such a way that it solves two tasks: charge prediction and relevant legal articles extraction. Following the same idea, Zhong et al. (2018) define the sub-tasks as learning (i) the applicable legal articles, (ii) the charges and (iii) the terms of penalty of a legal judgment, based on a textual fact description. In other words, from the fact description as sole input, multiple output are provided, such as the decision and the relevant articles supporting the decision. One should note that high-performing models used for multi-task learning are often not interpretable. For instance, in the work of Luo et al. (2017) and Zhong et al. (2018), black-box deep neural networks are used to solve the different tasks.

With the objective of making the automated judicial decisions interpretable, Li et al. (2018) use a Markov logic network (MLN) (Singla and Domingos, 2005) to predict the outcome of divorce judgments. Their algorithm first extracts logical rules, among other preprocessing steps, from case texts. Then, these rules are weighted and ordered in the MLN such that following the network of rules makes it possible to predict a case outcome. This model is interpretable as humans can understand how the decisions are made.

3.3.3 Explaining Judicial Decisions with Facts, Legal Articles and Arguments

Most of the time, machine learning techniques in the literature do not consider the parties arguments. Despite that, one should note that argument mining and generation is an ongoing work in the literature (Branting (2017), e.g. Palau and Moens (2009)). One next step could therefore be to design argument mining and generation as two new sub-tasks in a multi-task framework.

In the context of the European Court of Human Rights, Aletras et al. (2016) consider the three elements of interest (description of the situation, the applicable laws and the arguments of the parties) as a text in order to predict if a given article

of the European Convention of Human Rights has been violated. They use n-grams on the whole text to train a SVM with a linear kernel in order for their model to be interpretable. By using an interpretable model, they are able to provide how the elements of the case contributed to the decision. As these elements are legal articles and arguments, they can provide clues on how these articles and arguments were used (through the use of certain n-grams by the model) to make the decision. Ye et al. (2018) consider the generation of court views as a sequence to sequence (Seq2Seq) problem, where a fact description and the charges (which correspond to the decision of another model) are provided as input, and a corresponding court view corresponding to the rationale is generated. The Seq2Seq problem is commonly seen in language translation. In that context, a first sequence of words in a certain language is provided to the machine learning model and a second sequence of words corresponding to the first sequence but in another language is produced. Court view generation can therefore be seen as a machine translation problem, where the court view would be a "translation" of what can be read in a fact description.

4 Conclusion, Discussion and Research Directions

This paper presents how the law constrains machine learning models regarding their interpretability and explainability. The vocabulary used in law is not always determined, nor consistent in its strength. The constraints on explainability, in their weakest form, can be formulated in a four-level fashion: (i) providing the main features used to make a decision, (ii) providing all the processed features, (iii) providing a comprehensive explanation of the decision and (iv) providing an understandable representation of the whole model.

In the case of requirements related to administrative and judicial decisions, most of the work focuses on interpretable/explainable models, models that provide legal articles supporting their decisions, or both. However, models that provide answers to the arguments of the parties, alongside the decision, are not well studied in the machine learning literature. One clear direction, though, is the use of natural language processing (NLP) to solve the problem, as fact descriptions, legal articles and arguments are often in a text format. Even the explanation of a model's decision can be considered as an NLP problem through, e.g. Seq2Seq learning (Ye et al., 2018).

Note, that in Ye et al. (2018), the explainability of a model judicial decision is provided by the text generated by the Seq2Seq model. However, the Seq2Seq model, which is a deep neural network model, is not itself interpretable. Two different views on explainability are therefore to be put forward.

In the first view, the machine learning point of view, interpretability and explainability are defined on the abstract mathematical model that is used to make the decision (Bibal and Frénay, 2016). For instance, decision tree models are considered interpretable because their tree representation makes it easier for humans to understand the abstract mathematical model behind it. By following paths in the tree, users follow a mathematical formula, although in an easier way.

In the second view, that rather corresponds to the legal point of view, explainability can be defined as meaningful insights on how a particular decision is made.

In that second view, it is not necessarily required to provide an interpretable representation of a mathematical model, but most importantly to provide a train of thought that can make the decision meaningful for a user (i.e. so that the decision makes sense to him).

This distinction is crucial for drawing the future directions of administrative and judicial decisions made by machine learning models. Indeed, the problem is framed differently for machine learning researchers. In the first view, interpretable/explainable models are used to understand the mathematical processes behind decisions. This requires to develop interpretable models or to make it possible to explain black-box models (as developed in Section 3.2). In the second view, providing the human interpreter with an explanation of the decision that makes sense to him is the main objective, even if the output is not an explanation of the mathematics behind the decision as such. This is the point of view adopted by the Seq2Seq solution, and seems to be the explainability requirement wanted in law.

Following this analysis, we call for a close inter-disciplinary dialogue between the legal and machine learning communities in order, on the one hand, to specify the undetermined terms of the law in the light of their objectives and, on the other hand, to develop new techniques allowing machine learning models to comply with the different level of explainability required by law. Furthermore, this exchange may also help machine learning researchers to more clearly define (and solve) the new problems related to the strongest legal requirements.

References

N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016.

C. Alonso. La motivation des décisions juridictionnelles : exigence(s) du droit au procès équitable. In *Regards sur le droit au procès équitable*. Presses de l'Université Toulouse 1 Capitole, 2012.

K. D. Ashley and S. Brüninghaus. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2):125–165, 2009.

J.-L. Autin. La motivation des actes administratifs unilatéraux, entre tradition nationale et évolution des droits européens. *Revue française d'administration publique*, 1:85–99, 2011.

A. Bibal and B. Frénay. Interpretability of machine learning models and representations: an introduction. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 77–82, Bruges, Belgium, 2016.

L. K. Branting. Data-centric and logic-based models for automated legal problem solving. *Artificial Intelligence and Law*, 25(1):5–27, 2017.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

F. Doshi-Velez and M. Kortz. Accountability of AI under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*, 2017.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

L. Edwards and M. Veale. Enslaving the algorithm: From a "right to an explanation" to a "right to better decisions"? *IEEE Security & Privacy*, 16(3):46–54, 2018.

A. Fisher, C. Rudin, and F. Dominici. All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. *arXiv preprint arXiv:1801.01489*, 2018.

B. Frénay, D. Hofmann, A. Schulz, M. Biehl, and B. Hammer. Valid interpretation of feature relevance for linear data mappings. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 149–156, 2014.

B. Goodman and S. Flaxman. EU regulations on algorithmic decision-making and a "right to explanation". In *ICML Workshop on Human Interpretability in Machine Learning*, New York, USA, 2016.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):1–42, 2018.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning (ICML)*, pages 121–129, 1994.

Y. Kodratoff. The comprehensibility manifesto. *AI Communications*, 7(2):83–85, 1994.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

B. Lepri, N. Oliver, E. Letouzé, A. Pentland, and P. Vinck. Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology*, 31 (4):611–627, 2018.

J. Li, G. Zhang, L. Yu, and T. Meng. Research and design on cognitive computing framework for predicting judicial decisions. *Journal of Signal Processing Systems*, pages 1–9, 2018.

Z. C. Lipton. The mythos of model interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*, New York, USA, 2016.

B. Luo, Y. Feng, J. Xu, X. Zhang, and D. Zhao. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2727–2736, 2017.

G. Malgieri and G. Comandé. Why a right to legibility of automated decision-making exists in the general data protection regulation. *International Data Privacy Law*, 7(4):243–265, 2017.

B. Mittelstadt, C. Russell, and S. Wachter. Explaining explanations in AI. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT)*, pages 279–288, 2019.

R. M. Palau and M.-F. Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL)*, pages 98–107, 2009.

F. Pasquale. *Black Box Society. The Secret Algorithms That Control Money and Information*. Harvard University Press, 2015.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD*, pages 1135–1144, 2016.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1

(5):206–2015, 2019.

A. D. Selbst and S. Barocas. The intuitive appeal of explainable machines. *Fordham Law Review*, 87:1085–1139, 2018.

A. D. Selbst and J. Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, 2017.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

P. Singla and P. Domingos. Discriminative training of markov logic networks. In *National Conference on Artificial Intelligence (AAAI)*, pages 868–873, 2005.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

B. Ustun and C. Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

B. Ustun, S. Traca, and C. Rudin. Supersparse linear integer models for interpretable classification. *arXiv preprint arXiv:1306.6677*, 2013a.

B. Ustun, S. Traca, and C. Rudin. Supersparse linear integer models for predictive scoring systems. In *Proceedings of AAAI Late Breaking Track*, 2013b.

S. Wachter, B. Mittelstadt, and L. Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

C. Wiener. La motivation des décisions administratives en droit comparé. *Revue internationale de droit comparé*, 21(21):779–795, 1969.

H. Ye, X. Jiang, Z. Luo, and W. Chao. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1854–1864, 2018.

L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5(Oct):1205–1224, 2004.

H. Zhong, G. Zhipeng, C. Tu, C. Xiao, Z. Liu, and M. Sun. Legal judgment prediction via topological learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549, 2018.

CHAPTER 14

IMPACT OF LEGAL REQUIREMENTS ON EXPLAINABILITY IN MACHINE LEARNING

The extended abstract presented in this chapter was published at the International Conference on Machine Learning (ICML) workshop on Law & Machine Learning in 2020.

Impact of Legal Requirements on Explainability in Machine Learning

Adrien Bibal [*1] **Michael Lognoul** [*2] **Alexandre de Streel** [2] **Benoît Frénay** [1]

1. Legal Requirements on Explainability

The requirements on explainability imposed by European laws and their implications for machine learning (ML) models are not always clear. In that perspective, our research (Bibal et al., Forthcoming) analyzes explanation obligations imposed for private and public decision-making, and how they can be implemented by machine learning techniques.

For decisions adopted by firms or individuals, we mainly focus on requirements imposed by general European legislation applicable to all the sectors of the economy. The obligations of the General Data Protection Regulation (GDPR) (art. 13-15 and 22) as interpreted by the European Data Protection Board (EDPB) require the processors of personal data to provide "the rationale behind or the criteria relied on in reaching the decision," under certain circumstances, when a fully automated decision is made (EDPB Guidelines of 3 October 2017 on Automated individual decision-making and Profiling, p. 25; see also (Edwards & Veale, 2018; Wachter et al., 2017)). Consumer protection law imposes to online marketplaces to provide their consumers with "the main parameters determining ranking [...] and the relative importance of those parameters" (art. 6(a) of Directive 2011/83). The Online Platforms Regulation imposes very similar obligations to online intermediation services and search engines towards their professional users (art. 5 of Regulation 2019/1150).

Sectoral rules are also analyzed. For instance, financial regulators "may require the investment firm to provide [...] a description of the nature of its algorithmic trading strategies, details of the trading parameters or limits to which the system is subject, the key compliance and risk controls that it has in place [...]. The competent authority [...] may, at any time, request further information from an investment firm about its algorithmic trading and the systems used for that trading" (art. 17(2) of Directive 2014/65 on Markets in Financial Instruments).

[*]Equal contribution [1]PReCISE, Faculty of Computer Science, NADI, University of Namur, Belgium [2]CRIDS, Faculty of Law, NADI, University of Namur, Belgium. Correspondence to: Adrien Bibal <adrien.bibal@unaur.be>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

For decisions adopted by public authorities, two stronger requirements are studied: motivation obligations for administrations and for judges (imposed by European Convention on Human Rights). For administrative decisions, all factual and legal grounds on which the decision is based should be provided. For judicial decisions, judges have in addition to answer the arguments made by the parties in the litigation.

The objectives of those explanation requirements are twofold: first, allowing the recipients of a decision to understand it and act accordingly; second, allowing the public authority, before which a decision is contested, to exercise a meaningful effective control on the legality of the decision (European Commission White Paper of 19 February 2020 on Artificial Intelligence, p. 14).

2. Legal Requirements and Machine Learning

As explained in the previous section, legal texts do not always clearly identify the focus of the requirements. In private decision making, we identified that the explainability of four levels of machine learning entities or concepts are mentioned in legal texts (Bibal et al., Forthcoming): the main features used for a decision, all features used for a decision, how the features are combined for reaching a decision and the whole model (see Table 1).

The first and weaker level of requirements is to provide the main features used for a decision. Note that the main parameters mentioned in the legal texts refer to the features used by a ML model. While the main features used are natively provided by interpretable models such as linear models and decision trees, some works go further and provide weakly and strongly relevant features in linear models (John et al., 1994; Kohavi & John, 1997). In the context of black-box models, the feature importance provided by the out-of-bag error of random forests can pass these requirements, as well as the feature importance provided through the perturbation of input feature values (Fisher et al., 2019).

The second level of requirements is to provide all features involved in a decision. While providing all features used is again natively proposed by interpretable models, this requirement can be difficult to achieve when the number of features used by the model is huge. Sparsity penalties such as Lasso may be necessary to satisfy the requirement.

Main features • Directive 2011/83 on Consumer Rights, art. 6(a): obligation to provide the "main parameters" and their "relative importance" • Regulation 2019/1150 on promoting fairness and transparency for business users of online intermediation services, art. 5: obligation to provide "the main parameters" and "the relative importance of those parameters"
All features • Guidelines on automated individual decision-making and profiling: obligation to provide "the criteria relied on in reaching the decision" • Belgian law of 4 April 2014 on insurances, art. 46: obligation to provide "the segmentation criteria"
Combination of features Guidelines on Automated individual decision-making and Profiling: obligation to provide "the rationale behind the decision"
Whole model Directive 2014/65 on Markets in Financial Instruments, art. 17: obligation to provide "information [...] about its algorithmic trading and the systems used for that trading"

Table 1. Table reproduced from (Bibal et al., Forthcoming) containing the legal texts used as examples in this paper.

The third level of explainability requirements is to provide the combination of features that led to a particular decision. Again, interpretable models make it possible to check how the features have been combined to lead to a decision. In the context of black-box models, techniques like LIME (Ribeiro et al., 2016) have been developed to get insights on how models behave locally, i.e. for a particular decision.

Finally, the strongest requirement is to provide the whole model. In this case of strong requirement, only interpretable models can be used, as, by definition, black-box models cannot be provided (e.g. if the model is non-parametric) or understood (e.g. in the case of neural networks).

In addition to these four levels of explainability requirements for private decisions, requirements for public decisions impose two additional constraints. For administrative decisions, the legal motivation should also be provided with the decision. This means that all factual and legal grounds on which the decision is based must be provided. In the case of judicial decisions, in addition to the facts of the case and the motivation, which was already needed for administrative decisions, answers to the arguments of the parties to the litigation must also be provided. While some works try to tackle these requirements (e.g. (Ashley & Brüninghaus, 2009) explain decisions with facts only; (Zhong et al., 2018) introduce multi-task learning for dealing with legal articles, as well as facts; and (Ye et al., 2018) use sequence-to-sequence learning to propose answers to the arguments of the parties), legal requirements on the explainability of public decisions remain a challenge in machine learning, because ML algorithms are not designed to manipulate factual and legal grounds, as well as arguments, directly.

In conclusion, we call for an interdisciplinary conversation between the legal and AI research communities. In particular, legal scholars could benefit from better understanding the potential and the limitations of ML models and AI scholars from better understanding the objectives and ambiguities of the law.

References

Ashley, K. D. and Brüninghaus, S. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2):125–165, 2009.

Bibal, A., Lognoul, M., de Streel, A., and Frénay, B. Legal requirements on explainability in machine learning. *Artificial Intelligence and Law*, Forthcoming.

Edwards, L. and Veale, M. Enslaving the algorithm: From a "right to an explanation" to a "right to better decisions"? *IEEE Security & Privacy*, 16(3):46–54, 2018.

Fisher, A., Rudin, C., and Dominici, F. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *JMLR*, 20(177):1–81, 2019.

John, G. H., Kohavi, R., and Pfleger, K. Irrelevant features and the subset selection problem. In *Proceedings of ICML*, pp. 121–129, 1994.

Kohavi, R. and John, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of ACM SIGKDD*, pp. 1135–1144, 2016.

Wachter, S., Mittelstadt, B., and Floridi, L. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

Ye, H., Jiang, X., Luo, Z., and Chao, W. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In *Proceedings of NAACL*, pp. 1854–1864, 2018.

Zhong, H., Zhipeng, G., Tu, C., Xiao, C., Liu, Z., and Sun, M. Legal judgment prediction via topological learning. In *Proceedings of EMNLP*, pp. 3540–3549, 2018.

CHAPTER 15

USER EXPERIMENT GUIDELINES FOR MEASURING INTERPRETABILITY IN MACHINE LEARNING

The paper presented in this chapter was published at the Extraction et Gestion des Connaissances (EGC) workshop on Advances in Interpretable Machine Learning and Artificial Intelligence (AIMLAI) in 2019.

User-Based Experiment Guidelines for Measuring Interpretability in Machine Learning

Adrien Bibal, Bruno Dumas, Benoît Frénay

PReCISE - Faculty of Computer Science - NaDI - University of Namur
Rue Grandgagnage 21, 5000 Namur, Belgium
{adrien.bibal, bruno.dumas, benoit.frenay}@unamur.be

Abstract. With the advent of high-performance black-box models, interpretability is becoming a hot topic today in machine learning. While a lot of research is done on interpretability, machine learning researchers do not have precise guidelines for setting up user-based experiments. This paper provides well-established guidelines from the human-computer interaction community.

1 Introduction

Interpretability is a major concern nowadays in machine learning (Bibal and Frénay, 2016; Lipton, 2016). In several applications, such as credit scoring (Martens et al., 2011), machine learning models need to be interpretable in order to be accepted and used. However, despite being a natural way of evaluating interpretability (Doshi-Velez and Kim, 2017), user-based experiments are not widespread in the machine learning literature (Bibal and Frénay, 2016). This may be due to a lack of time or other resources, but also to a lack of guidelines on how to set up such experiments. Inspired by the human-computer interaction (HCI) literature, this paper provides guidelines on what to consider in order to set up user-based experiments.

2 User-Based Experiments on Interpretability in ML

As interpretability is about user comprehensibility of models, it may seem natural that machine learning experiments assessing interpretability involve users. Doshi-Velez and Kim (2017) stress the need to answer several questions when evaluating interpretability. One of the most important questions is how we should set up experiments involving users.

Doshi-Velez and Kim (2017) consider three experimental setups for answering this question. The first experimental setup concerns application-grounded metrics, in which the real task is sought to be evaluated. This kind of setup requires gathering users in order to evaluate the real performance of users on a real task. Second, human-grounded metrics consider experiments in which real task metrics are replaced by simplified tasks for measuring interpretability. For instance, asking users to compare two models may not be the real task, but the comparison makes it possible to get insights on interpretability. Finally, functionally-grounded metrics involve heuristics used to measure interpretability without the need to gather users. These are not user-based experiments, but may be considered when gathering users is too complex or if the resources needed for user-based experiments are not available for the researcher.

Several simplified tasks for the human-grounded metrics are listed by Piltaver et al. (2014a): "classify", "explain", "validate", "discover", "rate" and "compare". For instance, the model interpretability can be measured by asking users to manually classify an instance using the model. This "classify" metric provides an accuracy error representing the agreement between the classification manually made by the user and the one automatically made by the machine using the same model. Another example is "compare", for which two or more models are proposed to users, who are asked to choose the more interpretable among them. The authors evaluated the interpretability of decision trees based on their tasks in (Piltaver et al., 2014b).

Most user-based experiments on interpretability in the machine learning literature can be characterized given the Piltaver's categorization. Allahyari and Lavesson (2011) use a "compare" task for measuring the interpretability of decision trees and rules obtained by various algorithms. Huysmans et al. (2011) use a "classify" task by measuring accuracy, answer time and confidence of users. Other examples can be found in (Poursabzi-Sangdeh et al., 2018).

Despite these works on the classification of user-based experiments and user-based experimental tasks, no precise guidelines are provided to the machine learning researchers for setting up user-based experiments. The following section builds on guidelines established in the human-computer interaction (HCI) community in order to set up such kind of experiments.

3 Guidelines on User-Based Experiments

The guidelines proposed in this paper can be decomposed into three questions: "what do you want to measure" (Section 3.1), "who are your users" (Section 3.2) and "which type of metric can you use" (Section 3.3). Answering these questions may allow machine learning practitioners to better frame how to conduct a user-based experiment.

3.1 What do you Want to Measure?

As outlined in Doshi-Velez and Kim (2017)'s conclusion, it is important to note that "the claim of the research should match the type of the evaluation." This means that the research questions must be clearly stated before establishing the evaluation type.

On the one hand, one may want to get qualitative insights on the overall interpretability of a particular model. In this case, Nielsen and Landauer (1993) demonstrated that even just 5 users can identify 85% of usability problems, including most of the severe problems. The usual approach involves observing and taking notes of how the 5 users manipulate the model during the experiment. This can reveal a large part of the possible answers to questions such as "is the depth of my decision tree important regarding the interpretability", "does the balance of the tree play a role at all", etc.

On the other hand, if something specific, related to interpretability, is to be assessed, then a more specific experiment needs to be set up. First, the research questions must be clearly stated to allow the identification of the real task. Identifying the real task is important for designing an experiment that focuses on this real task (Doshi-Velez and Kim (2017)'s application-grounded metrics) or on the right simplified tasks (Doshi-Velez and Kim (2017)'s human-grounded metrics). Then, as a precise research question needs to be answered, as many users as needed for statistical significance have to be gathered. Finally, after the experiment is over, statistical tools can be used to analyze the results.

3.2 Who are your Users?

Echoing the "what do you want to measure" question, the question "who are your users" needs to be answered. Indeed, the real task is never realized in a vacuum, and users performing the task, in a real setting, have a particular profile. The goal of this question is to identify the user profile related to the task at hand. This identification is mandatory as the pool of users considered for the experiment should match as much as possible the work domain expert profile. This is a point considered by Doshi-Velez and Kim (2017) when they mention the nature of user expertise. Crowdsourcing platforms, such as Amazon Mechanical Turk [1] or CrowdCrafting [2], are valuable resources to gather users as long as they match the target profile.

In practice, users with the targeted profile may be hard to gather, especially when the required expertise is high and/or rare. This explains why students are often used in user-based experiments. For instance, in the examples considered in Section 2, Piltaver et al. (2014b), Allahyari and Lavesson (2011), and Huysmans et al. (2011) all enrolled students in their experiments. It has been shown that in certain cases, considering students in the evaluation, more than a choice by default, is in fact a good choice (Carver et al., 2010), as long as threats to validity are carefully addressed. One reason is the homogeneity of the student pool, limiting the difference between each profile and focusing the experiment on variables that are specific to the task. It also makes it easier to control the expertise background, as the same courses on the domain expertise have been taught to the student pool.

3.3 Which Type of Metric can you use?

The last question is about the different ways interpretability can be measured. Three non-exclusive possibilities can be mentioned: measuring users' errors, time and users' opinions.

First, the errors made by users can be measured. The error assessment can take several forms, such as the tasks identified in Piltaver et al. (2014a)'s design. For instance, the classify task can be used to assess if users can accurately use the model for prediction.

The second possibility is to consider the time taken by users to answer specific questions or the number of tasks performed in a given time. As an example, the time taken by users to classify a set of instances using two different models can be used to compare the interpretability of these two models (for a more extended discussion on the use of Piltaver's tasks for error and time measurement, see Piltaver et al. (2014a)). The duration can also be useful when an error measure is hard to define. For instance, for measuring the interpretability of an unsupervised model, it is not always possible to know what is a correct user answer. Instead, measuring the time needed for the user to grasp a clustering model may be more appropriate.

The third possibility is to consider users' opinions. This option can be combined with the others, and often takes the form of an experimental survey. After having measured the errors or the time taken by the users, questions can be asked about the interpretability of the model.

4 Conclusion

Based on the human-computer interaction (HCI) literature and by referring to the work of Doshi-Velez and Kim (2017) and of Piltaver et al. (2014a), this paper presents guidelines that

1. www.mturk.com
2. www.crowdcrafting.org

can be used by machine learning researchers interested in setting up user-based experiments to measure interpretability. These guidelines correspond to the minimal set of questions typically addressed in the HCI community. The three questions of this minimal set are: "what do you want to measure", "who are your users", and "which type of metric can you use." Through these questions, researchers can align themselves with the experimental settings that are standard in user-centric communities. Future works include finding how to choose between Piltaver's tasks regarding the questions presented in this paper.

References

Allahyari, H. and N. Lavesson (2011). User-oriented assessment of classification model understandability. In *Proc. SCAI*, pp. 11–19.

Bibal, A. and B. Frénay (2016). Interpretability of machine learning models and representations: an introduction. In *Proc. ESANN*, pp. 77–82.

Carver, J. C., L. Jaccheri, S. Morasca, and F. Shull (2010). A checklist for integrating student empirical studies with research and teaching goals. *ESEJ 15*(1), 35–59.

Doshi-Velez, F. and B. Kim (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Huysmans, J., K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems 51*(1), 141–154.

Lipton, Z. C. (2016). The mythos of model interpretability. In *Proc. ICML Workshop on Human Interpretability in Machine Learning*.

Martens, D., J. Vanthienen, W. Verbeke, and B. Baesens (2011). Performance of classification models from a user perspective. *Decision Support Systems 51*(4), 782–793.

Nielsen, J. and T. K. Landauer (1993). A mathematical model of the finding of usability problems. In *Proc. INTERACT and CHI*, pp. 206–213.

Piltaver, R., M. Luštrek, M. Gams, and S. Martinčić-Ipšić (2014a). Comprehensibility of classification trees - survey design. In *Proc. IS*, pp. 70–73.

Piltaver, R., M. Luštrek, M. Gams, and S. Martinčić-Ipšić (2014b). Comprehensibility of classification trees - survey design validation. In *Proc. ITIS*, pp. 5–7.

Poursabzi-Sangdeh, F., D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. Wallach (2018). Manipulating and measuring model interpretability. In *Proc. NIPS WiML Workshop*.

Résumé

Avec l'avancée des modèles "boîtes noires" hautement performants, l'interprétabilité est devenu un sujet de recherche majeur aujourd'hui. Alors que de plus en plus de recherches en apprentissage automatique portent sur l'interprétabilité, les chercheurs en apprentissage automatique n'ont pas de directives précises pour mettre en place des expériences utilisateurs. Cet article fournit une suite de directives à suivre, provenant de la communauté de l'interaction homme-machine, afin de mettre en place ce type d'expériences.

CHAPTER 16

LEARNING INTERPRETABILITY FOR VISUALIZATIONS USING ADAPTED COX MODELS THROUGH A USER EXPERIMENT

The paper presented in this chapter was published at the Conference on Neural Information Processing Systems (NIPS) workshop on Interpretable Machine Learning in Complex Systems in 2016.

Learning Interpretability for Visualizations using Adapted Cox Models through a User Experiment

Adrien Bibal
PReCISE Research Center
Faculty of Computer Science
University of Namur
Namur, 5000 - Belgium
adrien.bibal@unamur.be

Benoît Frénay
PReCISE Research Center
Faculty of Computer Science
University of Namur
Namur, 5000 - Belgium
benoit.frenay@unamur.be

Abstract

In order to be useful, visualizations need to be interpretable. This paper uses a user-based approach to combine and assess quality measures in order to better model user preferences. Results show that cluster separability measures are outperformed by a neighborhood conservation measure, even though the former are usually considered as intuitively representative of user motives. Moreover, combining measures, as opposed to using a single measure, further improves prediction performances.

1 Introduction

Measuring interpretability is a major concern in machine learning. Along with other classical performance measures such as accuracy, interpretability defines the limit between black-box and white-box models (Rüping, 2006; Bibal and Frénay, 2016). Interpretable models allow one to understand how inputs are linked to the output. This paper focuses on visualizations that map high-dimensional data to a 2D projection. In this context, interpretability refers to the ability of a user to understand how a particular visualization model projects data. When a user chooses a particular visualization, he or she implicitly states that he or she understands how the points are presented, i.e. how the model works. Interpretability is then defined through user preferences and no a priori definition is assumed.

Following Freitas (2014) and others, Bibal and Frénay (2016) highlights two ways to measure interpretability: through heuristics and user-based surveys. Tailored quality measures for visualizations are examples of the heuristics approach. Surveys can be used to qualitatively define the understandability of a visualization by asking for user feedback. Both approaches are complementary, but only a few works (e.g. Sedlmair and Aupetit (2015)) attempt to mix them to assess the relevance of several quality metrics for visualization. This paper bridges this gap through a user-based experiment that uses meta-learning to combine several measures of visualization interpretability.

Section 2 presents some visualization quality measures that are used during meta-learning. Section 3 introduces a family of white-box meta-models to find a score of interpretability. Then, Section 4 describes the user experiment that is used to model interpretability from user preferences. Finally, Section 5 discusses the experimental results and Section 6 concludes the paper.

2 Quality Measures of Visualizations

One can consider two types of quality measures for visualizations: one type uses only the data after projection and the other compares the points before and after projection. Typical measures of the first type focus on the separability of clusters in the visualization. Sedlmair and Aupetit (2015) reviewed, evaluated and sorted such measures in terms of algorithmic similarity and agreement with

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

human judgments. They confirmed the top position of distance consistency (DSC) as one of the best measures (Sedlmair and Aupetit, 2015). Let $\mathcal{P}$ be the set of points of the projection, $\mathcal{C}$ the set of classes and $centroid(c)$ the centroid of class c, then (Sips et al., 2009):

$$\text{DSC} = |\{x \in \mathcal{P} : (\exists c \in \mathcal{C} : c \neq c_x \wedge \text{dist}(centroid(c), x) < \text{dist}(centroid(c_x), x))\}| \,/\, |\mathcal{C}|.$$

Two other top measures in Sedlmair and Aupetit (2015) are the hypothesis margin (HM) and the average between-within (ABW). HM computes the average difference between the distance of each point x from its closest neighbor of another class and its closest neighbor of the same class (Gilad-Bachrach et al., 2004). ABW (Sedlmair and Aupetit, 2015; Lewis et al., 2012) computes the ratio of the average distance between points of different clusters and the average distance within clusters.

In order to compare visualization algorithms, Lee et al. (2015) propose a measure of the second type modeling neighborhood preservation. Their measure, NH_{AUC}, can be defined as follows. Let N be the number of points in the dataset, K the number of neighbors, v_i^K the K nearest neighbors of the ith point in the original dataset and n_i^K the K nearest neighbors of the ith point in the projection,

$$\text{Q}_{\text{NX}}(K) = \left(\sum_{i=1}^{N} |v_i^K \cap n_i^K| \right) / (KN)$$

measures the average preservation of neighborhoods of size K. Lee et al. (2015) then use the area under the $\text{Q}_{\text{NX}}(K)$ curve for different neighborhood sizes in order to compute NH_{AUC}.

3 Meta-Learning with Adapted Cox Models

The main goal of this paper is to evaluate whether combining state-of-the-art measures of different types improve the modeling of human judgment. To asses this, we set up an experiment asking users to express preferences between visualizations shown in pairs (see section 4 for more details) and then used these preferences to determine an interpretability score. Since our dataset is composed of preferences between visualizations, our learning problem is rooted in preference learning. For this kind of problem, an order must be learned based on preferences (Fürnkranz and Hüllermeier, 2011). Our dataset consists of a set of visualizations $\mathcal{V}$ and a set of user-given preferences $v_i \succ v_j$ expressing that v_i is preferred over v_j for some pairs of visualization $v_i, v_j \in \mathcal{V}$.

The preference learning algorithm considered for modeling user preferences must be interpretable, such as with a logistic regression (Arias-Nicolás et al., 2008), so that knowledge about the measures used as meta-features can be gained. To solve this problem, we consider a well-known interpretable model used in survival analysis, the Cox model (Cox, 1972; Branders, 2015). We adapted the Cox model to fit our preference learning problem. Indeed, in the case of pairwise comparisons of objects, the partial likelihood of a Cox model can be adapted as follows:

$$\text{Cox}_{\text{pref}}(\beta) = \prod\nolimits_{v_i \succ v_j} \Big[\frac{exp(\beta^T v_i)}{exp(\beta^T v_i) + exp(\beta^T v_j)} \Big] = \prod\nolimits_{v_i \succ v_j} \Big[\frac{1}{1 + exp(-\beta^T (v_i - v_j))} \Big].$$

This adapted Cox model learns a preference score using measures presented in section 2 as features of visualizations v_i and v_j. This regression differs from a true logistic regression in that there is no intercept term. The term $\beta^T v_i$ can be interpreted as an understandability score for visualization v_i.

4 User-Based Experiment

As mentioned in section 3, an experiment was set up to collect preferences from users. Visualizations presented to users were generated from the dataset MNIST with various numbers of classes (from 2 to 6) using t-SNE (van der Maaten and Hinton, 2008) with various perplexities between 5 and the dataset size in a logarithmic scale. Each user was interviewed after the experiment to discuss his or her strategies for choosing between visualizations. We then used this information to better understand cases where Cox_{pref} models were not in agreement with user preferences.

The population of our experiment consisted of 40 first-year university students. They were instructed to select, from two displayed visualizations, the one for which they best understood "how the computer had positioned the numbers". In addition to these two options, they could also select "no preference", in which case the comparison was not used for learning. Successive comparisons were assumed to be independent, meaning that no psychological learning bias was assumed to be involved.

Table 1: Average percentage of agreement with user preferences and 95% confidence interval thereof.

number of classes	ABW	HM	DSC	NH_{AUC}	Cox_{pref}
$63.6\% \pm 0.1$	$65.6\% \pm 0.1$	$67\% \pm 0.2$	$68.5\% \pm 0.2$	$71.5\% \pm 0.1$	$76.4\% \pm 0.2$

Table 2: Percentage of wins for every pairwise comparison between the five quality measures.

	number of classes	ABW	HM	DSC	NH_{AUC}	Cox_{pref}
ABW	84.5%					
HM	88.3%	67%				
DSC	97.5%	89.6%	70%			
NH_{AUC}	100%	99.3%	98.2%	87.1%		
Cox_{pref}	100%	100%	100%	100%	99.3%	

A total of 3294 preferences was collected. Because each user may have a different strategy while choosing visualizations, they were grouped into batches per user. For a given user, a random subset of his or her preferences was selected, with the total number of preferences being the same for all users. Thanks to this subsampling, all users had the same weight when modeling the overall strategy. The number of preferences per user was set at 30, which let aside 10 users that provided less than 30 preferences; our dataset was composed of 900 preferences. 1000 user permutations were performed. For each permutation, 2/3 of the users were used for training the Cox_{pref} model and 1/3 for testing. The performance measure was the percentage of agreement between users and the model. We used the same performance measure to individually compare the visualization measures used as meta-features.

5 Discussion

In addition to the two types of measures presented in section 2, the number of classes was also considered for meta-learning (Garcia et al., 2016). In the case of a tie (i.e same number of classes), one of the visualization was chosen randomly. Table 1 shows the means and standard deviations computed on the 1000 permutations and table 2 presents the percentage of win against other measures. Measure m_i^p wins against measure m_j^p if m_i has better performances than m_j for the permutation p.

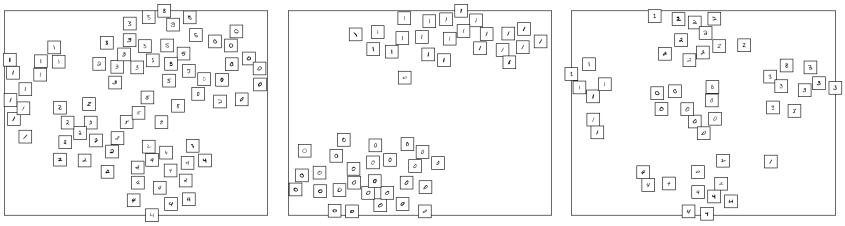
Among the measures of the first type discussed in section 2, DSC performs well in its group but is beaten by NH_{AUC}, the measure of the second type. Interestingly, NH_{AUC} obtains very good results despite the fact that it does not directly apply the well-known user-strategy of cluster separability (Sedlmair and Aupetit, 2015), a strategy that was confirmed during the interviews. Indeed, measures of the second type use the original high-dimensional data in their computation, which is not possible for a human. In both table 1 and 2, the Cox_{pref} model outperforms individual measures. Similar results were observed using all 3129 preferences from the same 30 users.

In order to understand why the Cox_{pref} models fail in 23.6% of the cases on average, we checked judgment errors from Cox_{pref} by referring to what users said during the interviews. We could observe that involving users open the opportunity for mistakes or unusual behaviors, as we can see in figure 1. Furthermore, in a few cases, when the user has no preference but distinguishes a semantic pattern that makes sense for him or her in the visualization, he or she tends to choose it (see figure 1).

In order to assess the importance of each visualization measure in the score of Cox_{pref}, we varied the L1 penalization to enforce sparsity. NH_{AUC} is selected first. Then ABW is added with an improvement of roughly 3.5%. The number of classes is added as a third measure, which improves the model by roughly 1.5%. Other additional measures only offer a minor improvement.

6 Conclusion

Using an adapted Cox model to handle the task of preference learning, we observed the modeling power of a measure taking into account elements that a human being cannot handle, such as NH_{AUC}. Furthermore, we confirmed the position of DSC as leader of its category. Finally, we showed that using a white-box model to aggregate state-of-the-art measures can improve the prediction of human judgment using information of measures from different families. Further work needs to confirm the results obtained with t-SNE for MNIST on a wide range of datasets and visualization schemes.



(a) (b) (c)

Figure 1: Examples of disagreement between users and Cox_{pref}. Among visualizations (a) and (b), Cox_{pref} prefers (b) where 0s and 1s are clearly separated, whereas the user preferred (a). Visualization (c) shows an example of semantic bias: two users reported that they preferred (c) when there is a tie because it looks like a clock (1s on the left, 2s at the top, 3s on the right and 4s at the bottom).

Acknowledgments

We are grateful to Prof. Bruno Dumas for his help for the design of the experiment involving users. We also thanks Dr. Samuel Branders for fruitful discussions and sharing resources on Cox models.

References

Arias-Nicolás, J., Pérez, C., and Martín, J. (2008). A logistic regression-based pairwise comparison method to aggregate preferences. *Group Decision and Negotiation*, 17(3):237–247.

Bibal, A. and Frénay, B. (2016). Interpretability of machine learning models and representations: an introduction. In *Proc. ESANN*, pages 77–82, Bruges, Belgium.

Branders, S. (2015). *Regression, classification and feature selection from survival data : modeling of hypoxia conditions for cancer prognosis*. PhD thesis, Université catholique de Louvain.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

Freitas, A. A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10.

Fürnkranz, J. and Hüllermeier, E. (2011). *Preference learning*. Springer.

Garcia, L. P., Lorena, A. C., Matwin, S., and de Carvalho, A. (2016). Ensembles of label noise filters: a ranking approach. *Data Mining and Knowledge Discovery*, 30(5):1192–1216.

Gilad-Bachrach, R., Navot, A., and Tishby, N. (2004). Margin based feature selection-theory and algorithms. In *Proc. ICML*, page 43, Banff, Canada.

Lee, J. A., Peluffo-Ordóñez, D. H., and Verleysen, M. (2015). Multi-scale similarities in stochastic neighbour embedding. *Neurocomputing*, 169:246–261.

Lewis, J. M., Ackerman, M., and De Sa, V. (2012). Human cluster evaluation and formal quality measures: A comparative study. In *Proc. CogSci*, pages 1870–1875, Sapporo, Japan.

Rüping, S. (2006). *Learning interpretable models*. PhD thesis, Universität Dortmund.

Sedlmair, M. and Aupetit, M. (2015). Data-driven evaluation of visual quality measures. *Computer Graphics Forum*, 34(3):201–210.

Sips, M., Neubert, B., Lewis, J. P., and Hanrahan, P. (2009). Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

CHAPTER 17

MEASURING QUALITY AND INTERPRETABILITY OF DIMENSIONALITY REDUCTION VISUALIZATIONS

The paper presented in this chapter was published at the International Conference on Learning Representations (ICLR) workshop on SafeML in 2019.

MEASURING QUALITY AND INTERPRETABILITY OF DIMENSIONALITY REDUCTION VISUALIZATIONS

Adrien Bibal & Benoît Frénay
PReCISE - Faculty of Computer Science - NADI
University of Namur
Namur 5000, Belgium
{adrien.bibal,benoit.frenay}@unamur.be

ABSTRACT

One first step to get insights about a dataset can be its visualization using dimensionality reduction (DR). However, DR processes induce a loss of information that needs to be quantified in order to evaluate the quality of their results. Furthermore, two DR visualizations with a similar loss value can be really different in the eyes of the user. This paper presents DR quality measures developed in the machine learning community, as well as visual quality measures considered in the information visualization community, which can be used to assess interpretability. We propose to combine several measures from these two categories in order to be able to predict and study users' understanding of DR visualizations.

1 INTRODUCTION

Given the high amount of data generated today, many techniques are developed and used to get insights about these data. Visualization is an important method for understanding hidden patterns in data and is often used as a first explanatory step before any processing or analysis. Indeed, when the studied dataset is high dimensional, the structures and patterns are hard to comprehend.

Dimensionality reduction (DR) is one of the different ways to transform high-dimensional (HD) data so as to allow a visualization (Lee & Verleysen, 2007). The objective of visualization through DR techniques is to find a low dimensional space, typically two or three dimensions, for representing high-dimensional data. Among all DR techniques, one can cite principal component analysis (PCA) (Hotelling, 1933), multidimensional scaling (MDS) (Kruskal & Wish, 1978) and t-distributed stochastic neighborhood embedding (t-SNE) (van der Maaten & Hinton, 2008).

In order to evaluate embeddings of HD data obtained with DR, two goals must be taken into account. On the one hand, it is necessary to define a measure of information preservation for the dimensionality reduction process. On the other hand, the user still needs to interpret the new space where data are projected, as it may serve as a basis for analyses. These two goals, ensuring information preservation and interpretability, should be considered together for measuring the overall quality of an embedding (Liu et al., 2017; Vellido et al., 2012; Frénay & Dumas, 2016; Dumas et al., 2018).

This paper proposes to bridge the gap between DR visualization quality metrics in machine learning and information visualization to measure the two facets of DR visualization quality. The paper is organized as follow. The background on dimensionality reduction is presented in Section 2. Then, Section 3 presents information preservation and interpretability measures in the literature. Propositions on how to bridge the gap between measures of the two categories, information preservation and interpretability, are presented in Section 4. Finally, Section 5 concludes the paper.

2 DIMENSIONALITY REDUCTION VISUALIZATION AND INTERPRETABILITY

Dimensionality reduction (DR) is the process of reducing the large number of dimensions d of a dataset to a lower number $m \ll d$. There are many reasons behind such a process, like the need to escape the curse of dimensionality (Bellman, 1961; Hastie et al., 2009). For instance, when the number of dimensions is too high, each pair of instances tend to have the same distance with respect

to all other pairs. This is a major difficulty when using algorithms with an objective function based on distances.

Another use of dimensionality reduction is to visually analyze the data at hand (Lee & Verleysen, 2007). When the number of reduced dimensions m is equal to 2, high-dimensional patterns can be seen and analyzed, as long as the information loss in the DR process is reasonable. The measures assessing this preservation of information and therefore characterizing the "accuracy" of the DR process are called *DR accuracy* measures in the remainder of this paper.

Among the possible DR techniques, linear DRs, such as PCA, are often considered to be methods providing interpretable embeddings because the way in which their parameters are combined can be easily understood. However, nonlinear DR (NLDR) embeddings, such as the ones computed by MDS or t-SNE, are hard to understand (Liu et al., 2017). Interpretability, in the context of DRs, is therefore understood as how easy it is to understand the mapping between the high and low dimensions. The measures assessing the presence of comprehensible visual patterns are called *DR interpretability* measures in this paper, even though the information visualization literature refers to them as visual quality measures. Indeed, we argue that the main way of assessing the interpretability of an NLDR mapping is through measuring meaningful visual patterns. Measures representing the two categories, DR accuracy and interpretability, are presented in the next section. Then, Section 4 presents a way to combine them in order to assess DR qualities globally.

3 Accuracy and Interpretability of DR Visualizations

As an introduction to this section, let us consider an analogy with regression analysis. Regression is a problem in which a relation must be found between a set of features $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_d$ and a target $\mathbf{t}$. In linear regression, a linear combination of the features $w_1\mathbf{x}_1 + w_2\mathbf{x}_2 + ... + w_d\mathbf{x}_d$ is used for predicting $\mathbf{t}$. The *mean squared error* (MSE) is an error measure often considered for evaluating the quality of the feature weights $w_1, w_2, ..., w_d$ found for predicting $\mathbf{t}$. However, the reduction of error may not be the sole objective to optimize. For instance, in Lasso (Tibshirani, 1996), another objective is to set as many weights $w_1, w_2, ..., w_d$ as possible to 0. In addition to overcoming overfitting problems, setting some weights to 0 makes the model more interpretable, as fewer features are used in the prediction.

Overall quality of DR visualizations can be considered in the same terms. The DR information preservation measures quantify how "accurate" the DR model is. DR "accuracy" corresponds to how well the patterns in the high-dimensional space, such as distances between instances or neighborhoods, are reproduced in the new low-dimensional space. The interpretability objective focuses on helping users to understand the model. In Lasso, this is performed by setting some feature weights w_i to 0. In DR visualizations, this second objective is related to how easily users visually understand the 2D or 3D embedding.

As Bertini et al. (2011) mention, quality metrics can evaluate any stage of the Card et al. (1999)'s information visualization pipeline (see Figure 1). In our case, the DR "accuracy" quantifies the information preservation of the DR transformation (first process of the pipeline: data transformation), while the DR interpretability metrics focus on the transformed data (second stage of the pipeline: transformed data). Because the two types of measure are grounded in different stages of the DR visualization process, the accuracy is measured with high-dimensional and low-dimensional data, while interpretability is only assessed using low-dimensional data. Note that further stages, such as the way 2D data are displayed, can influence the interpretability of the DR visualization result.

This section presents these two kinds of quality metrics. The measures quantifying the error made while reducing the dimensions are presented in Section 3.1. Measures assessing the presence of visual patterns in 2D representations of data are presented in Section 3.2.

3.1 On the Accuracy of Dimensionality Reduction

In machine learning, the quality of a DR embedding is defined by its faithful reproducibility of the projected high-dimensional structures and patterns. This quality needs to be objectively quantified, as it is hard for users to visually assess it. Indeed, by definition of the problem, users cannot visualize the high-dimensional patterns, which is why DR visualizations are needed (Mokbel et al., 2013).

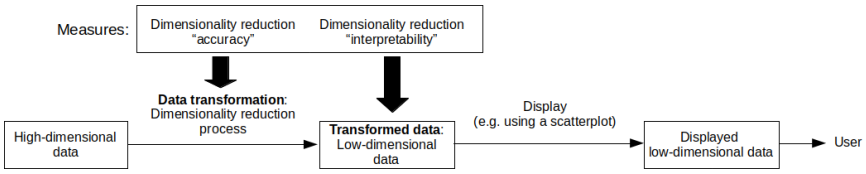


Figure 1: Figure adapted from Bertini et al. (2011) representing the Card's InfoVis pipeline.

DR accuracy metrics can be categorized by the aspect of information loss on which they focus. The two main categories for assessing DR accuracy are *distance preserving* and *neighborhood preserving* measures (Lee & Verleysen, 2009). Distance preserving measures have long been used as objective functions in algorithms such as *multidimensional scaling* (MDS). Under the name *stress function*, we find measures such as the famous *Kruskal's stress* (Kruskal & Wish, 1978), which measures how well pairwise distances in high dimension (HD) are preserved in the low-dimensional embedding (LD). The non-metric version of Kruskal's Stress (NMS) (Kruskal, 1964) measures how well the pairwise distance ranks are preserved, instead of the pairwise distances themselves. The *Sammon's non-linear mapping stress* (NLM) (Sammon, 1969) is another stress function, similar to the Kruskal's stress. The *Curvilinear component analysis* stress (CCA) (Demartines & Hérault, 1997) stands out from the other stress metrics by gradually focusing on small distances. Finally, the *correlation coefficient* (CC) (Geng et al., 2005) is a measure of correlation between the vector of pairwise distances in HD and the vector of pairwise distances in LD.

The second DR "accuracy" metric category focuses on neighborhood preservation. The *stochastic neighbor embedding* (SNE) (Hinton & Roweis, 2003), *t-distributed stochastic neighbor embedding* (t-SNE) (van der Maaten & Hinton, 2008) and *Jensen-Shannon embedding* (JSE) (Lee et al., 2013) algorithms include similar objective functions that focus on the neighborhood preservation of each instance, which is formalized by probability distributions. The size of the neighborhood to consider is controlled by a meta-parameter called the *perplexity*. *Neighbor retrieval visualizer* (NeRV) (Venna et al., 2010) is an algorithm that takes its inspiration from information retrieval, with a DR accuracy metric based on the precision/recall balance. *AUClogRNX* (Lee et al., 2015) is a widely used accuracy metric for DR. AUClogRNX is defined by a sum of the neighborhood preservation over all neighborhood sizes in logarithmic scale:

$$Q_{NX}(K) = \frac{1}{KN} \sum_{i=1}^{N} |\upsilon_i^K \cap n_i^K| \tag{1}$$

$$R_{NX}(K) = \frac{(N-1)Q_{NX}(K) - K}{N - 1 - K} \tag{2}$$

$$AUC_{lnK}(R_{NX}(K)) = \frac{\sum_{K=1}^{N-2} R_{NX}(K)/K}{\sum_{K=1}^{N-2} 1/K}, \tag{3}$$

where K is the number of neighbors, N is the number of instances, υ_i^K is the set of K nearest neighbors of instance i in HD and n_i^K is the set of K nearest neighbors of instance i in LD (Lee et al., 2015). Other neighborhood preservation measures include the *local continuity meta criterion* (LCMC) (Chen & Buja, 2009), *trustworthiness & continuity* (T&C) (Venna & Kaski, 2006) and Q_Y (Meng et al., 2011). LCMC is a penalized stress that increases the loss for close instances in LD that are not neighbors in HD. T&C compares the difference of neighborhood for each instance in HD and in LD. While LCMC and T&C are local measures, as they focus on neighborhoods, Meng et al. (2011) proposes to mix these local measures (called Q_{LC}) with a global measure

$$Q_{GB} = 1 - \frac{6 \sum_{i=1}^{k} d_i^2}{F}, \tag{4}$$

where d_i is a global comparison of ranks in LD and HD for instance i and F is used for normalization. They obtain the measure

$$Q_Y = \mu Q_{GB} + (1 - \mu) Q_{LC}. \tag{5}$$

Among the above DR accuracy metrics, some are intended to be used as objective functions of DR algorithms (e.g. the Kruskal's stress), and some others can only be used for measuring the DR "accuracy" (e.g. AUClogRNX). The advantage of the latter is that they can be mathematically defined without the constraint of being easy to optimize, such as being differentiable (Lee & Verleysen, 2010; Mokbel et al., 2013).

All the above metrics quantify the information preserved, with respect to distances or neighborhoods, by the DR embedding. However, if the DR is used for visualization, these metrics are not sufficient. Indeed, as with the Lasso analogy of Section 3, involving users implies adapting the result to them. This means that the way 2D data is presented in a scatterplot is crucial, even if it requires distorting the patterns present in HD a little bit more in LD. As Behrisch et al. (2018) write: "the essence of effectiveness resides in the identification of *interpretable visual patterns* that contribute to the overarching goal." The visualization interpretability metrics, considered here as the metrics assessing the presence of these interpretable visual patterns, are presented in the next section.

3.2 On the Interpretability of DR Visualizations

In the case of nonlinear dimensionality reduction (NLDR), the link between the new dimensions and the original ones is hard to understand. Liu et al. (2017) propose to see this as a trade-off between interpretability and the *intrinsic structure* of the reduction. Linear dimensionality reductions are often considered as easy to interpret because the new dimensions are linear combinations of the original ones. For NLDR, the intrinsic structure of the embedding is much more complex, resulting in a much less interpretable embedding. The difficulty of identifying the link between the high and the low dimensions is all the more important since many NLDR are non-parametric.

There are two main ways to solve the interpretability problem. First, techniques can be developed to interpret the new LD axes. For instance, regression analysis can be used to interpret the new axes with external variables. In psychology, some data obtained in an experiment A can be used to understand the dimensionality reduction performed by multidimensional scaling on data obtained from an experiment B (Koch et al., 2016; Bibal et al., 2018; Marion et al., 2019). Second, another way to get a better understanding of the embedding is to analyze the position of the instances in the scatterplot. If the instances are positioned such that users can understand these positions with the original dimensions in mind, then the embedding can be considered interpretable. Indeed, if a DR algorithm projects clusters of instances that users understand based on HD features, then the projection can be said to be interpretable, even for a non-parametric DR.

Metrics that measure the position of instances in the 2D space are called visual quality metrics in the information visualization literature. These measures, which help in interpreting the embedding, have different aspects. Among all possible measures, Bertini et al. (2011) present typical categories such as grouping/clustering, correlation, outliers and "complex patterns." Some measures that consider clusters in the 2D space use the instance labels (if present in the dataset) to measure if the 2D visual clusters correspond to those labels (see e.g., Sedlmair & Aupetit (2015); Aupetit & Sedlmair (2016)). For instance, one state-of-the-art supervised cluster measure is the distance consistency (DSC):

$$\text{DSC} = \frac{|x' \in v(X) : \text{CD}(x', centr'(c_{clabel(x)})) \neq true|}{N}, \tag{6}$$

where N is the number of instances, $v(X)$ is the 2D visualization of the dataset X, $centr'(c_i)$ is the 2D centroid of the class c_i, $clabel(x)$ is the provided label of the instance x and $\text{CD}(x, centr'(c_i))$ is true if the closest centroid to x is the one corresponding to the class c_i of x (Sips et al., 2009). This measure computes the proportion of instances for which the closest 2D centroid does not correspond to their label in the original dataset. In addition, measures based on graphs (graph-theoretic scagnostics), such as measures of density (by computing statistics on edge lengths of a minimal spanning tree) or the presence of outliers (detected by comparing edge lengths), can be found in Wilkinson et al. (2005). For more information on these measures, the reader is referred to the recent survey on visual quality measures by Behrisch et al. (2018).

All measures considered in this section assess the presence of patterns in the low-dimensional space. While the end goal is to measure if interesting visual patterns are present in the 2D space, it is not useful to produce a DR visualization with meaningful patterns if these patterns are not present in the high-dimensional space. In other words, a visualization must be both interpretable and accurate. This is why the gap between measures presented in the sections 3.1 and 3.2 should be filled.

4 BRIDGING THE GAP BETWEEN DR QUALITY MEASURE CATEGORIES

In order to globally assess DR visualization quality, accuracy and interpretability measures can be combined. Bibal & Frénay (2016) linearly combined visual clustering measures with AUClogRNX and found that AUClogRNX (the measure of DR "accuracy") outperforms the measures of visual patterns when predicting user preferences of embedding understandability. Johansson & Johansson (2009) also linearly combined some visual quality measures (presence of outliers, correlations and clusters) and estimated the weights through user interaction.

We propose to combine a large set of measures for each objective, DR "accuracy" and embedding "interpretability." By doing so, the combination would balance the two objectives, while considering different aspects of each objective. The overall quality measure could have the linear form

$$\begin{aligned}\text{overall quality measure} = (\alpha_1 * AM_1) + ... + (\alpha_i * AM_i) + ... + (\alpha_n * AM_n) \\ + (\beta_1 * IM_1) + ... + (\beta_j * IM_j) + ... + (\beta_k * IM_k),\end{aligned} \tag{7}$$

where AM_i is the i^{th} normalized accuracy measure, IM_j is the j^{th} normalized interpretability measure, n is the number of accuracy measures, k is the number of interpretability measures and the α's and β's are parameters to estimate. These parameters, which can be estimated based on a user-based experiment, allow the overall quality measure to be used for assessing other DR visualizations. The importance of each goal would be identified by comparing all α's with all β's. It would also be possible to rank α's (resp. β's) among all α's (resp. β's): the higher the α (resp. β) value, the greater the importance of its corresponding measure among the measures of accuracy (resp. interpretability). If a sparsity penalty is added, α's and β's set to 0 would allow us to know the measures that are not necessary for mimicking users. For instance, if the β_j associated with the measure of visual outliers is set to 0, that would mean that users may not consider visual outliers when assessing the overall quality of DR visualizations. Furthermore, collinearity between measures would highlight redundancies among them.

For estimating α and β values that best represent reality, a user-based experiment should be run. This means that a set of DR visualizations should be assessed by users who would give quality scores to these different visualizations. These scores would make up a vector $\mathbf{t}$ to predict. Optimizing α's and β's in Eq. 7 for predicting $\mathbf{t}$ would make it possible to get insights on the importance of DR accuracy with respect to interpretability for users when assessing visualizations, as well as on the importance of each quality measure for modeling users. Furthermore, multiple regressions can be considered to account for different user profiles. For instance, α's and β's can be estimated for a first profile (e.g. users accustomed to scatterplot analyses), and also for a second profile (e.g. novice users). Finally, it is possible that optimizing the overall quality measure based on user feedback results in bias in favor of the interpretability measures (i.e. users might not consider accuracy when evaluating visualizations). In order to avoid this issue, some information regarding the accuracy should be shown to users during the experiment. For instance, the information loss of the DR visualization or visual signals indicating local DR mistakes can be provided, e.g. (Aupetit, 2007; Lespinats & Aupetit, 2011).

5 CONCLUSION

In this paper, we presented how to approach the problem of interpretability in DR visualizations produced by dimensionality reduction (DR) techniques. Two kinds of measures were discussed. The first kind aims at assessing the quality of the DR process through the idea of information loss. These DR quality measures are mainly developed in the machine learning community, which rarely consider users as part of the evaluation. The other kind of measures, from the information visualization community, characterize the presence of meaningful visual patterns in the low-dimensional space. These measures focus on the visual patterns in 2D, even if these patterns are not present in HD.

We propose to combine these two categories of measures in order to account for the information loss, as well as the interpretability of DR visualizations. This would make it possible to highlight measures that best correspond to user's perception. In future works, we plan to set up a user-based experiment to find the parameters α's and β's from Eq. 7 that best fit user's understandability of DR visualizations. These parameters would allow us to compare state-of-the-art measures with each other and with respect to the real perception of users.

REFERENCES

Michaël Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304–1330, 2007.

Michaël Aupetit and Michael Sedlmair. Sepme: 2002 new visual separation measures. In *IEEE Pacific Visualization Symposium (PacificVis)*, pp. 1–8, 2016.

M. Behrisch, M. Blumenschein, N. W. Kim, L. Shao, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, H. Pfister, T. Schreck, D. Weiskopf, and D. A. Keim. Quality metrics for information visualization. *Computer Graphics Forum*, 37(3):625–662, 2018.

Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 1961.

Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

Adrien Bibal and Benoît Frénay. Learning interpretability for visualizations using adapted Cox models through a user experiment. *NIPS Workshop on Interpretable Machine Learning in Complex Systems*, 2016.

Adrien Bibal, Rebecca Marion, and Benoît Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, pp. 537–542, 2018.

Stuart Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485): 209–219, 2009.

Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, 1997.

Bruno Dumas, Benoît Frénay, and John A. Lee. Interaction and user integration in machine learning for information visualisation. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, pp. 97–104, 2018.

Benoît Frénay and Bruno Dumas. Information visualisation and machine learning: Characteristics, convergence and perspective. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, pp. 623–628, 2016.

Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, 2005.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: Data Mining, Inference, and Prediction*, volume 2. Springer-Verlag New York, 2009.

Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 857–864, 2003.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Sara Johansson and Jimmy Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15 (6):993–1000, 2009.

Alex Koch, Roland Imhoff, Ron Dotsch, Christian Unkelbach, and Hans Alves. The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion. *Journal of Personality and Social Psychology*, 110(5):675, 2016.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Joseph B Kruskal and Myron Wish. *Multidimensional scaling*. Sage, 1978.

John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

John A Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7-9):1431–1443, 2009.

John A Lee and Michel Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, 2010.

John A Lee, Emilie Renard, Guillaume Bernard, Pierre Dupont, and Michel Verleysen. Type 1 and 2 mixtures of Kullback–Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.

John A Lee, Diego H Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

Sylvain Lespinats and Michaël Aupetit. CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30(1):113–125, 2011.

Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization & Computer Graphics*, 23(3):1249–1268, 2017.

Rebecca Marion, Adrien Bibal, and Benoît Frénay. BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables. *Neurocomputing*, 342:83–96, 2019.

Deyu Meng, Yee Leung, and Zongben Xu. A new quality assessment criterion for nonlinear dimensionality reduction. *Neurocomputing*, 74(6):941–948, 2011.

Bassam Mokbel, Wouter Lueks, Andrej Gisbrecht, and Barbara Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013.

John W Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.

Michael Sedlmair and Michaël Aupetit. Data-driven evaluation of visual quality measures. In *Computer Graphics Forum*, volume 34, pp. 201–210, 2015.

Mike Sips, Boris Neubert, John P Lewis, and Pat Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum*, volume 28, pp. 831–838, 2009.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. Making machine learning models interpretable. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, pp. 163–172, 2012.

Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006.

Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.

Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization*, pp. 157–164, 2005.

CHAPTER 18

COMBINING QUALITY MEASURES FOR PREDICTING USER ASSESSMENT OF DIMENSIONALITY REDUCTION VISUALIZATION QUALITY

The paper presented in this chapter is a non-final version and is soon to be submitted in the journal IEEE Transactions on Visualization and Computer Graphics (TVCG).

Combining Quality Measures for Predicting User Assessment of Dimensionality Reduction Visualization Quality

Cristina Morariu, Adrien Bibal, Rene Cutura, Michael Sedlmair and Benoît Frénay

***Abstract*—A plethora of dimensionality reduction techniques have emerged over the past decades, leaving researchers and analysts with a wide variety of choices for reducing their data, all the more so given some techniques come with additional parametrization (e.g. t-SNE, UMAP, etc.). Recent studies are showing that humans use dimensionality reduction as a black-box regardless of the specific properties the method itself preserves. Hence, evaluating and comparing 2D projections is usually qualitatively decided, by setting projections side-by-side and letting human judgement decide which projection is the best. In this work, we propose a quantitative way of evaluating projections, that nonetheless places human perception at the centre. We run a comparative study, where we ask people to select 'good' and 'misleading' views between scatterplots of low-level projections of image datasets, simulating the way people usually select projections. We use the study data as labels for a set of quality metrics whose purpose is to discover and quantify what exactly people are looking for when deciding between projections. With this proxy for human judgements, we use it to rank projections on new datasets, explain why they are relevant, and quantify the degree of subjectivity in projections selected.**

***Index Terms*—Dimensionality Reduction, Machine Learning, Visualization, Quality Metrics**

I. Introduction

Some of the most wide-spread techniques for data exploration and visualization are dimensionality reduction (DR) methods, also known as projections. DR is a process that projects high-dimensional data to a lower-dimensional space, such that the resulting projection retains specific properties from the original data. A generic application of this mechanism is in visualization, where users can create scatterplots based on two retained dimensions as part of their data analysis process. DR methods are used in various domains ranging from biology and medical research to social sciences, and they are actively researched in both machine learning (ML) and visualization (VIS) communities.

An extensive amount of techniques exist to produce such projections, such as principal component analysis (PCA) [6], multidimensional scaling (MDS) [22], isometric feature mapping (Isomap or ISM) [42], t-distributed stochastic neighborhood embedding (t-SNE) [43] and, more recently, uniform manifold approximation (UMAP) [30]. These methods can produce widely different results, all the more so given that some have hyper-parameters (e.g. the perplexity of t-SNE).

Cristina Morariu and Adrien Bibal are co-first authors.

Evaluating the quality of these results is, however, the burden of users. In a typical process, a user generates a range of projections, visualizes them in scatterplots, and selects a suitable one from the line-up [2]. Several attempts have been made to improve our understanding of what users look for when evaluating projections. Some studies focus on investigating whether human judgment is indeed reliable for evaluating projections [27], while others focus on defining the tasks users perform when investigating projections [9]. Previous works also show that people use DR as a black-box mechanism without necessarily understanding what the objective of the specific technique is [26], [27]. To consolidate the evaluation of projections quantitatively, both the ML and VIS communities proposed quality metrics that can be used to select the best projections automatically.

In this paper, we aim at bridging previous research on quality metrics for dimensionality reduction and scatterplot visualization, with the work done on understanding human judgments of projection quality. We evaluate to what extent existing metrics in the literature can quantify user preferences. To this end, we gathered collections of images that we use to compute widely used DR techniques. Using this initial set of projections, a range of quality metrics are computed and preferences are collected during a user study. In total, 11 image collections are used, 25 projections are computed, resulting from different parametrizations of the DR techniques mentioned above, and a 54 person user study is run to collect preferences on these projections. Our aim is not to survey all DR methods, but rather to investigate whether quality metrics, or a combination thereof, can capture user preferences.

Our problem can be framed as a supervised learning problem, where the relationship between a combination of various quality metrics is used to predict human judgments. In order to solve this problem, machine learning models are used to compute how these metrics should be combined. The aim is to create and provide a model that can both predict projections users would most likely prefer, as well as to offer an explanation as to why they prefer them.

First, building a supervised model will allow us to derive a new metric based on user perception. Indeed, the new metric can be used to select projections that would have generally be considered interesting by users. This is of upmost importance when lots of DR techniques are considered, or for DR techniques that have several non-trivial hyper-parameters to tune. Second, this will enable us to compare which quality metrics are the more important. In particular, the following

research questions (RQ) are studied:

- RQ1: Can we predict user preferences over projections based on a set of quality metrics?
- RQ2: Are the metrics from both the machine learning community and the information visualization community necessary?
- RQ3: What are the most important metrics from each community?

While answering these research questions, we make the following contributions:

- a model that combines quality metrics and can be used to predict user preferences of projections on unseen data,
- a quantitative analysis that explains what users like and dislike when selecting DR projections;
- a new benchmark to evaluate the performance of future metrics when predicting human judgments;
- a proof-of-concept tool that can be used to rank projections for new datasets, as well as explain to its users what metrics drove the ranking of specific projection.

II. Background & Related Work

Our work brings together the two main types of evaluation in dimensionality reduction (DR): the quantitative one using visual and DR-specific quality metrics, and the qualitative evaluation based on human judgments. This section presents the latest work in these two areas, and explains how our contributions build on top of this knowledge.

A. DR Evaluation using Quality Metrics

Measuring the quality of projections is the work of two communities, and each brought quality measures that have their own properties. These different quality metrics are presented in this section.

1) Measures from the Machine Learning Community: The machine learning (ML) community has defined several measures that can be used as objective functions to optimize in dimensionality reduction (DR). For instance, the stress, the well-known objective function of multidimensional scaling, measures the preservation of pairwise distances between the instances in the high-dimensional (HD) and the low-dimensional (LD) spaces. However, a set of other metrics have additionally been defined, in this community, with the sole purpose of measuring the quality of the DR process. The rationale for this choice is that measures that are used in objective functions are constrained in their definition (e.g. being differentiable), constraints that may not be necessary if the sole purpose is to measure quality [24]. Examples of such measures are the local continuity meta-criterion (LCMC) [12], the measure of trustfulness and continuity (Truthfulness and Continuity) [45] and $AUC_{log}RNX$ [23]. These measures typically check if the neighborhoods in the HD space are preserved in the projection. For instance, LCMC computes, for each point, the average number of neighbors it has in common in HD and LD for a certain neighborhood size k.

Truthfulness is the trustworthiness of the visualization for a particular neighborhood of size k. Truthfulness is defined by roughly summing the rank of all pairwise distances from a point i in the original data to its nearest neighbors in the visualization that are not among the k nearest neighbors of i in the original data. This metric measures whether one can trust what can be seen in the visualization. The measure of Continuity is the exact opposite, as it tells how well the patterns from the original dataset are projected in the visualization. The Continuity for a particular neighborhood size k is defined by the rank of all pairwise distances from the point i in the visualization to the nearest neighbors of i in the original data that are not among the k nearest neighbors of i in the visualization.

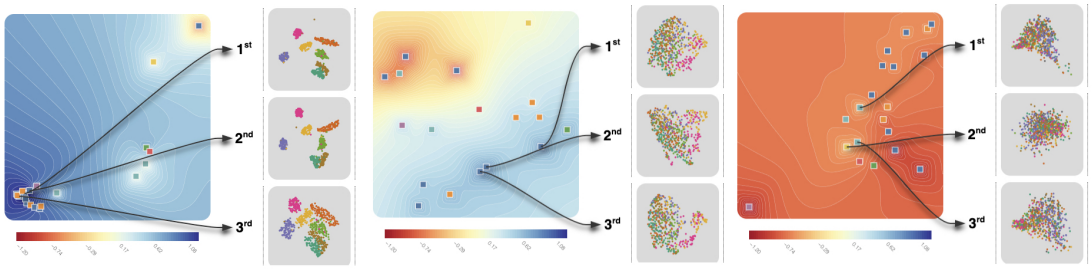
While the previously mentioned approaches focus on a specific neighborhood size k, $AUC_{log}RNX$ consider all neighborhood sizes, with a focus on smaller neighborhoods. In order to do so, $AUC_{log}RNX$ considers, for each point, the number of neighbors in common in LD and HD for all neighborhood sizes with a logarithmic importance.

2) Measures from the Visualization Community: The other community that tackles the measure of projection quality is the visualization (VIS) community. This community developed well-known measures for the detection of patterns in visualizations (e.g. the Scagnostics measures [49], [50]). These types of measures allow users to measure the sparsity, the skewness or even the presence of outliers in the visualization.

More recently, it has been shown that cluster measures can match user perception in visualizations [38]. These metrics are often supervised, meaning that labels about the instances must be provided in order to assess if the classes are well separated. Distance consistency (DSC), for instance, computes the number of instances that are closest to the centroid of their own class rather than an another class. Alternatively, SepMe [1] are an ensemble of separability metrics that use neighbourhood graphs to assess how well separated classes are. These metrics are currently the best performing separability metrics evaluated in literature.

Other popular measures in this category are the average between-within clusters (ABW) [26], the hypothesis margin (HM) [20], the neighborhood hit (NH) [34] and the Calinski-Harabasz index (CAL) [11]. All these metrics measure the separability between clusters, albeit differently. ABW measures the average distances between clusters on the average distances within clusters. HM uses the nearest point from a different cluster and the nearest point from the same cluster to define the notions of inter- and intra-cluster distances. NH makes use of a k-nearest neighbor (kNN) classifier to identify if the points in the visualization are close to their centroid (virtual central point of a cluster). NH corresponds to the accuracy of the classifier. Finally, CAL is a more complicated measure of the same idea: measuring the distances between clusters over the distances within the clusters.

Similar to our goals, several recent works [1], [2], [18], [25], [32] focused their attention on evaluating quality metrics against human perception, although with different use cases. Sedlmair and Aupetit [1], [38] examine perception of class separability in color-coded scatterplots, Pandey et al. [32] assess to what extent Scagnostics can be used as a proxi for human perception, and Lehmann et al. [25] evaluate whether Scagnostics can be used to filter perceptually interesting views for users.



a) MNIST a) Flowers a) Paintings

Fig. 1: The figure shows the top 3 best projections, as scored by our technique, for three of the datasets we have collected: MNIST handwritten digits, photos of flowers and, Art UK paintings. For each dataset, we provide metamaps where each square represents a projection for the particular dataset. The metamaps are calculated by applying dimensionality reduction on the quality metrics space and the color-coded contours represent the ranking score predicting human preferences. Well-liked projections tend to be generated from the same neighbourhood in the metamap manifold. The spread of the ranking score varies across the three datasets, informing the user that the best projections for a dataset are not necessarily great quality. For instance, the MNIST dataset produces stronger candidates than the paintings dataset.

3) Accuracy and Interpretability Measures: The main difference between the measures designed in ML and those in VIS is the object they measure. While ML metrics measure how well the information is preserved when reducing the dimensions, VIS metrics focus on the presence of patterns in the visualizations that make it possible for users to grasp their visualizations and get insights about their data. Following the parallel of Bibal and Frénay [4] with supervised learning, the ML measures would be "accuracy" measures, while VIS measures would be "interpretability" measures. And, as in supervised learning, the two types of measures should be balanced to obtain results that would satisfy users [3], [4]. Indeed, accuracy measures are necessary because visualizations with well-separated clusters are not useful if they are not faithful to the high-dimensional space. Likewise, interpretability measures are also necessary as if readable patterns are not provided, nothing may be taken from the visualization.

4) Combining the Different Quality Measures: One idea, which is the one followed by this paper, is to combine the two worlds by mathematically combining the metrics. For instance, Bibal and Frénay [4] formulated the linear combination of quality metrics as follows:

$$\begin{aligned}\text{combination} = &(\alpha_1 * AM_1) + ... + (\alpha_i * AM_i) + ... + (\alpha_m * AM_m) \\ &+ (\beta_1 * IM_1) + ... + (\beta_j * IM_j) + ... + (\beta_u * IM_u),\end{aligned} \quad (1)$$

where AM (resp. IM) means accuracy metric (resp. interpretability metric). The different α and β, which are learned, represent the contribution of the metric to which they correspond.

Ensembles or combinations of metrics were also discussed in the quantitative survey of DR methods of Espadoto et al. [17]. The authors surveyed 44 DR methods and computed the average of several metrics (truthfulness, continuity, neighborhood hit, normalized stress, Shepard goodness and local error) on 18 datasets in order to assess the global performance of individual DR techniques. However, in this case, the combination of measures is not learned. Similar to this survey are also the works of Nonato and Aupetit [31], and of van der Maaten et al. [44], which extensively review DR techniques alongside quality metrics for DR, albeit without actually computing quality metrics on projections.

5) Applications for Quality Metrics: Apart from the works mentioned here, the VIS community also focuses on bridging the gap between quality metrics and human judgments by designing visual analytics (VA) systems that aid users in comparing [13] or selecting [14], [21], [29] projections. The insights derived from our contribution can be used as part of a VA system that recommends projections, although this is outside the scope of this work.

Lehman et al. [25] also propose using specific quality metrics to automatically filter out easily rejected projections, as scored by users. Wang et al. [47] use previously evaluated quality metrics of subjective class separability to propose a new DR technique, which is implicitly optimized to model human perception of separability.

B. Evaluation Driven by Human Judgments

Despite the existence of quality metrics, the burden in the evaluation of projections remains mainly on users. This section discusses DR research that collects and/or uses human judgment to assess quality.

1) Taxonomies for high-level tasks related to DR: The work by Brehmer et al. [9] aims to define what high-level tasks users perform when they investigate projections. Following interviews, the authors introduce a characterization of tasks. These are *manifold tasks*, where users are trying to name the synthesized dimensions, and *cluster tasks*, where users verify, name, or match clusters with class names. These high-level tasks have been considered in the selection of our datasets to

ensure our study participants deal with different settings. An other closely aligned work is the one of Sedlmair et al. [40], which proposes a cluster analysis taxonomy, one of the most important analysis tasks in the DR data exploration process.

2) Assessing user preferences in DR: Lewis and van der Maaten [27] investigate whether human judgments are consistent by running a user study with groups of experts and novices. The participants are asked to select 2 good projections and a bad one from a line-up of 9 monochrome scatterplots, each representing a projection. They offer the users little information regarding the original dataset and find out different users prefer different projections, inferring that user preferences are vastly subjective. However, they also show that the more users have expertise, the more they are coherent in their judgement. Our study setup builds up on this one, as both studies focus on the real-life task of users selecting projections from a line-up. However, our goal is (i) to deepen the understanding about how users make their decisions and (ii) to model these for recommending projections. Our setup is detailed in Section III.

Bibal and Frénay [3] also ran a user study collecting user preferences of t-SNE projections of the MNIST dataset. The objective of the authors was to study how cluster separability measures and their combination (using a modified Cox model) could predict user preferences. The study presented in this paper is larger in scale at all levels: more datasets, more DR techniques (not only t-SNE), more quality metrics and different ways to frame the problem and to combine metrics. This enlargement in scope allows us to perform original analyses and to draw insightful conclusions.

3) Selecting DR projections: Oftentimes, when new DR methods are introduced, a comparative study to other techniques is proposed as evaluation. The projections get visualized in scatterplots and the reader is invited to assess the line-up and decide for themselves which is the superior projection. This can also be the case for the selection of hyper-parameter values inside a particular DR technique. For instance, the authors of t-SNE invite users to try various parametrizations and select the projection they prefer [43].

Some work [48] acknowledges that blindly trying pairs of hyperparameters and selecting the most appealing projection has downfalls, in that it can mislead users on the faithfulness of the projection. Furthermore, user guidelines given by authors often are technique-specific, in this particular case, for t-SNE. To overcome such issues, Sedlmair et al. [39] assessed what are the best visualization techniques to use during the DR exploration process, and provides guidelines on selecting DR techniques using visualization based on data collected in a user study.

Other work [18] designed a user study to assess which projections can best enhance users' abilities to detect clusters, outliers or estimate the underlying dataset density. These results were, however, not used to recommend better projections for specific tasks.

III. Data Collection

This section describes how the data needed for our models has been collected. Three main elements are needed to create the datasets from which our models will learn: the image collections used to generate projections from, the metrics evaluated on the said projections, and the user preferences associated to the projections.

A. Image Datasets Used & Projection Methods

A key issue when selecting datasets that are used in a DR process for which the result needs to be scored by users is how to provide users with information about the high-dimensional instances. Indeed, to be able to extract meaningful preferences of projections from users that go beyond the appealing aspect of scatterplots, one needs to make sure that users can process the high-dimensional data they are analyzing. From the work of Lewis et al. [27], it is known that assessing preferences by only supplying minimal information about the original data can result in highly subjective and inconsistent judgements across participants. Moreover, in a real-life situation, the user assessing his data often has some prior or intrinsic information that, at least in theory, gives him enough information to assess whether a projection makes sense or not.

In order to solve this issue, only collections of images were selected for our study. Under this setup, the projections visualized as scatterplots would not be simply monochrome scatterplots. Indeed, each dot encoding a 2D position is replaced by a thumbnail of the image getting projected at this location. For example, in the case of the COIL-100 dataset, a collection of objects photographed from different angles, the scatterplot would contain thumbnails of objects as shown in Figure 2a. By showing images as thumbnails, an access to the high-dimensional attributes (the pixels) is given in the same time has the position in the visualization.

Altogether, a total of eleven image datasets, listed in Table I, were collected. Multiple selection criteria were used when deciding which datasets should be included. First, datasets were chosen such that different tasks were performed, even if no task was explicitly defined in the experiment. For example, in the case of the MNIST dataset, the expected task is to match class names (the digits) to various clusters formed. In contrast, for the Stanford face dataset where a bust is photographed from different angles and at different lighting conditions, users can prefer a manifold where the lighting goes from light to dark, or one where the view angle changes smoothly.

Second, datasets of various difficulties were also collected, on the premise that it is much easier to state a preference on projections from an easy dataset like MNIST, as opposed to a more complex dataset like one consisting of photos of Paris. The original image size was used as a measure of dataset complexity, and during the study, users were asked to score the dataset difficulty. For each dataset, its difficulty is conveyed in Table I.

The dimensionality reduction techniques used to generate the projections are principal component analysis (PCA) [6], multidimensional scaling (MDS) [22], isometric feature mapping (Isomap) [42], t-distributed stochastic neighborhood projection (t-SNE) [43], uniform manifold approximation (UMAP) [30], locally linear projection (LLE) [36], and Gaussian random projection (GRP) [5]. For techniques with hyper-parameters,

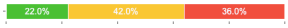
Dataset Name	Description	Difficulty (as scored by users)
COIL-100	Images of common objects photographed from different angles (128 x 128)	
MNIST	Handwritten Digist (28 x 28)	
Fashion MNIST	Images of clothes (28 x 28)	
Stanford Faces	One bust photographed from different angles, in different light conditions (50 x 50)	
Yale Faces	14 people displaying happy, neutral or sad faces (320 x 243)	
Flowers	Photos of 6 different species of flowers (500 x 500)	
Caltech plants	Photos/illustrations of 6 different species of plants (320 x 243)	
Caltech vehicles	Photos/illustrations of 6 different types of vehicles (320 x 243)	
Caltech instruments	Photos/illustrations of 6 different types of instruments (320 x 243)	
Paris Buildings	Photos of buildings in Paris (1024 x 768)	
Oxford Buildings	Photos of attractions in Oxford (1024 x 768)	

TABLE I: This table lists the datasets used in our experiment. The name, description, difficulty (easy - green, medium - amber, hard - red) and the amount of preference disagreements for each dataset are provided as scored by the users.

multiple projections were generated. One hundred projections were initially generated for each dataset and, then, 25 projections for each dataset were uniformly sampled based on the metric space to be used in the user experiment. The projections have chosen such that they have different values for each dataset. Projections that appeared very similar were also down-sampled manually, e.g. rotated variants, or duplicates of one another. This resulted in some datasets having 15 really distinct projections associated, which is above the number of projections presented to users (8 projections per datasets). An example of the projections shown to users can be seen in Figure 2.

B. User Preferences Dataset

This section describes the user experiment that has been set up to collect user preferences on projections.

1) Participants: In total, 54 users participated in our study, out of which 4 had finished a Ph.D., 38 had a master's degree and the remainder 12 had completed a bachelor's degree. We reached our user base by advertising the study within the university network of the co-authors. Participation was voluntary and unpaid. We asked participants for their domain expertise in machine learning, visualization and dimensionality reduction, and the majority of our user base reported familiarity with all these concepts.

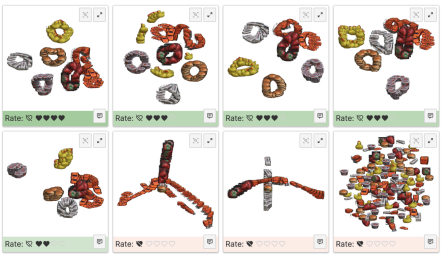
2) Study Procedure: We conducted a web-based user study that takes place completely online and on various display sizes. The study begins with an information page explaining the subject of the study, and the duration it takes (40 to 60 minutes). The participant can only access the study if their display size is larger than 700 x 500. After reading the information page, the users are presented a consent form, and a general introduction explaining what dimensionality reduction is and how the user interface of the study works. Prior to beginning the actual trials, users are asked their previous experience with machine learning, information visualization, and dimensionality reduction. They are also asked what their latest degree they graduated from is. The study then proceeds with the trials. Upon finishing, participants are asked of the overall difficulty of the setup, and any other feedback.

3) Trial Setup: Our study consists of multiple trials where users have to rate and rank projections. Each trial in the study uses as stimuli the projections generated from the dimensionality reduction algorithms applied to one of the datasets described in Section III-A.

A total of 8 projections are shown per trial in a 2-by-4 grid. The projections are randomly selected from the total amount of projections available for a particular dataset and are placed on the grid in a randomized order. The DR projections are shown as scatterplots of images on a white background. The eight views are connected by brushing and linking, so if a user hovers over one image within a scatterplot, this becomes highlighted across all eight plots. Additionally, the user can zoom in or enlarge one particular view. An auxiliary view is also included where the images in the scatterplots are replaced by dots that are color-coded by labels present in the dataset (but not used during the dimensionality reduction process).

At the beginning of each trial, users receive 15 points (represented by hearts in the interface) they should distribute across the eight projections. A higher number of points assigned to a projection signifies that the user prefers this projection more. One projection can receive a maximum of 4 points. A user may also mark a projection as bad, rather than distribute any point to it. After each rating, the user can sort the grid such that the projections get rearranged by preference in descending order. The sorting mechanism together with the restricted number of points per trials were designed to force the users into deciding which projections they liked more and which not. The intention was to avoid a situation where a user would award every projection an equal number of points. This also enables the user to only compare a projection with its neighbours on the left and right, rather than always taking into account 8 views simultaneously. A rated and sorted example of a trial is presented in Figure 2.

Upon completion of one trial, participants are asked to score



(a) Image scatterplot view of the interface. This view is used so that users can see, through thumbnails, how the images from the dataset have been projected in 2D.

(b) Point scatterplot view of the interface. This simpler view contains points instead of image thumbnails, with colors corresponding to class labels.

Fig. 2: Two different views of the same trial from the experiment for collecting user preferences. Each view contains 8 projections of COIL-100 built by different DR techniques. Black hearts correspond to the scores distributed among the projections.

the difficulty of the trial and whether they would like to score another dataset. Each user can complete up to 10 trials, each trial testing a dataset. The datasets across trials appear in a random order. Sampling with replacement is used to choose the next trial, meaning a user can see the same dataset twice but with a different selection of projections. The setup was implemented using a serverless architecture in JavaScript and can be accessed here [1]. The data collected during the setup is hosted in Germany.

4) Descriptive Results: An important aspect to analyze was the degree of consensus between users when it came to preferences. Previous work [27] showed that there is a high degree of subjectivity when it comes to users recording preferences of DR projections. Furthermore, users' ability to select good quality projections was called into question. In our study, however, we report that while there were disagreements in ratings, the majority converged towards well-defined preferences. From a descriptive perspective, only about 18.5% of the ratings were in disagreement with the majority. A breakdown of disagreement in conjunction with the difficulty of the dataset as scored by the user can be seen in Table I. Datasets perceived as harder also incurred a higher percentage of disagreements. One example is the dataset of building photos from Oxford. The same applies the other way around, where "easy" datasets such as MNIST and COIL-100 had low percentage of disagreements.

Based on the ratings awarded in each trial by each user, we calculated a preference matrix by counting how many times a projection was scored higher than another one. We further aggregated these results to assess whether particular DR techniques are systematically preferred over others. In Figure 3, we can see the user preferences aggregated on a DR technique level. The heatmap encodes how many times users agreed that one DR technique (mentioned row-wise) was better than another (column-wise). The bluer the cell the more people agree that the DR technique in the row was better than a technique in the column. There are clear winners and clear losers. For example, the Gaussian Random Projections (GRP) were universally disliked alongside bad parametrizations of UMAP (e.g. when only two neighbours are considered). Interestingly, there were no universally bad parametrization for t-SNE. UMAP with good parametrizations appeared to be systematically preferred over the other projections. A hierarchy can be observed: $PCA \leq Isomap \leq t-SNE \leq UMAP$, where $DR_i \leq DR_j$ means that the visualizations generated by DR_j were more often preferred to the visualizations generated by DR_i.

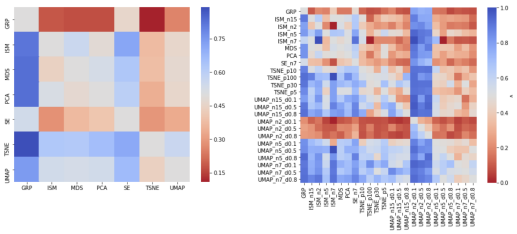


Fig. 3: User aggregated preferences of DR technique, overall (left) and parametrized (right). A score higher than 0.5, depicted in blue, means that more than 50% of the users preferred the DR technique specified in the row over the one specified in the coloumn. Scores lower than 50% are encoded in red. The heatmaps are roughly sorted by user preference in ascending order, with the exception of some parametrizations of UMAP which are universally disliked.

C. Quality Metrics Dataset

In order to predict user preferences, metrics from different communities that measure different aspects of visualizations have been gathered. All metrics were normalized such that their value is between 0 and 1.

[1]The user study is available here: https://kix2mix2.github.io/DumbleDR/public/index.html

Metric Name	Type	Applied on
Outlying [49], [50]	Scagnostics	LD
Skewed [49], [50]	Scagnostics	LD
Clumpy [49], [50]	Scagnostics	LD
Sparse [49], [50]	Scagnostics	LD
Striated [49], [50]	Scagnostics	LD
Convex [49], [50]	Scagnostics	LD
Skinny [49], [50]	Scagnostics	LD
Stringy [49], [50]	Scagnostics	LD
Monotonic [49], [50]	Scagnostics	LD
ABW [26]	Cluster separability	LD
CAL [11]	Cluster separability	LD
DSC [41]	Cluster separability	LD
HM [20]	Cluster separability	LD
NH [34]	Cluster separability	LD
SC [35]	Cluster separability	LD
CC [19]	Correlation btw distances	HD to LD
NMS [22]	Stress	HD to LD
CCA [16]	Stress	HD to LD
NLM [37]	Stress	HD to LD
LCMC [12]	Small neighborhoods	HD to LD
T&C [45]	Small neighborhoods	HD to LD
NeRV [46]	Small neighborhoods	HD to LD
$AUC_{log}RNX$ [23]	All neighborhoods	HD to LD

TABLE II: List of measures used in our analysis. If the metric is said to be applied on LD, then it only measures the quality (or check patterns in) the visualization. However, if it said to be applied from HD to LD, then it measures the accuracy of the DR process.

The list of metrics used, as well as whether they measure the correctness of the HD-to-LD mapping, or the quality of the LD visualization only, is presented in Table II.

Among the metrics in Table II that have not already been presented in Section II-A, one can find the silhouette coefficient (SC), the correlation coefficient (CC), the Kurskal's non-metric stress (NMS), the curvilinear component analysis (CCA), the non-linear mapping stress (NLM) and the neighbor retrieval visualizer (NeRV).

SC [35] is a classic metric in clustering that measures how clusters are separated to each other, versus how instances inside a same cluster are grouped together. This metric is similar to, e.g., ABW, except that it diverges a bit in its mathematical definition.

CC [19] is a metric that computes the correlation between the vector of all pairwise distances in the original dataset and the corresponding vector of pairwise distances in the visualization.

NMS [22], CCA [16] and NLM [37] are three stress measures that are considered in our study. Stress measures have in common that they measure how well pairwise distances in the high-dimensional space are preserved in the low-dimensional space. Each of the three measures have their particularities. For instance, NMS [22], as a non-metric measure, does not compare pairwise distances directly, but their ranking.

Finally, NeRV [46] is a metric based on information retrieval, in the sense that it translates the concepts of precision and recall to a measure similar to the Truthfulness and Continuity. Furthermore, similarly to the two sub-metrics of Truthfulness and Continuity, precision and recall are then combined by using, for instance, a simple mean. One particularity of NeRV is that it redefines the distances in the original dataset and in the visualization as probabilities, like t-SNE. It also contains a perplexity hyper-parameter that represents the size of the neighborhood to consider. In our experiments, NeRV perplexity has been fixed at 5.

In Figure 4, a correlation matrix heatmap of the calculated measures is presented. The separability metrics are all highly correlated. For this reason, for all analysis involved we decided to drop measures correlated at more than 95%. Between pairs of highly correlated measures, the most popular one in each pair was kept. In consequence, the metrics dropped from further analysis were: $SepMe_{mvf}$, $SepMe_{mvt}$, Continuity, NH, and CC. Additionally we also removed ABW and CAL, as they were low variance features.

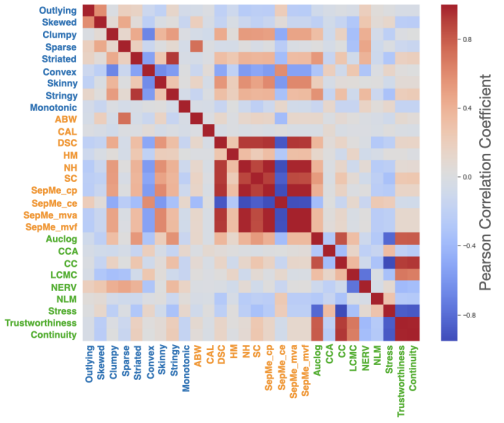


Fig. 4: Correlation matrix of the 2 categories of metrics calculated: interpretability (Scagnostics measures, in blue & separability measures, in amber) and accuracy measures (in green).

IV. User Perception Analysis

This section starts by explaining the general setup of our analysis, and follows by describing each of our three models. For each model, a description is provided, as well as an analysis of the results.

To model our data, three tasks are introduced with incremental levels of difficulty:

- The first aims to loosely classify "good" and "bad" projections, as decided by users. Here, the number of points awarded is not taken into account, only the whether the projection was crossed out or not.
- The second model aims to learn which projections are preferred by users, by answering the question "Would projection A be preferred to projection B?".
- The final model builds on top of the second by first learning the preferences, and also by providing an absolute measure of how good a projection is. Here we want to

examine whether a non-linear combination of the metrics can further improve the performance of our technique.

From all three models, we aim to extract the most important metrics that help predicting user preferences. Although there is disagreement in the data, no data and no participants were discarded from the training process. Hence, the models were trained on noisy annotations where the same projection may have conflicting annotations. With this setup, we were able to take into account the subjectivity in the data.

To conclude this section, we will decide which of the three models should be used as part of our technique and proof-of-concept tool presented in Section VI.

A. Modelling Setup

The evaluation of our models is operated on a leave-one-group-out basis. This is a cross-validation setup where the data is split into distinct groups and, then, a model is trained on the collected preferences related to all groups but one. The remaining group is used as a test set. The process is repeated for all combinations of groups. Throughout our modellings, we mainly use the datasets as our groups. We call this procedure leave-one-dataset-out (LODO).

Given that different datasets are used to generate our projections, and that they have different degrees of complexity (I), it is expected that all our models vary slightly in performance from dataset to dataset. Furthermore, computing a prediction score for each groups also enables the building of a measure of prediction uncertainty on unseen data, by calculating the confidence interval over all test dataset results $\mathbf{R}_{test}$ (i.e. $\bar{\mathbf{R}}_{test} \pm 1.96(\sigma(\mathbf{R}_{test})/\sqrt{n})$).

B. Model 1: Classifying Good and Bad Projections

In a first framing of our problem, a model is set up to learn the distinction between "good" and "bad" or misleading projections. During the collection of preferences, users were able to either assign points to a projection, meaning it is "good" to some extent depending on the number of points, or cross out the projection, meaning it is "bad". This can be seen in Figure 2, where some scatterplot score bars are highlighted in green and points are assigned, while others are highlighted in red.

For this setup, the metrics data are assigned as features and people's preferences are binarized to 1, if the projection was awarded any number of points, and 0, if the projection was marked as "bad". The data is fed to a random forest ensemble and evaluated on a LODO basis to determine the prediction performance for each dataset. As part of this task, we evaluated a boosted tree ensemble and as well as linear model, and cross-validated hyper-parameters to choose the best setup for each test fold of LODO. The ensemble got the best results as it corrects the decision trees' tendency to overfit their training sets. An additional advantage of using tree based models is the relative ease in deriving additive feature importance. The setup was implemented using the Scikit-learn and XGboost libraries in Python. When trained on the whole dataset, the random forest with 200 decision trees of a maximum depth of 10 nodes was our best setup. In total, the model used 3664 annotated projections to learn.

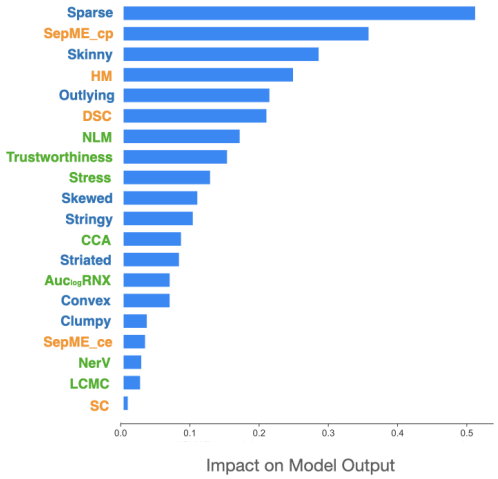


Fig. 5: These are the top features used by Model 1. The features are arranged in this plot in order of importance. The length of the bar represent the absolute impact on the model output. Sparsity is the most important feature, and its impact on the model output means that on average this feature could change the probability for "good" projections by 0.5 either positively or negatively.

SHAPley values [28] were used to explain the prediction for any instance x_i as a sum of contributions from its individual feature values. This interpretation is then similar to that of weights in a linear model, but in a model that can approximate more complex functions.

The area under the receiver operator curve (AUC) metric was optimized with the LODO setup, which provides the predictive performance of 78.36% with a confidence interval of $\pm$ 4.08%.

In terms of feature importance, Scagnostics features such as sparsity, skinny and outlying appear to be the most important ones. Feature importance is summarized in Figure 5. For the majority of the tested projections, low sparsity and high skinniness increase the chances of a projection to be disliked by participants. This makes sense as projections often selected by users as bad tend to be random projections, where points are scattered in the 2D visualizations, with no apparent meaning. An example of such a projection can be seen in the last position on the grid of Figure 2. A similar interpretation exists for very skinny projections such as the projections in the second and third to last places on the grid of Figure 2.

C. Model 2: Preference Learning

In a second modelling, the problem is re-defined as a preference learning problem. In order to do that, for each pair (v_i, v_j) of visualizations in a dataset, a percentage is associated

corresponding to the percentage of time v_i is preferred over v_j. For instance, 90% means that 90% of the time, when v_i and v_j were presented in the same trial to users, v_i received a larger number of hearts than v_j. Because the comparisons are aggregated to get percentages, the number of instances becomes 2268 for this dataset.

The goal of the preference learning model is then to reconstruct the preferences between visualizations, based on the percentage of time a particular visualization has been preferred to another visualization. As for all experiments, the quality metrics are used as explanatory variables for predicting the preferences.

In order to do such predictions, Bradley-Terry models (BTm) [8] are used. BTm linearly combines features to derive probabilities of being preferred:

$$P(v_i > v_j) = \frac{e^{w_0 + w_1 * m_{1,i} + ... + w_{23} * m_{23,i}}}{e^{w_0 + w_1 * m_{1,i} + ... + w_{23} * m_{23,i}} + e^{w_0 + w_1 * m_{1,j} + ... + w_{23} * m_{23,j}}}, \quad (2)$$

where w_0, w_1, ..., w_{23} are 24 weights to learn (one for each metric plus an eventual intercept), and $m_{k,i}$ (resp. $m_{k,j}$) are the k^{th} metric evaluated on the visualization v_i (resp. v_j). Furthermore, our BTm is trained with a Lasso penalty in order to encourage sparsity among the weights. This means that the model has to obtain the lowest error that it can, while using the fewest quality metrics. This results in discarding the metrics that have little to no effect in the prediction of participant preferences and those that are highly correlated. The BTm have been developed by modifying the package BradleyTerry2 in R to include the Lasso penalty.

The absolute value of the metric weights that have been found after learning a sparse BTm on our preference data are presented in Figure 6. The accuracy of the BTm is 62.3%, with the 95% confidence interval being [58.39%, 66.22%]. The accuracy is obtained by counting the number of time the model is right when it says $v_i > v_j$, over the total number of predictions. In order to obtain an accuracy from data that has not been used for training, the LODO strategy has been used. This means that each of the 11 datasets has been used to test a model trained on the other 10 datasets. The final accuracy, reported above, is the mean of the 11 test accuracy scores. This way, the reported final accuracy offers some guaranties on the use of the presented sparse linear model on new datasets. If only the data where users strongly agree on good and bad visualizations (at least 80% of agreement) is used, the accuracy becomes 65.93% [61.42%, 70.43%]. The lambda balancing the importance given to the error and the Lasso penalty was 0.021 for a BTm learned on the whole dataset.

D. Model 3: Ranking Projections

In our final setup, we expand on the previous by implementing a non-linear model to exploit further relationship amongst the metrics and potentially increase the performance of our technique. To that end, we implement a boosted tree ensemble to both rank the projections of each dataset, and to output a absolute measure of how good each projection is. This way, we can not only answer the question 'Is projection A better than projection B?', but also 'By how much?'. Furthermore,

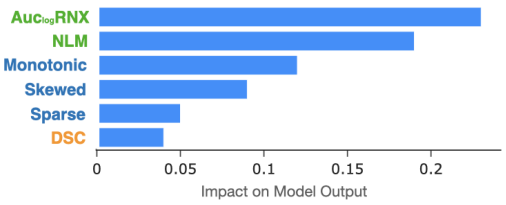


Fig. 6: These are the top features used by Model 2. Given the Lasso regularization, this model uses only 5 features compared to the other 2 models.

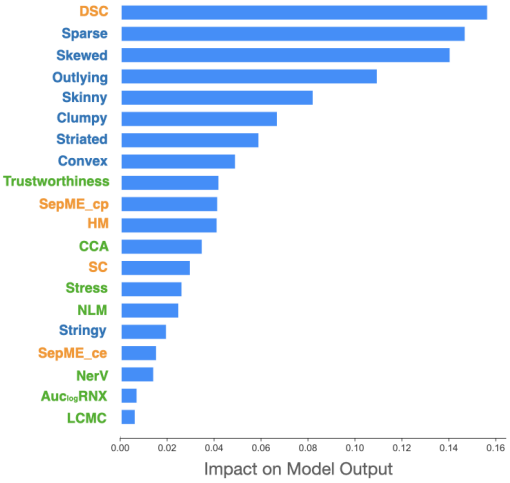


Fig. 7: These are the ranked features of Model 3. Three out of the top five features (DSC, Sparse and Skinniness) are in alignment with features from Model 2.

this popularity score can be used to compare if the projections generated for some datasets have a higher quality than for other datasets.

Boosted trees ensembles are a learning method that can be used for classification, regression and learning-to-rank tasks [10]. The general idea of most boosting methods is to train predictors sequentially, each trying to correct its predecessor's mistakes. The model recursively constructs a series of trees, each trying to improve where the previous made an error. After training all trees, the model outputs the class that is the mode of the classes of the individual trees.

Given boosted trees are widely considered state-of-the-art in supervised learning for tabular data, we expect that exploiting the non-linear relationships between our features could lead to performance improvement. That is, we would like to know whether a non-linear combination of our features, unlike the

one mentioned in Equation 1, can lead to better results.

If in the first modelling, the model had to classify each projection as "good" or "bad", while in the second modelling, pairwise comparisons are calculated for the model to learn. In this setup, projection lists sorted according to the points awarded by users are fed to the model. As such, the model learns again from 3664 instances as in the first setup, but these are sorted into 458 groups of 8 projections, as they were initially ranked by our participants. Based on the rankings submitted by the users, the Model 3's objective is to create a ranking for a new, unseen, dataset of projections. This set of projections can be of any length, not just 8 projections, and the model learns to minimize the number of incorrect pairwise comparisons. This learning is performed by giving the sorting of projections as objective for a boosted trees model. This extension of objectives for boosted trees is further described by the LambdaMART algorithm [10]. Following cross validation of our hyper-parameters, our model was trained using 15 sequentially trained decision trees. The best iteration was the 14th. The learning rate used in the setup was 0.3 and the maximum depth of each decision tree involved was 5. The setup was implemented using the XGboost library in Python.

The LODO error is calculated the same way as in the second modelling, by computing accuracy over the preference matrix of comparisons among the projections. Overall, the accuracy is 70%, with a confidence interval (CI) of $\pm$4.4%. When the LODO error is calculated only for comparisons where there was a strong agreement, such as 80% agreement, the accuracy increases to 78.09%, with a CI of $\pm$6.5%.

E. Model Selection

Theoretically, all three models can be used to reliably make predictions for the introduced tasks. If the users only wish to filter out bad projections, we recommend the first model as it has the higher accuracy on our datasets. However, given we have set out to rank predictions, we have selected the third model to use as part of our technique and tool presented in Section VI. The selection was made based on accuracy performance criterion as with the 3rd one, better result shall be expected in general.

V. Discussion & Limitations

This section presents analyses, discussions and limitations stemming from the three models of Section IV, as well as clear answers to our research questions.

A. Answers to our Research Questions

In the introduction of this paper, three research questions were presented:

- RQ1: Can we predict user preferences over projections based on a set of quality metrics?
- RQ2: Are the metrics from both the machine learning community and the information visualization community necessary?
- RQ3: What are the most important metrics from each community?

All three models also show that some metrics from both communities are important, which answer RQ2. Indeed, all our models use Scagnostics and cluster separability measures for detecting bad projections, and then use accuracy measures to find accurate projections among the ones that contain readable patterns. This conclusion logically stems from the fact that users will not pay too much attention to the semantics inside visualizations if the instances do not form readable patterns.

The answer to RQ3 lies in the Figures 5, 6, 7, presenting the feature importance in our models. While the VIS literature acknowledges the importance of some key Scagnostics measures (e.g. [25]) and of DSC as the best separability measure (e.g. [38]), our models (i) confirm the literature on their importance, (ii) show the importance of combining them and (iii) also show the importance to use accuracy metrics from the ML community. The last point logically follows from the fact that users will not select visualizations containing clear patterns, but which make no sense according to the high dimensions (e.g. clear clusters containing random images in them).

B. Consensus across the Models

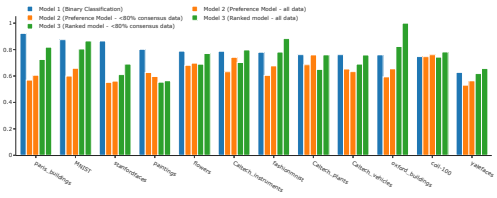


Fig. 8: Overview of the performance of the 3 experiments evaluated for each dataset separately. The first experiment (Binary Classification) is the best performing as it is concerned by an easier task.

The conclusions of all three experiments are very much in line with one another in terms of feature importance. Moreover, the accuracy of the two non-linear Models 1 and 3 is above 75% (for the relevant use cases against which it is optimized). Scagnostics features, like Sparse, Skewed and Skinny, alongside separability metrics, like DSC, are in all cases among the top 5 most important features. Features such as Sparse, Skewed, and Outlying are used to detect bad projections. These features tend to be high for projections where the positioning of the points appears random or uniformly distributed. These were universally disliked by humans, which can be seen in Figure 3, where the Gaussian random projection (GRP) was the most disliked DR technique. Previous work from Lehman et al. [25] also identified a subset of Scagnostics measures, namely stringy and striated, as measures which can be used to "early reject" projections that are not understandable for users.

In the second model, the accuracy metrics NLM and $AUC_{log}RNX$ have a large impact on the model (see Figure 6). They are not compensating each other, as dropping one of the two leads to a new model with a reduced performance. The

higher importance of $AUC_{log}RNX$ and the reliability of DSC, among cluster separability measures, to assess user preferences are aligned with similar experiments in the literature [3]. Indeed, all models use separability features, such as DSC, to detect the presence of semantically relevant clusters. A high DSC measure is a strong indicator of a liked projection. This is in line with the quantitative evaluation undertaken by Sedlmair et al. [40], which highlights class separability as one of the most important tasks people perform on DR. While Scagnostics measures are used like the other models, we can also see the Scagnostic measure clumpy, which identifies clusters regardless of their semantic composition. This points to the fact that people prefer looking more specifically for clusters that make sense semantically.

C. Performance on Unseen Datasets

Figure 8 displays the breakdown of performance accuracy for each dataset. Unsurprisingly, the model performs better on datasets which were rated as easier and with more consensus (see Table I). Given the LODO errors from all experiments, we can establish with a confidence interval of about 5% that our models will be able to generalize to new image datasets. A weakness introduced by our study is that we only use image based datasets. For this reason, we can only speculate that 1) users maintain their preferences for different dataset types, and, 2) that the metrics applied on different dataset types generate a similarly distributed metric dataset.

Even though very different datasets are used in the case study below, a future research direction would consist of extending the study with additional datasets of different types, such as tabular or text data as used in the quantitative survey from Espadoto et al. [17]. A novel way to present the high-dimensional space in the low-dimensional space, as the image thumbnails in the scatterplots of image datasets, will have to be found for tabular and text data.

D. On the Existence of Misleading Projections

A concern one can have is that people can select visually appealing projections that are nonetheless wrong with respect to the high-dimensional data. One possible analysis of our paper is to assess to what extent this can happen. Taking into account both the high-quality user sample and our study design that gives users a vast access to information regarding the high-dimensional space, we are confident that if any such "false positives" existed, they would have been caught and marked as misleading or bad. Given that, our different models show that the majority of projections flagged as bad by participants can be detected using Scagnostics and separability measures. Given that no accuracy metric is needed for spotting these bad projections, it rises the question of whether projections where meaningful clusters are formed in the visualization, even though these clusters do not exist in the high-dimensional space, is even possible.

E. Performance of DR techniques

An additional issue spanning from the type of datasets selected (i.e. image collections) is that linear techniques such as PCA get rated down. Given the fact that images lie on a non-linear manifold in the high-dimensional space, it makes sense that linear DR methods such as PCA or MDS under-perform in comparison to UMAP or t-SNE.

To evaluate the generalization to new DR techniques, a leave-one-dimensionality reduction-out (LODRO) error is calculated for Model 1. Rather than splitting by dataset during our cross validation, as in LODO, we train to detect "good" and "bad" projections by considering all dimensionality reduction techniques but one. The LODRO procedure allows us to check if our analysis applies to projection from new, unseen, DR techniques.

Overall, our generalization error to new DR techniques is settling at 59,8%, with a confidence interval of 9%. Figure 10 breaks down our results per DR technique for this analysis. GRP and MDS have the worst generalization error. The explanation can be that these particular methods bring very different projections than the other DR techniques. However, the users in our study graded the projections resulting from GRP, SE and some UMAP configurations as universally bad across all datasets (see Figure 3). Users have even commented about how these projections appear to be random. However, the visualizations that appear to be random to the human eye are in fact very different according to quality metrics, meaning that bad projections are not all bad in the same way. On the flip side, most configurations of UMAP, which is one of the newest proposed techniques in the literature, generalize very well. An interesting future direction could be to assess which minimal set of dimensionality reduction techniques could be jointly used to train models such as ours in order to ensure that the resulting projections are diverse enough to generalize well.

It should be noted that the LODRO strategy cannot be easily applied for the Models 2 and 3, since, in these setups, we would require more DR techniques, and more than 20 total projections per dataset in order to achieve significant results.

F. On the Limited Number of DR Techniques and Quality Metrics

To the best of our knowledge, the DR techniques and quality metrics presented in this paper are a representative set of what is popular in the literature. However, one can argue that DR techniques and quality metrics that are not yet popular are not used. Even more, one can argue that new DR techniques and quality metrics can be invented in the future. While this is true, one contribution of this paper is also to present a framework on the use of quality metrics to predict user preferences in projections. This means that new metrics can be plugged in our framework so that a new combination is automatically learned and then analyzed without needing additional user feedback. Similarly, the combination can be re-trained on projections produced by new DR techniques.

G. Predicting User Behavior when Comparing Projections

Potential future work can consist of using the characteristics from users in our models to derive a different combination of metrics per user profile. This can be done by using variants of

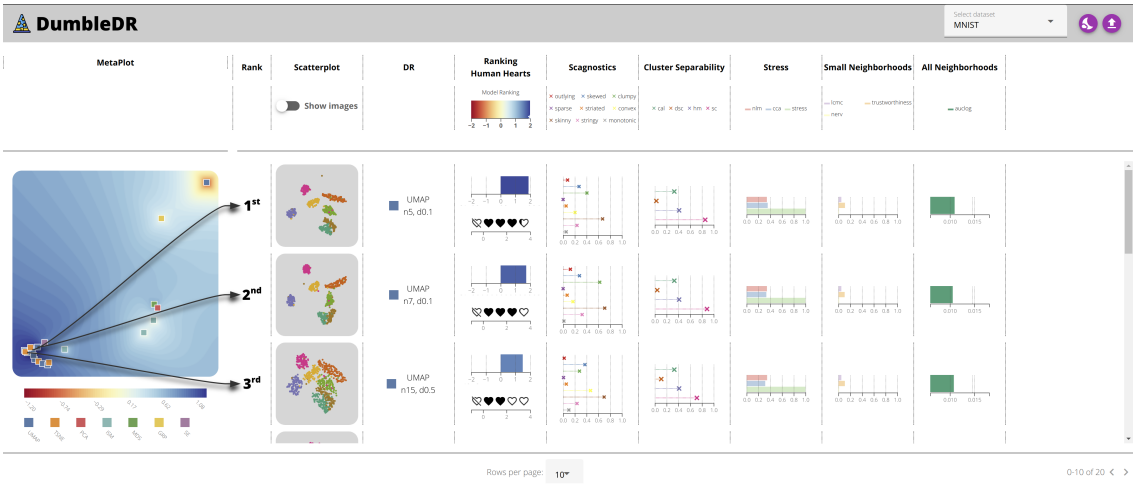


Fig. 9: Screen capture of the tool for ranking projections. The projections in the scatterplots column are ranked using Model 3. On the left of the ranking, a metamap shows similar projections close together and dissimilar ones far apart. The blue (resp. red) zone represent good (resp. bad) projections w.r.t. Model 3 scores. On the right of the ranking, the average number of hearts given by participants is shown, as well as individual quality metrics values.

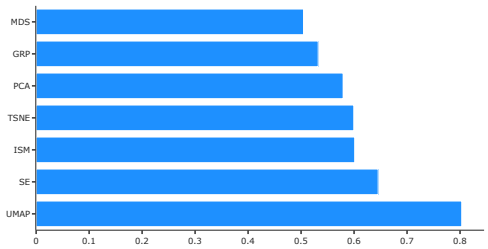


Fig. 10: Overview of the performance of the first experiment evaluated for each DR technique separately.

BTm. Indeed, the BTm presented in this paper can be used to analyze how user characteristics influenced their comparisons of projections. While BTm was used to predict the preferences based on features of the compared objects (the projections), BTm can also be used to predict the preferences based on the features of the ones that stated their preferences.

VI. Application

This section presents a visual analytics tool, named DumbleDR, containing an implementation of Model 3, in order to better showcase how to exploit the benefits of our technique. The following sections present the tool in more details, and two case studies showing the analysis of two new datasets with our proposed model.

A. Presentation of the Tool

Our tool's aim is to demonstrate how users could make sense of our model's outputs on novel datasets. To present the use of Model 3, introduced in Section IV, we designed and implemented a web application[2] that can intake new datasets, compute a range of projections and their associated metrics, and output the top projections. For the purpose of this demonstration, the web application only uses outputs from Model 3, although all 3 models introduced can be plugged in instead. The tool uses JavaScript, specifically the Druid package [15], to compute all projections and metrics, and D3 [7] for visualization.

By using this tool, users can (i) upload their dataset, (ii) have many projections of their dataset created, (iii) get a ranking of these projections based scores given to these projections from Model 3 and (iv) have numerous statistics about their dataset quality.

Figure 9 shows a screenshot of the tool where our technique is applied. After selecting a precomputed dataset or uploading a new one (on the top-right corner of the screen), projections and their respective quality metrics are computed. Without additional training required, Model 3 will output a score for each projection of the dataset. The output score is a real number which can be positive or negative. The higher the number, the better the projection is. The resulting projections are ranked in accordance to this output.

When uploading a novel dataset to our tool, the tool first computes a number of projections, then the associated quality metrics, and finally, the ranking. Of these three tasks, computing the metrics, in particular the accuracy ones, is the most expensive operation. This is because accuracy metrics use the high-dimensional space to compute distance-based neighbors in order to compare them with low-dimensional neighbors. If Scagnostics metrics take less than a minute to compute for 40

[2]The tool is available here: https://renecutura.eu/dumbledr/

projections of a dataset, separability measures take minutes, and accuracy measures can span hours. For this reason, the tool also contains the option to calculate ranking using a model trained on Scagnostics measures only, separability measures only, accuracy measures only, or any combination thereof. Throughout this paper, all the results are calculated using the complex combination of measures defined in Model 3.

On the left of the screen, in Figure 9, a metamap shows the similarity between the projections created based on a selected dataset. This metamap is a UMAP projection over the metrics calculated for each DR projection from the original dataset. this was originally introduced by Cutura et al. in their system VisCoder [14]. The colors in the metamap represent the ranking score of the visualizations: from dark blue for really good visualizations, according to Model 3, to dark red for really bad, low-ranked ones.

The spread of the ranking score outputed by Model 3 varies from dataset to dataset. This can be seen in Figure 1, which shows the metamap of three datasets, MNIST, Flower photography, and ART UK paintings, and the corresponding top three projections. The information encoded in the metamap contours can be used to deduct that the projections from MNIST are rated very high across the ranking (large blue zone) and, therefore, that lower ranked projections can also be considered good. On the contrary, for the paintings dataset, only few projections are good (large red zone), and Model 3 helps to find these good projections. The flower dataset, in the middle of Figure 1, is more balanced, as it contains both good and bad projections. In conclusion, not all produced projections are equal in terms of quality, and our ranking score, a combination of the metrics based on user preferences, is indicative of that.

On the right of the metamap in the tool (see Figure 9), the ranking of visualizations is presented, with arrows linking them to their position on the metamap. The DR column, which is on the right of the scatterplots column, provides all information about the embeddings used to obtain the visualization, along with their parametrization when relevant. The other columns show other information like the average number of points the visualization obtained during the user experiment and the scores from the individual quality metrics. The user can compare the ranking score with the average points awarded by people for each projection during the user study.

B. Use Cases

In this section, we present two use cases on two distinct and novel datasets that were not used in the user study or the previous analysis. The objective of the use cases is to present how to use our tool, and therefore the implementation of Model 3, to obtain projections ranked by quality.

1) Use Case 1: The Pets Dataset: In a first use case, let us consider a user who wants to get a visualization of the pets dataset [33]. This dataset contains 38 classes of various races of cats and dogs. All previous datasets used in this analysis contained a maximum of 7 classes. The reason was to avoid overwhelming users during our study. In this case study, we aim to see if our technique can be successfully applied on datasets with a much higher number of classes.

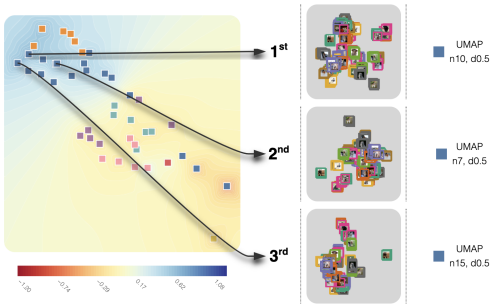


Fig. 11: Top 3 projections given by our tool on the pets dataset. The ranking is provided by Model 3 and shows that UMAP with some particular parametrizations offers visualizations of good quality.

Figure 11 shows the best projections that can be obtained on this dataset. Not only our tool, through our ranking model (Model 3), shows that UMAP can provide good visualizations of the pets dataset, but it also provides the parametrization to obtain these good UMAP projections. Getting the right parametrization is paramount as the worst visualizations, in the most red parts of the metamap, are also UMAP projections.

2) Use Case 2: Selecting Metamaps: Another use case that is exemplified by this very paper is the right choice of metamaps for comparing projections. As presented in this paper, metamaps are projections of projections. They are used to compare projections, and find similar or dissimilar projections. Another example of use, outside the use of metamaps in this paper, is to collect the most different projections in order to get different views of the data. In order to do that, one would project hundreds of projections, producing the metamp, and consider the projections that are the most distant from one another. However, in all applications, if the metamap used is not accurate, nor readable, no insight can be extracted from it.

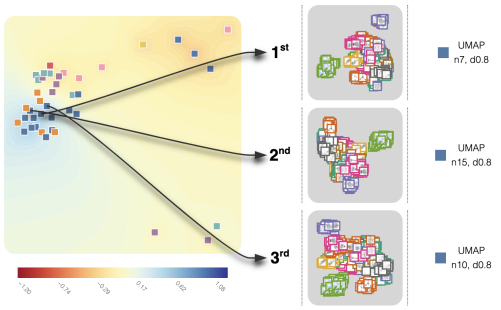


Fig. 12: Top 3 projections given by our tool on the set of metamaps. The ranking is provided by Model 3 and shows that UMAP with some particular parametrizations offers visualizations of good quality.

Figure 12 shows the best metamaps according to the combination of metrics from Model 3. As for the previous

example, UMAP provides the best metamaps when a certain parametrization is chosen. The metamaps presented in this paper are indeed produced by UMAP with a particular parametrization. Please also note that the parametrizations of UMAP that provide the best projections in this case study, the best metamaps, are not the same parametrizations as for the best projections of the pets dataset.

By using our tool and the combination of quality metrics implemented in it (i.e. Model 3), users can upload their dataset and get the techniques and parametrizations that provide the best projections. This eases the cumbersome process of (i) running many different DR techniques, (ii) testing many different parametrizations, (iii) finding, implementing and understanding many different quality metrics, and most importantly, (iv) selecting the best projection according to these quality metrics. Indeed, regarding (iv), our tool provides the combination of quality metrics that best predicts what users would consider as being a projection of quality.

VII. Conclusion

This paper tackles the problem of assessing the quality of dimensionality reduction (DR) visualizations using metrics from two research communities. The first group of metrics comes from the machine learning (ML) community and is used to assess the faithfulness of visualizations w.r.t. the high-dimensional (HD) data. The second group of metrics comes from the information visualization (VIS) community and is used to quantify the presence of readable patterns in the visualization. We proposed combining these different metrics in order to identify the important ones and draw conclusions for the two communities. We implemented a series of machine learning models to predict human preferences and examine to what extent metrics from both communities are used. The final model (Model 3) achieves 78.09% accuracy in predicting both well-liked and misleading projections. Furthermore, Model 3 was implemented in a tool to demonstrate the capabilities of the proposed technique to highlight high quality projections.

It was observed in all three models that Scagnostics and separability measures have a large impact for predicting users. In particular, these metrics were able to easily discriminate between visualizations deemed good or bad by users. It seems that accuracy metrics from the ML community are secondary, but they make it possible to discriminate between accurate and misleading visualizations with readable patterns.

References

[1] M. Aupetit and M. Sedlmair. SepMe: 2002 New Visual Separation Measures. In *Proceedings of the IEEE Pacific Visualization Symposium*, pp. 1–8, 2016. doi: 10.1109/PACIFICVIS.2016.7465244

[2] E. Bertini, A. Tatu, and D. Keim. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 17(12):2203–2212, 2011. doi: 10.1109/TVCG.2011.229

[3] A. Bibal and B. Frenay. Learning Interpretability for Visualizations using Adapted Cox Models through a User Experiment. In *NIPS Workshop on Interpretable Machine Learning in Complex Systems*, 2016.

[4] A. Bibal and B. Frénay. Measuring Quality and Interpretability of Dimensionality Reduction Visualizations. In *Safe Machine Learning Workshop at ICLR*, 2019.

[5] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proc. ACM Intl. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 245–250, 2001. doi: 10.1145/502512.502546

[6] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New-York, NY, USA, 2006.

[7] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data-Driven Documents. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[8] R. A. Bradley and M. E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.2307/2334029

[9] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing Dimensionally-Reduced Data: Interviews with Analysts and a Characterization of Task Sequences. In *Proc. of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)*, pp. 1–8, 2014. doi: 10.1145/2669557.2669559

[10] C. J. Burges. From RankNet to LambdaRank to LambdaMART: An Overview. Technical Report MSR-TR-2010-8, Microsoft Research, 2010.

[11] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101

[12] L. Chen and A. Buja. Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis. *Journal of the American Statistical Association (JASA)*, 104(485):209–219, 2009. doi: 10.1198/jasa.2009.0111

[13] R. Cutura, M. Aupetit, J.-D. Fekete, and M. Sedlmair. Comparing and Exploring High-Dimensional Data with Dimensionality Reduction Algorithms and Matrix Visualizations. In *Intl. Conf. on Advanced Visual Interfaces (AVI)*, 2020. doi: 10.1145/3399715.3399875

[14] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair. VisCoDeR: A Tool for Visually Comparing Dimensionality Reduction Algorithms. In *Euro. Symp. on Artificial Neural Networks (ESANN)*, 2018.

[15] R. Cutura, C. Kralj, and M. Sedlmair. DruidJS — A JavaScript Library for Dimensionality Reduction. In *Proceedings of the IEEE Information Visualization Symposium*, 2020. accepted for publication.

[16] P. Demartines and J. Hérault. Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets. *IEEE Trans. on Neural Networks and Learning Systems (TNNLS)*, 8(1):148–154, 1997. doi: 10.1109/72.554199

[17] M. Espadoto, R. M. Martins, A. Kerren, N. S. Hirata, and A. C. Telea. Towards a Quantitative Survey of Dimension Reduction Techniques. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 2019. doi: 10.1109/TVCG.2019.2944182

[18] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. De Oliveira, and L. Linsen. Perception-Based Evaluation of Projection Methods for Multidimensional Data Visualization. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 21(1):81–94, 2014. doi: 10.1109/TVCG.2014.2330617

[19] X. Geng, D.-C. Zhan, and Z.-H. Zhou. Supervised Nonlinear Dimensionality Reduction for Visualization and Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, 2005. doi: 10.1109/TSMCB.2005.850151

[20] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin Based Feature Selection - Theory and Algorithms. In *Proceedings of the International Conference on Machine Learning*, p. 43, 2004. doi: 10.1145/1015330.1015352

[21] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. DimStiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pp. 3–10. IEEE, 2010. doi: 10.1109/VAST.2010.5652392

[22] J. B. Kruskal. Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika*, 29(1):1–27, 1964. doi: 10.1007/BF02289565

[23] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015. doi: 10.1016/j.neucom.2014.12.095

[24] J. A. Lee and M. Verleysen. Scale-independent quality criteria for dimensionality reduction. *Pattern Recognition Letters*, 31(14):2248–2257, 2010. doi: 10.1016/j.patrec.2010.04.013

[25] D. J. Lehmann, S. Hundt, and H. Theisel. A Study on Quality Metrics vs. Human Perception: Can Visual Measures Help us to Filter Visualizations of Interest? *it Inf. Technol.*, 57(1):11–21, 2015. doi: 10.1515/itit-2014-1070

[26] J. Lewis, M. Ackerman, and V. de Sa. Human Cluster Evaluation and Formal Quality Measures: A Comparative Study. In *Proc. of the Annual Meeting of the Cognitive Science Society*, vol. 34, 2012.

[27] J. Lewis, L. Van der Maaten, and V. de Sa. A Behavioral Investigation of Dimensionality Reduction. In *Proc. of the Annual Meeting of the Cognitive Science Society*, vol. 34, 2012.

[28] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From Local Explanations to Global Understanding with Explainable AI for Trees. *nature machine intelligence*, 2(1):2522–5839, 2020. doi: 10.1038/s42256-019-0138-9

[29] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, 41:26–42, 2014. doi: 10.1016/j.cag.2014.01.006

[30] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[31] L. G. Nonato and M. Aupetit. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 25(8):2650–2673, 2018. doi: 10.1109/TVCG.2018.2846735

[32] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards Understanding Human Similarity Perception in the Analysis of Large Sets of Scatter Plots. In *ACM Conf. on Human Factors in Computing Systems (SIGCHI)*, pp. 3659–3669, 2016. doi: 10.1145/2858036.2858155

[33] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats And Dogs. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092

[34] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 14(3):564–575, 2008. doi: 10.1109/TVCG.2007.70443

[35] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: 10.1016/0377-0427(87)90125-7

[36] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000. doi: 10.1126/science.290.5500.2323

[37] J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, 100(5):401–409, 1969. doi: 10.1109/T-C.1969.222678

[38] M. Sedlmair and M. Aupetit. Data-driven Evaluation of Visual Quality Measures. In *Computer Graphics Forum*, vol. 34, pp. 201–210. Wiley Online Library, 2015. doi: 10.1111/cgf.12632

[39] M. Sedlmair, T. Munzner, and M. Tory. Empirical Guidance on Scatterplot and Dimension Reduction Technique Choices. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 19(12):2634–2643, 2013. doi: 10.1109/TVCG.2013.153

[40] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory. A Taxonomy of Visual Cluster Separation Factors. In *Computer Graphics Forum*, vol. 31, pp. 1335–1344. Wiley Online Library, 2012. doi: 10.1111/j.1467-8659.2012.03125.x

[41] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum*, vol. 28, pp. 831–838, 2009. doi: 10.1111/j.1467-8659.2009.01467.x

[42] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000. doi: 10.1126/science.290.5500.2319

[43] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

[44] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research (JMLR)*, 10(66-71):13, 2009.

[45] J. Venna and S. Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006. doi: 10.1016/j.neunet.2006.05.014

[46] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization. *Journal of Machine Learning Research (JMLR)*, 11(2), 2010.

[47] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen. A Perception-Driven Approach to Supervised Dimensionality Reduction for Visualization. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 24(5):1828–1840, 2017. doi: 10.1109/TVCG.2017.2701829

[48] M. Wattenberg, F. Viégas, and I. Johnson. How to Use t-SNE Effectively. *Distill*, 2016. doi: 10.23915/distill.00002

[49] L. Wilkinson, A. Anand, and R. Grossman. Graph-Theoretic Scagnostics. In *Proceedings of the IEEE Information Visualization Symposium*, pp. 157–164, 2005. doi: 10.1109/INFVIS.2005.1532142

[50] L. Wilkinson, A. Anand, and R. Grossman. High-Dimensional Visual Analytics: Interactive Exploration Guided by Pairwise Views of Point Distributions. *IEEE Trans. Visualization & Computer Graphics (TVCG)*, 12(6):1363–1372, 2006. doi: 10.1109/TVCG.2006.94

Cristina Morariu was a researcher at the University of Stuttgart (Germany) in the group of Professor Michael Sedlmair until October 2020. Currently, she works as a Machine Learning Scientist at Amazon. She received an M.Sc. degree in Operational Research with Data Science from Univeristy of Edinburgh in 2017.

Adrien Bibal is a Ph.D. student at the Université de Namur (Belgium) under the supervision of Professor Benoît Frénay. He received an M.S. degree in Computer Science and an M.A. degree in Philosophy from the Université catholique de Louvain (Belgium) in 2013 and 2015 respectively. His Ph.D. thesis in machine learning is on the interpretability of dimensionality reduction mappings.

Benoît Frénay is associate professor at the Université de Namur. He received his Ph.D. degree from the Université catholique de Louvain (Belgium) in 2013. His main research interests in machine learning include interpretability, interactive machine learning, dimensionality reduction, label noise, robust inference and feature selection. In 2014, he received the Scientific Prize IBM Belgium for Informatics for his PhD thesis on Uncertainty and Label Noise in Machine Learning.

VIII. Supplementary Material: Quality Measure Equations

LCMC is defined as

$$LCMC(k) = \frac{1}{n}\sum_{i=1}^{n}|\nu_i^k \cap \rho_i^k|, \tag{3}$$

where n is the number of points and ν_i^k (resp. ρ_i^k) is the set of the k nearest neighbors of the point i in the original data (resp. in the visualization). T&C combines two measures. The first one is the trustfulness of the visualization for a neighborhood size k, which is defined by

$$T(k) = 1 - \frac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in U_k(i)}(r^{HD}(i,j)-k), \tag{4}$$

where $r^{HD}(i,j)$ is the rank of the j^{th} point in terms of distance to the point i in the original data and $U_k(i)$ is the set of the k nearest neighbors of point i in the visualization that are not among the k nearest neighbors of point i in the original data. This metric measures whether we can trust what can be seen in the visualization. The measure of continuity is the exact opposite, as it tells how well the patterns from the original dataset are projected in the visualization. The continuity for a particular neighborhood size k is defined by

$$C(k) = 1 - \frac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in V_k(i)}(r^{LD}(i,j)-k), \tag{5}$$

where $r^{LD}(i,j)$ is the rank of the j^{th} in terms of distance to the point i in the visualization and $V_k(i)$ is the set of the k nearest neighbors of point i in the original data that are not among the k nearest neighbors of point i in the visualization.

While the previously mentioned approaches focus on a specific neighborhood size k, $AUC_{log}RNX$ consider all neighborhood sizes, with a focus on smaller neighborhoods. In order to do so, $AUC_{log}RNX$ considers the neighborhood sizes with a logarithmic importance:

$$AUC_{log}RNX = \left(\sum_{k=1}^{n-2}\frac{R_{NX}(k)}{k}\right) \Big/ \left(\sum_{k=1}^{n-2}\frac{1}{k}\right), \tag{6}$$

where

$$R_{NX}(k) = \frac{(n-1)Q_{NX}(k)-k}{n-1-k}, \tag{7}$$

and where

$$Q_{NX}(k) = \frac{1}{nk}\sum_{i=1}^{n}|\nu_i^k \cap \rho_i^k|. \tag{8}$$

The other community that tackles the measure of visualization quality is the visualization (VIS) community. This community developed well-known measures for the detection of patterns in visualizations (e.g. Scagnostics measures [49], [50]). These types of measures allow users to measure the sparsity in the visualization, the skewness or even the presence of outliers.

More recently, it has been shown that cluster separability measures can match user perception in visualizations [38]. In particular, distance consistency (*DSC*) [41] has been shown to be one of the most performing measures to predict user preferences [38]. These metrics are often supervised, meaning that labels about the instances must be provided in order to assess if the clusters are well separated. *DSC*, for instance, computes the number of instances that are closest to the centroid of another class label then their own. More formally,

$$DSC = \frac{|\mathbf{y}_i \in \mathbf{Y} : CD(\mathbf{y}_i, centr(c_{clabel(\mathbf{y}_i)})) = true|}{n}, \tag{9}$$

where $\mathbf{Y}$ is the set of points in the visualization, n is the total number of points, $centr(c_{clabel(\mathbf{y}_i)})$ computes the virtual point that is at the center of all points with the same class label of $\mathbf{y}_i$) and $CD(\cdot,\cdot)$ computes the distance between two points.

Other popular measures in this category are the average between-within clusters (*ABW*) [26], the hypothesis margin (*HM*) [20], the neighborhood hit (*NH*) [34] and the Calinski-Harabasz index (*CAL*) [11]. All these metrics measure the separability between clusters, albeit differently. *ABW* measures the average distances between clusters on the average distances within clusters:

$$\text{ABW} = \frac{avg_{\mathbf{y}_i \overset{C}{\not\sim} \mathbf{y}_j} dist(\mathbf{y}_i,\mathbf{y}_j)}{avg_{\mathbf{y}_i \overset{C}{\sim} \mathbf{y}_j} dist(\mathbf{y}_i,\mathbf{y}_j)} \quad \forall \mathbf{y}_i,\mathbf{y}_j \in \mathbf{Y}, \tag{10}$$

where $\mathbf{y}_i \overset{C}{\not\sim} \mathbf{y}_j$ means that y_i is not in the same cluster as y_j and $\mathbf{y}_i \overset{C}{\sim} \mathbf{y}_j$ means that the two points are in the same cluster.

HM uses the nearest point from a different cluster (nearmiss) and the nearest point from the same cluster (nearhit) to define the notions of inter and intra cluster distances:

$$HM = \sum_{\mathbf{y}_i \in \mathbf{Y}} \frac{1}{2}(\text{dist}(\mathbf{y}_i, \text{nearmiss}(\mathbf{y}_i)) - \text{dist}(\mathbf{y}_i, \text{nearhit}(\mathbf{y}_i))). \tag{11}$$

NH makes use of a k-nearest neighbor (kNN) classifier to identify if the points in the visualization are close to their centroid (virtual central point of a cluster). *NH* corresponds to the accuracy of the classifier.

Finally, *CAL* is a more complicated measure of the same concepts:

$$CAL = \frac{\text{BG}}{(k-1)} \Big/ \frac{\text{WG}}{(n-k)} = (\bar{d}^2 + \frac{(n-k)}{(k-1)}A_k)/(\bar{d}^2 - A_k), \tag{12}$$

where BG (resp. WG) means between groups (resp. within groups). WG is defined by

$$\text{WG} = \frac{1}{2}\sum_{C_k}(n_{C_k}-1)\bar{d}^2_{C_k}, \tag{13}$$

where C_k is k^{th} class label and $d^2_{C_k}$ is the squared distances of points belonging to the class C_k. BG is defined by

$$\text{BG} = \frac{1}{2}((k-1)\bar{d}^2 + (n-k)A_k), \tag{14}$$

where $\bar{d}^2$ is the average of the squared distances between all points, and with

$$A_k = \frac{1}{(n-k)}\sum_{C_k}((n_{C_k}-1)(\bar{d}^2 - \bar{d}^2_{C_k})). \tag{15}$$

A_k is simply "a weighted mean of the differences between the general and the within-group mean squared distances" [11].

CHAPTER 19

CONSTRAINT PRESERVING SCORE FOR AUTOMATIC HYPERPARAMETER TUNING OF DIMENSIONALITY REDUCTION METHODS FOR VISUALIZATION

The paper presented in this chapter is a non-final version and is soon to be submitted in the journal IEEE Transactions on Artificial Intelligence (TAI).

Constraint Preserving Score for Automatic Hyperparameter Tuning of Dimensionality Reduction Methods for Visualization

Viet Minh Vu, Adrien Bibal, and Benoît Frénay, *Member, IEEE,*

Abstract—In data analysis, visualization through dimensionality reduction (DR) is one of the most effective ways to understand a dataset. However, the hyperparameters of those visualization algorithms are sometimes difficult to tune for end-users. This paper proposes a solution to ease the choice of hyperparameter values for several widely used DR methods like t-distributed stochastic neighbor embedding (t-SNE), LargeVis and uniform manifold approximation and projection (UMAP). We present the *constraint preserving score*, a computationally efficient score, to measure the quality of a visualization. The idea is to measure how well a visualization preserves the information encoded in input pairwise constraints like group information or similarity/dissimilarity relationships between instances. Based on this quantitative measure, we use Bayesian optimization to effectively explore the solution space of all visualizations to find the most suitable one. The proposed score is flexible as it can measure quality in different ways depending on the provided constraints. Experiments show its interest for end-users, its complementarity with existing visualization quality measures and its flexibility to easily express different quality aspects.

Index Terms—Dimensionality Reduction, Visualization, Pairwise Constraints, Hyperparameter Tuning, Bayesian Optimization

1 Introduction

Dimensionality reduction (DR) methods transform the data from a high dimensional (HD) space into a low dimensional (LD) space while preserving relevant structures of the original data. Modern DR methods like t-distributed stochastic neighbor embedding (t-SNE) [1], LargeVis [2] and uniform manifold approximation and projection (UMAP) [3] aim to visualize HD data in order to help users to get insights about their data. These techniques are powerful but they require to carefully tune different hyperparameters, which are often hard to understand for the end-users. Choosing a good hyperparameter value is crucial, since it predetermines the quality and usefulness of the obtained visualization [4], [5]. Typically, the desired visualization result has to be chosen through trial-and-error. This process is tedious, which makes it difficult for the user to find the best suitable visualization.

This paper tackles the problem of automatically choosing the hyperparameter values of DR techniques, such as the *perplexity* of t-SNE. The two major difficulties arising from this problem are the measure of the visualization quality and the search through the hyperparameter space to find the best values. The idea presented in this paper is to use the semantic information encoded in pairwise constraints to measure the quality of visualizations. This is done by transforming the constraints in the form of relationships between object pairs into a quantitative measure. The contributions of this paper to address the above difficulties are the followings. First, we propose a reliable measure called *constraint preserving score* to measure the quality of the embedding of any DR method. This score provides a different aspect of quality w.r.t. to the state-of-the-art visualization quality measures, while being computationally more efficient and flexible. Second, we apply the proposed score under a Bayesian optimization framework [6], [7] to automatically find a range of hyperparameter values corresponding to the best visualizations that respect the user needs.

The approach to find the best hyperparameter values with a score, instead of modifying DR methods, allows us to apply existing techniques to any DR methods. In that sense, the approach is DR-method agnostic. Furthermore, when using constraints for choosing the best hyperparameter values, visualizing these constraints makes it possible to explain the choice of visualization. By explaining how the visualization is chosen, a step towards interpretability of the DR process is also made. The end-users can also use our method as a black-box hyperparameter tuning toolbox. Furthermore, the approach can also be used by DR experts to analyze the impact of hyperparameters on the quality of visualizations.

This paper is organized as follows. Sec. 2 presents the background on DR methods, visualization quality metrics, pairwise constraints in unsupervised learning and an overview of how the automatic hyperparameter selection for DR methods is handled in the literature. Sec. 3 present how to transform the knowledge in the input constraints into the constraint preserving score. The experimental setting for evaluating our proposed method is described in Sec. 4. The main characteristics of the proposed score are empirically proved through the experiments in Sec. 5. We compare our score to other visualization quality metrics in Sec. 6 and show how to apply Bayesian optimization on this score to automate the hyperparameters tuning task in Sec. 7.

• *V.M.Vu, A.Bibal, B.Frénay are with the University of Namur, Belgium. E-mail: { vuvietminh, adrien.bibal, benoit.frenay }@unamur.be*

Manuscript received Xxx xx, 20xx; revised Xxx xx, 20xx.

Finally, Sec. 8 concludes our work.

2 Background and Related Work

This section presents the background and methods related to our work. Sec. 2.1 presents the dimensionality reduction (DR) techniques used in our evaluation (t-SNE [1], LargeVis [2] and UMAP [3]). Sec. 2.2 presents the quality measures used in the literature to assess DR embeddings. Sec. 2.3 describes how user constraints are used in clustering and in DR. Finally, Sec. 2.4 reviews the techniques to choose the hyperparameters DR algorithms.

2.1 Dimensionality Reduction for Visualization

The three visualization methods t-SNE, LargeVis and UMAP are widely used in practice and have the same characteristic of preserving the local structures in the data. They can be summarized in two main steps. First, a neighborhood graph is constructed from the high-dimensional (HD) data. This step requires a parameter to determine the size of the set of k-nearest neighbors (KNN), called *n_neighbors* in UMAP and *perplexity* in t-SNE and LargeVis. This KNN graph is weighted in different ways to transform similarity in the data space into a probability density. Second, the weighted graph represented by a probability density is projected into a low-dimensional (LD) space to obtain the visualization.

Constructing the KNN graph requires pairwise distances between all n instances in d-dimensional space and has a complexity of $O(dn^2)$. t-SNE [1] constructs the exact KNN graph and thus cannot scale with large datasets. Its accelerated version, called Barnes-Hut t-SNE [8], uses a tree-based algorithm to reduce the complexity to $O(dn \log n)$. LargeVis [2] approximates a very accurate KNN graph by using the random projection trees technique to build neighborhood candidates for each data point. In t-SNE and LargeVis, edges in the KNN graph are weighted by an isotropic Gaussian kernel with an adapted bandwidth derived from the perplexity parameter. UMAP [3] has the same idea but with a different theoretical foundation. Indeed, it uses a more sophisticated topological data analysis technique to model local connectivity (similar to the neighborhood graph) by a fuzzy topological structure.

In the embedding space, all three methods simply create a neighborhood graph and, then, transform it to probability density using the Student's t-distribution (UMAP uses a similar but more general function). The second step is to solve the graph layout problem to match the probability densities in HD and LD spaces. t-SNE solves it by minimizing the forward Kullback-Leibler divergence. LargeVis models the probability of obtaining an edge between neighborhood nodes in the LD space and maximizes the log likelihood of this model. UMAP considers the graphs in HD and LD as fuzzy sets and minimizes the cross entropy of those fuzzy sets. All methods use stochastic gradient descent for optimization.

The quality of the output embedding depends heavily on the hyperparameters of these methods, which control the construction of the KNN graph in the HD space and the structure of the KNN graph in the LD space. The perplexity/*n_neighbors* determines the approximate number of neighbors for each data point: small values reveal more local structures, while large values reveal more global structures in the data. UMAP also uses another hyperparameter (*min_dist*) to determine the minimum distance between points in the embedding in order to directly control how tight the groups are formed in the visualization. Our goal in this paper is to automatically tune the hyperparameters for these three methods to find the best visualization (Sec. 3).

TABLE 1: Properties of the five cluster-label-agnostic quality metrics considered in this paper to assess visualizations.

Metric	Range	Description
CC	$[0, 1]$	Pearson correlation coefficient between pairwise distance vectors
NMS	$[0, +\infty[$	Stress based on comparison of pairwise distance orders
CCA	$[0, +\infty[$	Stress with emphasis put on LD
NLM	$[0, +\infty[$	Stress with emphasis put on HD
$AUC[R_{NX}]$	$[-1, 1]$	How neighbors in HD are preserved in LD

2.2 Visualization Quality Metrics

Several metrics exist to evaluate the quality of embeddings. In this paper, clustering-based quality measures are not considered because they need labeled data for measurement. Table 1 summaries the reviewed metrics and some mathematical details are provided in Appendix A. *Correlation coefficient* (CC) [9] compares the pairwise distances in the HD and LD spaces by computing the correlation between the pairwise distance vectors that comprise the distances between all pairs of points in HD and LD. The well-known *Kruskal's non-metric stress* (NMS) [10], often used as the objective function of non-metric multidimensional scaling, is used to compare the pairwise distance orders between the HD and LD spaces. The *curvilinear component analysis stress* (CCA) [11] is a kind of Kruskal's stress with an emphasis on the embedding pairwise distances. This metric evaluates the embedding quality by looking whether if instances in the LD space are close to each other. The *Sammon's non-linear mapping stress* (NLM) [12] is a measure similar to CCA, but focusing on the closeness of instances in the HD space. Finally, the rank-based criteria are used to measure how the neighborhood in HD space is preserved in LD space [13]. Average normalized intersection of the neighborhood sets in the two spaces is calculated for different neighborhood sizes K. These values are arranged with logarithmic scale of K. The area under this curve gives a final score $AUC[R_{NX}]$ that assesses the average DR quality on all scales [14].

2.3 User Constraints for Clustering and DR

Clustering is a machine learning problem whose goal is to find groups (called *clusters*) in the data. User constraints can incorporate domain expertise with the goal of explicitly defining the property of the expected clusters. The pairwise constraints are first introduced in constrained K-Means [15]. Must-link and cannot-link constraints indicate that two instances must be in the same cluster or cannot be in the same cluster, respectively. The popular survey by Davidson et

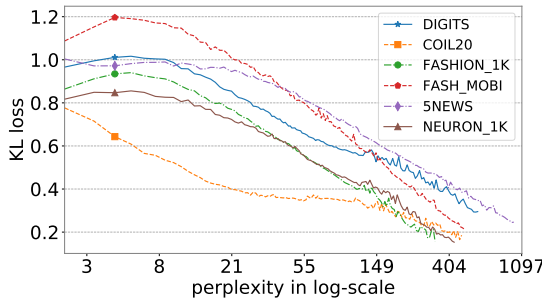


Fig. 1: The KL losses for several datasets tend to decrease systematically when the perplexity increases. The perplexities are shown in logarithmic scale in the range $[2, n/3]$, where n is the number of instances in each dataset.

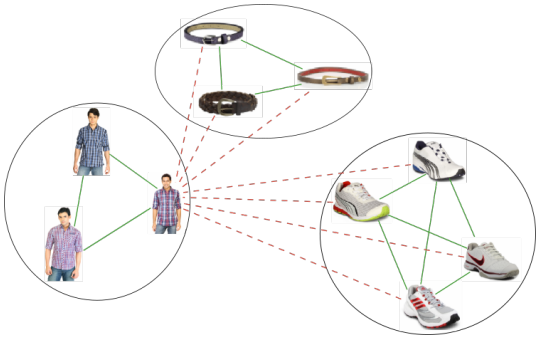


Fig. 2: Examples of the generated pairwise constraints from three different groups of FASHION_1K dataset. Similar link (in green) indicates images in the same groups. Dissimilar link (in red) indicates images of different groups.

al. [16] focuses on *constraint-based* and *distance-based* clustering methods with instance-level constraints. In constraint-based methods, the clusters are formed in such a way that the given constraints are preserved as much as possible [17], [18]. In distance-based methods, the constraints are first used to train a distance function that is later used by a clustering algorithm [19], [20].

Users can also inject constraints in DR methods to force the output visualization to have some expected properties. These *objective* constraints can be partial labels as in semi-supervised latent Dirichlet allocation [21], or constraints on the value of features as in bounded PCA [22]. If users interact with the visualization, they can give feedback in form of instance-level *subjective* constraints. Pairwise constraints are often used to attract points connected by similar links and repulse points connected by dissimilar links. Such constraints are used in pairwise constraint-guided feature projection [23], semi-supervised DR [24], graph-driven constrained DR via linear projection [25] and constrained locality preserving projections [26]. Sacha et al. [27] review more methods for integrating user interaction into DR techniques. Endert et al. [28] propose a wider survey on integrating machine learning into visual analysis.

2.4 Choosing Hyperparameter Values of DR Methods

Choosing hyperparameters of the DR methods depends on characteristics of the dataset such as the number of instances (size), the topology (structure) or the density (distribution) of instances, which makes it hard to select the best one. For instance, the suggested values for t-SNE's perplexity are between 5 and 50 [1]. However, in practice, the embedding can change drastically between two different perplexity values. Therefore, there is no evidence to ensure that the suggested perplexities are good for all datasets. The original t-SNE paper also proposes a simple method to select a good perplexity by looking at the Kullback-Leibler (KL) loss produced by several perplexities and choose the lowest one. However, the KL loss tends to decrease when the perplexity increases [29], which is confirmed by our experiments, as shown in Fig. 1. For this reason, it is not suitable to use the KL loss for evaluating the embedding quality since a very high perplexity would always be chosen. In practice, the users have to manually choose this hard-to-understand hyperparameter, which is often tedious and error-prone.

Few papers in the literature attempt to derive the best hyperparameter values for DR methods automatically. Strickert [30] suggests using rank-based data to avoid perplexity calculation. Lee et al. [31] use a multi-scale approach by averaging all neighborhood sizes. Despite providing visualizations by bypassing the perplexity selection problem, these two solutions do not solve the selection problem itself. Cao and Wang [29] try to tackle the problem by selecting the perplexity of t-SNE that minimizes a modified *Bayesian information criteria* (BIC):

$$BIC = 2KL(P||Q) + \frac{perplexity}{n}\log(n), \qquad (1)$$

where $KL(P||Q)$ is the KL loss in the objective function of t-SNE and n is the number of instances. However, this method is designed for t-SNE only and does not make it possible to inject user knowledge through constraints. In summary, the problem of tuning hyperparameters for complex methods like UMAP or t-SNE is still not solved completely.

3 Constraint Preserving Score

This section presents the proposed constraint preserving score. We first illustrate the pairwise constraints used in this work (Sec. 3.1), then explain how to quantify these constraints to use it as a score (Sec. 3.2).

3.1 Introduction to the User Pairwise Constraints

Humans can often distinguish similar and dissimilar high-dimensional objects (e.g. comparing images by visual features such as the shape, colors or objects in the image) and group these objects by their similarities. For instance, we can easily identify three different groups from the clothing images of the FASHION_1K dataset in Fig. 2, because we can find that the three man's T-shirts look similar, while being different from the shoes and the belts. Consciously, we go from low-level comparison between individual objects

to the higher-level abstraction such as groups of similar objects.

Our idea is to use the information encoded in the pairwise link between objects to evaluate the quality of a visualization. Many modern visualization methods such as t-SNE, LargeVis and UMAP preserve the local structures in the dataset, i.e., similar data points in the HD space should be close together in the embedding space. These methods are considered as successful when they reveal distinguishable groups of similar data points in the resulting visualization. If one knows in advance some patterns in the dataset (e.g., groups of points with the same labels or groups of similar points annotated by human), the quality of the patterns found by these visualization methods can be assessed.

Two types of pairwise constraints can be defined. First, a similar-link constraint (*similar link* for short) indicates that two instances are similar and should be in a same group. Second, a dissimilar-link constraint (*dissimilar link* for short) indicates that two instances are dissimilar and should be in different groups. These pairwise constraints are used to measure how well the local structures are preserved in the embedding space, or in other words, to measure the quality of the visualization.

Local structure preserving property can be also evaluated by measuring how well the neighborhoods in the HD space are preserved in the LD space (e.g. the $AUC[R_{NX}]$ metric). The differences between the two approaches of preserving the pairwise constraints and preserving the neighborhoods are highlighted in Sec. 5.1 and Sec. 6 gives an empirical comparison.

3.2 Defining the Constraints Preserving Score

Given a set of user pairwise constraints, the *constraint preserving score*, called f_{score}, measures how well the pairwise constraints are preserved in a particular embedding. We first propose how to quantify the satisfaction of individual constraints and we then formulate f_{score} based on the set $\mathcal{S}$ of similar links and the set $\mathcal{D}$ of dissimilar links.

Constraint Measurement

We first measure the *strength* of the input pairwise constraints in a given embedding. A similar link should have a high strength and a dissimilar link should have a low strength. Therefore, the strength of a constraint can be measured as the inverse of the distance between two connected points. If a Student's t distribution is placed at the point $\mathbf{y}_i$ in the embedding, the strength of the constraint connecting $\mathbf{y}_i$ to another point $\mathbf{y}_j$ is defined as

$$q_{ij} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}, \quad (2)$$

where the denominator is a normalization constant calculated from all pairs $\{(\mathbf{y}_k, \mathbf{y}_l)\}$ in the embedding.

This formula (or a similar formulation) is used in t-SNE, LargeVis and UMAP to model the neighborhood relationship in the embedding space. q_{ij} can be interpreted as the probability of $\mathbf{y}_i$ and $\mathbf{y}_j$ being neighbors in the embedding space. Therefore, for each similar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}$, q_{ij} should be high. Inversely, q_{ij} is expected to be low for each dissimilar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}$.

Constraint Preserving Score

The amount of information encoded in the similar links that are preserved in a given embedding is measured as a log-likelihood for all similar links $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}$:

$$f_{score}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \log \prod_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}} q_{ij} = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{S}} \log q_{ij}. \quad (3)$$

If all pairs of points connected by a similar link are close, the log-likelihood is high, and so is $f_{score}(\mathcal{S})$.

In contrast, the probability q_{ij} for each dissimilar link $(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}$ should be low. In other words, the negative log-likelihood over all dissimilar links should be large.

$$f_{score}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \log \prod_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}} q_{ij} = -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{y}_i, \mathbf{y}_j) \in \mathcal{D}} \log q_{ij}. \quad (4)$$

The better the embedding respects the dissimilar links, the higher $f_{score}(\mathcal{D})$ is. Another way to measure how well a dissimilar link $(\mathbf{y}_i, \mathbf{y}_j)$ is preserved is to use $1 - q_{ij}$. However, in practice, the value of q_{ij} is very small, meaning that $1 - q_{ij}$ is close to one, which makes the log-likelihood of all dissimilar links vanish.

The final constraint preserving score is defined as a combination with equal contribution of both similar links and dissimilar links

$$f_{score}(\mathcal{S}, \mathcal{D}) = \frac{1}{2} f_{score}(\mathcal{S}) + \frac{1}{2} f_{score}(\mathcal{D}). \quad (5)$$

$f_{score}(\mathcal{S}, \mathcal{D})$ is written as f_{score} for short. An embedding that retains as much as possible the constraint information f_{score} is considered to have a good quality with respect to the user pairwise constraints.

4 Experimental Setup

We propose to use f_{score} to assess the visualizations and we demonstrate how to use this score to find the best hyperparameters of three DR methods (t-SNE, LargeVis and UMAP). Sec. 4.1 describes the experimental setup for evaluating f_{score} with six datasets. Sec. 4.2 presents the pairwise constraints used in our experiments. Sec. 4.3 then presents the evaluation protocol to analyze the characteristics of f_{score} and to compare it with other quality metrics.

4.1 Experimental Datasets

Six datasets of gray-scale and color images, text and gene expressions are used to evaluate our score. DIGITS is a subset of the optical recognition of handwritten digits dataset of 8x8 gray-scale images [32]. COIL20 is a dataset of 32x32 gray-scale images of 20 rotated objects [33]. FASHION_1K contains 1000 gray-scale 28x28 images sampled from the Fashion-MNIST clothing dataset [34]. *FASHION_MOBILENET* (FASH_MOBI for short) contains 1494 color images of the seven most numerous classes sampled from the real-world fashion product images dataset [35]. To extract features for this dataset, MobileNet [36] is used with pre-trained weights where the last fully connected layer is replaced by a global average pooling layer to obtain a flattened output vector of 1280 dimensions. For these four image datasets, PCA is applied to keep 90% variance of the

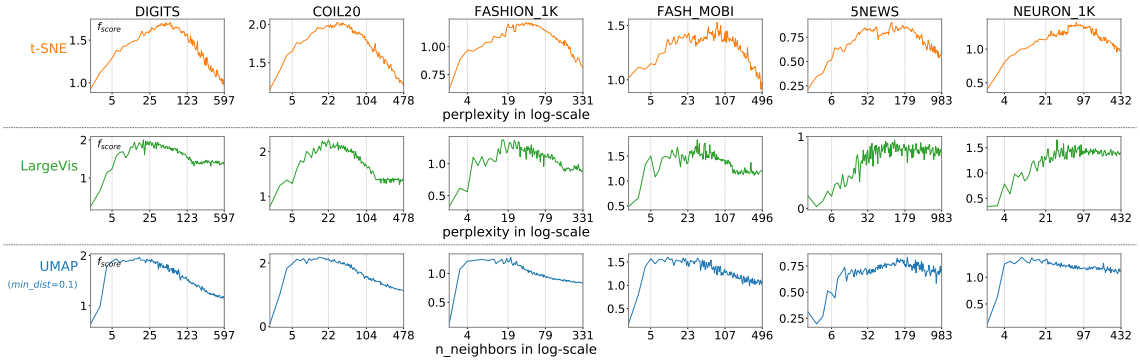


Fig. 3: Evolution of f_{score} with respect to the hyperparameter of three DR methods for six experimental datasets.

data. This helps us speed up the computation of pairwise distances and reduce the noise of outliers if they exist.

5NEWS dataset contains the text of 5 groups selected from the 20 Newsgroups dataset. The text is converted into a matrix of token counts via the term frequency inverse document frequency method. The count vectors are then fed into a latent Dirichlet allocation model [37] to extract 15 hidden topics, which are the 15 features used by the DR methods.

The last real-world dataset is the open NEURON_1K dataset [38], which contains 1301 brain cells from an E18 mouse. These cells have been processed and provided by 10X Genomics.The processed data have 10 PCA features and 6 labels found by a graph-based clustering method.

4.2 Constraint Generation

The input of our constraint preserving score is a set of constraints in the form of similar and dissimilar links. As shown in Sec. 3.2, the pairwise constraints can be generated from the groups of selected instances. Users can group the instances that they find similar to indicate that the instances in the same group should be connected by similar links. Similarly, instances in different groups indicates that they should be connected by dissimilar links. In order to objectively evaluate the proposed score, pairwise constraints generated from labeled instances are used throughout our experiments.

Given a dataset of C classes, k labeled instances are randomly selected for each class. Similar links are created for all possible pairs of these k instances, leading to $|\mathcal{S}| = Ck(k-1)/2$ pairs of constraints.

The dissimilar links are formed by first choosing two different classes among the C classes ($\binom{C}{2}$ ways), and then choosing a pair of two instances from these classes ($k^2/2$ unique pairs). The number of all possible dissimilar links is therefore given by $|\mathcal{D}| = C(C-1)k^2/4$.

4.3 Evaluation Protocol

This section demonstrates the characteristics of f_{score} and compares it with other scores. First, a grid of hyperparameters is created for each of the three methods (t-SNE, LargeVis and UMAP). For t-SNE and LargeVis, the grid is an integer vector of perplexity values in $[2, n/3]$. For UMAP, the two-dimensional grid is created from an integer vector of *n_neighbors* values in $[2, n/3]$ and a vector of 10 real values of *min_dist* in $[0.001, 1.0]$. All hyperparameter values are sampled in a natural logarithmic scale. Second, the embedding for each combination of hyperparameters in each grid is calculated. Third, f_{score}, $AUC[R_{NX}]$ and the BIC-based score (if applicable) are computed for each embedding.

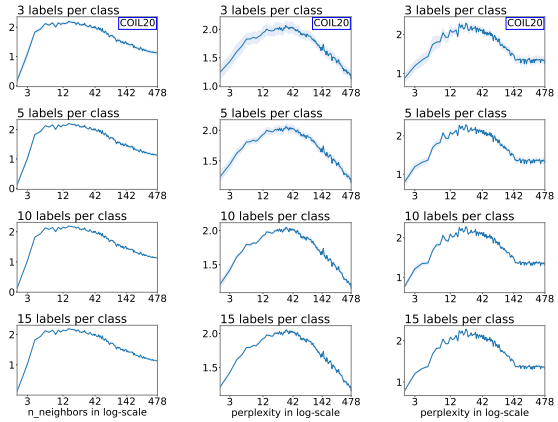
The grid of hyperparameters is only used to empirically analyze the characteristics of f_{score} (Sec. 5) and to compare it with two other scores (Sec. 6). After showing that f_{score} is reliable, the score is used to measure the quality of the visualization. Finding the best visualization is done by searching for the hyperparameter that maximizes f_{score}. Instead of greedily searching through the grid of hyperparameters, the use of Bayesian optimization is proposed (Sec. 7).

5 Characteristics of f_{score}

Our experiments show that f_{score} has three important characteristics: it is a *well-behaved* function of the input visualization (Sec. 5.2), it is stable w.r.t. the number of input labeled instances (Sec. 5.3) and it is flexible w.r.t different sets of input constraints (Sec. 5.4).

5.1 Computational Efficiency

f_{score} requires an extra amount of labeled data but the computation is simple and efficient. Supposing that we have a dataset of n data points in d-dimensional space. f_{score} has a computational complexity of $O(n^2)$ since it only requires access to the pairwise distances between embedded points. Furthermore, the summation over all input pairwise constraints can be efficiently vectorized via matrix slicing operations. In contrast, $AUC[R_{NX}]$ must access to both HD data and the embedding. It has a high complexity of $O(dn^2log(n))$ and may not be applicable for large datasets. The BIC-based score, despite its simplicity, can only be used for t-SNE. For an embedding not generated by t-SNE, it requires to compute t-SNE's KL loss, which involves the HD data and has a complexity of $O(dn^2)$. The proposed f_{score} is agnostic w.r.t. the DR method and is computationally more efficient.



(a) UMAP (b) t-SNE (c) LargeVis

Fig. 4: Stability of f_{score} with the embeddings of UMAP (a), t-SNE (b) and LargeVis (c) for COIL20 dataset. The mean (blue line) and variance (filled region around the line) is calculated for each perplexity/n_neighbors with different number of labeled instances per class (3, 5, 10 and 15).

5.2 Well-behaved Function

In the previous section, f_{score} was analyzed as a function of a given embedding. f_{score} is now analyzed as a function of perplexity/*n_neighbors*, the most important hyperparameter of the studied DR methods. Fig. 3 shows the behavior of f_{score} for t-SNE, LargeVis and UMAP on six datasets. The hyperparameter values are shown in logarithmic scale. The score values for the embeddings of UMAP are shown only with a fixed *min_dist* = 0.1. Pairwise constraints are generated from 10 labeled instances per class.

We found that f_{score} has the form of a convex-like function of perplexity/*n_neighbors*. f_{score} is a well-behaved function that increases when the number of neighbors (perplexity/*n_neighbors*) increases, then reaches its maximum value, and finally decreases when the number of neighbors is too large. This result is also true when evaluating f_{score} as a function of two parameters (*n_neighbors* and *min_dist*) for UMAP embeddings. This function is not smooth but it is feasible to find a global maximum.

In the case of LargeVis, there are flat regions where f_{score} does not change too much. The reason is that LargeVis is designed for large datasets and, thus, when applied to medium-sized datasets, the impact of the perplexity is not that important. In contrast, t-SNE and UMAP are very sensitive to their hyperparameters. The experiment results of Sec. 6 and Sec. 7 are focused on t-SNE and UMAP.

5.3 Stability

This section investigates the number of labeled instances per class that are needed to obtain a reliable f_{score}. f_{score} is evaluated with different numbers of labeled instances (3, 5, 10 and 15) per class. The sets of labeled instances are independent and not accumulated. In each setting, f_{score} is repeatedly evaluated 20 times with different sets of pairwise constraints. Mean and variance values of f_{score} for t-SNE, LargeVis and UMAP (with *min_dist* of 0.1) embeddings for the COIL20 dataset are shown in Fig. 4. When the number of labeled instances increases, f_{score} is more stable since the variance decreases. One can also observe that the region where f_{score} has a high value is stable for different number of constraints. This result is shown for COIL20, but also holds for the other datasets. In other words, f_{score} is stable w.r.t the number of input labeled instances. For the remaining of this paper, 10 labeled instances per class will be used to calculate f_{score} in all experiments, since it is a reasonable small number of labels and the variance of score value is negligible.

5.4 Flexibility

In contrast to other DR quality measures, f_{score} is flexible, in the sense that it changes with the input constraints. In most cases, the constraints generated from class labels reflect naturally the class-relationship between the instances. However, if users want to see different patterns from their data, they can specify different constraints to describe what they need. This section provides concrete examples with t-SNE embeddings for three real-world datasets. 10 labeled instances per class/group are used.

A first example is for the FASH_MOBI dataset with seven sub-categories. The best visualization (perplexity of 60) presents seven detached sub-groups as shown in the top-left plot of Fig. 5a. If the user wants to see more abstract, general groups, they can form higher-level groups:

- { *Bag, Jewellery, Watches* } $\rightarrow$ *Accessories*,
- { *Sandal, Shoes* } $\rightarrow$ *Footwear*,
- { *Topwear, Bottomwear* } $\rightarrow$ *Apparel*.

The previously-chosen visualization did not reveal these three higher-level groups. The new best perplexity (113) better reveals this structure as shown in the bottom-right corner of Fig. 5a.

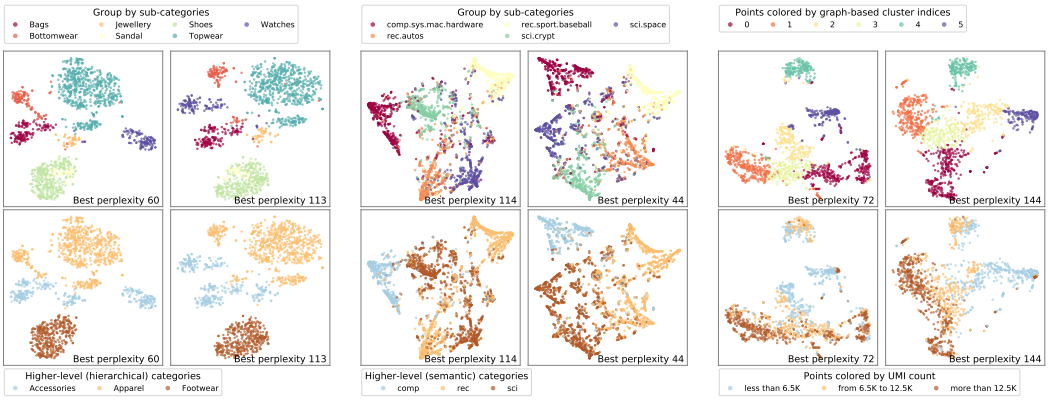
A second example focuses on semantic labels for the textual 5NEWS dataset. The five original classes can be regrouped into three semantic general topics:

- { *rec.autos, rec.sport.baseball* } $\rightarrow$ sportive records (*rec*),
- { *sci.space, sci.crypt* } $\rightarrow$ scientific group (*sci*),
- *comp.sys.mac.hardware* stays in its own group (*comp*).

The problem of the visualization found with the constraints generated from the original class labels is that the global structure is not always revealed. For instance, two subgroups of the same topic can be placed far apart (bottom-left of Fig. 5b). By using the new constraints generated from the three above semantic groups, f_{score} finds a better visualization in which elements in these semantic groups are placed close to each other (bottom-right of Fig. 5b).

The last example is for the genetic NEURON_1K dataset. The original 1301 cells are grouped into 6 classes found by a graph-based clustering algorithm. These classes are characterized by the transcriptome profiles of individual cells (presented in the RNA sequences). However, another important aspect to characterize individual cells is the count of absolute number of molecules: the *unique molecular identifier* (UMI) [39]. Therefore, the cells can be regrouped into three new groups:

- the ones with less than 6.5K molecules,



(a) FASH_MOBI dataset (b) 5NEWS dataset (c) NEURON_1K dataset

Fig. 5: Flexibility of f_{score} demonstrated with t-SNE embeddings for three selected datasets. Each dataset is shown in four plots. Two plots on the left are the same visualization: the best one found by f_{score} with the original labels. Two plots on the right are the same visualization: the best one found by f_{score} with the labeled instances from the higher-level categories. The plots in the top row are colored by the original categories, while the ones in the bottom row are colored by the higher-level categories. It can be observed that different sets of labels makes it possible to select different kinds of visualizations.

- the ones having from 6.5K to 12.5K molecules,
- the ones with more than 12.5K molecules.

Fig. 5c illustrates the visualizations found by f_{score} with the constraints generated from the original graph-based clusters and from the new groups. One should note that f_{score} finds can find a perplexity value reflecting the *UMI count* in the visualization, other quality scores cannot.

6 Comparison with Other Quality Scores

This section qualitatively compares the best visualizations found by f_{scores} and by two other metrics. f_{score} is compared with $AUC[R_{NX}]$ and the BIC-based score for evaluating t-SNE embeddings (Sec. 6.1). f_{score} is also compared with $AUC[R_{NX}]$ for evaluating UMAP embeddings (Sec. 6.2).

6.1 Comparison of f_{score} with $AUC[R_{NX}]$ and the BIC-based Score for t-SNE

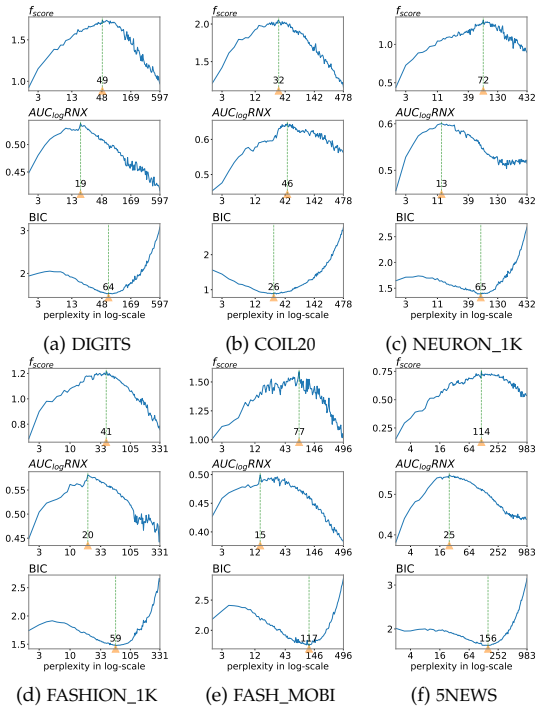
Fig. 6 shows that, for the six selected datasets, f_{score} agrees with $AUC[R_{NX}]$, the BIC-based score or both of them. In order to compare thoroughly the best solutions found by these scores, metamaps are used for visualizing the solution space of DR methods. Each point in the metamap is a t-SNE embedding corresponding to a perplexity value. Two points close to each other in the metamap correspond to perplexities that provide similar visualizations. (See *VisCoDer* [40], a tool using metamaps to discover and compare embeddings of different DR methods). The metamaps are built using UMAP (*n_neighbors*=50, *min_dist*=0.1).

Fig. 8 shows the metamaps for NEURON_1K and highlights several visualizations selected by different scores. The four metamaps are colored by the values of perplexity, f_{score}, $AUC[R_{NX}]$ and the BIC-based score. The 10% of embeddings with the highest scores are highlighted. It is clear that the three scores reveal different visualizations. This means that the different scores can select visualization with different qualities. This is in line with Wattenberg et al. [4], who state that we need more than one visualization to understand the hidden patterns in HD data. The visualizations at the bottom of Fig. 8 serve as a qualitative evaluation of the best visualizations found by the three scores.

6.2 Comparison of f_{score} with $AUC[R_{NX}]$ for UMAP

The following analysis considers two hyperparameters *n_neighbors* and *min_dist* to evaluate f_{score} and $AUC[R_{NX}]$ for UMAP embeddings (Fig. 7). For three datasets (DIGITS, COIL20 and FASHION_1K), the evolution of f_{score} is clearer and smoother than the one of $AUC[R_{NX}]$. For NEURON_1K, the two scores discover different optimal regions. For FASH_MOBI and 5NEWS, $AUC[R_{NX}]$ reveals clearer regions of best hyperparameters, but it likely gives the same score for different *min_dist*s while *n_neighbors* is fixed. In contrast, f_{score} discovers the influence of *min_dist* in conjunction with *n_neighbors*. The combination of these two hyperparameters is important for UMAP embeddings, since while *n_neighbors* controls local structures (the size of local neighborhoods), *min_dist* controls directly how tight are the groups in the visualization.

Fig. 9 shows the metamaps for UMAP embeddings of COIL20 and several selected visualizations. f_{score} considers the first visualization (a) as the best one. The next two visualizations are considered good by $AUC[R_{NX}]$, but not by f_{score}. In the second one (b), the groups clearly highlight the local structures, but are not tight enough to reveal the global structures. In the third one (c), the groups are



(a) DIGITS (b) COIL20 (c) NEURON_1K

(d) FASHION_1K (e) FASH_MOBI (f) 5NEWS

Fig. 6: Comparison of f_{score}, $AUC[R_{NX}]$ and the BIC-based score for t-SNE embeddings. (b),(d): f_{score} agrees with both two other scores. (e): It agrees with neither of them. (a), (c) and (f): It agrees only with the BIC-based score.The best perplexity selected by each score marked by the green vertical line gives an idea of what is the good range of perplexity according to each score.

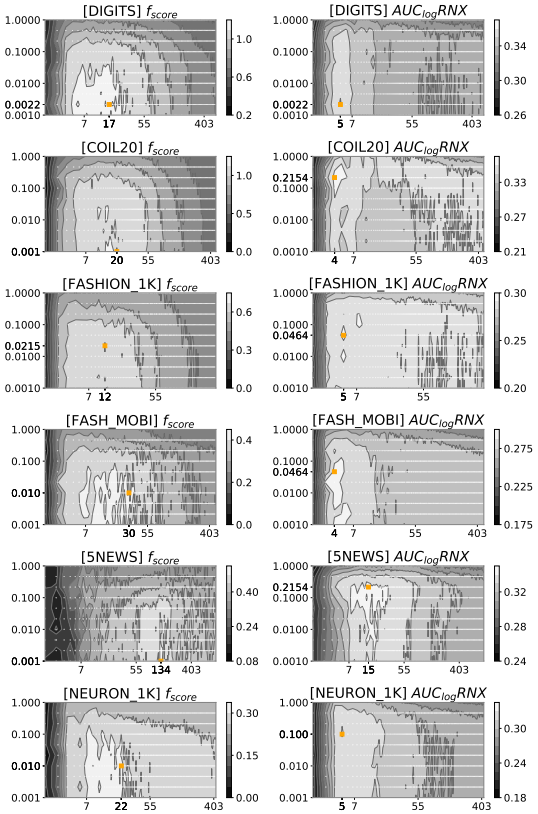


Fig. 7: Comparison of f_{score} (on the left) and $AUC[R_{NX}]$ (on the right) for UMAP embeddings. The best combination of hyperparameters found by each score is denoted by the orange point in each dataset. In each plot, *n_neighbors* (on the horizontal axis) and the *min_dist* (on the vertical axis) are shown in logarithmic scale. The light/dark region corresponds to the large/small values of the two scores.

retracted and heavily overlapped. This visualization has a high $AUC[R_{NX}]$ score since the neighborhood information is well preserved, while the visualization is actually not clear. However, this same visualization is discouraged by f_{score}. The last visualization (d) belongs to the low score region in the metamap (considered by both two scores), which corresponds to the combination of too large *n_neighbors* and/or a too large *min_dist*.

In the previous experiments, it has been shown that f_{score} differs from other scores in its simplicity, efficiency and flexibility. Assuming that we have small amount of labeled data (10 labeled points for each class) to generate a fixed set of input pairwise constraints, we now tackle the problem of finding the best hyperparameters of DR methods by using f_{score}.

7 Bayesian Optimization for Hyperparameter Tuning with f_{score}

This section considers how to search through all combinations of hyperparameters to find the one with a maximum score. We propose to use Bayesian optimization (BayOpt) to solve this problem. Sec. 7.1 introduces the advantages of this approach. Sec. 7.2 and Sec. 7.3 evaluate the task of tuning one hyperparameter for t-SNE and two hyperparameters for UMAP using the proposed f_{score}.

7.1 Hyperparameter Tuning and Bayesian Optimization

Hyperparameters of DR methods can be tuned by trial-and-error or through a naive grid search. A better approach exists, such as random search [41], which randomly samples combinations of hyperparameters. However, the parameter space in which the search takes place grows exponentially w.r.t. the number of hyperparameters.

Bayesian optimization (BayOpt) is a strategy for finding the extremum (minimum or maximum) of an objective function f with as few evaluations as possible [6]. The objective function can be any complex non-convex black-box function that does not have a closed-form expression, or its derivative may not be accessible. The goal of BayOpt is not to approximate this unknown function, but instead to estimate its maximum from a set of observed input samples and func-

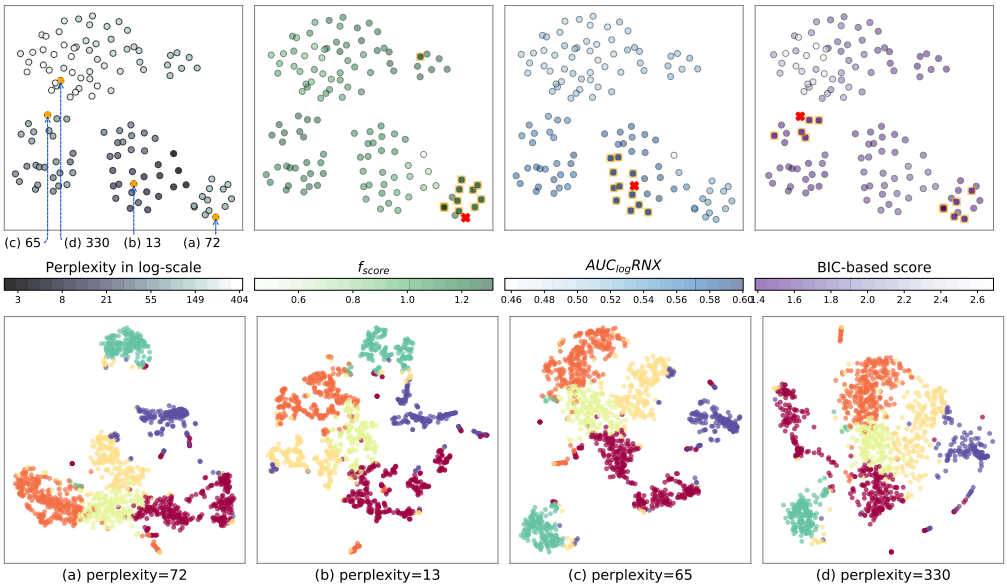


Fig. 8: Metamaps and sample visualizations for NEURON_1K. The top 10% highest scores in the metamap according to each metric are highlighted on the top row. On the bottom row, the visualizations are chosen using f_{score} (a), using $AUC[R_{NX}]$ (b), using the BIC-based score (c). The last one (d) is not considered good by any of the four scores.

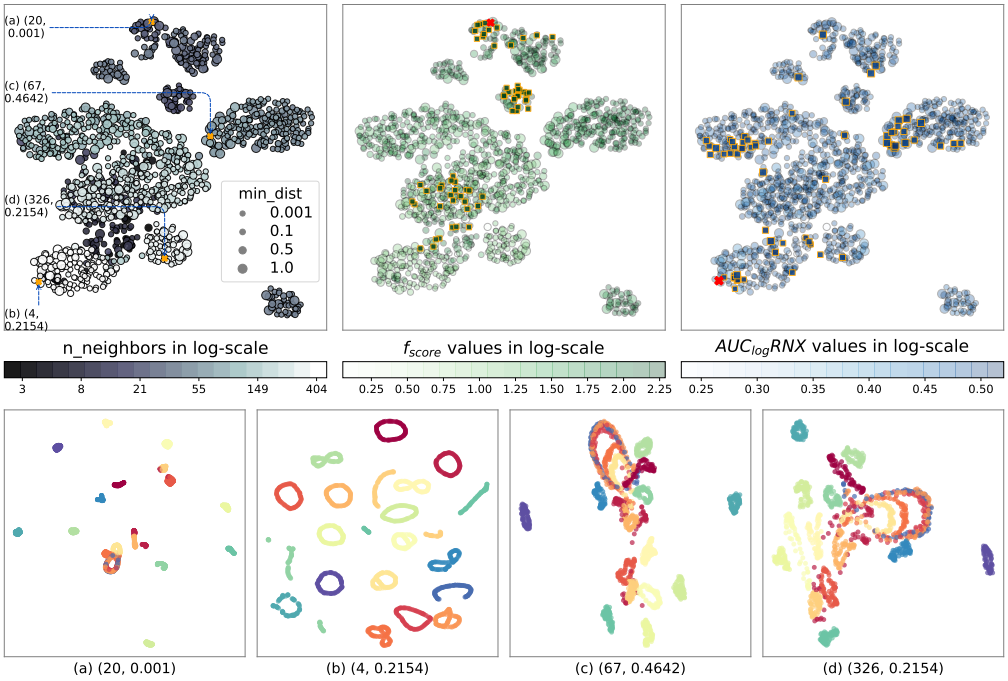


Fig. 9: Metamaps and sample visualizations for COIL20. The top 5% highest scores in the metamap according to each metric are highlighted on the top row. On the bottom row, (a) is chosen by f_{score}, (b) is chosen by $AUC[R_{NX}]$. (c) is considered good by $AUC[R_{NX}]$ but not by f_{score} and (d) is not considered good by any score.

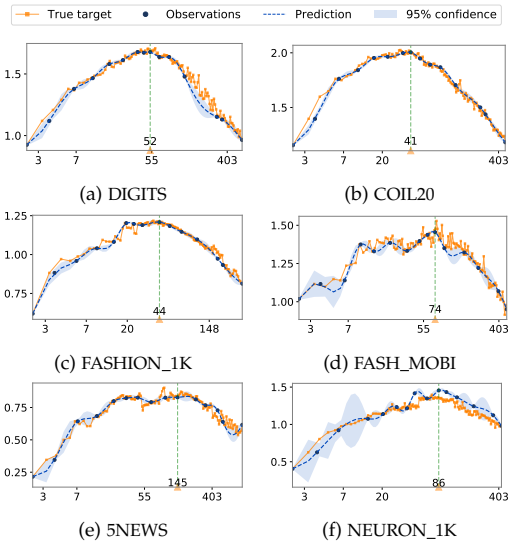


Fig. 10: Tuning t-SNE's perplexity for six datasets using BayOpt. f_{score} is evaluated only for the embeddings of 15 selected perplexities shown by the dark blue points. The dotted blue line presents the predicted f_{score} for all other perplexities. The filled blue region represents the uncertainty of the prediction. The green vertical line indicates the best predicted perplexity. The orange lines are the true values of f_{score}, only used as references to see how well the BayOpt prediction approximates the true target values.

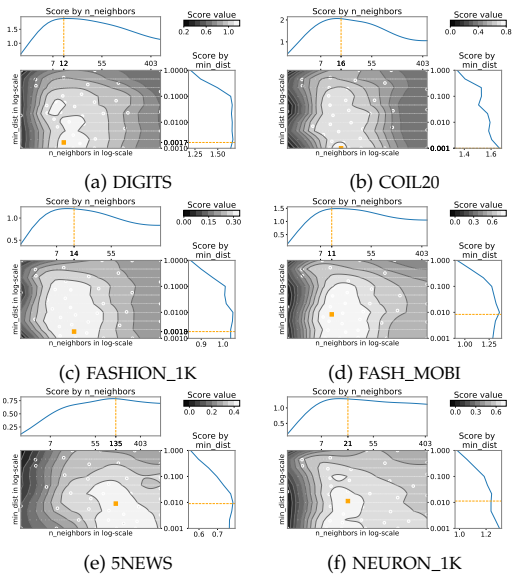


Fig. 11: Tuning two hyperparameters of UMAP using BayOpt. In each plot, 40 points (combinations of *n_neighbors* and *min_dist*) are evaluated and shown by the white dots. The contour plots are constructed from the predicted f_{score} for all other points in the grid. The light/dark region corresponds to the large/small values of f_{score}. The orange points indicate the best predicted hyperparameters.

tion values. BayOpt constructs a statistical model describing the relationship between the tuned hyperparameters and the target function. Based on past observations, BayOpt predicts the most promising hyperparameters to evaluate. There is a trade-off between exploration and exploitation, several strategies exist to guide the optimization process to discover the parameter space: maximum probability of improvement, expected improvement and lower or upper confidence bound [42]. BayOpt successfully solves the problem of hyperparameters tuning for classification [43] or experimental design/randomized experiments [44].

In this work, the objective function to maximize under the BayOpt framework is f_{score}. The exploration strategy is chosen such that it ensures the discovery of the largest parameter space possible. Expected Improvement (EI) acquisition function is thus a good choice for the surrogate function of BayOpt. This function maximizes the expected improvement over the current best parameters and has proven its efficiency in practice [43]. In EI, the parameter ξ controls the trade-off between global search (exploration) and local optimization (exploitation). ξ is set to a large value (0.1) in order to put the importance on the exploration strategy. Since there is always a small variance in the f_{score} value, the BayOpt approach takes into account this type of uncertainty by adding small values to the diagonal of the kernel function of the underlying Gaussian process model in order to make it more robust to noise.

7.2 Tuning One Hyperparameter for t-SNE

Fig. 10 demonstrates how BayOpt works for tuning t-SNE's perplexity for all six selected datasets. The *true target* is the score values for each perplexity and is used only as a reference to compare with the estimate score values predicted by BayOpt. For datasets of various sizes (from 1000 to around 3000 instances), the scores needs to be evaluated for only 15 selected perplexities. These perplexity values are selected by BayOpt iteratively, starting with five random perplexities. The pairs of perplexity and the corresponding f_{score} are used to update the BayOpt model at each iteration. The next predicted perplexity to evaluate is the most promising perplexity value that does not decrease f_{score}. It should be noted that BayOpt does not explicitly approximate the score function, but it tries to find the maximum value instead. BayOpt does not only find the best hyperparameter values, but also indicates the region in which it is not certain about its prediction, which is usually the region of too high or too low perplexity values.

7.3 Tuning Two Hyperparameters for UMAP

Tuning hyperparameters for UMAP is a more difficult task, since the hyperparameter grid is much larger than the one of t-SNE. Instead of evaluating thousands of combinations of values for two hyperparameters, BayOpt converges only after 40 iterations for all six experimented datasets. Fig. 11 demonstrates how BayOpt works to find the region of best combinations for the six datasets. The uncertainty of BayOpt

prediction is not shown in this plot. In comparison with the fully evaluated grid of f_{score} in Fig. 7, it is clear that BayOpt can approximate the region of highest score efficiently with a very limited number of evaluations.

In practice, BayOpt is used to tune multiple hyperparameters. Contour plots of every pair of hyperparameters are used to investigate the region with the best combinations. One advantage of the BayOpt approach is that it does not only maximize the target score function, but it also gives predicted scores for all hyperparameter combinations. Indeed, in each plot in Fig. 11, only 40 points are exactly evaluated. The contour is calculated upon the predicted value of the BayOpt's underlying Gaussian process model for all other points. Without spending too much resources to obtain a full grid, the estimated score given by BayOpt is reliable enough to point out the best hyperparameters.

8 Conclusion and Future Work

This work tackles the problem of automatically tuning the hyperparameters of DR methods, which requires to search through all visualizations and rank them by their quality in order to find the best one. A new constraint-based score is introduced to measure the quality of visualizations by evaluating how well the information encoded in input pairwise constraints is preserved in this visualization. Our proposed score, f_{score}, is a simple, efficient and flexible quality metric. It does not require to calculate neighborhood information in the HD space or the expensive objective function of a non-linear DR method. Furthermore, we show that this score is complementary to other quality metrics, while being flexible (as the score can change w.r.t. to what users expect to see) and cheaper to compute. Based on this score, we propose to use Bayesian optimization to efficiently find the best hyperparameters instead of traditional search-based methods. With an additional information of labeled data, the proposed workflow facilitates the use of DR methods by making the choice of difficult-to-understand hyperparameters easier. In practice, our methodology helps users to discover different visualizations with various perspectives on the structure of data.

In future work, we plan to evaluate the quality of the selected visualization through a user-based experiment. Users' feedback could also be directly incorporated into the BayOpt framework to accelerate the convergence of the optimization. Another perspective is to modify f_{score} in order to consider new types of constraints based on a contrastive loss [45] or a triplet loss [46].

References

[1] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[2] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. WWW*, Montréal, Canada, Apr. 2016, pp. 287–297.

[3] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[4] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, vol. 1, no. 10, p. e2, 2016.

[5] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[6] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, G. I. Marchuk, Ed., 1975, pp. 400–404.

[7] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, pp. 117–128, 1978.

[8] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014.

[9] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1098–1107, 2005.

[10] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[11] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, 1997.

[12] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.

[13] J. A. Lee, E. Renard, G. Bernard, P. Dupont, and M. Verleysen, "Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation," *Neurocomputing*, vol. 112, pp. 92–108, 2013.

[14] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, "Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure," *Neurocomputing*, vol. 169, pp. 246–261, 2015.

[15] K. Wagstaff, C. Cardie, S. Rogers, and S. e. a. Schrödl, "Constrained k-means clustering with background knowledge," in *Proc. ICML*, Williamstown, MA, USA, Jun. 2001, pp. 577–584.

[16] I. Davidson and S. Basu, "A survey of clustering with instance level constraints," in *Proc. ACM TKDD*, 2007, pp. 1–41.

[17] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proc. SIAM SDM*, Brighton, UK, Nov. 2004, pp. 333–344.

[18] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proc. SIAM SDM*, Houston, Texas, USA, Nov. 2005, pp. 138–149.

[19] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. ICML*, Washington DC, USA, Aug. 2003, pp. 11–18.

[20] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. NIPS*, Vancouver, Canada, Dec. 2003, pp. 521–528.

[21] M. Sugiyama, T. Idé, S. Nakajima, and J. Sese, "Semi-supervised local fisher discriminant analysis for dimensionality reduction," *Machine Learning*, vol. 78, pp. 35–61, Jan. 2008.

[22] P. Giordani and H. A. Kiers, "Principal component analysis with boundary constraints," *Journal of Chemometrics*, vol. 21, no. 12, pp. 547–556, Oct. 2007.

[23] W. Tang and S. Zhong, "Pairwise constraints-guided dimensionality reduction," in *Computational Methods of Feature Selection*. Chapman & Hall, 2007, pp. 295–312.

[24] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proc. SIAM SDM*, Minnesota, USA, Apr. 2007, pp. 629–634.

[25] I. Davidson, "Knowledge driven dimension reduction for clustering." in *Proc. IJCAI*, California, USA, Jul. 2009, pp. 1034–1039.

[26] H. Cevikalp, J. Verbeek, F. Jurie, and A. Klaser, "Semi-supervised dimensionality reduction using pairwise equivalence constraints," in *Proc. VISAPP*, Funchal, Portugal, Jan. 2008, pp. 489–496.

[27] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 241–250, 2017.

[28] A. Endert, W. Ribarsky, C. Turkay, B. Wong, I. Nabney, I. D. Blanco, and F. Rossi, "The state of the art in integrating machine learning into visual analytics," *Computer Graphics Forum*, vol. 36, no. 8, pp. 458–486, 2017.

[29] Y. Cao and L. Wang, "Automatic selection of t-SNE perplexity," in *ICML AutoML Workshop*, Sydney, Australia, Oct. 2017, pp. 1–7.

[30] M. Strickert, "No perplexity in stochastic neighbor embedding," in *Workshop New Challenges in Neural Computation*, Graz, Austria, Aug. 2012, pp. 68–115.
[31] J. A. Lee, D. H. Peluffo-Ordóñez, and M. Verleysen, "Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction," in *Proc. ESANN*, Bruges, Belgium, Apr. 2014, pp. 177–182.
[32] C. Kaynak, "Methods of combining multiple classifiers and their applications to handwritten digit recognition," *Unpublished master's thesis, Bogazici University*, 1995.
[33] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Tech. Rep., 1996.
[34] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
[35] Kaggle Open Datasets. (2019) Fashion product images dataset. [Online]. Available: https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset
[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[38] 10X Genomics. (2018) 1k brain cells from an e18 mouse. [Online]. Available: https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/neuron_1k_v3
[39] T. Kivioja, A. Vähärautio, K. Karlsson, M. Bonke, S. Linnarsson, and J. Taipale, "Counting absolute number of molecules using unique molecular identifiers," *Nature Proceedings*, pp. 1–18, 2011.
[40] R. Cutura, S. Holzer, M. Aupetit, and M. Sedlmair, "Viscoder: A tool for visually comparing dimensionality reduction algorithms," in *Proc. ESANN*, Bruges, Belgium, Apr. 2018, pp. 105–110.
[41] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. NIPS*, Grenada, Spain, Dec. 2011, pp. 2546–2554.
[42] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint*, 2010.
[43] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proc. NIPS*, Nevada, USA, Dec. 2012, pp. 2951–2959.
[44] B. Letham, B. Karrer, G. Ottoni, E. Bakshy *et al.*, "Constrained bayesian optimization with noisy experiments," *Bayesian Analysis*, vol. 14, no. 2, pp. 495–519, 2019.
[45] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," in *Proc. ICLR*, Vancouver, Canada, May. 2018.
[46] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Boston, USA, Jun. 2015, pp. 815–823.

APPENDIX

Let d_{ij}^x and d_{ij}^y be, respectively, the distance between instances i and j in HD and LD. Let d^x and d^y be the distances matrices for all pair of points in HD and LD. Here are the mathematical formulas for the five selected metrics.

- The Correlation Coefficient is defined as:

$$\text{CC} = \text{pearson\_correlation}(d^x, d^y) = \frac{\text{Cov}(d^x, d^y)}{\sigma(d^x)\sigma(d^y)}$$

- For measuring the distance order in NMS, an isotonic transformation d^{iso} is performed on d^x. The Kruskal's stress is then computed using this transformation:

$$\text{NMS} = \sqrt{\frac{\sum_{ij}(d_{ij}^{iso} - d_{ij}^y)^2}{\sum_{ij} d_{ij}^y}}$$

- The Curvilinear Component Analysis Stress function is defined as:

$$\text{CCA} = \sum_{ij}(d_{ij}^x - d_{ij}^y)^2 F_\lambda(d_{ij}^y),$$

in which $F_\lambda(d_{ij}^y)$ is a decreasing-weighting function of d_{ij}^y. Examples of weighting functions include the step function or $1 - sigmoid(d_{ij}^y)$.

- The stress function of Sammon's Nonlinear mapping is:

$$\text{NLM} = \frac{1}{\sum_{ij} d_{ij}^x} \sum_{ij} \frac{(d_{ij}^x - d_{ij}^y)^2}{d_{ij}^x}$$

- The quality measure $AUC[R_{NX}]$ can be defined as follows. Let k be the number of neighbors considered, n the number of instances, ν_i^k the set of the k closest neighbors of i in the embedding and ρ_i^k the set of the k closest neighbors of i in the HD space, Q_{NX} is defined as:

$$Q_{NX}(k) = \frac{1}{nk} \sum_{i=1}^{n} |\nu_i^k \cap \rho_i^k|$$

$R_{NX}(k)$, the rescaled version of $Q_{NX}(k)$, is defined as:

$$R_{NX}(k) = \frac{(n-1)Q_{NX}(k) - k}{n - 1 - k}$$

$AUC[R_{NX}]$ is computed by taking the area under the $R_{NX}(k)$ curve in the log-scale of k:

$$AUC[R_{NX}] = \left(\sum_{k=1}^{n-2} \frac{R_{NX}(k)}{k}\right) / \left(\sum_{k=1}^{n-2} \frac{1}{k}\right)$$

CHAPTER 20

AN INTERACTIVE TECHNIQUE FOR EXPLAINING VISUAL CLUSTERS IN DIMENSIONALITY REDUCTION VISUALIZATIONS WITH DECISION TREES

The paper presented in this chapter is currently under review for the journal IEEE Transactions on Visualization and Computer Graphics (TVCG).

IXVC: An Interactive Pipeline for Explaining Visual Clusters in Dimensionality Reduction Visualizations with Decision Trees

Adrien Bibal, Antoine Clarinval, Bruno Dumas, and Benoît Frénay, *Member, IEEE*

***Abstract*—High-dimensional data with many features are usually challenging to represent with standard visualization techniques. Usually, one has to resort to dimensionality reduction techniques such as PCA, MDS or t-SNE to represent such data. Such dimensionality reduction techniques make it possible to highlight the high-dimensional structures of data. In many of such visualizations, comparable instances appear to form visual clusters. However, no feedback is directly given by these techniques to the user about the features that make the instances cluster together in the visualization. As such, the interpretation of which features define a given visual cluster is a complicated task. In this paper, we propose a novel interactive approach (called Interactive eXplanation of Visual Clusters – IXVC) to explain dimensionality reduction visualizations by mapping their clusters to explanations provided by decision trees. The decision trees use features in high-dimensional data to explain two-dimensional clusters, filling the gap between the dimensionality reduction visualization and the original data.**

***Index Terms*—Nonlinear Dimensionality Reduction, Explainability, Interactivity, Decision Trees**

I. Introduction

Data in today's world are often high dimensional, as the collected elements are characterized by many features (also called *dimensions*). While getting insights of the high-dimensional (HD) data is important, it is not always easy to use traditional visualization tools and techniques.

In machine learning, dimensionality reduction (DR) techniques are designed to reduce the number of features of the original HD data. Reducing the number of features provides low-dimensional (LD) data that can be useful for many machine learning algorithms. Furthermore, if the number of dimensions in LD is low enough (e.g. 2 dimensions), the LD data can be visually presented to users.

DR techniques are used in many different fields for visualizing data. For instance, multidimensional scaling (MDS) [1] is often used in psychology to generate visualizations that are analyzed in order to explore data or validate hypotheses [2], [3] (for an example, see [4]). Principal component analysis (PCA) [5] is another famous technique that can produce visualizations if the first components are kept. One of the main differences between MDS and PCA is their interpretability. *Interpretability* is defined by the intrinsic capacity of a model to be understandable [6], [7]. In the context of DR visualizations, the link between the dimensions of a PCA visualization (also called *principal components*) and the corresponding HD data is commonly considered interpretable. This is due to the fact that the principal components are linear combinations of the HD features. By looking at the weights in the linear combinations, features from HD data that are used for defining LD dimensions can be identified. On the contrary, the mapping between the HD dimensions and the LD dimensions produced by MDS or other nonlinear DR (NLDR) techniques is not always clear. This lack of interpretability is an issue. In order to overcome this interpretability problem, methods can be developed to *explain* such black-box models or mappings [8].

In some cases of NDLR visualizations (e.g. with t-SNE), the dimensions have no meaning and therefore cannot be used as a basis for explanation. Instead, the analysis must rely on visual cluster present in the visualization. However, there are issues pertaining to visual cluster analysis such as arbitrary cluster shapes and the analyst's intuitiveness injected in the explaining process. Currently, none of the approaches proposed in the literature address most issues related to explaining NLDR through visual clusters. This paper aims to fill this gap and studies the following research question:

> If visual clusters clearly appear in a given DR visualization, can we explain these visual clusters based on the original dimensions?

In order to handle this research question, corresponding to explaining black-box DR mappings through visual clusters, an interactive pipeline is proposed in this paper. This pipeline, called *Interactive eXplanation of Visual Clusters* (IXVC), explains the link between clusters visually present in LD (visual clusters) and the original HD features by using a decision tree. Decision trees are considered for providing explanations because they can non-linearly predict the clusters while being interpretable. Furthermore, decision trees stay interpretable even in the case where the original data are high-dimensional (as opposed to, e.g., linear models), as the decisions in the trees consider features one by one. This makes the proposed solution scalable in terms of the number of HD dimensions. The pipeline is interactive and therefore involves the analyst in the selection of clusters to be explained. IXVC is implemented in a web application that has been used for the pipeline evaluation.

In order to present IXVC, Section II first presents how the literature tackles the explainability of NLDR visualizations

All authors are with the Namur Digital Institute, University of Namur, Namur, Belgium.
Adrien Bibal and Antoine Clarinval are co-first authors.
Contact: {*adrien.bibal;antoine.clarinval*}*@unamur.be*

through dimensions and clusters. Section III motivates the need for the explanation of visual clusters in NLDR by using t-SNE as an example. Then, Section IV introduces IXVC. Section V presents the tool that implements the pipeline and an example of usage. A user-based experiment has been conducted for evaluating the pipeline and the tool, and is presented in Section VI. A discussion on the evaluation results is presented in Section VII. Directions for future work are proposed in Section VIII and Section IX concludes the paper.

II. Explaining Dimensionality Reduction Visualizations

Dimensionality reduction (DR) is the process of reducing the number of features that are available in high dimension. DR is often considered when the dimensionality of data is too high for processing with certain algorithms, or to explore high-dimensional (HD) data. For instance, the curse of dimensionality makes some machine learning (ML) algorithms unusable if the number of features (or *dimensions*) in the original dataset is too high. Another example is data exploration through visualization techniques, which is made easier when the number of features is reduced. For instance, scatter plots can be used when the number of dimensions is reduced to two (2D). Such a 2D visualization obtained through DR is called a DR visualization. Figure 1 presents an example of a visualization resulting from a DR process.

Through the DR process, information is inevitably lost. In multidimensional scaling (MDS), for instance, the measure of this loss, called the *stress*, is defined as the difference between pairwise distances between instances in HD and in LD. More formally, let d_{ij}^{HD} be the distance between the instances i and j in HD, and d_{ij}^{LD} the distance between the instances i and j in LD, the Kruskal's stress [1] is defined as

$$Stress = \sqrt{\frac{\sum_{ij} \left(d_{ij}^{\mathrm{HD}} - d_{ij}^{\mathrm{LD}}\right)^2}{\sum_{ij} {d_{ij}^{\mathrm{HD}}}^2}}.$$

The DR loss of information, called *DR errors* in this paper for the sake of generality, is essential to consider while interpreting or explaining DR visualizations. Indeed, because of DR errors, some instances are not positioned correctly in LD, with respect to their position in HD. These DR errors make the task of analyzing the visualization more difficult. Some visual techniques have already been developed to hint the presence of DR errors in visualizations (see e.g. [9]–[12]).

In the context of DR visualizations, interpreting a particular DR means understanding the mapping between the instances in HD and the corresponding instances in 2D. There are two main ways to interpret or explain such mappings [1]. First, the mapping can be interpreted by focusing on the interpretation of the two new dimensions. The literature concerned by the interpretation and the explanation of the reduced dimensions is developed in Section II-A. Second, the visual clusters in the two-dimensional visualization can also be used to find an interpretation or explanation, as developed in Section II-B.

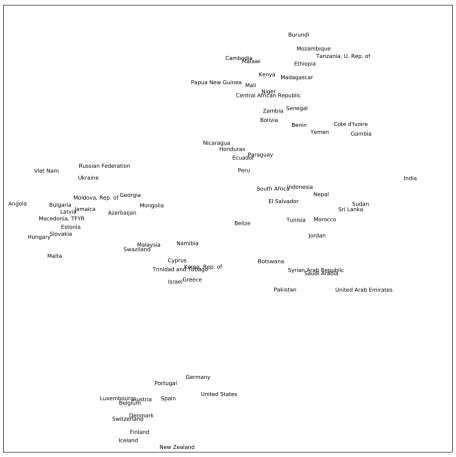


Fig. 1: DR visualization (generated by the DR algorithm t-SNE [13]) of the 2006 Human Development Report [14]. This visualization is composed of 76 sampled countries from the dataset. Based on the HD features, which are socio-economic features, the DR derives two dimensions. Even if visual clusters can intuitively be identified, it is not clear how the HD features have been used to generate them.

A. Explaining DR Visualizations using Dimensions

Among the two ways to interpret DR visualizations, interpretation or explanation of the LD dimensions is the most widespread in the literature. First of all, some DR techniques, such as the *principal component analysis* (PCA) [5], are interpretable because the mapping between the HD space and the 2D space dimensions is linear. This means that the new dimensions are defined as linear combinations of the HD features. One classical way to link reduced dimensions from a linear DR and the original HD features is by using axis legends [15]. These legends are often represented as bar charts representing the contribution of HD features for each new dimension. In the case of linear DR, these contributions are contained in the model parameters (i.e. the weights). Another way to visualize the contribution of the HD features to the embedding constructed by a linear DR technique is by using biplots [16], [17]. Biplots are plots that make it possible to visualize the instances, such as in traditional scatter plots, as well as HD features. Indeed, the HD features are visualized as vectors in the plot. The direction and the length of a vector $\mathbf{v}_i$, corresponding to an HD feature $\mathbf{f}_i$, is defined according to the contribution of $\mathbf{f}_i$ to the two reduced dimensions.

In the case of nonlinear DR (NLDR), the contribution of each HD feature to the reduced dimensions is not given, which makes NLDR mappings hard to interpret [18]. For transferring biplots to the NLDR case, Coimbra et al. make uniform perturbations in the values of each HD feature, while setting the unperturbed HD features to their mean value [19]. By doing so, they obtain curved axes representing the tendencies of HD features in the 2D plot. Following the same idea, Cavallo and Demiralp propose to draw *prolines* for each point

of interest $\mathbf{x}^{LD}$ in a scatter plot and each feature x_j^{HD} [20]. A proline is drawn in LD by creating new samples in which the value of x_j^{HD} varies but all other HD feature values are fixed, and then by computing the projection of all generated samples to LD. The proline corresponds to the line that connects all created samples projected in LD.

Coimbra et al. also present axis legends for NLDR, based on their curved biplot axes [19]. They define the height of the bar h_d^i corresponding to the contribution of the HD feature $\mathbf{f}_i$ in the bar chart of the reduced dimension $\mathbf{d}$ as

$$h_d^i = \left| \left((\mathbf{q}_S^i - \mathbf{q}_1^i) \cdot \mathbf{d} \right) \left(1 - \frac{\left| ||c_i|| - ||\mathbf{q}_S^i - \mathbf{q}_1^i|| \right|}{||c_i||} \right) \right|,$$

where $||c_i||$ is the length of the curved biplot axis c_i, $\mathbf{q}_1^i$ is a point at one end of the curve c_i and $\mathbf{q}_S^i$ is a point at the other end of the curve c_i [19]. The first term roughly tells how parallel the curve c_i is to the scatter plot axis $\mathbf{d}$, whereas the second term is used to approximate the linearity of c_i.

Another way to deal with the interpretability issue of NLDR is by transforming the mapping to be linear. For instance, Gisbrecht et al. apply a linear kernel to the NLDR algorithm t-SNE [13], in order to make the mapping linear [21].

External resources can also be used to explain the dimensions. For instance, social scientists often use *property fitting* (PROFIT) [22] to find trends in a visualization (e.g. [4]). These trends are created and explained using linear combinations of features that have not been used to make the visualization (i.e. external features). *Best interpretable rotation* (BIR) is also a solution that explains DR visualization dimensions by using linear combinations of external features [23], [24].

B. Explaining DR Visualizations using Clusters

As shown in Figure 1, visual clusters can be formed in a DR visualization. Explaining the DR visualization mapping using visual clusters is another way to get insights about the HD-to-2D mapping. This task is related to the combination of the *verify clusters* task of Brehmer et al., and the *name clusters* task [25]. Indeed, this task can be called *map synthesized clusters to original dimensions*, echoing Brehmer et al.'s *map synthesized to original dimensions* task. These visual clusters can either be identified manually by a user or automatically by a clustering algorithm.

When the clusters are identified, an explanation of these clusters can be provided. Most of the time, the explanation is provided by experts (e.g. [26]). This makes the *name clusters* task subjective [2]. One drawback of this approach is the extra knowledge experts may inject during the explanation that does not come from the data used to generate the visualization. Furthermore, even if they do not inject extra knowledge (e.g. by restricting their explanation to the original HD features), it is still difficult to explain how the HD features are combined to form clusters in 2D.

If one wants to consider an explanation that is more informative of how the clusters have been mapped, it is useful to link the visual clusters to combinations of HD features. For instance, if k-means [5] is used on HD instances and the labels of the clusters found are shown in 2D, it is possible to get a more informative explanation of the visual clusters through the centroids of k-means. However, the 2D visual clusters do not necessarily correspond to the HD clusters found by k-means. Furthermore, the HD centroids provided to the user by k-means are defined in terms of each and every HD feature, which makes them difficult to interpret in practice.

Approaches exist in the literature to detect and rank features according to their significance regarding a classification, a regression or a clustering procedure [27]. For instance, *recursive feature elimination* (RFE) can be used to remove features one by one iteratively by using the model coefficients in the case of linear models or the feature importance score in, e.g., random forests [27]. In the case of clustering, several different metrics of feature interest can be used to prune the initial set of features [28] (see e.g. [29]–[32]).

For explaining visual clusters by using HD features, da Silva et al. propose to rank HD features according to an euclidean ranking and a variance ranking [33]. For each instance in the dataset, a set of neighbors is chosen by the user in LD and the euclidean distance between each instance and their LD neighbors is computed for each HD feature. The visualization is then colored by following the top ranked HD features in the different neighborhoods.

After automatically detecting clusters using a grid in LD, Kandogan proposes to rank HD features by labeling each cluster according to a score associated to each HD feature [34]. This score is computed for each automatically detected LD cluster and for each HD feature as a linear combination of measures on properties of the LD cluster (e.g. its density). The weights of the linear combinations, i.e. the importance of the properties, are set by the user.

Joia et al. use a *singular-value decomposition* (SVD) on the transposed matrices containing the HD features of instances in each automatically found cluster to compute the importance of HD features for those clusters [35].

Rauber et al. propose to let the user select a group of instances and a ranking of HD features is provided following a *discrimination* criterion (i.e. how individual HD features explain the separation of selected instances from the rest) or a *coherence* criterion (i.e. how the *compactness* of the selected instances are explained by individual HD features) [36]. In contrast, the interactive ML pipeline proposed in this paper provides an explanation of visual clusters based on only a few HD features that are combined by using a decision tree. Indeed, in our pipeline, (i) the user must select at least two groups (or clusters) and the explanation is provided by confronting the groups, (ii) the selection is performed using a lasso instead of a rectangle for considering any cluster shape and (iii) the features are combined using a decision tree for explaining the selected groups instead of having a ranking.

Parisot et al. use an evolutionary algorithm in order to find the dataset preprocessing that leads to a new dataset for which a clustering result is easier to interpret [37]. In order to find such a new dataset, the objective of the evolutionary algorithm is to find a small decision tree that is used to explain the clustering of the preprocessed dataset, while having a clustering on the preprocessed dataset that is as similar as possible to the clustering on the original dataset.

van Ham et al. consider a scatter plot made from two HD features and use a decision tree to explain a selection of instances in the scatter plot by using HD features that are not used in the scatter plot [38]. Their decision tree is a binary classification tree that has the task of explaining the selected instances versus all others, and that does not address the issue of explaining visual clusters in DR visualizations.

t-viSNE is a tool that includes different techniques for getting insights about t-SNE visualizations [39]. In particular, the authors use the first component of a PCA on user-selected points in order to know the main HD features that describe the selected points. However, (i) this selection has no link with the projection, as it explains the HD points instead of how the HD points are projected in LD, and (ii) the explanation is linear because of the PCA, while the projection is nonlinear. In order to take into account these limits, another tool that considers polylines is proposed in t-viSNE. The idea is to draw lines in the visualization and then rank the HD features according to how they explain, for each LD dimension, the order of the LD points on the polylines. Concerning the problem we address in our work, the shortcomings of this tool are that (i) the dimensions are explained instead of the clusters and (ii) the relative importance of the HD features is known, but not how they are combined to explain the dimensions.

III. Explaining DR through Clusters: the Case of t-SNE

Explaining DR through clusters is needed, but the existing solutions that could be used face several challenges. In order to make these challenges explicit, we take t-SNE [13], a state-of-the-art nonlinear DR (NLDR) algorithm, as an example. The objective of t-SNE is to preserve HD proximity by making neighbors two instances in 2D if they are neighbors in HD. More precisely, the closer the instances are in HD, the more t-SNE tries to put them close in 2D. An example of a t-SNE visualization is shown in Figure 1.

First, given its focus on neighborhoods in HD, t-SNE naturally tends to accentuate clusters in 2D, which makes it a good candidate for the explanation through clusters. Furthermore, the dimensions of t-SNE visualizations have no meaning [40] and cannot be used as a basis for explanations. Because of that, only the cluster approach for explaining can be used, and no technique for explaining through dimensions can make t-SNE more interpretable [40], unless t-SNE is modified (e.g. [21]).

Second, visual clusters resulting from t-SNE can have complex shapes, which make clustering algorithms with predefined cluster shapes, such as k-means, not suitable. Indeed, the predefined shapes of such clustering algorithms restrain the possible cluster explanations to the clusters that can be possibly formed by the clustering algorithm.

Third, cluster analysis performed manually by experts, or automatically by clustering algorithms, can be misleading because of t-SNE propensity to show clusters. Indeed, despite its strength in the detection of real HD clusters, t-SNE is also known for sometimes presenting clusters in 2D that do not exist in HD [40].

All the issues presented in this section are not restricted to t-SNE. Other state-of-the-art NLDR algorithms, such as UMAP [41] and LargeVis [42], share these issues. To the best of our knowledge, no techniques addressing all these issues exist in the literature. IXVC, the solution proposed in this paper to the task *map synthesized clusters to original dimensions* tackles these issues.

IV. Interactive Explanation of Clusters using Decision Trees

Section IV-A proposes answers to the issues discussed in Section II-B and III. Then, Section IV-B builds on the proposed answers and presents an ML pipeline that helps data analysts to explain DR visualizations through clusters.

A. Answers to Cluster Interpretability Issues

A first issue, mentioned in Section II-B, is the intuitive explanation of clusters. This issue arises when data analysts use their intuition to explain clusters that are made of errors from the DR algorithm. Having access to objective reasons behind visual clusters, based on the HD features, would help data analysts to overcome this issue of intuitive assessment.

A second issue, presented in Section III, concerns the possibility for 2D clusters to have arbitrary complex shapes. The hypotheses made on the form of clusters by clustering algorithms may not be suitable when clusters take complex shapes. However, in the case of a 2D visualization analysis, data analysts can draw the limits of visual clusters themselves (e.g., with a hand-made selection).

A third issue is about the role of DR errors in cluster explanation. In this case, having a feedback on DR errors may help the data analyst to explain the mapping. Indeed, if individual errors, for each instance in 2D, are provided to data analysts, it would be possible to decide whether to discard or not some instances during the visual analysis. This issue is important, since the presence of instances erroneously placed in visual clusters because of DR errors can mislead the analyst.

Ideally, in addition to an easier explanation of visual clusters, providing feedback on visual clusters would also help data analysts to decide to take several actions. For instance, in order to improve the interpretation, they may want to choose a more appropriate DR algorithm, to change the *hyperparameters* (or *meta-parameters*) of the DR algorithm or to remove instances that make the DR process difficult.

B. IXVC: the Interactive Machine Learning Pipeline

In this section, the interactive pipeline developed for explaining clusters in DR visualizations is presented. The pipeline is called IXVC for *Interactive eXplanation of Visual Clusters*.

The pipeline is used in the context of an exploration of a DR visualization. As such, the first step is to consider a particular DR visualization. This first step corresponds to the scatter plot (1) in Figure 2. The error made by the DR algorithm for each instance in the visualization should be provided, in order for the data analyst to unselect elements that have a DR error that is too high.

The second step is to manually select the visual clusters for which the analyst wants an objective explanation (see (2) in

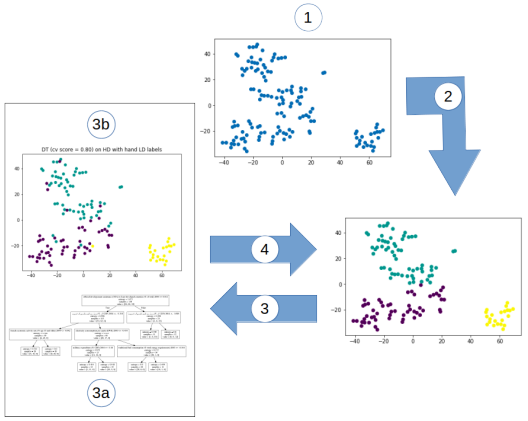


Fig. 2: From a given DR visualization (1), the data analyst manually selects visual clusters (2) (the colors correspond to the selected clusters), and a visualization of the errors made by a decision tree (3a) explaining the manual clusters using the HD features is provided (3b). Given the feedback provided, a new manual clustering can be done by the data analyst (4).

Figure 2). All instances in the visualization do not have to be selected and the number of 2D clusters can be arbitrary.

Next, a decision tree (DT) is built based on the 2D clusters provided in the second step (see DT (3a) in Figure 2). The 2D cluster memberships serve as labels for the decision tree and the original features, the ones of the HD data, are the criteria on which the decisions in the decision tree are made. One can note that decision trees can be used to explain HD clusters (see, e.g., [37], [43], [44]). However, our task is different because the goal is not to cluster data in HD, but to understand a given DR visualization through 2D visual clusters. Therefore, we propose to understand the visualization by interactively querying the meaning of visual clusters of interest.

A k-fold cross validation is then used to select the best hyperparameters for the decision tree (see (3a) in Figure 2). By doing so, the decision tree provides its best possible solution for interpreting the DR visualization by explaining the manually selected clusters with the original features. While the first information provided by the decision tree to the analyst is the explanation in terms of the original features, the second information comes from the prediction errors of the decision tree (see the scatter plot (3b) in Figure 2). Indeed, visualizing the errors made by the tree when predicting the selected visual clusters allows the analyst to see where, in the visualization, the analyst clustering cannot be explained. This information may hint that the DT cannot help in the explanation of the selected visual clusters, but can also hint the need to cluster the instances in a different way.

Finally, the analyst may either stop the analysis or choose to cluster the instances differently. In the latter case, the analyst proceeds with the second step again by selecting other visual

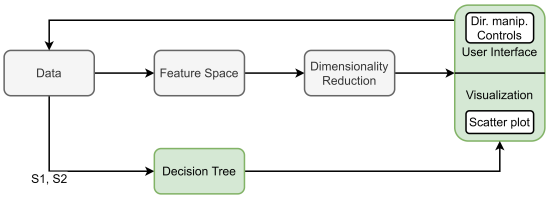


Fig. 3: In our approach, the data, feature space, and the DR in the process model from [45] are considered as given, and the interaction augments the data in order to train a DT that will, in turn, augment the visualization.

clusters (see (4) in Figure 2), which are then explained with a new decision tree. In the former case, the analyst stops the analysis and accepts the explanation. The analysis can also stop because the multiple iterative explanations of clusters have provided enough information to the analyst.

Sacha et al. [45] developed a process model describing interactive DR and have defined seven scenarios of interaction. Although our work proposes to combine user interaction and DR, this interaction takes place after the DR was performed rather than during its computation. In our approach, the data, feature space, and the DR in the process model from [45] are considered as given, and the interaction augments the data in order to train a DT that will, in turn, augment the visualization (Figure 3). These augmentations will help the analyst understand how clusters are mapped onto the embedding. Scenarios S1 (i.e. data selection) and S2 (i.e. annotation and labelling) from [45] are supported in the explaining process, as users can filter instances and define visual clusters (hence, assigning a label to instances) to build the DT.

Note that as the goal is to help understanding visual clusters in a visualization, and not an automatically computed clustering, users can draw the frontiers of the clusters they see and assess the explanation received via the DT. This particular setup explains why our pipeline is interactive (users must be in the loop), as well as iterative (users can try other explanations in order to expand their understanding of the visualization).

V. Interactive Explanation Interface

This section introduces the web interface implemented to evaluate IXVC. Figure 4 shows the IXVC interface. The top part of the interface shows the DR scatter plot from which the user selects the clusters (top-left), a list of the selected clusters (top-middle) and a scatter plot (top-right) showing the predictions resulting from the DT (bottom of the interface). The idea of generating a DT from a user selection of clusters in a scatter plot was previously considered by Ware et al. [46]. In IXVC, however, instead of representing two HD features, the scatter plot is the result of a DR process. Moreover, the decision tree of Ware et al. is built manually by defining splits through the scatter plot, whereas it is automatically generated from the selection of clusters in IXVC.

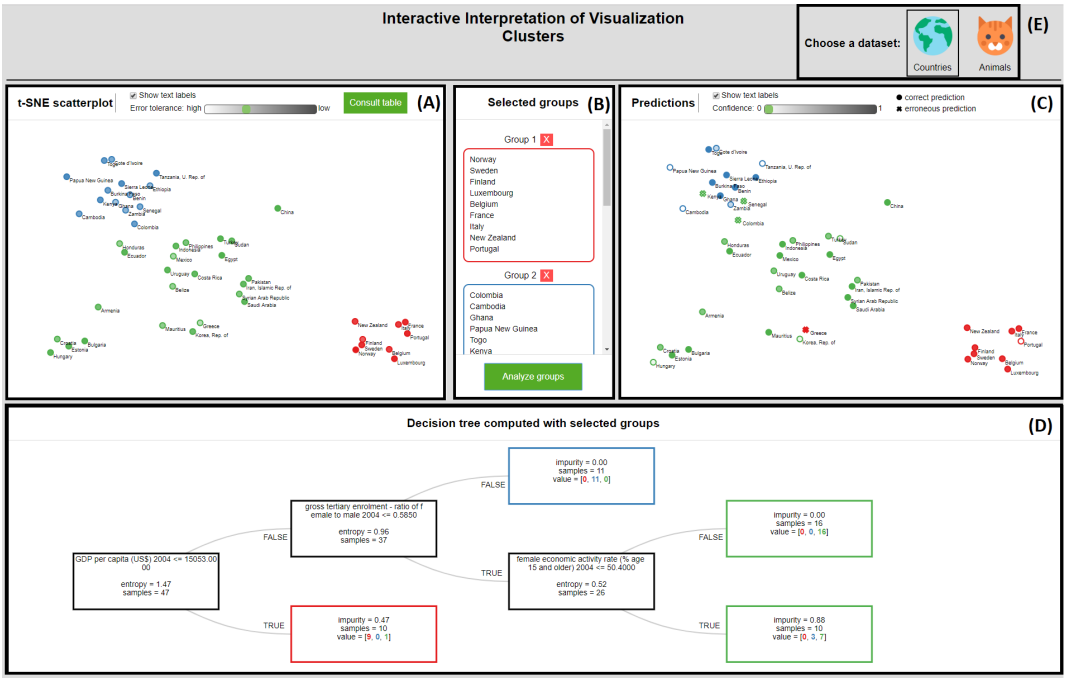


Fig. 4: IXVC interface. Top left scatter plot (A) corresponds to the DR visualization. Instances are colored with respect to the user selection of clusters shown in the top middle part (B). On the bottom, the decision tree (D) explaining the user selection is provided. The colors corresponding to the tree predictions are presented in the top right visualization (C). For the evaluation (see Section VI), users can switch between the *country* and the *zoo* datasets by using the earth and cat icons (E).

A. *Interface Design*

The IXVC interface is implemented as a web application running on a Python web server. The visuals were developed in Javascript using the D3.js library [47]. The Python web server handles the execution of the ML algorithms using scikit-learn [48]. For the evaluation of IXVC, visualizations generated by t-SNE are used.

When launching the IXVC interface, the user is presented with a scatter plot (located at the top-left part of the interface) generated by running t-SNE (without PCA preprocessing) on the dataset at hand. Each instance is represented as a black dot with an associated text label showing its name, thus allowing the identification of individual instances. The individual errors resulting from t-SNE (measured using the individual Kullback-Leibler divergence loss) are depicted by the opacity of each dot, with the whitest dots representing the highest error. Instances can be filtered out according to a DR error tolerance threshold defined by the user. The DR error threshold is labelled as *error tolerance* instead of *loss tolerance* in the interface, as a preliminary evaluation (see Section VI-A) of the interface suggested that it is more meaningful for users and more generic formulated as such.

A major challenge when displaying a scatter plot is the visual clutter that can occur when there are numerous data points to show, which can impede the analyst's work and cause delay in the rendering of the visualization in the interface. Previous work in the literature suggests to implement techniques, including interaction features, in order to tackle visual clutter (e.g. [49], [50]) and to ensure that the visualization at hand possesses desirable properties such as scalability and individual data point localization. Ellis and Dix [50] have identified eight properties and eleven clutter reduction techniques that can be used to achieve these desirable properties. In the context of the IXVC interface, three clutter reduction techniques, namely the sampling (discussed in Section VI), the filtering and the opacity, were implemented. This combination allows us to obtain all the desirable properties listed in [50] that are necessary to the interface. In particular, the scalability regarding the number of data points (achieved through sampling and filtering), and the ability to discriminate individual points on the visual representation (achieved through opacity) are of upmost importance.

The scatter plot provides a lasso-like interaction allowing users to select visual clusters. The selected instances are subsequently colored alike to mark their belonging to the same cluster following a categorical color scale generated with ColorBrewer [51]. The clusters thus defined by the user are displayed in a pane to the right of the scatter plot. The interface uses the word *groups* instead of *clusters* in order for the evaluation participants to avoid the confusion with clusters that would be obtained from an automatic clustering technique.

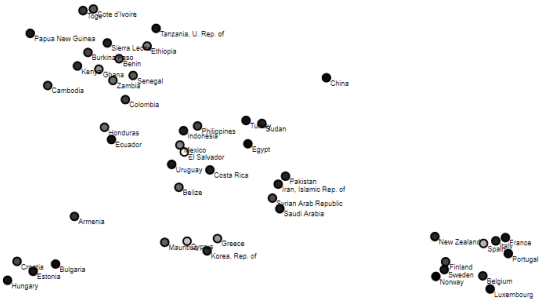


Fig. 5: Initial scatter plot (step 1 of the pipeline) showing 50 instances. The luminance of the dots indicates the individual DR errors (the whiter the dots are, the higher the error is).

When the user is finished selecting visual clusters, a decision tree generated from the selection is displayed under the scatter plot. The decision tree attempts to predict the cluster of each selected instance using the HD features. The representation of the decision tree shows the features selected to build the tree as well as the entropy (named *impurity* in the leaves of the decision tree). The entropy characterizes the distribution of instances by cluster in a specific node. It is equal to 0 if only elements of one cluster are present in the node, and to $\log_2(\#\text{ of clusters})$ if elements are spread equally between all the visual clusters to predict. For each leaf in the tree, the number of instances predicted for each cluster is presented. The prediction for each instance is shown in a second scatter plot in the upper right part of the interface. Whereas the decision tree gives the number of incorrect predictions in each leaf, this scatter plot makes it possible to identify the incorrectly predicted instances in question. The instances are colored according to their predicted cluster and are shaped as a dot if the prediction is consistent with the user selection, and as a cross otherwise. The level of confidence of the predictions made by the decision tree is denoted by the opacity of the points on the scatter plot. Again, instances can be filtered out according to a threshold defined by the user on the minimal confidence provided by the DT.

Based on the decision tree and the scatter plot showing the predictions, the user can reflect on the explanations and draw a new cluster selection. In turn, he can cycle through the visual cluster explanation pipeline again by generating a new decision tree with new selected visual clusters. This iterative process is repeated until the user feels that he has a sufficient understanding of the DR visualization.

B. Example Case Study: Explaining Clusters of Countries

In this section, a case study demonstrates a step-by-step application of the IXVC pipeline. The data analyst works with 50 instances extracted from the 138 countries of the 2006 Human Development Report [14], hereafter called the *country* dataset. The countries are characterized by 45 socio-economical indicators such as *GDP* and *population growth*.

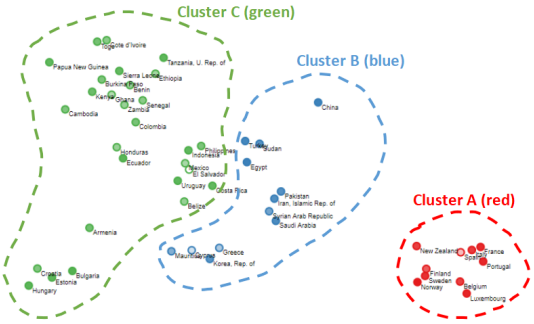
In the first step, the data analyst discovers the scatter plot shown in Figure 5. In the second step, he makes a selection of clusters. In the example of Figure 6a, three clusters have been selected. In the third step, a decision tree and a second scatter plot are generated based on the cluster selection. The decision tree (Figure 6c) shows that the red cluster can be flawlessly explained by the *money spent on assisting least developed countries*, with the countries in red spending more. The decision tree separates the 40 remaining instances according to their *GDP*. It also explains the blue cluster as the set of countries above 64.9 billion USD of GDP. Among the 15 concerned instances, 10 are correctly predicted as belonging to the blue cluster. However, 5 instances selected in the green cluster by the user are erroneously predicted as blue. All the remaining instances are predicted to belong to the green cluster. The decision tree uses the *primary exports* feature to separate the 25 remaining instances. Among the countries under 57% of primary export, 2 instances being in the blue cluster are erroneously predicted as green. The scatter plot in Figure 6b shows the predictions and highlights the errors.

Unsatisfied with the 7 erroneous predictions of the decision tree, the data analyst undertakes a second iteration of the pipeline, setting aside the red cluster, for which there was no prediction error. The analyst divides the remaining 40 instances into two significantly changed clusters (Figure 7a). In this new selection, the former blue cluster (cluster B in Figure 6a) has been enlarged to include the erroneously predicted countries in Figure 6b. The resulting decision tree (Figure 7b) explains the new clusters using the *GDP per capita* feature. It results in only 2 prediction errors instead of the 7 errors that occurred in the first iteration that used the *GDP* to separate the 40 instances. This new explanation, which would have been tedious, or even impossible, to reach without the IXVC pipeline, leaves the data analyst satisfied with the cluster explanation. The selected clusters can be explained by the *money spent on assisting least developed countries* and the *GDP per capita*. 48 instances out of 50 are correctly predicted by the corresponding decision tree.

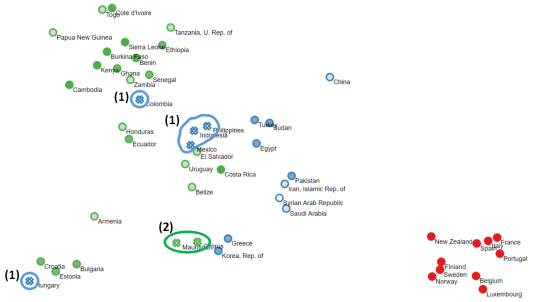
VI. Evaluation

This section presents the evaluation of the IXVC pipeline and interface. Note that the datasets that are relevant with the pipeline need to contain understandable features in order to be used with decision trees. For evaluation purposes, two datasets are available for analysis with the interface. First, the *country* dataset presented in Section V-B. Second, the *zoo* dataset [52] characterizes 101 animals with 16 features such as the number of legs and whether they have feathers or not. A table displaying the whole datasets is available to users via a button on the interface. The t-SNE perplexities for generating the DR visualizations of the *country* and *zoo* datasets are 6 and 18 respectively.

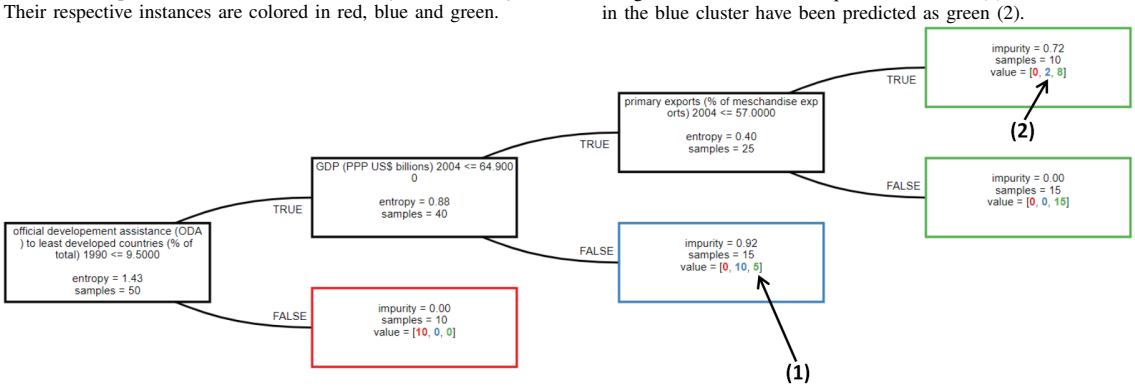
For each dataset, 50 instances were randomly sampled. The goal of the evaluation was to evaluate the IXVC pipeline rather than the interface developed to implement it. The participants would have been confronted to a barrier not related to the pipeline, which would have thus tweaked the evaluation results. Showing more than 50 instances at once would hinder the readability of the scatter plot and sampling is one technique commonly suggested to tackle such visual clutter issues [50].



(a) Cluster selection (step 2 of the pipeline). Three clusters of non-trivial shape (A, B, C) have been selected by the data analyst. Their respective instances are colored in red, blue and green.

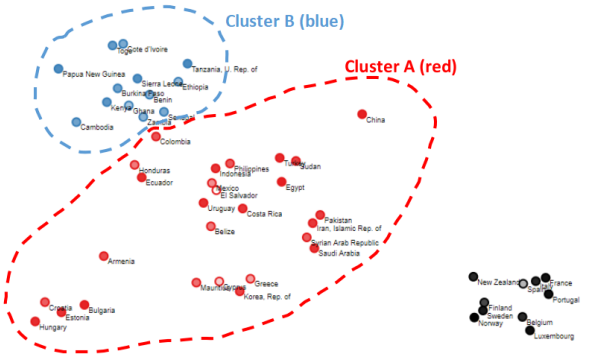


(b) Individual predictions (step 3 of the pipeline). Among the 50 instances, 43 have been correctly predicted. 5 countries selected in the green cluster have been predicted as blue (1), and 2 selected in the blue cluster have been predicted as green (2).

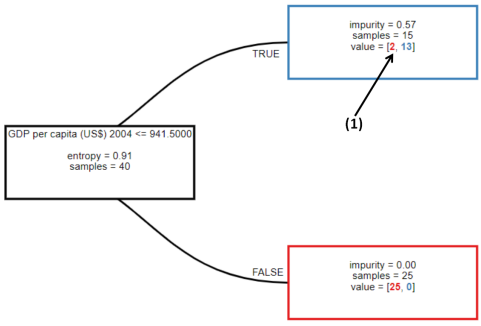


(c) Decision tree (step 3 of the pipeline). For each leaf of the tree, the *value* field shows how many instances have been predicted as red, blue and green. The red cluster is flawlessly explained by the *money spent on assisting least developed countries* feature. The remaining 40 instances are then successively separated according to the *GDP* and *primary exports* features. The prediction of the tree is erroneous for 5 instances from the green cluster (1) and 2 from the blue cluster (2).

Fig. 6: Example case study: first iteration of the IXVC pipeline to explain the visual clusters of countries.



(a) Cluster selection (step 2 of the pipeline, second iteration). As a follow-up to Figure 6b, the data analyst selected two clusters (A, B). The cluster A corresponds to the cluster B in Figure 6a, enlarged to include the erroneously predicted elements. The 10 elements at the bottom right of the plot have not been considered in this iteration as they formed a flawlessly explained cluster in the first iteration.



(b) Decision tree (step 3 of the pipeline, second iteration). The 40 instances are almost perfectly separated according to the *GDP per capita*. 2 instances selected in the cluster A have been predicted as belonging to cluster B by the decision tree (1).

Fig. 7: Example case study: second iteration of the IXVC pipeline to explain the visual clusters of countries

A. Preliminary Feedback

During the development of the IXVC interface, early feedback has been sought from two researchers that are non-experts, but knowledgeable, in ML and information visualization. The goal was to detect usability flaws in the interface and to determine if any information was missing for the cluster explanation. Overall, the two researchers made a few usability suggestions such as adding captions and changing labels in order to make the interface clearer.

B. Evaluation Methodology

The objective of the evaluation was to measure whether IXVC helps to conduct an analysis of a DR visualization, and if so, with more objectivity. As a tool supporting analysis through a t-SNE visualization and decision trees, IXVC is destined to users knowledgeable of these techniques. 16 students (13 males and 3 females) following a graduate-level data science program in which t-SNE and decision trees are taught were recruited. The age of the participants ranged from 20 to 53 years (two participants are older students resuming their studies), with a median of 22. The students had previously carried out a class project on intuitive visual cluster explanation (without any tool) with the *country* dataset.

The evaluation consisted of 45-minute sessions following quasi-empirical evaluation practices [53]. The sessions began with a brief introduction to the goal of the IXVC pipeline. No explanation on how the interface works was provided at this point. Two researchers were present throughout the session in order to answer participants' questions and to take note of their remarks as well as observations. Then, the two datasets participants were asked to work on were introduced. The *country* dataset was presented as a set of countries characterized by various socio-economical indicators and the *zoo* dataset was presented as a set of animals described by biological traits. It was essential to provide only the minimum, but necessary, information in order not to guide the explanations towards specific HD features. Observation and questionnaire filling were used to collect data.

1) Observations: Observations were conducted throughout the sessions by two researchers to detect usability issues and to see whether the analysis behavior of the participants was consistent with the pipeline. Observations were mainly passive with questions from participants answered when asked.

2) Questionnaire: When participants were finished with the analysis of the two datasets, they were invited to fill a short three-part questionnaire. First, the initial perceived expertise of the participants was measured. The second part of the questionnaire was about the data analysis process with IXVC. Lastly, the general usability of the IXVC interface was measured in order to control the impact of the interface in the evaluation of the pipeline. For this latter part, the System Usability Scale (SUS) questionnaire [54] was used. The SUS is a questionnaire scoring the usability of a system with 10 questions measured on a 5-point Likert scale. It has the advantage of being quick to complete and highly reliable. Following literature recommendations, two adaptations were made to the original SUS. In the *eighth item of the SUS*, "cumbersome" was replaced by "awkward", as the participants are non-native English speakers. The word "cumbersome" in the SUS has been reported to cause confusion [55], especially among non-native English speakers [56]. The *first item of the SUS* measures the extent to which users would like to use a system frequently. Since the participants of the evaluation are students, who only perform cluster explanation in the context of a class, the first item of the SUS formulated as such was not relevant, and would have tweaked the SUS score. Instead, an adapted version of this question was included in the questionnaire, formulated as "I would like to use the tool in the future if I need to analyze a visualization generated by t-SNE." Although the answers to this question are of interest to the evaluation, it was not included in the computation of the SUS score in order to preserve its reliability. Rather, the score was computed from the 9 other questions, as [57] showed that removing an item inducts a negligible deviation from the results of the 10-item scale and has no impact on reliability.

C. Evaluation Results

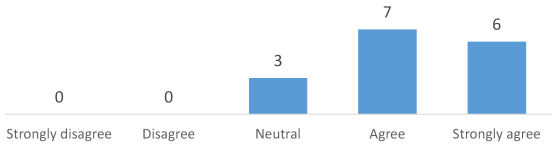
The following section presents the results of the user evaluation. The results obtained from the observations and from the answers to the questionnaire are successively discussed.

1) Observations: Most participants intuitively followed the pipeline to get explanations about visual clusters. However, some usages of the IXVC interface that differed from how participants were expected to apply the pipeline were observed. First, the analysis process was more exploratory than expected. The pipeline describes an iterative process in which a selection of clusters is refined by adjusting the manual clustering. However, a few participants tended to try many different cluster selections instead of iteratively refining one.

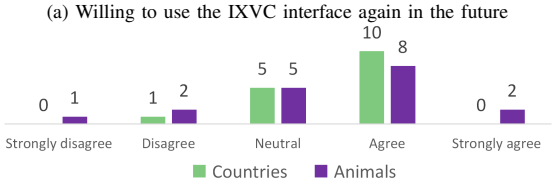
Second, one participant (P11) was attempting to generate a decision tree involving an intuitive feature not necessarily present among the HD features. Indeed, in the scatter plot of the zoo dataset, the participant selected clusters separating aquatic animals from another. At this point, P11 did not consult the dataset table to see if there was indeed an feature distinguishing aquatic animals. In doing so, P11 tried to build a decision tree where this feature appears. P11 made repeated attempts until such a tree was displayed on the screen.

2) Questionnaire: Overall, participants rated, on a scale from 1 to 5, their knowledge of t-SNE as intermediate (median = 3) and of decision trees as good (median = 4). They felt familiar with the dataset on countries (median = 4), but not with the one on animals (median = 2). In both cases, access to the dataset tables during the analysis was useful to the participants (median = 4 and 4.5 for countries and animals).

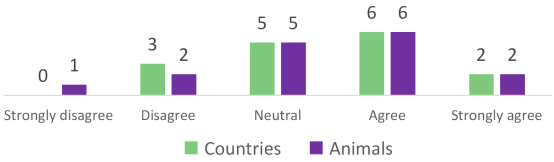
75% of the participants stated that they prefer the IXVC interface to the tool-free approach they used in the prior class project. Furthermore, 81% reported they would like to use the IXVC interface again if they have to work on a t-SNE generated visualization in the future (see Figure 8a). Participants felt that the interface helped them to gain a better understanding of the datasets (median = 4 (agree) for both datasets, see Figure 8b). The IXVC interface scored 3.5 (between neutral and agree) as support to a more objective analysis for both datasets (see Figure 8c).



(a) Willing to use the IXVC interface again in the future



(b) IXVC helped to understand the datasets better



(c) IXVC helped to conduct more objective explanations

Fig. 8: Distribution of the answers from the 16 participants

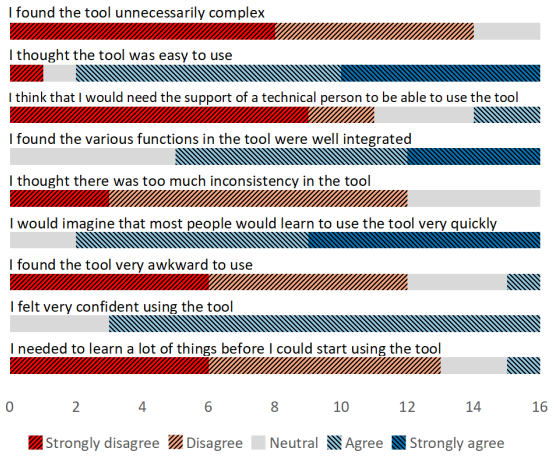


Fig. 9: Likert distribution for the 9 items used to compute the SUS score.

The SUS score of IXVC is 77 (95% confidence interval is [72, 82]), which is above the "good usability" threshold defined by [58] at 71.4. Figure 9 shows the Likert distribution for the 9 items used to compute the SUS score. Furthermore, Lewis et al. [59] showed that the data gathered through the SUS can also be used to reliably derive a learnability score that can be interpreted in the same way as the SUS score. It measures the extent to which an interface enables its users to learn how to use it. The learnability score is computed by considering the fourth and tenth items of the questionnaire. For the IXVC interface, the score stands at 78, very close to the SUS score, which indicates that it has a good learnability.

VII. Discussion

The first observation from the experimental results is that the interface did not alter the evaluation of the pipeline. Indeed, with a SUS score of 77, the interface usability has been considered good, meaning that the implementation has not, for the most part, interfered with the evaluation of the pipeline.

Considering the pipeline itself, it has been considered more useful than the intuitive analysis performed without it by 75% of the participants. This indicates that providing information on the explanation of the selected clusters is important when explaining DR visualizations. Moreover, the participants showed great enthusiasm towards the explanations given by the decision tree and the interactivity of the IXVC interface. In the questionnaire, P1 wrote that "the decision tree is great to visualize how the different groups can be divided and what differentiates them the most." P9 wrote "the real added value is in the decision tree. I think it is very valuable to have an objective reason for clusters."

However, while some participants felt that IXVC brings added-value, the question "I feel that the tool helped me to conduct a more objective analysis" scored between neutral and agree. In an open field of the questionnaire, one participant wrote: "the analysis, in my point of view, isn't more objective since there is a great part of [intuitivity] when choosing the clusters." It seems that some participants interpreted the question as "is the whole process objective" and did not understand that the objectivity resides in the explanations given by the decision tree.

Finally, although the familiarity of the participants was quite different between the two datasets, the same results were observed for both. This leads us to conclude that IXVC is beneficial to the DR explainability process, irrespective of the prior knowledge of the data at hand.

VIII. Future Work

The evaluation results and open fields in the questionnaire allow considering future directions to improve IXVC. A first future work is the development of IXVC for making it an educational tool. As mentioned in Section VI, IXVC is destined to users knowledgeable on t-SNE and decision trees. However, several participants pointed out that the use of IXVC, in fact, needs little knowledge of these techniques. Since the participants are students following a data science program, they are eager to use tools that may ease the understanding of techniques such as t-SNE. Moreover, the playful character of IXVC was emphasized by one participant.

Another future work can be identified following the observation of P11 described in Section VI-C1. P11 used IXVC for checking if a feature he had in mind played a role in the visual cluster separation. Based on this use, the IXVC interface could propose all features as clickable buttons, which would show how the selected feature would make it possible to separate the visual clusters. In the IXVC pipeline, this corresponds to providing all possible decision trees with only one decision based on each HD feature. This pre-exploration step may indicate to the analyst how to manually cluster the 2D instances in step 2 of IXVC.

Another participant (P16) suggested to generate several decision trees to have the opportunity to consider different possible explanations. This means that, in another future work, several trees could be suggested in step 3 of IXVC. For instance, after building the first decision tree, a second one could be built by removing, from the possible features to choose for a decision, the feature that is chosen as first node in the tree.

IX. Conclusion

In this paper, we proposed an interactive machine learning pipeline called IXVC (for *Interactive eXplanation of Visual Clusters*). The pipeline provides explanations on visual clusters manually selected by a data analyst in a dimensionality reduction (DR) visualization. The explanatory feedback on the manually selected clusters is provided by a decision tree whose decisions are based on the high-dimensional (HD) features. Interactively, the data analyst can thus select clusters in the visualization and receive an explanation of the selected clusters through a decision tree. IXVC is a need for data analysts [25] and handles a task that can be called *map synthesized clusters to original dimensions* in Brehmer et al.'s typology [25].

IXVC was implemented as a web application for its evaluation. Results of the evaluation suggest that using the proposed interactive pipeline helps users to explain how visual clusters in a DR visualization are related to the HD features that have been used to create the visualization, even when the mapping between the high and the low dimensions is not provided. It is also suggested by the evaluation results that the usefulness of the pipeline does not depend on the prior knowledge the analyst has on the dataset.

Acknowledgements

The authors want to thank the participants of the experiment for their time, as well as for the interesting discussions on the pipeline and its implementation. We also acknowledge our two colleagues, Laurent Evrard and Gonzague Yernaux, who kindly agreed to take part in the preliminary evaluation. The authors also thank the European Regional Development Fund (ERDF) for their financial support through the Wal-e-Cities LIV project with award number [ETR121200003138].

References

[1] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. Newbury Park, CA, USA: Sage, 1978.

[2] M. C. Hout, M. H. Papesh, and S. D. Goldinger, "Multidimensional scaling," *WIREs Cogn. Sci.*, vol. 4, no. 1, pp. 93–103, Jan 2013.

[3] N. Jaworska and A. Chupetlovska-Anastasova, "A review of multidimensional scaling (MDS) and its utility in various psychological domains," *Tut. Quantitative Methods for Psychol.*, vol. 5, no. 1, pp. 1–10, 2009.

[4] A. Koch, R. Imhoff, R. Dotsch, C. Unkelbach, and H. Alves, "The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion," *J. Personality and Social Psychol.*, vol. 110, no. 5, pp. 675–709, May 2016.

[5] C. Bishop, *Pattern Recognition and Machine Learning*. New-York, NY, USA: Springer, 2006.

[6] A. Bibal and B. Frénay, "Interpretability of machine learning models and representations: an introduction," in *Proc. 24th Eur. Symp. Artif. Neural Netw., Comput. Intell. and Mach. Learn. (ESANN 2016)*, Bruges, Belgium, Apr. 27–29, 2016, pp. 77–82.

[7] Z. C. Lipton, "The mythos of model interpretability," in *Proc. ICML Workshop Human Interpretability of Mach. Learn. (WHI 2016)*, New-York, NY, USA, Jun. 23, 2016.

[8] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Jan. 2019.

[9] M. Aupetit, "Visualizing distortions and recovering topology in continuous projection techniques," *Neurocomputing*, vol. 70, no. 7–9, pp. 1304–1330, Mar. 2007.

[10] S. Lespinats and M. Aupetit, "Checkviz: Sanity check and topological clues for linear and non-linear mappings," *Comp. Graph. Forum*, vol. 30, no. 1, pp. 113–125, Mar. 2011.

[11] T. Schreck, T. Von Landesberger, and S. Bremm, "Techniques for precision-based visual analysis of projected data," *Inf. Visualization*, vol. 9, no. 3, pp. 181–193, Sep. 2010.

[12] R. M. Martins, D. B. Coimbra, R. Minghim, and A. C. Telea, "Visual analysis of dimensionality reduction quality for parameterized projections," *Comput. & Graph.*, vol. 41, pp. 26–42, Jun. 2014.

[13] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[14] United Nations Development Programme, "Human development report," 2006.

[15] B. Broeksema, A. C. Telea, and T. Baudel, "Visual analysis of multi-dimensional categorical data sets," *Comput. Graph. Forum*, vol. 32, no. 8, pp. 158–169, Dec. 2013.

[16] J. C. Gower and D. J. Hand, *Biplots*. London, U.K.: Chapman & Hall, 1995.

[17] M. J. Greenacre, *Biplots in practice*. Bilbao, Spain: Fundaciòn BBVA, 2010.

[18] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Visualization and Comput. Graph.*, vol. 23, no. 3, pp. 1249–1268, Mar. 2017.

[19] D. B. Coimbra, R. M. Martins, T. T. Neves, A. C. Telea, and F. V. Paulovich, "Explaining three-dimensional dimensionality reduction plots," *Inf. Visualization*, vol. 15, no. 2, pp. 154–172, Apr. 2016.

[20] M. Cavallo and Ç. Demiralp, "A visual interaction framework for dimensionality reduction based data exploration," in *Proc. ACM SIGCHI Conf. Human Factors in Comput. Syst. (CHI 2018)*, Montréal, QC, Canada, Apr. 21–26, 2018, pp. 635:1–635:13.

[21] A. Gisbrecht, B. Mokbel, and B. Hammer, "Linear basis-function t-SNE for fast nonlinear dimensionality reduction," in *Proc. Int. Joint Conf. Neural Netw.*, Brisbane, Australia, Jun. 10–15, 2012, pp. 1–8.

[22] J. J. Chang and J. D. Carroll, "How to use PROFIT, a computer program for property fitting by optimizing nonlinear or linear correlation," unpublished.

[23] A. Bibal, R. Marion, and B. Frénay, "Finding the most interpretable MDS rotation for sparse linear models based on external features," in *Proc. 26th Eur. Symp. Artif. Neural Netw., Comput. Intell. and Mach. Learn. (ESANN 2018)*, Bruges, Belgium, Apr. 25–27, 2018, pp. 537–542.

[24] R. Marion, A. Bibal, and B. Frénay, "BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables," *Neurocomputing*, vol. 342, pp. 83–96, 2019.

[25] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences," in *Proc. Workshop Beyond Time and Errors: Novel Eval. Methods for Visualization (BELIV 2014)*, Berlin, Germany, Oct. 20–21, 2014, pp. 1–8.

[26] A. Lebel, M. Cantinotti, R. Pampalon, M. Thériault, L. A. Smith, and A.-M. Hamelin, "Concept mapping of diet and physical activity: uncovering local stakeholders perception in the Quebec City region," *Social Sci. & Medicine*, vol. 72, no. 3, pp. 439–445, 2011.

[27] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, Jan. 2002.

[28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[29] E. P. Xing and R. M. Karp, "Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts," *Bioinf.*, vol. 17, no. suppl_1, pp. S306–S315, Jun. 2001.

[30] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.

[31] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Aug. 2004.

[32] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery and Data Mining (KDD 2010)*, Washington, DC, USA, Jul. 24–28 2010, pp. 333–342.

[33] R. R. O. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. C. Telea, "Attribute-based visual explanation of multidimensional projections," in *Proc. 6th EuroVis Workshop Vis. Analytics (EuroVA 2015)*, Cagliari, Italy, May 25–26, 2015, pp. 134–139.

[34] E. Kandogan, "Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations," in *Proc. IEEE Conf. Vis. Analytics Sci. and Technol. (VAST 2012)*, Seattle, WA, USA, Oct. 14–19, 2012, pp. 73–82.

[35] P. Joia, F. Petronetto, and L. G. Nonato, "Uncovering representative groups in multidimensional projections," in *Proc. Eurographics/IEEE Conf. Visualization (EuroVis 2015)*, Cagliari, Italy, May 25–26, 2015, pp. 281–290.

[36] P. E. Rauber, R. R. O. da Silva, S. Feringa, M. E. Celebi, A. X. Falcão, and A. C. Telea, "Interactive image feature selection aided by dimensionality reduction," in *Proc. 6th EuroVis Workshop Vis. Analytics (EuroVA 2015)*, Cagliari, Italy, May 25–26, 2015, pp. 54–61.

[37] O. Parisot, M. Ghoniem, and B. Otjacques, "Decision trees and data preprocessing to help clustering interpretation," in *Proc. 3rd Int. Conf. Data Manage. Technol. and Appl. (DATA 2014)*, Vienna, Austria, Aug. 29–31, 2014, pp. 48–55.

[38] F. van Ham, M. Petitclerc, and R. Pisters, "Guiding multidimensional analysis using decision trees," in *Proc. 23rd Conf. Center for Adv. Stud. Collaborative Res. (CASCON 2013)*, Toronto, ON, Canada, Nov. 18–20, 2013, pp. 200–214.

[39] A. Chatzimparmpas, R. M. Martins, and A. Kerren, "t-viSNE: Interactive assessment and interpretation of t-SNE projections," *IEEE Trans. Visualization and Comp. Graph.*, 2020.

[40] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-SNE effectively," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/misread-tsne

[41] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," Dec. 2018, arXiv:1802.03426.

[42] J. Tang, J. Liu, M. Zhang, and Q. Mei, "Visualizing large-scale and high-dimensional data," in *Proc. 25th ACM Int. Conf. World Wide Web (WWW 2016)*, Montréal, QC, Canada, Apr. 11–15, 2016, pp. 287–297.

[43] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proc. 1st Pacific-Asia Conf. on Knowl. Discovery and Data Mining (PAKDD 1997)*, Singapore, 1997, pp. 21–34.

[44] M. Qiu, S. Davis, and F. Ikem, "Evaluation of clustering techniques in data mining tools," *Issues in Inf. Syst.*, vol. 5, no. 1, 2004.

[45] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Trans. Visualization and Comp. Graph.*, vol. 23, no. 1, pp. 241–250, 2016.

[46] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, "Interactive machine learning: letting users build classifiers," *Int. J. Human–Comput. Stud.*, vol. 55, no. 3, pp. 281–292, Sep. 2001.

[47] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Visualization and Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[49] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *IEEE Trans. Visualization and Comput. Graph.*, vol. 19, no. 9, pp. 1526–1538, Sep. 2013.

[50] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *IEEE Trans. Visualization and Comput. Graph.*, vol. 13, no. 6, pp. 1216–1223, Nov. 2007.

[51] M. Harrower and C. A. Brewer, "Colorbrewer.org: an online tool for selecting colour schemes for maps," *The Cartographic J.*, vol. 40, no. 1, pp. 27–37, 2003.

[52] *Zoo Data Set*, UCI Machine Learning Repository, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Zoo

[53] R. Hartson and P. S. Pyla, *The UX Book: Process and guidelines for ensuring a quality user experience*. Amsterdam, Netherlands: Elsevier, 2012.

[54] J. Brooke, "SUS – A quick and dirty usability scale," *Usability Eval. in Industry*, vol. 189, no. 194, pp. 4–7, 1996.

[55] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Human–Comput. Interact.*, vol. 24, no. 6, pp. 574–594, Jun. 2008.

[56] K. Finstad, "The system usability scale and non-native English speakers," *J. Usability Stud.*, vol. 1, no. 4, pp. 185–188, Aug. 2006.

[57] J. R. Lewis and J. Sauro, "Can I leave this one out?: the effect of dropping an item from the SUS," *J. Usability Stud.*, vol. 13, no. 1, pp. 38–46, Nov. 2017.

[58] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, May 2009.

[59] J. R. Lewis and J. Sauro, "The factor structure of the system usability scale," in *Proc. 1st Int. Conf. Human Centered Des. (HCD 2009)*, San Diego, CA, USA, Jul. 19–4, 2009, pp. 94–103.

Adrien Bibal is a Ph.D. student at the Université de Namur (Belgium) under the supervision of Professor Benoît Frénay. He received an M.S. degree in Computer Science and an M.A. degree in Philosophy from the Université catholique de Louvain (Belgium) in 2013 and 2015 respectively. His Ph.D. thesis in machine learning is on the interpretability of dimensionality reduction mappings.

Antoine Clarinval received a master degree in computer science in 2017 from the University of Namur, Belgium. He is currently pursuing the PhD degree at the University of Namur. His research interests include smart city education, citizen participation in smart cities, and how public displays and information visualization can support it. In this regard, he is especially interested in traffic data visualization and open data.

Bruno Dumas received his PhD in 2010 from the University of Fribourg, Switzerland. His PhD thesis focused on the creation of multimodal interfaces, following three axes: software architectures, modeling languages and multimodal fusion algorithms. He then worked for three and a half years at the Vrije Universiteit Brussel as a post-doc. His research areas focus on human-machine interaction, multimodal interfaces and more broadly on how the expansion of computing in everyday life influences usage.

Benoît Frénay is associate professor at the Université de Namur. He received his Ph.D. degree from the Université catholique de Louvain (Belgium) in 2013. His main research interests in machine learning include interpretability, interactive machine learning, dimensionality reduction, label noise, robust inference and feature selection. In 2014, he received the Scientific Prize IBM Belgium for Informatics for his PhD thesis on Uncertainty and Label Noise in Machine Learning.

CHAPTER 21

FINDING THE MOST INTERPRETABLE MDS ROTATION FOR SPARSE LINEAR MODELS BASED ON EXTERNAL FEATURES

The article presented in this chapter was published in the proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) in 2018.

Finding the Most Interpretable MDS Rotation for Sparse Linear Models based on External Features

Adrien Bibal[1]*, Rebecca Marion[2]* and Benoît Frénay[1]

1- NADI Institute - PReCISE Research Center
University of Namur - Faculty of Computer Science
Rue Grandgagnage 21, 5000 Namur - Belgium

2- ISBA - Université catholique de Louvain
Voie du Roman Pays 20, 1348 Louvain-la-Neuve - Belgium

Abstract. One approach to interpreting multidimensional scaling (MDS) embeddings is to estimate a linear relationship between the MDS dimensions and a set of external features. However, because MDS only preserves distances between instances, the MDS embedding is invariant to rotation. As a result, the weights characterizing this linear relationship are arbitrary and difficult to interpret. This paper proposes a procedure for selecting the most pertinent rotation for interpreting a 2D MDS embedding.

1 Introduction

In many applications, the usability of machine learning techniques depends on their interpretability [1]. This paper deals with the problem of understanding, or interpreting, a multidimensional scaling (MDS) embedding using features that were not used to compute the MDS (i.e. "external" features). This is a kind of *multi-view learning* task based on data from multiple sources [2]. The goal here is to characterize the relationship between two views: one taking the form of (dis-)similarities between instances and the other expressing features of these instances.

For example, in psychology, two independent experiments are sometimes run where one is used to interpret the result of the other. This is the case for implicit measure studies, which aim to understand human decisions encoded in one database by using another database. A first database is composed of similarity ratings for a set of instances, whereas the second database contains characterizations of the same instances with respect to a set of features. The research question is then: how can the feature matrix be used to explain the comparisons in the first database? Another field of application is the medical sciences, where clinical features can be used to interpret patient similarity with respect to gene expression, protein abundance, etc.

This work proposes an approach that strikes a balance between interpretability and performance: it finds an optimal rotation of an MDS embedding that can be used to identify a small subset of features necessary for accurately explaining that embedding. In this work, we focus on 2D MDS embeddings, constraining the rotation to revolve around a single axis.

*Both authors have contributed equally.

2 State of the Art

The problem of interpreting an MDS representation of a set of instances is frequently encountered in the social sciences (see, e.g., [3]). Let $\mathbf{Y}$ $(n \times K)$ be a matrix resulting from the application of MDS to an $n \times n$ (dis-)similarity matrix. Some authors interpret this embedding by clustering the instances in $\mathbf{Y}$ [4]. For 2D MDS embeddings ($K = 2$), another more popular approach is to regress a set of external features $\mathbf{f}_j$, $j : 1, .., d$, onto the MDS matrix $\mathbf{Y}$ through *property fitting* [5]: $\mathbf{f}_j = \mathbf{Y}\mathbf{w}_j + \boldsymbol{\xi}_j$, where $\mathbf{w}_j$ is a vector of weights and $\boldsymbol{\xi}_j$ is an error vector. A subset of features important for explaining the MDS dimensions are identified based on some measure of model fit, such as the coefficient of determination R^2. If the model for a given feature $\mathbf{f}_j$ has a sufficiently adequate fit with respect to some threshold, its line of fit is plotted in the MDS space. As a result, the MDS can be interpreted based on a subset of external features.

Unfortunately, because each feature is regressed separately onto $\mathbf{Y}$, potential dependence between features is ignored. In order to account for all features at once, some authors apply Principal Component Analysis (PCA) to a feature matrix $\mathbf{F}$ $(n \times d)$, then regress each principal component l onto the MDS matrix: $\text{PCA}(\mathbf{F})_l = \mathbf{Y}\mathbf{w}_l + \boldsymbol{\xi}_l$, for $l : 1, ..., q$, where q is the total number of principle components. Extra processing steps have also been proposed in order to allow the PCA components to be non-orthogonal (see [3] for an applied example).

While the weights for each dimension of $\text{PCA}(\mathbf{F})$ are still estimated independently of each other, this method has the advantage of accounting for dependence between features: each component regressed onto $\mathbf{Y}$ is a linear combination of features. However, the PCA components of $\mathbf{F}$ are estimated independently of the MDS embedding $\mathbf{Y}$. This means that the PCA components are not optimal, in terms of precision, for a regression onto the MDS space. In addition, the solution does not necessarily improve model interpretability, as a single principle component l could depend on all of the features in $\mathbf{F}$.

3 Proposed Approach

As seen in Section 2, there is a need for a method that identifies a small subset of features that best explain two MDS dimensions $\mathbf{y}_1$ and $\mathbf{y}_2$ while accounting for dependence between features $\mathbf{f}_j$. In order to allow features to jointly explain the MDS dimensions, we propose performing a linear regression where the MDS dimensions $\mathbf{y}_1$ and $\mathbf{y}_2$ are response variables, rather than predictors, and thus the predictors are the features in $\mathbf{F}$. This section presents our motivation and goals, as well as our proposed approach, which is then evaluated in Section 4.

3.1 Motivation and Goals

Let the multivariate regression model be defined as $\mathbf{Y} = \mathbf{F}\mathbf{W} + \boldsymbol{\Xi}$, where $\mathbf{W}$ $(d \times 2)$ is a matrix containing the regression weights to be estimated and $\boldsymbol{\Xi}$ $(d \times 2)$ is an error matrix. Variables with non-zero weights for a given dimension of $\mathbf{Y}$ are considered to be explicative of the corresponding axis in the 2D MDS space.

Unfortunately, the orientation of the MDS embedding $\mathbf{Y}$ is arbitrary, meaning that the weights $\mathbf{W}$ are also arbitrary, and thus difficult to interpret. Indeed, $\mathbf{Y}$ is found by minimizing a measure of the degree to which distances between n instances in the $n \times n$ (dis-)similarity space are preserved in the new $n \times 2$ space. A popular measure of this kind is the Kruskal stress [6]. Minimizing this criterion results in MDS solutions with arbitrary orientations because distances between instances in the resulting space remain the same for any rotation.

Rather than simply regressing the arbitrarily rotated MDS solution $\mathbf{Y}$ onto $\mathbf{F}$, it could be more relevant to find a rotation of $\mathbf{Y}$ that optimizes some criterion related to the analysis goals at hand. Let the 2D rotation matrix $\mathbf{R}^\theta$ for a given angle θ be defined as

$$\mathbf{R}^\theta = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}.$$

The regression model of interest is thus: $\mathbf{YR}^\theta = \mathbf{FW}^\theta + \mathbf{\Xi}$, where $\mathbf{W}^\theta$ is a weight matrix that depends implicitly on the rotation angle θ.

For our particular case, we are interested in finding the rotation angle θ that optimizes some trade-off between interpretability and model error. We assume that the model is most interpretable when the number of non-zero weights in $\mathbf{W}^\theta$ is minimal, i.e. the model is "sparse."

Without considering sparsity, the ordinary least squares (OLS) solution for $\theta = 0$ is given by $\mathbf{W}^0 = (\mathbf{F}^\top\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{Y}$. It can be shown that the OLS solution for any θ is $\mathbf{W}^\theta = (\mathbf{F}^\top\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{YR}^\theta = \mathbf{W}^0\mathbf{R}^\theta$. Thus, the effect of rotating $\mathbf{Y}$ is to rotate $\mathbf{W}^0$ with the same angle, and the mean squared error (MSE) of the model, which is the sum of the MSE for $\mathbf{y}_1$ and $\mathbf{y}_2$, is invariant under rotation.

The OLS solution, however, does not guarantee interpretability as defined above. In order to encourage interpretability, some model constraint must be included so that unimportant variables are excluded from the model. A natural constraint for this purpose is the L_0 norm, which counts the number of non-zero weights in the model. The function to minimize is

$$\frac{1}{2n}\sum_{k=1}^{2}||\mathbf{Yr}_k^\theta - \mathbf{Fw}_k^\theta||_2^2 + \sum_{k=1}^{2}\lambda||\mathbf{w}_k^\theta||_0, \tag{1}$$

where λ is a tuning parameter that controls the trade-off between model error and interpretability. Optimizing Eq. (1) with respect to $\mathbf{W}^\theta$ is an NP-Hard problem [7], so in practice, the L_1 norm is often used as an approximation [8]:

$$\frac{1}{2n}\sum_{k=1}^{2}||\mathbf{Yr}_k^\theta - \mathbf{Fw}_k^\theta||_2^2 + \sum_{k=1}^{2}\lambda||\mathbf{w}_k^\theta||_1. \tag{2}$$

For a given θ, the solution $\mathbf{W}^\theta$ is found using any Lasso implementation. However, in contrast to the OLS solution, for a given λ, the model error and sparsity of the Lasso solution depend on the rotation angle (see Section 4.3). The optimal rotation angle θ^* being unknown, it must be optimized.

3.2 Finding the Best Rotation with the $\mathbf{L}_0$ Norm

The proposed procedure for finding the best interpretable rotation (BIR) provides an optimal angle θ^* and associated weight matrix $\mathbf{W}^{\theta^*}$ for which the number of non-zero weights and the model error are minimized. In an approach inspired by [9], the procedure finds an angle θ whose corresponding Lasso solution $\mathbf{W}^{\theta}$ minimizes Eq. (1). This procedure is formalized by

$$\theta^* = \arg\min_{\theta} \sum_{k} \left(\frac{1}{2n} ||\mathbf{Y}\mathbf{r}_k^{\theta} - \mathbf{F}\mathbf{w}_k^{\theta}||_2^2 + \lambda ||\mathbf{w}_k^{\theta}||_0 \right), \tag{3}$$

where $\mathbf{W}^{\theta} = \text{Lasso}(\mathbf{F}, \mathbf{Y}\mathbf{R}^{\theta}, \lambda)$, which is found by minimizing Eq. (2). The univariate function to minimize in Eq. (3) being non-convex, any generic solver for non-convex optimization may be used.

4 Evaluation

This section evaluates the performance of the Lasso solution when the matrix $\mathbf{Y}$ is rotated with the angle found using the BIR selection procedure. This is then compared to (i) the average performance of angles resulting in the least sparse Lasso solutions, as well as (ii) the estimated performance when $\mathbf{Y}$ is rotated with a random angle from the set $\Theta = \{0.1, 0.2, ..., 360\}$ degrees. The first case demonstrates the worst case scenario and the second represents the estimated expected performance obtained for an arbitrary MDS orientation.

4.1 Data and Pre-Processing

We evaluated the performance of the proposed BIR selection procedure on five popular datasets: Hepatitis, Dermatology, Heart (Statlog), and Pima Indians Diabetes from [10] and Diabetes from [11]. These datasets were chosen because their features can be easily split into two different, meaningful data views. For example, Hepatitis can be split into a view with basic clinical features (e.g. age, family history, etc.) and another view with more complex histopathological features (e.g. melanin incontinence, etc.). For each dataset, we removed all instances with missing values. We used the view with the most complex features to compute a dissimilarity matrix based on Euclidean distances, then applied 2D metric MDS. We used the other view (normalized) to interpret the MDS space.

4.2 Evaluation Criteria

We evaluated the BIR procedure using two criteria. The first criterion, referred to as s^{θ}, measures the degree of model sparsity (i.e. interpretability), and is calculated as $\sum_{k=1}^{2} ||\mathbf{w}_k^{\theta}||_0$, the number of non-zero weights in $\mathbf{W}^{\theta}$. $\text{Prob}(s^{\theta}) = \frac{1}{|\Theta|} \left| \left\{ \theta' \in \Theta \mid \sum_{k=1}^{2} ||\mathbf{w}_k^{\theta'}||_0 = s^{\theta} \right\} \right|$ represents the approximate probability that Lasso obtains a degree of sparsity s^{θ} when θ is chosen at random. The second criterion is the overall model error $MSE = \frac{1}{2n} \sum_{k=1}^{2} ||\mathbf{Y}\mathbf{r}_k^{\theta} - \mathbf{F}\mathbf{w}_k^{\theta}||_2^2$.

Dataset	Angle Selection	θ (°)	s^θ	Prob(s^θ)	MSE
Hepatitis	least sparse case		11	8.9%	0.169
$d = 15$	average case		9.1		0.170
30 weights	BIR procedure	59.8	6	2.1%	0.168
Dermatology	least sparse case		12	3.1%	0.098
$d = 17$	average case		9.5		0.092
34 weights	BIR procedure	36.4	7	12.7%	0.086
Heart	least sparse case		3	71.9%	0.180
$d = 4$	average case		2.7		0.180
8 weights	BIR procedure	0.9	2	28.1%	0.180
Diabetes	least sparse case		7	42.5%	0.195
$d = 5$	average case		5.7		0.194
10 weights	BIR procedure	68.2	3	10.1%	0.191
Pima	least sparse case		5	8.6%	0.220
$d = 5$	average case		3.4		0.219
10 weights	BIR procedure	20.3	2	0.7%	0.224

Table 1: Comparison of BIR selection with the least sparse and average cases. The total number of weights is twice the number of external features ($= 2 \times d$).

4.3 Results

For each dataset, the results presented here correspond to (i) the least sparse rotations, which highlights the importance of choosing an appropriate angle, (ii) the expected value estimated by averaging the criterion values for all θ in the set $\Theta = \{0.1, 0.2, ..., 360\}$ degrees, and (iii) the rotation chosen by the BIR procedure. The relative performance of i-iii was similar for a variety of λ values. Experimental results for one of these values, $\lambda = 0.1$, can be found in Table 1.

4.4 Discussion

For all datasets, the BIR procedure yields a solution that is 1.5-2.5 times more sparse than the least sparse solution with a negligible computational cost (a few seconds). Selecting a random angle results, on average, in models that are also less sparse than for the BIR procedure, with a greater or equal error for all but one dataset. These results suggest that using a rotation selection procedure is advantageous for someone requiring interpretability. Furthermore, the probability of randomly choosing a solution with the least sparsity can be high relative to a sparser solution. For Diabetes, there is a 42.5% chance of randomly selecting a rotation yielding 7 non-zero weights, whereas a solution with only 3 non-zero weights can be found using the BIR procedure.

5 Conclusion

This paper demonstrates the importance of choosing a rotation angle for a 2D MDS embedding that makes it easier to interpret. A procedure is provided for

selecting a rotation and estimating a sparse linear regression model that finds a compromise between interpretability and model error.

In the current procedure, an optimal rotation angle θ^* is chosen by minimizing a function that depends on Lasso solutions $\mathbf{W}^\theta$. In a future work, it would be interesting to develop a more direct and simultaneous optimization of the angle and weight matrix. Another extension would be to tackle the problem of rotating an MDS space with more than two dimensions, which would require the optimization of a vector $\boldsymbol{\theta}$. Moreover, a more nuanced definition of interpretability could be used to encourage both overall sparsity and an equal distribution of non-zero weights among the MDS dimensions.

Acknowledgment

The authors want to thank Nathan Nguyen from the Université catholique de Louvain for having pointed to the need for this kind of procedure in psychology. The second author gratefully acknowledges financial support from the IAP Research Network P7/06 of the Belgian State (Belgian Science Policy), the Fonds spécial de recherche (Fédération Wallonie-Bruxelles) and the Belgian Fund for Scientific Research (F.R.S.-FNRS, FRIA grant).

References

[1] A. Bibal and B. Frénay. Interpretability of machine learning models and representations: an introduction. In *Proceedings of ESANN*, pages 77–82, 2016.

[2] S. Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013.

[3] A. Koch, R. Imhoff, R. Dotsch, C. Unkelbach, and H. Alves. The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion. *Journal of Personality and Social Psychology*, 110(5):675, 2016.

[4] M. L. Davison and S. G. Sireci. Multidimensional scaling. In *Handbook of applied multivariate statistics and mathematical modeling*, pages 323–352. Elsevier, 2000.

[5] J. J. Chang and J. D. Carroll. How to use PROFIT, a computer program for property fitting by optimizing nonlinear or linear correlation. *Unpublished manuscript, Bell Laboratories*, 1968.

[6] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[7] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[8] C. Ramirez, V. Kreinovich, and M. Argaez. Why L1 is a good approximation to L0: A geometric explanation. *Journal of Uncertain Systems*, 7, 2013.

[9] W. Herlands, M. De-Arteaga, D. Neill, and A. Dubrawski. Lass0: sparse nonconvex regression by local search. *NIPS Workshop on Optimization*, 2015.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

CHAPTER 22

BIR: A METHOD FOR SELECTING THE BEST INTERPRETABLE MULTIDIMENSIONAL SCALING ROTATION USING EXTERNAL VARIABLES

The article presented in this chapter was published in the journal Neurocomputing in 2019.

BIR: A Method for Selecting the Best Interpretable Multidimensional Scaling Rotation using External Variables

Rebecca Marion[a,*], Adrien Bibal[b,*], Benoît Frénay[b]

[a] *ISBA, IMMAQ, Université catholique de Louvain, Voie du Roman Pays 20, B-1348 Louvain-la-Neuve, Belgium*
[b] *PReCISE, NADI, Faculty of Computer Science, University of Namur, Rue Grandgagnage 21, B-5000 Namur, Belgium*

Abstract

Interpreting nonlinear dimensionality reduction models using external features (or external variables) is crucial in many fields, such as psychology and ecology. Multidimensional scaling (MDS) is one of the most frequently used dimensionality reduction techniques in these fields. However, the rotation invariance of the MDS objective function may make interpretation of the resulting embedding difficult. This paper analyzes how the rotation of MDS embeddings affects sparse regression models used to interpret them and proposes a method, called the Best Interpretable Rotation (BIR) method, which selects the best MDS rotation for interpreting embeddings using external information.

Keywords: Interpretability, Dimensionality Reduction, Multidimensional Scaling, Orthogonal Transformation, Multi-View, Sparsity, Lasso Regularization

1. Introduction

Dimensionality reduction consists of mapping instances from a certain space into a lower-dimensional space. For *nonlinear dimensionality reduction* (NLDR), this mapping is nonlinear, meaning that the new representation of the instances is not a linear transformation of the instances in the original space. NLDR is especially useful when the relationship between features is not linear, for instance in psychology [2] and ecology [3]. However, the nonlinear mapping of instances from high to low dimension makes it difficult to interpret the resulting embedding, whose axes do not have an easily apparent meaning.

In many cases, interpretability is essential to the use of machine learning models [4]. In the context of NLDR, the model of interest is the nonlinear mapping function, which is sometimes interpreted based on an additional set of features. By studying the relationship between the NLDR output and this set of features, the model that generated the output can be interpreted. For example, in implicit measure studies in psychology [5], data describing a given set of instances are collected in two, often independent, experiments. The instances from one experimental dataset are mapped into a reduced space using multidimensional scaling, and then the features from the other dataset are used to interpret the mapping by finding trends with linear functions.

Using a second set of features to interpret an NLDR embedding is also a popular approach in ecology [3]. For instance, a collection of abiotic features – such as soil acidity, temperature and altitude – may be used to interpret similarities and differences between sampling sites in terms of species abundance. A dataset of species abundance for a variety of sampling sites is mapped to a lower-dimensional space using an NLDR method, and then a dataset of abiotic features for these same sites is used to identify a link between abiotic environmental conditions and species abundance.

This approach to the interpretation of NLDR is an example of *multi-view learning*, also known as *data fusion*, or *coupled*, *linked*, *multiset*, *multiblock* or *integrative data analysis* [6], where different feature sets are used to solve a machine learning problem [7]. In this particular case, one view (the m-dimensional NLDR embedding of n instances) is interpreted using another view (d features of the same n instances, i.e. "exter-

*Corresponding authors. Both authors contributed equally. Name order reversed with respect to [1].

Email addresses: rebecca.marion@uclouvain.be (Rebecca Marion), adrien.bibal@unamur.be (Adrien Bibal), benoit.frenay@unamur.be (Benoît Frénay)

nal" variables, which were not used to compute the embedding). Similar two-view problems are encountered in a variety of fields, including, but not restricted to, psychology [2], epidemiology [8], ecology [9], biology [10] and chemometrics [11].

In this work, we are interested in NLDR methods whose objective function is rotation-invariant, particularly *multidimensional scaling* (MDS) [12]. MDS is an NLDR technique that takes an $n \times n$ (dis)similarity or distance matrix $\mathbf{D}$ as input and outputs an embedding (or configuration) $\mathbf{X}$ of these n instances in an m-dimensional space, with $m \ll n$ [12, 13]. More precisely, MDS finds a matrix $\mathbf{X}$ such that (dis)similarities d_{ij} in $\mathbf{D}$ can be mapped to distances between m-dimensional vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ with minimal loss.

In order to find this mapping, the MDS algorithm must minimize a loss function often called the *stress function*. This stress function can take many forms, but one of the most frequently used functions is Kruskal's stress function [12, 13]:

$$\text{stress} = \sqrt{\frac{\sum_{i,j}[d_{ij} - \text{dist}(\mathbf{x}_i, \mathbf{x}_j)]^2}{\sum_{i,j} d_{ij}^2}}. \quad (1)$$

One particular property of this stress function is that by preserving the relative distances between each pair of instances, the stress function is invariant to a variety of transformations of $\mathbf{X}$. Indeed, the same stress score can be obtained under transformations such as translation, reflection and rotation [13]. The indetermination of the embedding rotation is the motivation for this work.

In practice, MDS is used in psychology and other fields as a means of projecting data into a viewable space, often in two or three dimensions [14]. MDS is also useful for processing data that is stored as (dis)similarity pairs, and it can handle ordinal (dis)similarity values (processed with non-metric MDS) or continuous ones (processed with metric MDS). The widespread use of MDS is supported by its implementation in various social science tools such as SPSS and ANTHROPAC. As the purpose of MDS in practice is to understand data, interpreting the MDS embedding is a crucial step, which is carried out by experts, machine learning techniques or both.

However, the MDS embedding rotation is an issue when the arbitrarily oriented MDS axes must be interpreted. This paper, which is an extended version of [1], analyzes how the rotation of MDS embeddings affects their interpretation and proposes a method for handling this rotational indeterminacy.

This paper is structured as follows. Section 2 reviews how MDS embeddings are interpreted in the literature. Section 3 exposes issues related to embedding orientation when the embedding axes are interpreted using multiple regression models. Section 4 presents several machine learning and statistical methods that can be used to solve such a problem. The *Best Interpretable Rotation* (BIR) selection method that we developed to select the best MDS embedding orientation for interpretation is described in Section 5. Section 6 presents the results of two experiments evaluating the performance of BIR and shows how it compares to the methods listed in Section 4. Discussions about these results are presented in Section 7. Finally, we conclude our paper and provide directions for future work in Section 8.

2. Interpreting an MDS Embedding

Two different and complementary uses of multidimensional scaling (MDS) stand out: *exploratory* and *confirmatory* uses [13, 14]. For the former, the MDS embedding is used as a means of discovering hidden structures in (dis)similarity data [2]. Expert knowledge is therefore needed for analyzing the MDS embedding. For the latter use, the MDS embedding is used to confirm hypotheses the researcher has in mind *a priori* [14]. In this case, *external features* (or *external variables*) are used to discover patterns in the embedding. As the confirmatory process must remain objective, the user lets machine learning techniques find the patterns for him.

For each of these two purposes, there are two main ways to interpret MDS embeddings: *neighborhood interpretation* and *dimensional interpretation* [12]. Clustering (or cluster analysis) is the machine learning problem associated with the first type of interpretation. The goal of clustering is to group instances in a given dataset. The groups found by clustering algorithms are called *clusters*. For instance, Lebel et al. [15] use an agglomerative hierarchical clustering technique for exploring their MDS embedding. They then ask experts to provide an interpretation for each cluster found, as well as each dimension. Therefore, they combine neighborhood interpretation and dimensional interpretation for the purpose of exploration. For hypothesis confirmation, the clustering of instances based on external features is not often used in the literature.

In the context of hypothesis confirmation, the most frequently used technique to link external features with an embedding is linear regression [12, 14]. More precisely, let $\mathbf{X}$ be an $n \times m$ embedding and $\mathbf{F}$ an $n \times d$ matrix of external features. The goal is to estimate the weights (or parameters) $\mathbf{W}$ in

$$\mathbf{F} = \mathbf{X}\mathbf{W} + \mathbf{E}, \quad (2)$$

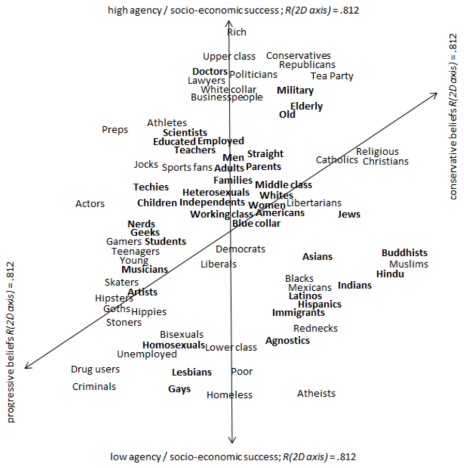


Figure 1: Figure reproduced from Koch et al. [17] presenting two stereotype trends in an MDS embedding of social groups: socio-economic success (vertical line) and beliefs (oblique line).

with $\mathbf{E}$ being an error term [12]. In most cases, $m = 2$ to allow visualization of the embedding $\mathbf{X}$. Indeed, a line representing the trend explained by a given feature $\mathbf{f}_j$ can be drawn in a 2D plot of the embedding $\mathbf{X}$. This line is given by the unit vector $\hat{\mathbf{w}}_j$, whose m elements are normalized versions of w_{jk}, also called direction cosines $\hat{w}_{jk}$, where k is a given dimension of embedding $\mathbf{X}$ [12]:

$$\hat{w}_{jk} = \frac{w_{jk}}{\sqrt{w_{j1}^2 + w_{j2}^2 + ... + w_{jm}^2}}, \tag{3}$$

with m being the total number of dimensions in $\mathbf{X}$.

In the literature, such an approach is often called PROFIT [16]. PROFIT stands for *PROperty FITting*, with the external features understood as properties. Many articles in the literature use this kind of approach for interpreting MDS embeddings (e.g. [17, 18, 19, 20, 21]). Often, the coefficient of determination R^2 is used to select the fitted properties to keep. Figure 1 shows an example of the regression of external features onto an MDS embedding. The two stereotype trends "socio-economic success" and "type of beliefs" are drawn on an MDS embedding containing social groups as instances.

One drawback of such an approach is that each feature is independently regressed onto the MDS embedding, making it impossible to relate the MDS dimensions to combinations of features. This is problematic because each MDS dimension might best be described by a linear combination of features rather than an individual feature. In order to address this issue, *principal component analysis* (PCA) is often run on the external feature matrix $\mathbf{F}$ in order to extract principal components that are then interpreted as meta-features. For instance, Koch et al. [17] in Figure 1 extract their "agency/socio-economic success" stereotype feature from a linear combination of six other stereotypes: powerless-powerful, dominated-dominating, low status-high status, poor-wealthy, unconfident-confident and unassertive-competitive.

As a complement to this, rotation of these components can overcome some limitations of PCA. Indeed, rotation may be useful for either achieving a more understandable distribution of the features in the PCA components (with e.g. a *varimax rotation* [22]) or, if orthogonality of the PCA components is not desired or required, for breaking the orthogonality of the components (with e.g. an *oblimin rotation* [23]).

Nonetheless, the interpretation problem is not fully addressed by these approaches, as the combination of features is not optimized with respect to the information in the MDS embedding. It would be more appropriate to find the best combination of external features for explaining the embedding. The next section presents the problem of reversing the regression direction in order to account for linear combinations of external features, as well as subsequent issues raised by this problem.

3. Problem Statement

In this paper, we are interested in using a *multi-view learning* approach (see Section 1) in order to interpret an MDS mapping model. In particular, a matrix of external features (view 1) is used to interpret the dimensions of an MDS embedding (view 2). In this context, it seems natural to model each MDS dimension as a linear combination of these external features, rather than modeling the features as linear combinations of the MDS dimensions, as was seen in Section 2. The problem of interest is thus to estimate $\mathbf{W}$ in

$$\mathbf{X} = \mathbf{F}\mathbf{W} + \mathbf{E}, \tag{4}$$

where $\mathbf{X}$ is an $n \times m$ MDS embedding, $\mathbf{F}$ is an $n \times d$ matrix of external features and $\mathbf{W}$ is a $d \times m$ matrix of regression weights. The following sections focus on a two-dimensional embedding ($m = 2$) in order to simplify the optimization of the proposed method, and we assume that $d > m$.

As seen in Section 1, the MDS solution is only uniquely determined up to some transformations, including orthogonal transformations such as rotation. The orientation of MDS embeddings is thus arbitrary, and as a consequence, the magnitude of the weights in $\mathbf{W}$ is also arbitrary. Let $\mathbf{W}$ be the ordinary least squares (OLS) weights for a model where $\mathbf{X}$ is not rotated, and let $\mathbf{W}^\theta$ be the OLS weights for a model where $\mathbf{X}$ is rotated by any angle $\theta \in [0, 360]$ degrees. We have that

$$\mathbf{W}^\theta = \mathbf{W}\mathbf{R}^\theta, \tag{5}$$

where $\mathbf{R}^\theta$ is an orthogonal rotation matrix defined as

$$\mathbf{R}^\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{6}$$

This follows from the fact that the OLS objective function is invariant to rotation. Indeed,

$$\arg\min_{\mathbf{W}} \|\mathbf{X}\mathbf{R}^\theta - \mathbf{F}\mathbf{W}\mathbf{R}^\theta\|_F^2 = \arg\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{F}\mathbf{W}\|_F^2. \tag{7}$$

Let $\mathbf{M} = \mathbf{X} - \mathbf{F}\mathbf{W}$. By expressing the Frobenius norm as a trace, and using the fact that $\mathbf{R}^\theta\mathbf{R}^{\theta\top} = \mathbf{I}$ and that the trace is invariant under cyclic permutation, we can show that

$$\begin{aligned} &\|\mathbf{X}\mathbf{R}^\theta - \mathbf{F}\mathbf{W}\mathbf{R}^\theta\|_F^2 \\ &= \|\mathbf{M}\mathbf{R}^\theta\|_F^2 \\ &= \text{trace}(\mathbf{R}^{\theta\top}\mathbf{M}^\top\mathbf{M}\mathbf{R}^\theta) \\ &= \text{trace}(\mathbf{R}^\theta\mathbf{R}^{\theta\top}\mathbf{M}^\top\mathbf{M}) \quad \textit{cyclic permutation} \\ &= \text{trace}(\mathbf{M}^\top\mathbf{M}) \qquad\qquad \mathbf{R}^\theta\mathbf{R}^{\theta\top} = \mathbf{I} \\ &= \|\mathbf{M}\|_F^2 \\ &= \|\mathbf{X} - \mathbf{F}\mathbf{W}\|_F^2. \end{aligned} \tag{8}$$

As shown in Figure 2, rotating the matrix $\mathbf{X}$ results in weight magnitudes that are a sinusoidal function of the rotation angle θ. While the model error remains constant for all rotations, some rotations yield models that are easier to interpret than others (i.e. rotation angles yielding more model weights equal to zero). This means that the arbitrary rotation of an embedding generated by MDS may not be the best rotation for interpretation. Thus, modeling the MDS dimensions as a function of the feature matrix introduces a new problem: the determination of a non-arbitrary rotation that facilitates the interpretation of the MDS dimensions.

The analyses in this paper are applied to MDS embeddings, but the rotation problem exists for any $\mathbf{X}$ generated using an NLDR method with a rotation-invariant objective function (e.g. t-SNE [24]).

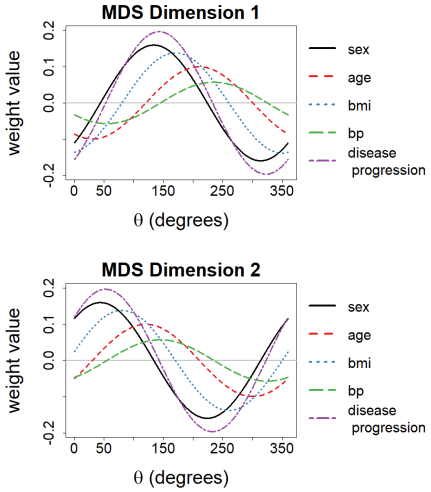


Figure 2: Example of regression weights for OLS models estimated when a 2D MDS embedding is rotated with different angles θ.

4. Existing Methods for View Rotation

The MDS objective function preserves Euclidean distances between all pairs of points, which makes it invariant to orthogonal transformations. As such, any orthogonal transformation of a given MDS embedding is an equally valid solution to the MDS problem. While this paper is primarily concerned with the problem of rotating an MDS embedding, any orthogonal transformation, including rotation and/or reflection, could be applied to an embedding. This section presents several approaches from the statistics and machine learning literature for orthogonally transforming a data "view" – in this case, an embedding generated by MDS. In what follows, $\mathbf{R}$ is an orthogonal transformation matrix of any kind, not exclusively a rotation matrix.

4.1. Principal Component Analysis

The most well known and frequently used single-view rotation method is principal component analysis (PCA). As mentioned in Section 2, PCA can be applied to the matrix of features $\mathbf{F}$ to generate principal components that are then regressed onto an MDS embedding $\mathbf{X}$. However, in this work, we are primarily interested in an orthogonal transformation of $\mathbf{X}$, not $\mathbf{F}$.

In this context, the goal of PCA is to find an orthogonal transformation matrix $\mathbf{R}$ ($m \times m$) that maximizes the

variance in successive columns of $\mathbf{Z} = \mathbf{XR}$. As such, $\mathbf{Z}$ is a rotation of $\mathbf{X}$ such that each successive column of $\mathbf{Z}$ captures a maximum of the variance in $\mathbf{X}$ not already represented in the previous columns.

4.2. Orthogonal Procrustean Transformation

Procrustean transformation [13] is one of the most frequently used MDS embedding transformations. This transformation aims to align an MDS embedding with another matrix. Most of the time, it is used to align two 2D or 3D embeddings in order to visually compare them and remove indeterminacies linked to their orientation or dilation. However, the problem can be generalized to the case where the two matrices do not have the same dimensionality, e.g. by adding columns of zeros [13, 25].

Let $\mathbf{X}$' be the concatenation of $\mathbf{X}$ and a matrix with $d - m$ columns of zeros, such that $\mathbf{X}$' ($n \times d$) and $\mathbf{F}$ ($n \times d$) have the same dimensionality. In the orthogonal Procrustes problem, $\mathbf{X}$' is transformed with a matrix $\mathbf{R}$ in order to minimize the squared distance between $\mathbf{X}'\mathbf{R}$ and the $n \times d$ target matrix $\mathbf{F}$ [13]:

$$\begin{aligned} \arg\min_{s,\mathbf{R}} \ & \text{tr}\,[(\mathbf{F} - s\mathbf{X}'\mathbf{R})^\top(\mathbf{F} - s\mathbf{X}'\mathbf{R})] \\ & \text{s.t. } \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \end{aligned} \tag{9}$$

where $\mathbf{R}$ is the $d \times d$ Procrustean transformation matrix and s is a scaling factor. The trace calculated is the sum of the squared distances between each point i in $\mathbf{F}$ and the corresponding point i in $\mathbf{X}'\mathbf{R}$, which are found in the diagonal of $(\mathbf{F} - s\mathbf{X}'\mathbf{R})^\top(\mathbf{F} - s\mathbf{X}'\mathbf{R})$ [13].

4.3. Eigenvector Partial Least Squares (PLS)

Eigenvector partial least squares (PLS) [11], also known as Bookstein PLS [26], is a two-view matrix factorization method. The goal of eigenvector PLS is to find orthogonal transformation matrices $\mathbf{P}$ and $\mathbf{R}$ such that the covariance between $\mathbf{T} = \mathbf{FP}$ and $\mathbf{Z} = \mathbf{XR}$ is maximal. Both $\mathbf{T}$ and $\mathbf{Z}$ are of dimension $n \times p$, where $p = \min(m, d)$, d is the number of features in $\mathbf{F}$ and m is the number of columns in $\mathbf{X}$.

4.4. Eigenvector PLS Regression (PLS-R)

Eigenvector PLS Regression (PLS-R) is an extension of eigenvector PLS to regression. Orthogonal transformation matrices $\mathbf{R}$ and $\mathbf{P}$ are first found using eigenvector PLS. Then, the matrix $\mathbf{Z} = \mathbf{XR}$ is regressed onto $\mathbf{T} = \mathbf{FP}$ using ordinary least squares (OLS). The model is defined as

$$\mathbf{Z} = \mathbf{TB} + \mathbf{E}, \tag{10}$$

where $\mathbf{E}$ is an error term and $\mathbf{B}$ is a matrix of regression weights, calculated as follows:

$$\mathbf{B} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Z}. \tag{11}$$

The orthogonally transformed view $\mathbf{XR}$ can thus be interpreted as a linear combination of the features in $\mathbf{F}$:

$$\mathbf{XR} = \mathbf{TB} + \mathbf{E} = \mathbf{FW} + \mathbf{E}, \tag{12}$$

where $\mathbf{W} = \mathbf{PB}$ is a matrix of regression weights describing the linear relationship between each feature in $\mathbf{F}$ and each dimension of $\mathbf{XR}$.

4.5. Sparse Reduced Rank Regression (SRRR)

Unlike eigenvector PLS-R, Sparse Reduced Rank Regression (SRRR) [27] introduces a constraint to encourage $\mathbf{W}$ to be sparse. Both $\mathbf{R}$ ($m \times p$) and $\mathbf{W}$ ($d \times p$) are constrained to have rank $p \leq \min(m, d)$, p being a hyperparameter that must be selected. $\mathbf{R}$ and $\mathbf{W}$ are found by optimizing the objective function

$$\begin{aligned} \arg\min_{\mathbf{R},\mathbf{W}} \ & \|\mathbf{XR} - \mathbf{FW}\|_F^2 + \gamma \sum_{j=1}^{d} \|\mathbf{w}_j\|_2 \\ & \text{s.t. } \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \end{aligned} \tag{13}$$

where $\mathbf{w}_j$ is the j^{th} row of $\mathbf{W}$ and $\gamma > 0$. The second term in Equation (13) is a type of *group regularization*, as groups of weights are penalized together. Note that the L_2 norm $\|\mathbf{w}_j\|_2$ is not squared, and as a result, it forces the elements of $\mathbf{w}_j$ to be either all zero or non-zero (see [28] for more details). As γ increases, more and more rows of $\mathbf{W}$ are set to zero, meaning that the associated features are no longer active in the model. Equation (13) is optimized by alternating between the optimization of $\mathbf{R}$ for fixed $\mathbf{W}$ and $\mathbf{W}$ for fixed $\mathbf{R}$.

4.6. Summary and Shortcomings

The most frequently used orthogonal transformation, PCA, maximizes explained variance by considering only the matrix to which the transformation is applied, making it a single-view method. For our problem setting, the multi-view methods presented above are more appropriate than PCA because the transformation of $\mathbf{X}$ with respect to $\mathbf{F}$ is directly optimized: it is learned using the external feature matrix $\mathbf{F}$ that will later be used to build the model linking the two views.

While orthogonal Procrustean transformation considers both matrices $\mathbf{X}$ and $\mathbf{F}$ for transforming the former, it requires the two matrices to have the same dimensionality. If this is not the case, the number of dimensions

in the smaller matrix must be artificially increased before the transformation is applied. In our case, $m < d$, meaning that both the augmented matrix **X**' and its transformed version **X**'**R** have d columns. Because of this, it is difficult to compare Procrustean transformation to the other methods in this section, which find a m-dimensional orthogonal matrix **R**.

Eigenvector PLS aligns two matrices **X** and **F** using orthogonal transformations such that the dimensionality of **X** ($n \times m$) is preserved. However, for eigenvector PLS-R, the weights linking the two matrices are not sparse, making the interpretation of **XR** difficult in most cases.

SRRR yields a more easily interpretable model than the other multi-view methods because it encourages sparsity in the matrix of regression weights. However, when features are included in the model, they have non-zero-valued weights for each dimension of **XR** due to the group penalty. This is problematic for the interpretation of the MDS axes in **X**, because this group penalization implies that all of the axes are explained by the same features.

Thus, while a few existing multi-view methods are able to find orthogonal transformations adapted for subsequent regression problems (eigenvector PLS-R and SRRR), the sparsity of the models generated is insufficient, in the case of eigenvector PLS-R, and the distribution of non-zero-valued weights is ill adapted to the problem at hand, in the case of SRRR.

5. Proposed Method: BIR Selection

Among all possible MDS embedding rotations, the rotation that interests us is the one making it possible to understand the embedding. In order to do so, some methods, such as SRRR, presented in Section 4, regularize the regression model used to understand the embedding. However, as observed in Section 3, regression weights change depending on the chosen rotation, which implies different possible interpretations of these weights. For a better understanding of how rotation affects regularized regression weights, Section 5.1 analyzes weight changes for Ridge regularization. Note that this type of penalization may not be adapted to our problem because it shrinks all weight values towards zero, yielding small but non-zero weight values. Section 5.2 analyzes weight changes for sparse regression performed using Lasso regularization. After having analyzed various rotation effects, Section 5.3 presents the Best Interpretable Rotation (BIR) selection method, and Section 5.4 presents an extension of BIR, BIR Lasso regression (BIR-LR), which learns a sparse regression model based on the rotation chosen by BIR.

5.1. Effect of Rotation on Ridge Regularization

Ridge regression adds a term to the OLS objective function that penalizes weight values through a squared Euclidean norm (also called the L_2 norm):

$$\arg\min_{\mathbf{W}} \|\mathbf{XR} - \mathbf{FW}\|_F^2 + \lambda \sum_{j=1}^{d} \|\mathbf{w}_j\|_2^2, \tag{14}$$

where the hyperparameter λ controls the balance between error and regularization. The squared L_2 norm shrinks weight values towards zero.

As this work is concerned with the rotation of **X** and its effect on a subsequent regression model, Figure 3a shows how Ridge regression weights depend on rotation angle. As with OLS, the Ridge objective function is rotation invariant, so the regression weights are a sinusoidal function of the rotation angle θ. Note that $\sum_{j=1}^{d} \|\mathbf{w}_j\|_2^2$ can be rewritten using a squared Frobenius norm, $\|\mathbf{W}\|_F^2$, which is rotation invariant. Using the same logic as in Equation (8), we can show that

$$\begin{aligned} &\arg\min_{\mathbf{W}} \|\mathbf{XR}^\theta - \mathbf{FWR}^\theta\|_F^2 + \lambda \|\mathbf{WR}^\theta\|_F^2 \\ &= \arg\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{FW}\|_F^2 + \lambda \|\mathbf{W}\|_F^2. \end{aligned} \tag{15}$$

As with OLS, $\mathbf{W}^\theta$, the weights for a given rotation θ, are equal to $\mathbf{WR}^\theta$, and the error is constant for all rotations.

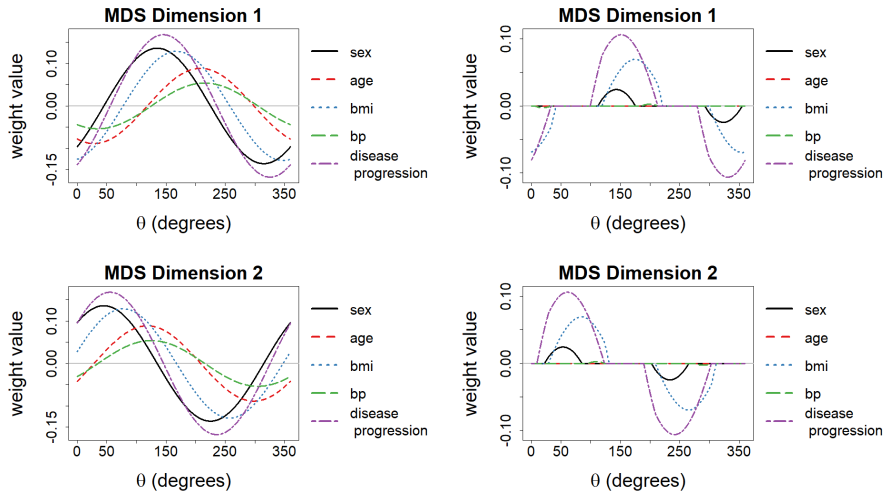
5.2. Effect of Rotation on Lasso Regularization

Another famous penalty is the Lasso penalty. The Lasso penalty regularizes regression weights using an L_1 norm:

$$\arg\min_{\mathbf{W}} \|\mathbf{XR} - \mathbf{FW}\|_F^2 + \lambda \sum_{j=1}^{d} \|\mathbf{w}_j\|_1. \tag{16}$$

The Lasso induces a thresholding effect based on λ, setting all weights under this λ-dependent threshold to zero. This effect can be observed in Figure 3b, where, for certain rotation angles, several features have zero-valued weights. As the Lasso simultaneously sets many weights to zero, the regression model is generally less complex, and thus more interpretable, than OLS and Ridge models.

However, in contrast to OLS and Ridge, the Lasso objective function is not invariant to rotation. Indeed, as shown in Figure 4, the error and the number of weights set to zero change as the MDS embedding is rotated. This means that failing to rotate the MDS embedding before applying the Lasso may yield a model that is suboptimal in terms of model error and sparsity. If one wants to use the Lasso to interpret an MDS embedding, the embedding orientation should be carefully selected.



(a) Example of regression weights for Ridge models estimated when a 2D MDS embedding is rotated with different angles θ ($\lambda = 0.15$).

(b) Example of regression weights for Lasso models estimated when a 2D MDS embedding is rotated with different angles θ ($\lambda = 0.15$).

Figure 3: Effect of rotation on weight values for Ridge and the Lasso.

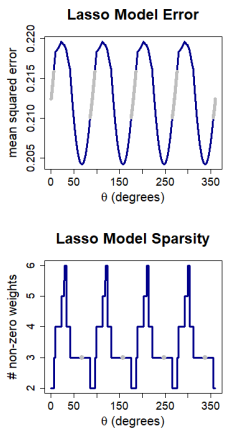


Figure 4: Error and sparsity of Lasso models estimated when a 2D MDS embedding is rotated with different angles θ ($\lambda = 0.15$). Line segments highlighted in gray indicate θ values minimizing sparsity (top plot) or minimizing model error (bottom plot). The different minima for model sparsity and error do not overlap. Model weights are represented in Figure 3b.

.

5.3. Selecting the Best Rotation for Interpretation

Among all possible MDS embedding orientations, we are interested in selecting the one yielding a Lasso regression model with the best balance between error and interpretability. In what follows, we measure model error using the mean squared error (MSE), and we quantify interpretability by counting the number of non-zero-valued weights (or active features) in the model (L_0 norm). This leads to the Best Interpretable Rotation (BIR) selection criterion, which selects the best angle θ^* as

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \frac{1}{2n} \|\mathbf{X}\mathbf{R}^{\theta} - \mathbf{F}\mathbf{W}^{\theta}\|_F^2 + \lambda \sum_{k=1}^{2} \|\mathbf{w}_k^{\theta}\|_0 \\ &= \arg\min_{\theta} \sum_{k=1}^{2} \left(\frac{1}{2n} \|\mathbf{X}\mathbf{r}_k^{\theta} - \mathbf{F}\mathbf{w}_k^{\theta}\|_2^2 + \lambda \|\mathbf{w}_k^{\theta}\|_0 \right), \end{aligned} \tag{17}$$

where $\mathbf{R}^{\theta}$ is a rotation matrix dependent on θ, $\mathbf{w}_k^{\theta}$ is the weight vector obtained when the Lasso is applied to the k^{th} column of an embedding rotated by an angle θ and λ strikes a balance between the MSE and the L_0 norm. The solution θ^* is then used to calculate a rotation matrix $\mathbf{R}$ based on Equation (6).

5.4. Lasso Regression based on a BIR-Selected Angle

Similar to some of the methods summarized in Section 4, the BIR selection method finds an orthogonal transformation matrix $\mathbf{R}$ for a view $\mathbf{X}$, given another view $\mathbf{F}$. However, in contrast to methods like eigenvector PLS-R and SRRR, the model used for interpreting $\mathbf{XR}$ based on the external feature view $\mathbf{F}$ is not learned.

The purpose of BIR Lasso regression (BIR-LR) is to learn a sparse linear model linking these two views by applying Lasso regression to a target matrix $\mathbf{X}$ rotated by the angle θ^* found using BIR. Note that the optimization of $\mathbf{R}^\theta$ using BIR involves the L_0 norm in Equation (17), while the Lasso involves the L_1 norm when optimizing $\mathbf{W}$ (see Equation (16)).

BIR-LR is similar to SRRR (see Section 4) in that the optimization of the regularized weight matrix $\mathbf{W}$ depends on the transformation matrix $\mathbf{R}$. For SRRR, $\mathbf{W}$ is regularized using an L_2 penalty,

$$\gamma \sum_{j=1}^{d} \|\mathbf{w}_j\|_2, \tag{18}$$

whereas the $\mathbf{W}$ optimized in BIR-LR is regularized using an L_1 (Lasso) penalty,

$$\lambda \sum_{j=1}^{d} \|\mathbf{w}_j\|_1. \tag{19}$$

The disadvantage of using the L_2 penalty is that each given feature has non-zero-valued weights for all dimensions of the MDS or none of them. Using the L_1 penalty makes it possible to learn models where a given feature has a non-zero-valued weight for one dimension and a zero-valued weight for another, which greatly simplifies interpretation.

6. Evaluation of the BIR Selection Method

This section presents our evaluation procedure and results. The problem at hand involves two tasks: (i) finding an optimal orthogonal transformation matrix $\mathbf{R}$ for interpreting an MDS embedding and (ii) learning an interpretable model $\mathbf{W}$ that accurately relates external features to the orthogonally transformed embedding. Two experiments are run to compare the performance of different methods with respect to these two tasks.

The purpose of the first experiment is to evaluate whether the orthogonal transformation $\mathbf{R}$ found using the BIR selection method yields a better Lasso solution than the orthogonal transformations found using other methods (task 1). PCA, eigenvector PLS and SRRR are used to generate the competitor transformation matrices (see Section 4). The purpose of the second experiment is to compare BIR-LR with two existing methods that combine view transformation with regression: SRRR and eigenvector PLS-R (tasks 1 and 2).

We compare the performance of each method with respect to two baselines: (i) the performance of the *least sparse rotation*, calculated as the average performance of the Lasso for the set of rotation angles yielding the least sparse solution and (ii) the expected performance of a *random rotation*, calculated as the average performance of the Lasso for all rotation angles $\theta \in \Theta = \{0.1, 0.2, ..., 360\}$ degrees.

6.1. Datasets and Pre-Processing

The performance of BIR and the other methods is evaluated using seven popular, publicly available datasets that can easily be split into two meaningful, distinct views: Hepatitis, Dermatology, Heart (Statlog) from [29], Insurance Company Benchmark from [30, 29], Community and Crimes from [31, 32, 33, 29], Pima Indians Diabetes from [34] and Diabetes from [35]. As an example, the features in Diabetes are divided into a view containing blood serum measurements – such as glucose and cholesterol levels – and another view composed of simple patient traits – such as age, sex and disease progression (see Table 1 for all split details). For each dataset, instances with missing values are removed, and non-ordinal categorical features are binarized using one-hot encoding. The total number of instances in each dataset, as well as the number of features in each view, is summarized in Table 2.

For each dataset, a view containing interpretable features is used as the external feature set $\mathbf{F}$. A dissimilarity matrix $\mathbf{D}$ of pairwise Euclidean distances between instances is constructed based on the other view $\mathbf{Q}$, which has been normalized. A 2D metric MDS embedding $\mathbf{X}$ is calculated using $\mathbf{D}$. All MDS embeddings $\mathbf{X}$ are centered and all external feature matrices $\mathbf{F}$ are normalized.

6.2. Evaluation Procedure

For both experiments, 10-fold cross-validation is used to assess the average performance of the different methods for each of the seven datasets. The MDS embeddings $\mathbf{X}$ are trained using all instances in $\mathbf{Q}$, then the instances in $\mathbf{X}$ and $\mathbf{F}$ are split into 10 folds. For each instance in $\mathbf{X}$ assigned to a given fold, the corresponding instance in $\mathbf{F}$ is assigned to the same fold.

Dataset	Features in Q	External Features in F
Hepatitis	**Histopathological features:** bilirubin, alk.phosphate, sgot, Albumin, protime	**Patient clinical information:** hist, age, sex, steroid, antivirals, fatigue, malaise, anorexia, big.liver, firm.liver, spleen.palp, spiders, ascites, varices, class
Dermatology	**Features measured through microscope analysis:** melanin, eosinophils, PNL, fibrosis, exocytosis, acanthosis, hyperkeratosis, parakeratosis, clubbing, elongation, thinning, spongiform, munro.microabcess, hypergranulosis, dis.granular, vacuolisation, spongiosis, saw.tooth, follic.horn.plug, perifolli.parakeratosis, inflam.monoluclear, band.like	**Patient clinical information:** erythema, scaling, def.borders, itching, koebner, polyg.papules, follic.papules, oral.musocal, knee.elbow, scalp, family.hist, age, disease
Heart	**Features measured at a consultation:** rest.BP, cholest, fast.sugar, rest.ECG, max.HR, ex.angina, ST.depress, ST.slope, blood.vessels, thal	**Patient clinical information:** age, sex, pain.type, disease
Diabetes	**Blood serum measurements:** s1, s2, s3, s4, s5, s6 (hdl, ldl, glucose, etc.)	**Patient clinical information:** age, sex, body mass index, blood pressure, disease.prog
Pima	**Features measured at a consultation:** glucose, pressure, triceps, insulin	**Patient clinical information:** pregnant, mass, pedigree, age, diabetes
Crimes	**Criminality features:** e.g. murders, robberies, autoTheft, arsons, etc.	**Socio-demographic features:** e.g. household-size, racePctWhite, medIncome, RentMedian, etc.
Insurance	**Insurance product usage features:** e.g. PPERSAUT (contribution car policies), ALEVEN (number of life insurances), etc.	**Socio-demographic features:** e.g. MHKOOP (home owners), MRELGE (married), MINKGEM (average income), etc.

Table 1: Division of dataset features into two views: **Q**, which contains the features used for computing the MDS, and **F**, the set of external features used to interpret the MDS. For datasets with more than 50 features (Crimes and Insurance), only a few feature examples are provided.

Dataset	Instances	Features		
		Total	Q	F
Hepatitis	80	20	5	15
Dermatology	358	35	22	13
Heart	270	14	10	4
Diabetes	442	11	6	5
Pima	768	9	4	5
Crimes	302	142	18	124
Insurance	5822	134	43	91

Table 2: Dimensions of evaluation datasets.

6.3. Experiment 1: Orthogonal Transformations

In this experiment, the quality of different orthogonal transformations is studied by evaluating the sparsity and test error of Lasso models where **F** is the feature matrix and transformed embedding **XR** is the target. As the Lasso is used for all evaluated methods, only the quality of the embedding transformation (with respect to the learned Lasso model) is measured.

BIR, PCA, eigenvector PLS and SRRR, as well as the baseline rotations (least sparse and random rotations), are applied to each training fold to produce orthogonal transformation matrices **R**. Then, Lasso models with varying values of λ are trained on the same folds. For SRRR, 25 γ values in the interval $[1, 3000]$, equally spaced in logarithmic scale, were tested. In the evaluated datasets, two distinct trends were observed among the SRRR transformation matrices trained with these γ values. In what follows, the γ values 1 and 208 have been selected because they yield models representative of the two observed trends for all of the datasets. Thirty equally spaced λ values in the interval $[0.01, 0.45]$ are used for the Lasso models. This range was chosen in order to cover a large range of sparsity degrees (as calculated using Equation (20)).

Several evaluation criteria are calculated based on the Lasso model weights **W**: the degree of sparsity

$$s = \sum_{k=1}^{2} \|\mathbf{w}_k\|_0, \tag{20}$$

and the mean squared error (MSE) of prediction on the

test fold, calculated as

$$\text{MSE} = \frac{1}{2n}\sum_{k=1}^{2}\|\mathbf{X}\mathbf{r}_k - \mathbf{F}\mathbf{w}_k\|_2^2, \tag{21}$$

where $\mathbf{X}$ and $\mathbf{F}$ contain only instances in the test fold.

Figure 5 shows the results of the first experiment. For each method, the average MSE over 10 folds is plotted against the average number of non-zero-valued weights (over the same folds). For all datasets except Hepatitis, and for a given number of non-zero-valued weights, BIR angles result in model error that is less than or equal to all other methods. In contrast to BIR, the least sparse rotation always has the worst test error for these datasets, probably because of overfitting during training. Hepatitis is the only exception, where, for non-sparse models, the average MSE of the least sparse case is the smallest and the average MSE of BIR is the largest. However, we argue that the most interesting models for ease of interpretation are the ones with few non-zero-valued weights, in which case BIR yields smaller model error than the other methods. Overall, BIR outperforms all other methods and baseline rotations for sparse and interpretable regression models (left part of the plots).

Transforming the MDS embedding based on information in only one view, the MDS embedding itself, seems less optimal for subsequent regression. For Hepatitis and Dermatology, the MDS orientation selected by PCA is always worse than a random rotation on average, and for the other datasets, the results are sometimes better and sometimes worse than a random rotation (but always worse than BIR).

The same conclusion can be drawn for eigenvector PLS and SRRR. Indeed, despite using both matrices $\mathbf{X}$ and $\mathbf{F}$ to find an orthogonal transformation of $\mathbf{X}$, the performance of the regression of $\mathbf{XR}$ onto $\mathbf{F}$ fluctuates. As with PCA, the results for eigenvector PLS and SRRR are sometimes better than a random rotation and sometimes worse, while always being worse than BIR, except for the non-sparse solutions for Hepatitis. Note that eigenvector PLS is the main competitor of BIR for some datasets (e.g. Hepatitis, Dermatology and Insurance), while SRRR is its principal competitor for other datasets (e.g. Pima and Crimes). This suggests that the transformation quality of eigenvector PLS and SRRR depends heavily on the dataset, while BIR consistently provides good transformations for all datasets tested.

6.4. Experiment 2: Multi-View Regression Models

In this experiment, BIR-LR weights (Lasso weights computed on rotations selected by BIR) are compared to the weights estimated using methods that simultaneously transform the target and estimate weights with a regression method other than the Lasso. These weights are also compared to Lasso weights computed on the least sparse and random baseline rotations. The same set of λ values from the first experiment is used for BIR-LR and the baseline rotations. A sequence of 25 values in the interval [1, 3000], equally spaced in logarithmic scale, is used for the hyperparameter γ in SRRR.

Like in the first experiment, for each method, both the orthogonal transformation matrix $\mathbf{R}$ and the matrix of regression weights $\mathbf{W}$ are estimated using the training folds. However, the weights $\mathbf{W}$ for the methods from the literature are estimated using the specific regression approach of these methods, rather than by applying the Lasso. The degree of sparsity s is calculated for each $\mathbf{W}$, and the MSE of prediction is calculated for instances in the test fold (see Equations (20) and (21)).

Figure 6 shows the results of the second experiment. Note that eigenvector PLS-R has only one data point because it has no extra hyperparameters. Eigenvector PLS-R, which does not explicitly encourage model sparsity, appears to the far right in all plots. Despite its low average MSE, the obtained weights, which are all non-zero-valued, do not meet our need for interpretable solutions. For all datasets except Hepatitis and Insurance, SRRR has a greater average MSE than a random rotation or BIR-LR for all γ values tested. For the Hepatitis dataset, SRRR is only better than a random rotation for complex models with at least 18 active features. For the Insurance dataset, SRRR is comparable to a random rotation. For all datasets, BIR-LR has a lower average MSE than the other methods and baseline rotations for models with fewer than 10 non-zero-valued weights. Note that the weights for BIR-LR, as well as the least sparse and random rotations, were obtained using the Lasso, so they are the same as in the first experiment.

6.5. Analysis of Model Interpretability

In this section, we compare models from experiment 2 in order to assess the interpretability of BIR-LR. To simplify visualization of the models, we focus on the three datasets with the smallest number of external features: Diabetes, Heart and Pima. For BIR-LR, the hyperparameter λ selected for each dataset is the point (average MSE, average degree of sparsity) in the elbow of the corresponding plot in Figure 6 (i.e. the point closest to the origin). For the other methods, we select the hyperparameters yielding an average MSE closest to the chosen BIR-LR average MSE. All models are trained on all instances in $\mathbf{X}$ and $\mathbf{F}$.

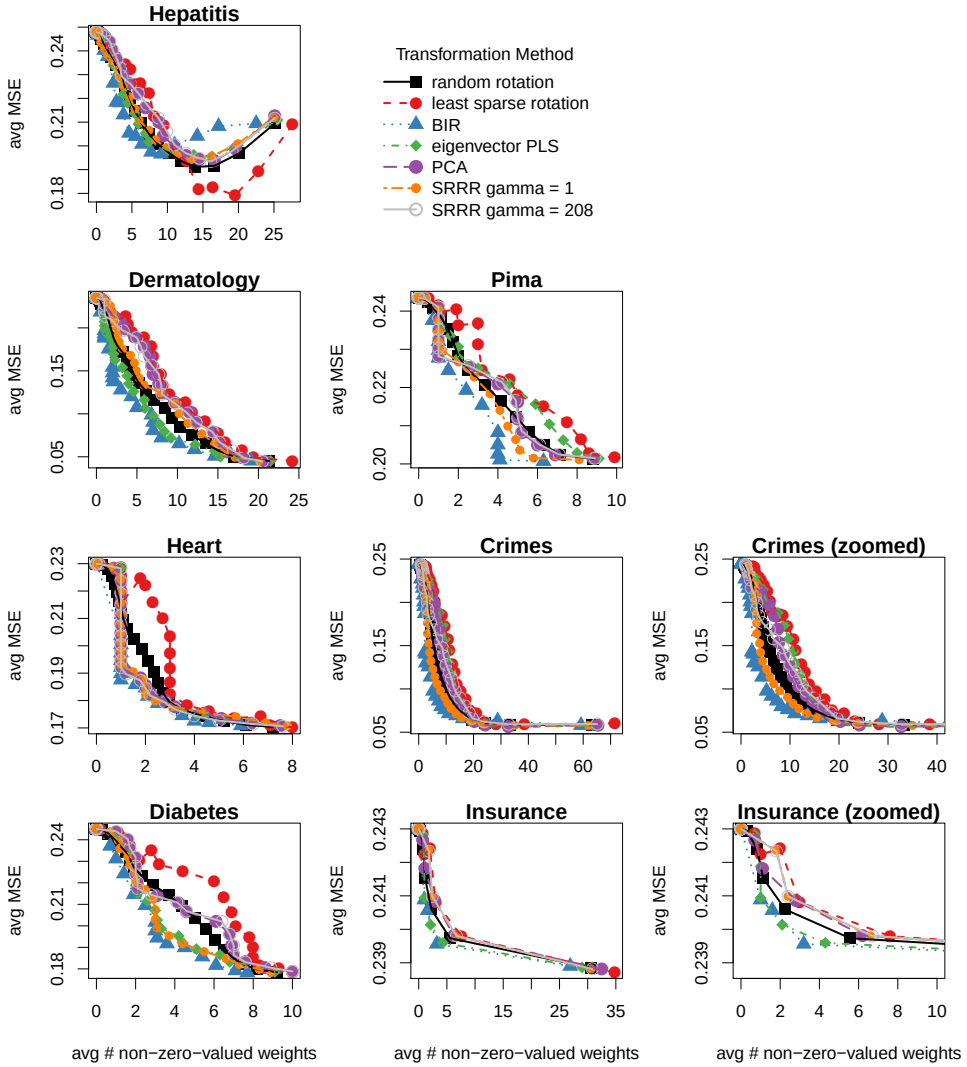


Figure 5: **Experiment 1**. Mean squared error (MSE) and degree of sparsity for Lasso models learned based on different embedding transformations. Each point represents an average value over 10 folds for a given λ, where λ is the hyperparameter used when training the Lasso models. Two SRRR curves are shown here, each representing different γ values. See the text for more details on the selection of γ. Crimes (zoomed) (resp. Insurance (zoomed)) is a zoomed version of the Crimes plot (resp. Insurance plot), showing the average number of non-zero-valued weights in the interval [0, 40] (resp. [0, 10]).

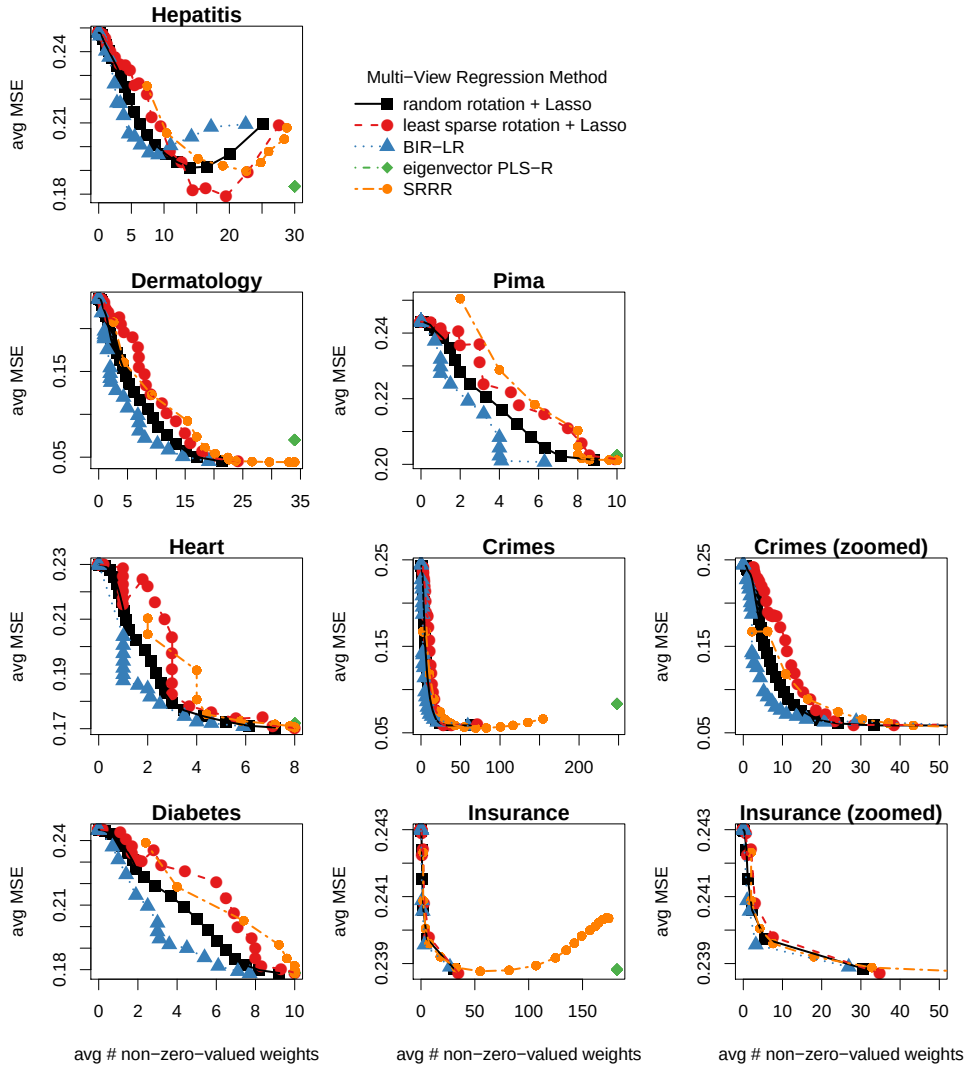


Figure 6: **Experiment 2**. Mean squared error (MSE) for different degrees of sparsity. Each point represents an average value over 10 folds for a particular hyperparameter setting, e.g. a value λ for the Lasso model. Note that eigenvector PLS-R does not have any hyperparameters. Crimes (zoomed) and Insurance (zoomed) are zoomed versions of the Crimes and Insurance plots, showing the average number of non-zero-valued weights in the interval [0, 50].

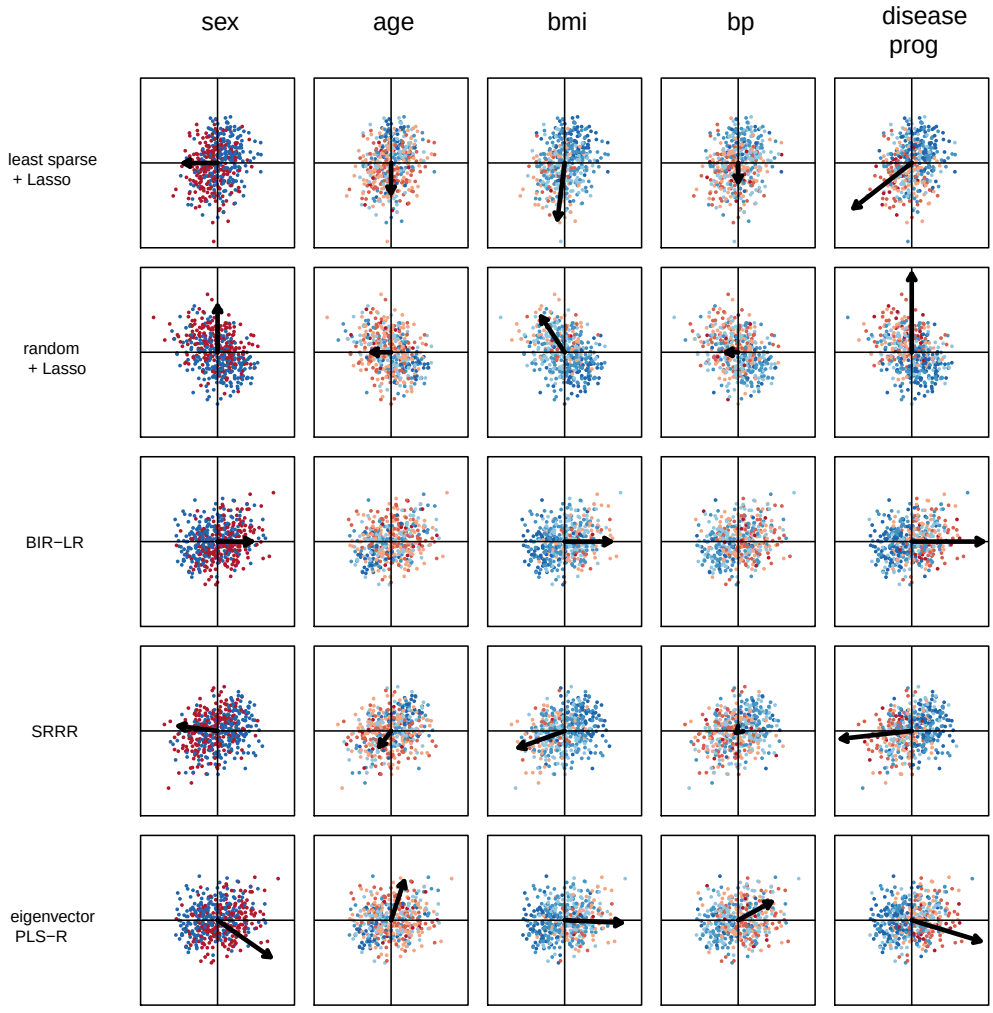


Figure 7: **Experiment 2 (Diabetes)**: Each row represents a specific multi-view regression method and each column corresponds to a feature in the Diabetes dataset. Each scatterplot depicts an MDS embedding transformed by the method in the corresponding row. Each instance in the scatterplots is colored according to its value for the feature in question, using a scale from blue (minimum) to red (maximum). Finally, each arrow direction represents the regression weights $\mathbf{w}_j$ for the corresponding column feature. The arrow length is proportional to the L_2-norm of $\mathbf{w}_j$. Note that the "least sparse" (resp. "random") row presents a single example of a rotation yielding the least (resp. average) model sparsity. Eigenvector PLS-R does not have any hyperparameters, so the error level for this method does not necessarily match the others. It is included here to be consistent with previous figures.

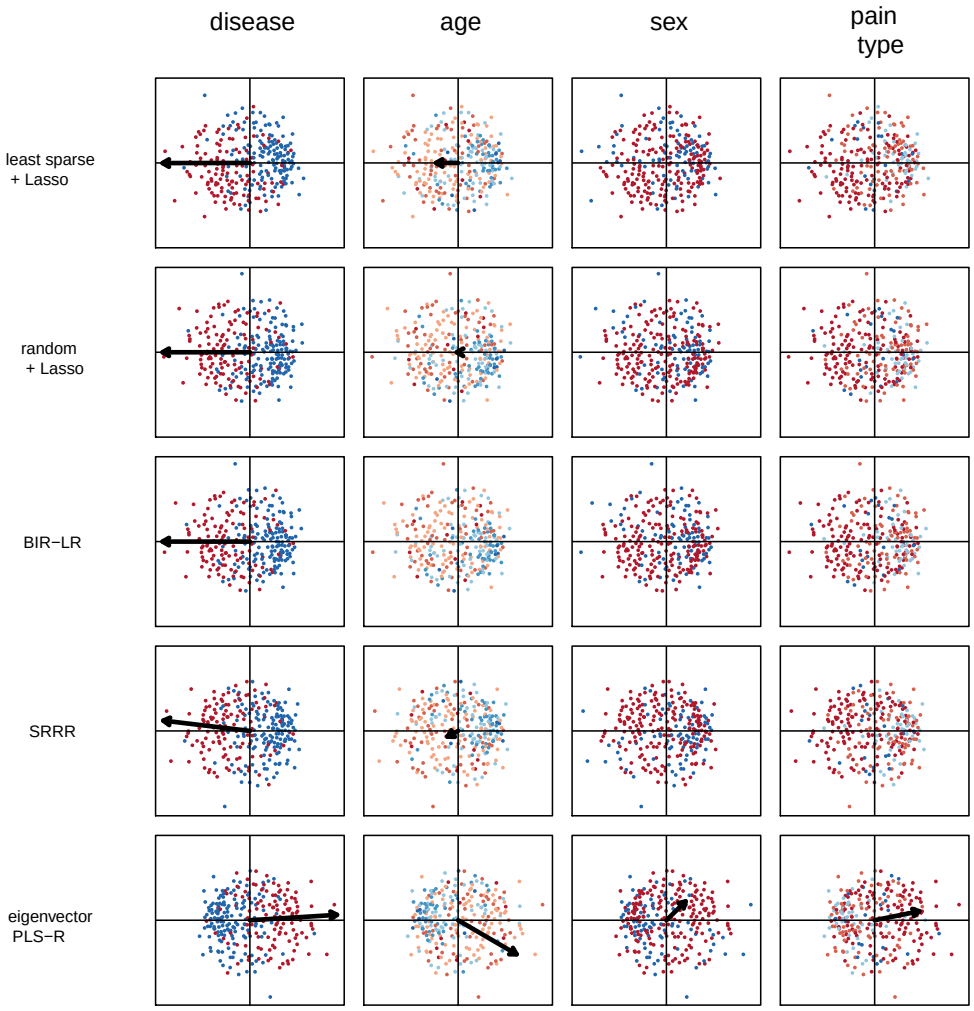


Figure 8: **Experiment 2 (Heart)**: Each row represents a specific multi-view regression method and each column corresponds to a feature in the Heart dataset. Each scatterplot depicts an MDS embedding transformed by the method in the corresponding row. Each instance in the scatterplots is colored according to its value for the feature in question, using a scale from blue (minimum) to red (maximum). Finally, each arrow direction represents the regression weights $\mathbf{w}_j$ for the corresponding column feature. The arrow length is proportional to the L_2-norm of $\mathbf{w}_j$. Note that the "least sparse" (resp. "random") row presents a single example of a rotation yielding the least (resp. average) model sparsity. Eigenvector PLS-R does not have any hyperparameters, so the error level for this method does not necessarily match the others. It is included here to be consistent with previous figures.

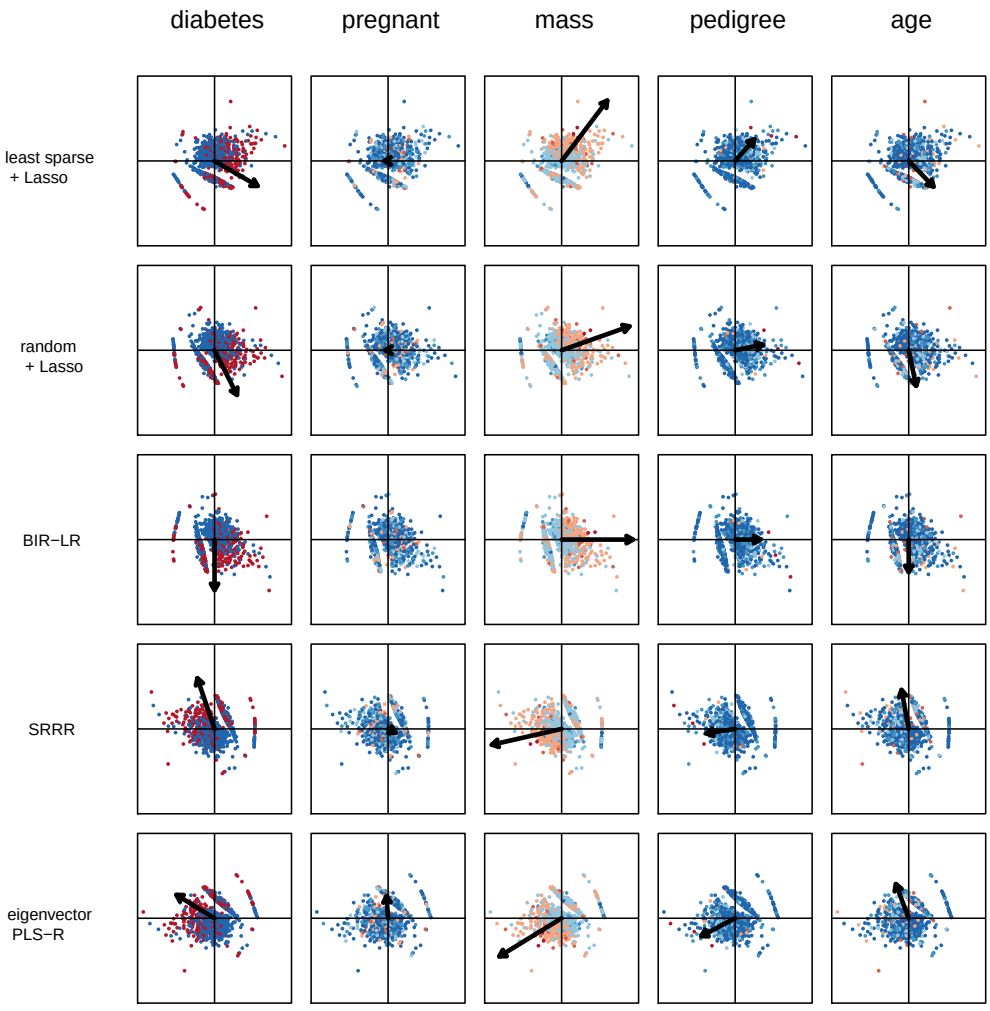


Figure 9: **Experiment 2 (Pima)**: Each row represents a specific multi-view regression method and each column corresponds to a feature in the Pima dataset. Each scatterplot depicts an MDS embedding transformed by the method in the corresponding row. Each instance in the scatterplots is colored according to its value for the feature in question, using a scale from blue (minimum) to red (maximum). Finally, each arrow direction represents the regression weights $\mathbf{w}_j$ for the corresponding column feature. The arrow length is proportional to the L_2-norm of $\mathbf{w}_j$. Note that the "least sparse" (resp. "random") row presents a single example of a rotation yielding the least (resp. average) model sparsity. Eigenvector PLS-R does not have any hyperparameters, so the error level for this method does not necessarily match the others. It is included here to be consistent with previous figures.

Figures 7, 8 and 9 present the transformations and weights learned by different multi-view regression methods applied to Diabetes, Heart and Pima, respectively. Each row contains scatterplots of an MDS embedding transformed by a given method, and each column contains a different coloration of the instances for each feature. Each instance is colored based on the value of the feature for that instance, where dark blue is the minimum value and dark red is the maximum value. For instance, the first scatterplot in the second column of Figure 7 is an MDS embedding rotated by an angle yielding the least sparse solution, and it is colored according to the age of each patient in the scatterplot.

The arrows in the figures represent the weight vectors $\mathbf{w}_j$ for the different features j. Their length is proportional to $\|\mathbf{w}_j\|_2$. Arrows that are vertical or horizontal indicate that the feature in question is used to explain only the vertical or horizontal dimension of the MDS. For example, the first scatterplot in the second column of Figure 7 has a vertical arrow, meaning that age is not used to explain the horizontal dimension. For the third scatterplot in the second column of Figure 7, which corresponds to the age weights in the BIR-LR model, there is no arrow, meaning that the weights for age are equal to zero for both of the rotated MDS dimensions.

These figures allow us to show that, for the same level of error as the other methods, BIR-LR models often have more zero-valued weights (vertical or horizontal arrows, or no arrows at all). This means that the two dimensions can often be interpreted using small, disjoint sets of features. In these figures, we observe that BIR-LR finds rotations resulting in models that only include the features displaying a strong relationship with one of the rotated dimensions. For instance, in Figure 7, clear visual color trends are observed for the features sex, bmi and disease progression, which are the only features selected by BIR-LR. Because the weights for these features take the value zero for the vertical dimension, the resulting model is much sparser than for the other methods. Features like age, for which blue and red instances are mixed in all directions, are not selected by BIR-LR. Thus, for this dataset (Diabetes), the BIR-LR model suggests that a horizontal trend can be captured by three distinct features but that no feature in $\mathbf{F}$ can explain the vertical axis in the MDS embedding. This seems to be confirmed by the observation that no top-down color change is apparent for this orientation.

Similar observations can be made for Figures 8 and 9. Based on these three figures, we demonstrate that BIR-LR can provide models facilitating the interpretation of MDS embeddings, thanks to rotations resulting in sparse models.

7. Discussion on the Performance of BIR

As discussed in Section 5 and demonstrated by the experiments in Section 6, choosing an angle for interpreting an MDS embedding with sparse regression is important. Bibal, Marion and Frénay [1] have shown that choosing an angle at random leads to a worse solution on average than BIR-LR in terms of both model sparsity and error.

In this paper, we have shown that selecting the orientation of an MDS embedding using single-view rotation methods such as PCA, or two-view orthogonal transformation methods in the case of eigenvector PLS and SRRR, does not necessarily lead to the most interpretable regression models. Indeed, for all datasets evaluated, except Hepatitis, BIR-LR proposes solutions that have lower or equal test error for all degrees of sparsity. For the Hepatitis dataset, we observe that these conclusions may only hold for sparser solutions. However, for the purpose of interpretation, these may be precisely the solutions that are most desirable.

The degree of sparsity in BIR-LR weights is controlled by the hyperparameter λ, which must be selected by the user according to his needs. One possible heuristic for choosing this hyperparameter could be the elbow method, but this should only be used in cases where the plot of average MSE with respect to degree of sparsity has an elbow shape. The chosen λ would be the point with the smallest distance from the origin. The selection of an optimal λ is beyond the scope of this paper, but several potential strategies can be found in [36].

8. Conclusion and Future Works

This paper was concerned with the problem of interpreting a nonlinear dimensionality reduction (NLDR) model using a set of external features. In particular, we studied the use of linear regression to model multidimensional scaling (MDS) embedding dimensions as linear combinations of external features. This approach makes it possible to explain how the MDS model mapped instances into to the new, lower-dimensional space as a linear function of external features.

As MDS embeddings are only uniquely determined up to certain transformations, including rotation, we studied how the rotation of an MDS embedding affects subsequent linear regression models. While Lasso regression generally yields a model that is sparser and more interpretable than ordinary least squares (OLS) or Ridge regression, its model error and sparsity are both dependent on the rotation angle of the MDS embedding.

Thus, when using the Lasso to model the linear relationship between an MDS embedding and a set of external features, the rotation of the MDS embedding should not be chosen arbitrarily.

In this paper, we proposed the Best Interpretable Rotation (BIR) selection method for choosing an angle that rotates a 2D embedding such that a subsequent Lasso model strikes a balance between model error and sparsity. BIR Lasso regression (BIR-LR), which consists of Lasso regression where the target is rotated with the angle found using BIR, was also introduced. Using BIR-LR to interpret an MDS embedding model is an example of *post hoc interpretation* [37]. Indeed, sparse linear regression is used after the MDS embedding has been generated in order to interpret the way in which the instances were mapped into the embedding.

We compared BIR and BIR-LR to methods in the machine learning and statistics literature that also search for an orthogonal transformation, either based on information in the two data views available (two-view transformation) or information in only one of the two views (single-view transformation). For sparse models (i.e. models with fewer than 10 non-zero-valued weights in our experiments), BIR-LR had smaller test error than all methods tested, for all datasets.

The proposed BIR-LR method does not depend on visualization for interpretation, meaning that it would be possible to extend it to the case of more than two dimensions. In future work, the restriction to the case of two-dimensional embeddings could be lifted.

Finding an objective function integrating the optimization of θ and the weights $\mathbf{W}$ is also a subject of future work. For the moment, the best rotation is found on the basis of possible Lasso solutions. However, an optimal or near optimal rotation could be found while simultaneously learning a sparse regression model.

Finally, in this work and in the literature, constraints are used to encourage overall sparsity s; however, other definitions of sparsity could be developed that are more directly related to model interpretability. Furthermore, a more nuanced measure of interpretability that goes beyond sparsity is a subject of future research.

Acknowledgment

The authors would like to thank Nathan Nguyen from the Université catholique de Louvain for having pointed to the need for this kind of method in psychology, as well as Prof. Bernadette Govaerts and Prof. Rainer von Sachs from the Université catholique de Louvain for their insights on the subject. The first author gratefully acknowledges financial support from the Belgian Fund for Scientific Research (F.R.S.-FNRS, FRIA grant).

References

[1] A. Bibal, R. Marion, B. Frénay, Finding the most interpretable MDS rotation for sparse linear models based on external features, in: Proceedings of the European Symposium on Artificial Neural Networks, Bruges, Belgium, 2018, pp. 537–542.

[2] N. Jaworska, A. Chupetlovska-Anastasova, A review of multidimensional scaling (MDS) and its utility in various psychological domains, Tutorials in Quantitative Methods for Psychology 5 (1) (2009) 1–10.

[3] P. Legendre, L. Legendre, Numerical ecology: second English edition, Vol. 20 of Developments in Environmental Modeling, Elsevier Science, 1998.

[4] A. Bibal, B. Frénay, Interpretability of machine learning models and representations: an introduction, in: Proceedings of the European Symposium on Artificial Neural Networks, Bruges, Belgium, 2016, pp. 77–82.

[5] B. Gawronski, J. De Houwer, Implicit measures in social and personality psychology, Handbook of Research Methods in Social and Personality Psychology 2 (2014) 283–310.

[6] I. Van Mechelen, A. K. Smilde, A generic linked-mode decomposition model for data fusion, Chemometrics and Intelligent Laboratory Systems 104 (1) (2010) 83–94.

[7] S. Sun, A survey of multi-view machine learning, Neural Computing and Applications 23 (7-8) (2013) 2031–2038.

[8] J.-a. Lin, H. Zhu, R. Knickmeyer, M. Styner, J. Gilmore, J. G. Ibrahim, Projection regression models for multivariate imaging phenotype, Genetic Epidemiology 36 (6) (2012) 631–641.

[9] J. Thioulouse, Simultaneous analysis of a sequence of paired ecological tables: A comparison of several methods, The Annals of Applied Statistics (2011) 2300–2325.

[10] W. Lee, D. Lee, Y. Lee, Y. Pawitan, Sparse canonical covariance analysis for high-throughput data, Statistical Applications in Genetics and Molecular Biology 10 (1) (2011) 1–24.

[11] K. Varmuza, P. Filzmoser, Introduction to multivariate statistical analysis in chemometrics, CRC press, 2016.

[12] J. B. Kruskal, M. Wish, Multidimensional scaling, Sage, 1978.

[13] I. Borg, P. J. Groenen, Modern multidimensional scaling: Theory and applications, Springer, 2005.

[14] M. C. Hout, M. H. Papesh, S. D. Goldinger, Multidimensional scaling, Wiley Interdisciplinary Reviews: Cognitive Science 4 (1) (2013) 93–103.

[15] A. Lebel, M. Cantinotti, R. Pampalon, M. Thériault, L. A. Smith, A.-M. Hamelin, Concept mapping of diet and physical activity: uncovering local stakeholders perception in the Quebec City region, Social Science & Medicine 72 (3) (2011) 439–445.

[16] J. J. Chang, J. D. Carroll, How to use PROFIT, a computer program for property fitting by optimizing nonlinear or linear correlation, Unpublished Manuscript, Bell Laboratories (1968).

[17] A. Koch, R. Imhoff, R. Dotsch, C. Unkelbach, H. Alves, The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion, Journal of Personality and Social Psychology 110 (5) (2016) 675–709.

[18] S. Pattyn, Y. Rosseel, A. Van Hiel, Finding our way in the social world, Social Psychology 44 (2013) 329–348.

[19] P. I. Armstrong, S. X. Day, J. P. McVay, J. Rounds, Holland's RIASEC model as an integrative framework for individual differences, Journal of Counseling Psychology 55 (1) (2008) 1–18.

[20] G. M. Levine, J. B. Halberstadt, R. L. Goldstone, Reasoning and the weighting of attributes in attitude judgments, Journal of Personality and Social Psychology 70 (2) (1996) 230–240.

[21] D. Farrell, Exit, voice, loyalty, and neglect as responses to job dissatisfaction: A multidimensional scaling study, Academy of Management Journal 26 (4) (1983) 596–607.
[22] H. F. Kaiser, The varimax criterion for analytic rotation in factor analysis, Psychometrika 23 (3) (1958) 187–200.
[23] H. H. Harman, Modern factor analysis, University of Chicago Press, 1976.
[24] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.
[25] E. R. Peay, Multidimensional rotation and scaling of configurations to optimal agreement, Psychometrika 53 (2) (1988) 199–208.
[26] H. Abdi, Partial least squares regression and projection on latent structure regression (PLS regression), Wiley Interdisciplinary Reviews: Computational Statistics 2 (1) (2010) 97–106.
[27] L. Chen, J. Z. Huang, Sparse reduced-rank regression for simultaneous dimension reduction and variable selection, Journal of the American Statistical Association 107 (500) (2012) 1533–1545.
[28] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.
[29] M. Lichman, UCI machine learning repository (2013). URL `http://archive.ics.uci.edu/ml`
[30] P. van der Putten, M. van Someren, Coil challenge 2000: The insurance company case, Tech. rep., Leiden Institute of Advanced Computer Science (2000).
[31] D. o. C. Bureau of the Census, Census Of Population And Housing 1990 United States: Summary Tape File 1a & 3a, U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan (1992).
[32] D. o. J. Bureau of Justice Statistics, Law Enforcement Management and Administrative Statistics, U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan (1992).
[33] D. o. J. Federal Bureau of Investigation, Crime in the United States (1995).
[34] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, R. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Annual Symposium on Computer Application in Medical Care, Washington, D.C., USA, 1988, pp. 261–265.
[35] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, The Annals of Statistics 32 (2) (2004) 407–499.
[36] D. Homrighausen, D. J. McDonald, A study on tuning parameter selection for the high-dimensional lasso, Journal of Statistical Computation and Simulation (2018) 1–28.
[37] Z. C. Lipton, The mythos of model interpretability, in: ICML Workshop on Human Interpretability of Machine Learning, New York, USA, 2016.

Rebecca Marion is a Ph.D. student at the Université catholique de Louvain (UCLouvain) in Belgium, under the supervision of Professors Rainer von Sachs and Bernadette Govaerts. She received an M.S. in Statistics, concentration in Biostatistics, from UCLouvain in 2016. Her Ph.D. thesis is on multi-view learning and feature selection for data with grouped features.

Adrien Bibal is a Ph.D. student at the Université de Namur (Belgium) under the supervision of Professor Benoît Frénay. He received an M.S. degree in Computer Science and an M.A. degree in Philosophy from the Université catholique de Louvain (Belgium) in 2013 and 2015 respectively. His Ph.D. thesis in machine learning is on the interpretability of dimensionality reduction models.

Benoît Frénay is associate professor at the Université de Namur. He received his M.S. and Ph.D. degrees from the Université catholique de Louvain (Belgium) in 2007 and 2013, respectively. His main research interests in machine learning include interpretability, interactive machine learning, dimensionality reduction, label noise, robust inference and feature selection. In 2014, he received the Scientific Prize IBM Belgium for Informatics for his PhD thesis on Uncertainty and Label Noise in Machine Learning.

CHAPTER 23

BIOT: EXPLAINING MULTIDIMENSIONAL MDS EMBEDDINGS USING THE BEST INTERPRETABLE ORTHOGONAL TRANSFORMATION

The paper presented in this chapter is currently under review for the journal Neurocomputing.

BIOT: Explaining Multidimensional MDS Embeddings using the Best Interpretable Orthogonal Transformation

Adrien Bibal[b,*], Rebecca Marion[a,*], Rainer von Sachs[a], Benoît Frénay[b]

[a] *ISBA, LIDAM, Université catholique de Louvain, Voie du Roman Pays 20, B-1348 Louvain-la-Neuve, Belgium*
[b] *PReCISE, NADI, Faculty of Computer Science, University of Namur, Rue Grandgagnage 21, B-5000 Namur, Belgium*

Abstract

Dimensionality reduction (DR) is a popular approach to data exploration in which instances in a given dataset are mapped to a lower-dimensional representation or "embedding." For nonlinear dimensionality reduction (NLDR), the dimensions of the embedding may be difficult to understand. In such cases, it may be useful to learn how the different dimensions relate to a set of external features (i.e., relevant features that were not used for the DR). A variety of methods (e.g. PROFIT and BIR) use external features to explain embeddings generated by NLDR methods with rotation-invariant objective functions, such as multidimensional scaling (MDS). However, these methods are restricted to two-dimensional embeddings. In this paper, we propose BIOT, which makes it possible to explain an MDS embedding with any number of dimensions without requiring visualization.

Keywords: Multidimensional Scaling, Explainability, Lasso, Orthogonal Transformations

1. Introduction

Interpretability and explainability are hot topics in machine learning. Interpretability refers to the intrinsic capacity of a model to be understandable for a user [1, 2], and the problem of explainability arises for non-interpretable (i.e. black-box) models [3]. Indeed, when machine learning models are black boxes, techniques that are external to the model must be used to provide explanations.

While most of the machine learning literature on interpretability and explainability is framed for a supervised learning context, the need for such concepts also exists in unsupervised learning. For instance, in clustering (or cluster analysis), users may want to understand the meaning behind the clusters found. Similarly, users that perform dimensionality reduction (DR) on their data may be interested in understanding the meaning of the reduced dimensions.

DR is often used when the high-dimensionality of the original dataset makes it difficult to perform data exploration and/or makes data analysis victim to the curse of dimensionality [4, 5], among other problems. However, some of the most effective DR techniques (i.e. UMAP [**?**], *t*-SNE [6], MDS [7], etc.) are nonlinear, which makes the embeddings they generate difficult to interpret. One solution to this problem is to use a set of additional features to explain the dimensions of the low-dimensional embedding.

For example, in psychology, nonlinear dimensionality reduction is commonly applied to datasets containing pairwise comparisons between objects (e.g. the perceived (dis)similarity between pairs of social groups [8]). Additional interpretable features are then used to determine the meaning of the embedding dimensions [8]. In sensometrics, it is also common to study the relationship between embedding dimensions and an external feature set. For instance, some studies seek to identify sensory attributes in one dataset (e.g. flavor, smell of products) that could be used to explain embeddings of consumer preferences in a second dataset (e.g. product appreciation scores) [9].

The aforementioned examples depend on the assumption that the embedding dimensions are them-

*Corresponding authors. Both authors contributed equally.
Email addresses: adrien.bibal@unamur.be (Adrien Bibal), rebecca.marion@uclouvain.be (Rebecca Marion), rainer.vonsachs@uclouvain.be (Rainer von Sachs), benoit.frenay@unamur.be (Benoît Frénay)

selves meaningful. This is not necessarily the case for neighborhood-preserving methods such as t-SNE or UMAP, which do not preserve small distances in the same way as large distances, generating embedding dimensions that can be spatially misleading. However, methods that seek to preserve all pairwise distances between instances are good candidates for this explanation approach.

Multidimensional scaling (MDS) [7] is a very popular nonlinear dimensionality reduction (NLDR) method [10] in this category, especially in fields like psychology and ecology, and it is well-developed in the literature. Explanation techniques, such as property fitting (PROFIT), exist to explain MDS embeddings by regressing external features onto the embedding dimensions [11]. PROFIT has several shortcomings [12], but these limitations can be overcome by regressing the embedding dimensions onto the external features using sparse regression techniques such as the Lasso. However, for NLDR methods with objective functions invariant to rotation, such as MDS, this approach requires the optimization of the embedding orientation. Indeed, all rotations of an MDS embedding are equivalent for MDS, but can result in very different regression models in terms of sparsity, interpretability and error.

Best interpretable rotation (BIR) is a state-of-the-art method for solving this problem [13, 12], but it (i) involves exhaustively exploring all possible rotation angles and (ii) is restricted to explanations of two-dimensional (2D) embeddings. In this paper, we propose best interpretable orthogonal transformation (BIOT), a new method that tackles these two issues. We show that (i) the performance of BIOT is better than BIR and other state-of-the-art techniques and that (ii) BIOT makes it possible to easily explain embeddings with more than two dimensions. This last feature of BIOT lifts the requirement of having two dimensions to explore the data, which makes, e.g., 5D and 6D embeddings now useful. Thanks to this, embeddings that have a lower DR loss, and are thus more faithful to the original high-dimensional data, can be studied.

This paper is structured as follows. Section 2 motivates the need for explaining NLDR embeddings and highlights the potential explainability of MDS. Section 3.1 introduces the notations used in this paper. The problem tackled in this paper is formally stated in Section 3.2. BIOT, the method proposed to solve this problem, even for embeddings with more than two dimensions, is introduced in Section 3.3. Section 4 presents how regressing embedding dimensions onto external features can be performed using state-of-the-art techniques. A numerical evaluation of the proposed method and state-of-the-art methods is presented in Section 5. In order to clearly highlight the usefulness of BIOT, a case study demonstrates the application of BIOT to explain MDS embeddings in Section 6. Finally, Section 7 concludes the paper.

2. Motivation

The nonlinear dimensionality reduction (NLDR) methods used today produce embeddings that are not always understandable. To compensate for this lack of understandability, or interpretability, NLDR embeddings are often restricted to two or three dimensions so that the data can be explored and analyzed visually. Furthermore, some methods are not even designed to produce higher-dimensional embeddings. For example, Barnes-Hut, the widely used approximation for accelerating the optimization of t-distributed stochastic neighbor embedding (t-SNE) [6], is technically restricted to produce embeddings with three or fewer dimensions (because it uses quadtree for two-dimensional embeddings and octtree for three-dimensional embeddings) [**?**].

One problem with using visualization to analyze NLDR embeddings is that it inherently limits the amount of information from the original dataset that can be represented in the embedding. Moreover, the relative positions of instances in the visualization are not always easy to explain (e.g. why some instances are close together or far apart). This is especially true for neighborhood-preserving NLDR methods (such as t-SNE [6] and uniform manifold approximation and projection (UMAP) [**?**]). These techniques can provide interesting visual results, but are not completely faithful to the original space, as large distances in the original space are less well preserved than small distances [**?**]. As a result, the axes of the visualization (i.e. the embedding dimensions) have no particular meaning.

In contrast, methods that attempt to preserve all pairwise distances (e.g. multidimensional scaling (MDS) [7]) are able to generate more spatially meaningful embedding dimensions. As a result, the embedding dimensions can be used as features for characterizing the instances. Moreover, if the meaning of these dimensions is identified, the data can be explored without necessarily resorting to visualization: similarities and dissimilarities between instances can be explained by the embedding dimensions that characterize them.

In this paper, we are interested in the problem of exploring high dimensional datasets using NLDR embeddings with more than two or three dimensions. In particular, we focus on embeddings generated by MDS,

a popular distance-preserving method in the literature. The next section describes this problem in detail.

3. Proposed Method

3.1. Notations

Matrices are indicated with bold, upper-case letters (e.g. $\mathbf{X}$) and vectors are indicated using bold, lower-case letters with dot notation, where $\mathbf{x}_{\bullet,j}$ is the j-th column vector in $\mathbf{X}$ and $\mathbf{x}_{i,\bullet}$ is the i-th row vector. Scalar elements from a matrix or vector are indicated using lower-case letters (e.g. x_{ij}). Instances are indexed with the letter $i \in \{1, ..., n\}$, external features with the letter $j \in \{1, ..., d\}$ and embedding dimensions with the letter $k \in \{1, ..., m\}$.

3.2. Problem Definition and Background

Multidimensional scaling (MDS) [7] is a nonlinear dimensionality reduction (NLDR) [10] technique that is widely used in academia (e.g. in psychology), as well as in industry. Given an $n \times n$ (dis)similarity matrix, where n is the number of instances, MDS produces an $n \times m$ embedding $\mathbf{X}$ for a chosen number of dimensions m.

In its most classical form, the objective of MDS is to minimize the stress, a measure of reconstruction error. This means maximizing the match between the dissimilarities of instances in the high-dimensional (HD) space and the pairwise distances in the low-dimensional (LD) space. For instance, Kruskal's stress is defined as

$$\text{Stress} = \sqrt{\frac{\sum_{ii'} (d_{ii'}^{\text{HD}} - d_{ii'}^{\text{LD}})^2}{\sum_{ii'} d_{ii'}^{\text{HD}^2}}}, \tag{1}$$

where $d_{ii'}^{\text{HD}}$ (resp. $d_{ii'}^{\text{LD}}$) is the dissimilarity (resp. distance) between the i-th and i'-th instances in HD (resp. LD).

The embedding $\mathbf{X}$ obtained when minimizing the stress is usually used to visually explore the data when $m = 2$. This latter case is called visualization through NLDR [10]. In either case, it is often important to understand the meaning of the MDS dimensions in order to draw conclusions about the data.

One approach for explaining MDS embeddings consists of using an $n \times d$ matrix $\mathbf{F}$ of external features (i.e. features that were not involved in the NLDR process). These external features also allow users to test whether they can explain the embedding with features that were not used to produce it. One popular technique for explaining MDS embeddings with external features is to regress each external feature $\mathbf{f}_{\bullet,j}$ in $\mathbf{F}$ onto the embedding $\mathbf{X}$:

$$\mathbf{f}_{\bullet,j} = \mathbf{X}\mathbf{w} + \mathbf{e}, \tag{2}$$

where $\mathbf{w}$ is a vector of regression weights and $\mathbf{e}$ is an error vector [7]. Property fitting (PROFIT) is based on this idea of fitting external features (called properties) to the induced embedding [11].

Two main issues arise from classical approaches like PROFIT [12]. First, rather than using combinations of external features to explain the embedding, external features are used one by one, thereby providing less insight about the dimensions. Second, the solution requires that the embedding $\mathbf{X}$ be visualized. Indeed, the goal of PROFIT is to show trends in an NLDR visualization. However, one may be interested in explaining an NLDR embedding with more than two dimensions.

One approach to solving the first issue is (i) to reverse the regression direction in order to explain each dimension of the embedding $\mathbf{X}$ on the basis of a linear combination of the external features $\mathbf{F}$, and (ii) to apply a sparsity penalty to the regression weights $\mathbf{W}$ so that each dimension of $\mathbf{X}$ is explained by as few features in $\mathbf{F}$ as possible [13, 12]:

$$\mathbf{X} = \mathbf{F}\mathbf{W} + \mathbf{E}, \tag{3}$$

where $\mathbf{W}$ is sparse. However, the authors in [13, 12] demonstrate that the arbitrary orientation of an MDS embedding is often not the best for balancing model error with sparsity. They show that it is necessary to simultaneously optimize both the sparse weight matrix $\mathbf{W}$ and a rotation matrix $\mathbf{R}$ that controls the orientation of the embedding. The model of interest becomes

$$\mathbf{X}\mathbf{R} = \mathbf{F}\mathbf{W} + \mathbf{E}, \tag{4}$$

where $\mathbf{W}$ is constrained to be sparse. In other words, one must find the rotation $\mathbf{R}$ leading to the sparse regression model that best explains the rotated MDS embedding $\mathbf{XR}$.

In principle, any transformation matrix $\mathbf{R}$ that preserves all meaningful structure from the original embedding could be used in this framework. Orthogonal transformations, which preserve all pairwise Euclidean distances between instances, are thus good candidates. In this paper, we are interested in the problem of finding the best orthogonal transformation of MDS embeddings of any number of dimensions such that they can be explained with sparse linear models based on external features. The next section introduces our proposed method, Best Interpretable Orthogonal Transformation (BIOT), for solving this problem.

3.3. BIOT, the Proposed Method

The overall objective of Best Interpretable Orthogonal Transformation (BIOT) is to explain the dimensions of an embedding $\mathbf{X}$ ($n \times m$) using a matrix of external features $\mathbf{F}$ ($n \times d$). BIOT does this by finding an orthogonal $m \times m$ matrix $\mathbf{R}$ such that the transformed embedding can be explained by a sparse weight matrix $\mathbf{W}$ ($d \times m$). Given a hyperparameter $\lambda > 0$, the optimization problem for BIOT is

$$\arg\min_{\mathbf{R},\mathbf{W}} \frac{1}{2n}\|\mathbf{XR} - \mathbf{FW}\|_F^2 + \lambda \sum_{k=1}^{m} \|\mathbf{w}_{\bullet,k}\|_1 \qquad (5)$$

s.t. $\mathbf{R}$ is an orthogonal matrix, i.e. $\mathbf{RR}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}_m$.

The orthogonality constraint for $\mathbf{R}$ ensures that the transformed embedding $\mathbf{XR}$ retains all meaningful structure from the original embedding: pairwise euclidean distances and the dimensionality of the embedding are preserved. The Lasso penalty on the columns of $\mathbf{W}$ (i.e. $\sum_{k=1}^{m} \|\mathbf{w}_{\bullet,k}\|_1$) encourages the selection of fewer features per embedding dimension. As a result, the transformed dimensions can be explained by potentially distinct sets of features. The best $\mathbf{R}$ for this problem is the orthogonal transformation that results in the model with the best balance between model error and sparsity, as controlled by the hyperparameter λ.

3.3.1. Optimizing $\mathbf{W}$ for Fixed $\mathbf{R}$

Given a fixed embedding orientation $\mathbf{R}$, the optimization of the weights $\mathbf{W}$ is a Lasso problem. For a particular embedding dimension k, the optimal weight vector is

$$\arg\min_{\mathbf{w}_{\bullet,k}} \frac{1}{2n}\|\mathbf{Xr}_{\bullet,k} - \mathbf{Fw}_{\bullet,k}\|_2^2 + \lambda\|\mathbf{w}_{\bullet,k}\|_1. \qquad (6)$$

Following cyclic coordinate descent optimization [14], all values of $\mathbf{w}_{\bullet,k}$ are fixed, except a certain value w_{jk} at each iteration. The problem to solve can therefore be rewritten as

$$\arg\min_{w_{jk}} \frac{1}{2n}\|\mathbf{e}_{-jk} - \mathbf{f}_{\bullet,j}w_{jk}\|_2^2 + \lambda\|\mathbf{w}_{-jk}\|_1 + \lambda|w_{jk}|, \qquad (7)$$

where $\mathbf{e}_{-jk} = \mathbf{Xr}_k - \mathbf{F}_{-j}\mathbf{w}_{-jk}$, $\mathbf{F}_{-j}$ is $\mathbf{F}$ without its j-th column $\mathbf{f}_{\bullet,j}$ and $\mathbf{w}_{-jk}$ is the weight vector $\mathbf{w}_{\bullet,k}$ without its j-th value w_{jk}. The optimal w_{jk} can be calculated using soft thresholding [14]:

$$w_{jk} = \frac{\text{sign}(\mathbf{f}_{\bullet,j}^\top\mathbf{e}_{-jk})(|\mathbf{f}_{\bullet,j}^\top\mathbf{e}_{-jk}| - n\lambda)_+}{\mathbf{f}_{\bullet,j}^\top\mathbf{f}_{\bullet,j}}. \qquad (8)$$

3.3.2. Optimizing $\mathbf{R}$ for Fixed $\mathbf{W}$

When $\mathbf{W}$ is found, the next step is to adjust the orientation of the embedding. Since $\mathbf{RR}^\top = \mathbf{I}_m$, for fixed $\mathbf{W}$, Eq. (5) can be rewritten as

$$\arg\min_{\mathbf{R}} \frac{1}{2n}\|\mathbf{X} - \mathbf{FWR}^\top\|_F^2 + \lambda \sum_{k=1}^{m} \|\mathbf{w}_{\bullet,k}\|_1 \qquad (9)$$
$$\text{s.t. } \mathbf{R} \text{ is an orthogonal matrix.}$$

Finding the optimal matrix $\mathbf{R}$ is an orthogonal Procrustes problem [15]. Indeed, for a fixed $\mathbf{W}$, Eq. (9) can be rewritten as

$$\arg\min_{\mathbf{Q}} \|\mathbf{A} - \mathbf{BQ}\|_F^2 \text{ s.t. } \mathbf{QQ}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_m, \qquad (10)$$

where $\mathbf{A} = \mathbf{X}/\sqrt{2n}$, $\mathbf{B} = \mathbf{FW}/\sqrt{2n}$ and $\mathbf{Q} = \mathbf{R}^\top$. The matrix $\mathbf{Q}$ that minimizes Eq. (10) can then be found by decomposing the matrix $\mathbf{C} = \mathbf{B}^\top\mathbf{A} = \frac{1}{2n}(\mathbf{FW})^\top\mathbf{X}$ using SVD, such that $\mathbf{C} = \mathbf{U\Sigma V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ contain the left- and right-singular vectors of $\mathbf{C}$ and $\mathbf{Q} = \mathbf{UV}^\top$ [16]. The transformation matrix $\mathbf{R}$ optimizing Eq. (9) is thus $\mathbf{R} = \mathbf{Q}^\top = \mathbf{VU}^\top$.

3.3.3. Optimization Algorithm

Algorithm 1, inspired by [17], presents BIOT. It is composed of two repeated steps: 1) optimizing $\mathbf{W}$ given an embedding transformation $\mathbf{R}$ and 2) optimizing $\mathbf{R}$ given regression weights $\mathbf{W}$. These steps are repeated until the change of $\mathbf{W}$ from one iteration to another is lower than a predefined threshold[1].

Algorithm 1: BIOT algorithm, inspired by [17].

```
Data: MDS embedding X and feature matrix F
Result: Explanation of X with sparse weights W
R = I_m;
X = XR;
W is obtained by solving Eq. (6) for each k of X;
while W changes do
    // Optimizing R
    R is obtained by solving Eq. (9);
    X = XR;
    // Optimizing W
    for each dimension k of X do
        w_{•,k} is obtained by solving Eq. (6);
return W and R
```

[1] The implementation of BIOT in R can be found at https://github.com/rebeccamarion/BIOT.

3.3.4. Selecting the Hyperparameters λ and m

BIOT requires the selection of two hyperparameters: the λ used for the Lasso penalty, which represents the relative importance of sparsity with respect to error, and the number m of embedding dimensions to analyze. The first hyperparameter, λ, is common to all Lasso problems and can be set according to the same strategies. For instance, the λ leading to the smallest test mean squared error (test MSE) can be considered. Alternatively, the "one-standard error" rule [5] may be used, whereby the largest λ within one-standard deviation of the minimum test MSE is chosen. This corresponds to a sparser model than for the minimum test MSE model, without resulting in a significantly different level of error.

While sparsity helps avoid issues like overfitting, it is mainly used, in this work, as a means to obtain interpretable regression models (i.e. models with a reasonable number of non-zero weights). Therefore, while the above heuristics can be used to select λ, the final choice remains with the user. In practical settings, it may be interesting to increase the sparsity of regression models, and thus their interpretability, even at the cost of increasing their test MSE. For the evaluation of BIOT in this paper (Section 5), however, one of the methods presented in the previous paragraph is used to maintain objectivity.

The number m of embedding dimensions is more similar to the hyperparameters used in unsupervised learning. In clustering, for instance, different numbers of clusters must be tested and analyzed, given the knowledge of experts, to see which choice makes sense. Similarly, for BIOT, different numbers of dimensions can be tested to observe how the analysis changes. Oftentimes, increasing m results in explanations with more and more nuance, as seen in the example provided in Section 6.

In addition to increasing the granularity of the explanation, increasing the number of dimensions reduces the information loss in the embedding, making it more faithful to the original dataset. It also makes the explanation of each individual dimension easier, as less information must be explained by the external features. However, these advantages come at the cost of increasing the cognitive load for the user: understanding 10 dimensions simultaneously may be difficult, even if each dimension is explained by only two or three features. Therefore, some balance must be found between cognitive ease and the level of nuance and faithfulness. Gradually increasing the number of dimensions provides a practical means of evaluating when the number of dimensions m becomes too high for cognitive processing.

The next section presents methods that can be seen as competitors to BIOT.

4. Related Work

Best interpretable rotation (BIR) finds rotations of 2D MDS embeddings that can be explained by sparse multiple regression models [13, 12]. The authors of BIR demonstrated that both the model error and number of non-zero weights of Lasso multiple regression models depend on the rotation of the response matrix $\mathbf{X}$. In order to find a rotation balancing model error with interpretability, they proposed finding the best rotation angle θ^* as follows:

$$\theta^* = \arg\min_{\theta} \frac{1}{2n} \|\mathbf{X}\mathbf{R}^{\theta} - \mathbf{F}\mathbf{W}^{\theta}\|_F^2 + \lambda \sum_{k=1}^{m} \|\mathbf{w}^{\theta}_{\bullet,k}\|_0, \quad (11)$$

where $m = 2$, $\mathbf{R}^{\theta}$ is the 2D rotation matrix for a given angle θ and $\mathbf{w}^{\theta}_{\bullet,k}$ is the Lasso solution explaining the k^{th} dimension of $\mathbf{X}$ rotated by $\mathbf{R}^{\theta}$. While the matrix of weights $\mathbf{W}^{\theta}$ is the solution to a regression problem with an ℓ_1-norm penalty, BIR's objective function is minimized with respect to a scalar θ, making it feasible to impose an ℓ_0-norm penalty.

Looking for such a θ^* results in better solutions than other potential competitors from the literature [12], but BIR suffers from two important issues. First, θ is optimized by performing an exhaustive search. In practice, an optimization method for non-convex objective functions is used, such as simulated annealing. The solution for this kind of optimization depends on how long users accept to wait for a solution, as a time-stopping threshold is provided as input. Second, BIR can only find a rotation matrix for 2D MDS embeddings.

BIOT addresses both of these weaknesses. It relaxes BIR's constraint that $\mathbf{R}$ be a rotation matrix, allowing $\mathbf{R}$ to be any type of orthogonal matrix (which includes rotation and reflection matrices as special cases). This makes it possible to apply the method to higher-dimensional embeddings, while preserving the meaningful structure in the transformed embedding. BIOT also relaxes the ℓ_0 norm in BIR's objective function to an ℓ_1 norm applied to the columns of $\mathbf{W}$. This makes the objective function bi-convex, and the solution can be found using alternating optimization instead of an exhaustive search.

For MDS embeddings with two or more dimensions m, sparse reduced rank regression (SRRR) [17] could potentially be used to regress transformed embedding dimensions on external features. SRRR was originally introduced as a method for predicting an untransformed response matrix using a weight matrix $\mathbf{C} = \mathbf{W}\mathbf{R}^{\top}$ of

fixed rank r. The original problem presented in [17] is to find $\mathbf{R}$ ($m \times r$) and $\mathbf{W}$ ($d \times r$) by solving

$$\arg\min_{\mathbf{R},\mathbf{W}} \frac{1}{2n}\|\mathbf{X} - \mathbf{F}\mathbf{W}\mathbf{R}^\top\|_F^2 + \lambda \sum_{j=1}^{d} \|\mathbf{w}_{j\bullet}\|_2 \quad \text{s.t. } \mathbf{R}^\top\mathbf{R} = \mathbf{I}_r \text{ and } \text{rank}(\mathbf{W}\mathbf{R}^\top) = r, \tag{12}$$

where $\mathbf{w}_{j\bullet}$ is the j^{th} row of $\mathbf{W}$, $\lambda > 0$ and $r \in \{1, ..., \min(d, m)\}$. The second term in Eq. (12) is a Group-Lasso penalty that forces the elements of $\mathbf{w}_{j\bullet}$ to be either all zero or non-zero [18]. As λ increases, more and more rows of $\mathbf{W}$ are set to zero, meaning that fewer and fewer features are used to explain the response matrix.

The objective function in Eq. (12) can be reformulated to show that the matrix $\mathbf{W}$ in SRRR contains the regression weights for predicting a transformed response matrix $\mathbf{XR}$. Indeed, thanks to the rotational invariance of the Frobenius norm, the first term in Eq. (12) can be rewritten as follows:

$$\frac{1}{2n}\|\mathbf{X} - \mathbf{F}\mathbf{W}\mathbf{R}^\top\|_F^2 = \frac{1}{2n}\|\mathbf{X}\mathbf{R} - \mathbf{F}\mathbf{W}\|_F^2. \tag{13}$$

For the current application, the meaningful structure from the original embedding $\mathbf{X}$ must be preserved. Therefore, SRRR is only applicable when its hyperparameter r (the rank of $\mathbf{W}\mathbf{R}^\top$ and the number of columns in $\mathbf{R}$) is fixed to $r = m$, the number of embedding dimensions. The setting $r = m$ is the only one that ensures that $\mathbf{R}$ is an orthogonal matrix and that the transformed embedding $\mathbf{XR}$ retains the same number of dimensions as the original embedding $\mathbf{X}$.

Despite its potential relevance for the problem at hand, the sparsity constraints in SRRR are less well adapted than the constraints in BIOT. Indeed, for SRRR, the same set of features would be selected for each transformed embedding dimension, making it difficult to attribute a distinct meaning to each dimension. In contrast, BIOT makes it possible to select potentially distinct sets of features for each embedding dimension, providing greater model interpretability.

Other methods in the literature address either the problem of finding an orthogonal transformation or finding a sparse multiple regression model, but not both. Sparse multi-task regression methods (e.g. multi-task Lasso [19], adaptive multi-task Lasso [20], robust feature selection [21] and joint rank and row selection [22]) find a sparse weight matrix but do not transform the response matrix in any way. Latent variable methods, such as eigenvector partial least squares regression (eigen PLS-R) [23, 24], find an orthogonal transformation of a response matrix that improves the prediction of subsequent multiple regression models, but the models are entirely non-sparse. Sparse latent variable approaches such as sparse canonical correlation analysis (SCCA) [25, 26, 27, 28] and sparse partial least squares regression (SPLS-R) [29] estimate sparse regression models, but the transformation of the response matrix is not orthogonal.

The next section evaluates BIOT by comparing its performance with state-of-the-art methods.

5. Evaluation of BIOT

This section compares BIOT with competitors from the literature for MDS embeddings of two or more dimensions.

5.1. Evaluation Datasets

Three real-world datasets are drawn from the field of ecology: the Doubs river fish communities dataset (Doubs) [30], the Oribatid mites dataset (Mite) [31, 32] and the hunting spider dataset (Spider) [33]. Each dataset is made up of two distinct feature sets. The first feature set $\mathbf{Q}$ contains abundances of p different species (of fish, mites and spiders, respectively) measured at n different sampling sites. The second feature set $\mathbf{F}$, in each dataset, corresponds to d features measured at the n sites, such as Cartesian coordinates, water pH and altitude. For each dataset, ordinal MDS is applied to the first feature set $\mathbf{Q}$ in order to produce several embeddings with a number of dimensions m ranging from 2 to $\min(p, d) - 1$.

The fourth dataset used in our evaluation comes from an experiment in psychology about stereotypes (Stereotypes) [8]. In this dataset, the first feature set $\mathbf{Q}$ contains similarity comparisons made by participants between n social groups (e.g. students, homeless and athletes). The second feature set $\mathbf{F}$ contains features that encode stereotypes about these social groups (e.g. degree of smartness, trustworthiness and sincerity). The first feature set $\mathbf{Q}$ ($n \times n$) is used to generate several MDS embeddings, as for the other datasets.

For all datasets, the second feature set $\mathbf{F}$ (normalized) is used to explain the mean-centered embeddings produced by the MDS of feature set $\mathbf{Q}$. Table 1 summarizes the characteristics of the datasets used in the experiments.

Table 1: Characteristics of the evaluation datasets

dataset	instances	features		
		Q	**F**	**total**
Doubs	30	27	13	40
Mite	70	35	16	51
Spider	28	12	15	27
Stereotypes	80	80	31	111

5.2. Experimental Protocol

Four methods are compared in this study: BIOT, BIR (for 2D embeddings only), SRRR and eigen PLS with Lasso regression (ePLS+Lasso). For ePLS+Lasso, we add a Lasso step to ordinary eigen PLS in order to benchmark BIOT and to make the results comparable. Eigen PLS is used to estimate a transformation matrix $\mathbf{R}$, then Lasso regression is performed based on the transformed embedding. A range of 20 values for λ ($[0.0001, 3.5]/\sqrt{d}$ in logarithmic scale) was chosen such that each method produces solutions ranging from entirely sparse to entirely non-sparse. For SRRR, the rank r of the matrix $\mathbf{WR}^\top$ is fixed to the number of embedding dimensions, as explained in Section 4.

For each method, embedding and value of λ, 10-fold cross-validation is performed to evaluate the average test prediction mean squared error (MSE). The average test MSEs for each method and embedding are plotted with respect to the average number of non-zero weights in $\mathbf{W}$ per dimension, where each point represents a value of λ. The minimum of each plotted curve is the minimum test MSE.

In order to statistically analyze the results obtained for a given dataset and number of dimensions m, the performance of the methods is compared for a particular choice of λ. For each method, λ is chosen as the value with the smallest average test MSE.

5.3. Results and Discussion

In this experiment, BIOT and the competing methods are applied to embeddings of different numbers of dimensions. The curves for 2D, 4D and 6D embeddings are presented in Fig. 1. For 2D embeddings (Figs. 1a, 1d, 1g and 1j), BIOT finds solutions that are generally as sparse or sparser than those of the other methods, for a similar MSE. Indeed, if a horizontal line is drawn in the graphs (representing a fixed MSE value), BIOT is almost always to the left of the other curves.

Similar trends can be observed for embeddings of more than two dimensions, as shown in the second and third columns of Fig. 1. Note that BIR is not present in these plots, as it can only be applied to 2D embeddings. Interestingly, the difference between the curves is accentuated as the number of embedding dimensions increases. This can be observed as a shifting pattern in the 4D and 6D embeddings of Stereotypes in Fig. 1k and Fig. 1l, compared to a similar but less clear pattern in the 2D embedding of Stereotypes in Fig. 1j. This observation is important, as BIOT is designed for use on higher-dimensional embeddings, where reconstruction error (like the stress) is lower.

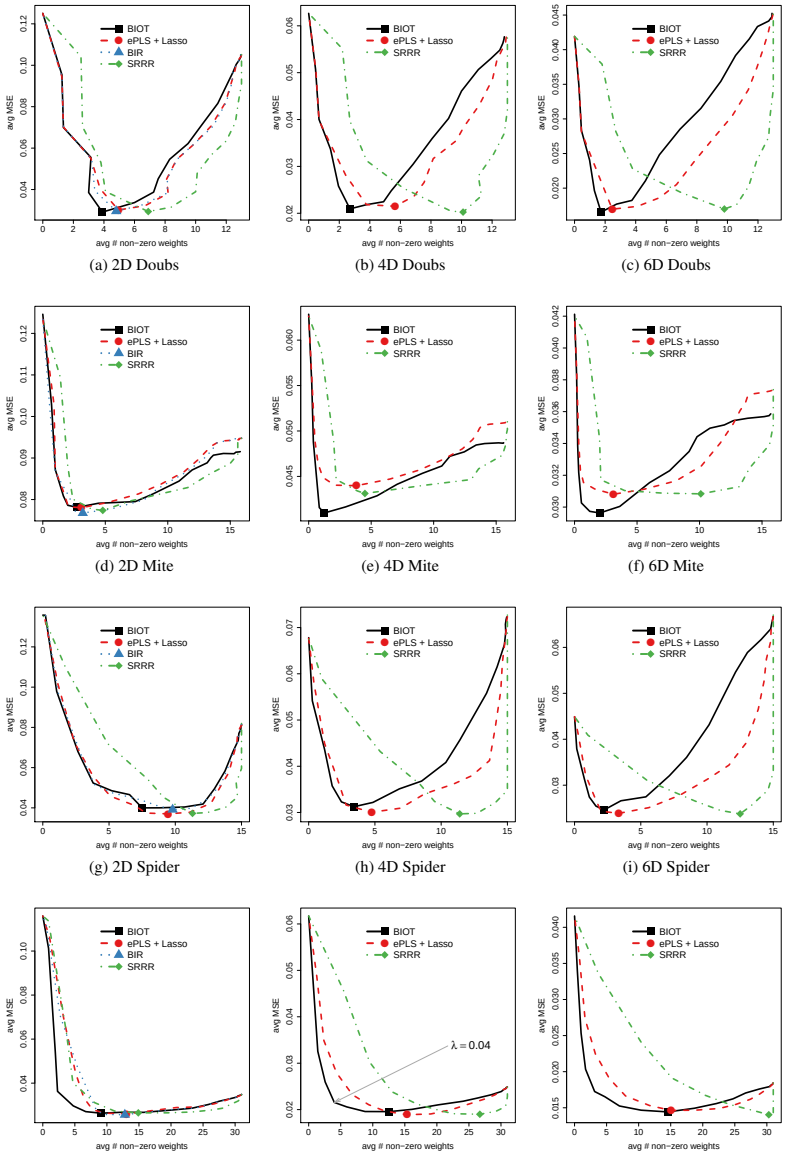
The comparison of all methods applied to all embeddings are shown in Table 2. The last embeddings of Stereotypes ($m > 13$) are omitted, as they have stress levels equivalent to the stress for $m = 13$. The results for each method are shown as a pair of values (average number of non-zero weights per dimension, average test MSE $\times 10^3$). On each line, results with the highest sparsity (resp. lowest MSE) are highlighted in bold (resp. italics), as well as all other results that are not significantly different according to a pairwise Wilcoxon signed-rank test ($\alpha = 0.05$). Any results not shown in bold or italics are significantly worse than the best results across the different folds.

As seen in Table 2, the best MSE is generally not significantly different for all methods, but the average number of non-zero weights often is. Most of the time, BIOT provides solutions with a lower number of features per dimension, while having a test MSE similar to its competitors. The average number of features used to explain a dimension generally decreases as the number of dimensions m increases. This can be explained by the fact that each new embedding dimension adds less information than previous ones (the stress decreases less). Therefore, fewer and fewer features are needed to explain each additional dimension.

In the next section, several embeddings of Stereotypes are analyzed using BIOT to demonstrate the interpretation of an MDS embedding with more than two dimensions.

6. Case Study: Applying BIOT to Stereotypes

The Stereotypes dataset was collected in order to study how people (in the US) implicitly assign stereotypes to social groups. In a first experiment, participants ranked the similarity between social groups, such as celebrities, students and criminals (feature set **Q**). In a second experiment, participants scored these social groups with respect to stereotypes, such as wealthy, altruistic and skillful (external features **F**). The goal was then to see how these stereotypes could explain perceived similarities between social groups.



(a) 2D Doubs (b) 4D Doubs (c) 6D Doubs

(d) 2D Mite (e) 4D Mite (f) 6D Mite

(g) 2D Spider (h) 4D Spider (i) 6D Spider

(j) 2D Stereotypes (k) 4D Stereotypes – The arrow indicates the λ used in the case study in Section 6. (l) 6D Stereotypes

Figure 1: Performance of BIOT, BIR, ePLS+Lasso and SRRR for several λ values. The average test MSE is plotted against the average number of non-zero weights per dimension. The three columns represent 2D, 4D and 6D embeddings, and the four rows represent the datasets Doubs, Mite, Spider and Stereotypes. The minimum test MSE is highlighted for each method.

Table 2: Results for four datasets Doubs (Do), Mite (Mi), Spider (Sp) and Stereotypes (St). Each result is a pair (average number of non-zero weights, average MSE $\times 10^3$) corresponding to the λ with the smallest average test MSE.

	m	stress	BIR	BIOT	ePLS	SRRR
Do	2	0.070	4.8, *30*	**3.9**, *29*	5.0, *30*	6.9, *29*
	3	0.038		**3.1**, *25*	4.0, *25*	7.8, *24*
	4	0.026		**2.7**, *21*	5.7, *21*	10.1, *20*
	5	0.018		**2.2**, *19*	2.9, *19*	9.5, *18*
	6	0.013		**1.7**, *17*	2.5, *17*	9.8, 17
	7	0.012		**1.5**, *14*	2.1, *15*	9.7, 16
	8	0.008		**1.3**, *13*	1.9, *14*	10.1, 15
	9	0.006		**1.1**, *12*	1.7, *12*	10.1, 14
	10	0.005		**1.0**, *11*	1.5, *11*	10.1, 13
	11	0.004		**0.9**, *10*	1.4, *11*	10.1, 12
	12	0.003		**0.8**, *9*	1.3, *10*	10.1, 11
Mi	2	0.144	3.2, *77*	**2.8**, *78*	**3.1**, *78*	4.8, *77*
	3	0.112		**2.0**, *51*	4.6, *53*	5.1, *53*
	4	0.091		**1.2**, *41*	3.8, *44*	4.5, *43*
	5	0.077		**1.1**, *35*	3.0, *37*	5.2, *36*
	6	0.065		**2.0**, *30*	3.1, *31*	10.1, *31*
	7	0.057		**1.8**, *26*	2.6, *27*	10.2, *27*
	8	0.049		**1.6**, *23*	2.6, *24*	10.9, *24*
	9	0.044		**1.4**, *21*	2.5, *22*	11.1, *22*
	10	0.040		**1.3**, *19*	3.7, *20*	11.4, *20*
	11	0.036		**1.1**, *17*	3.3, *18*	8.0, *18*
	12	0.032		**1.0**, *16*	3.1, *17*	11.4, *17*
Sp	2	0.089	9.8, *39*	**7.5**, *40*	9.4, *37*	11.3, *37*
	3	0.055		**4.1**, *36*	7.3, *33*	10.9, *37*
	4	0.037		**3.4**, *31*	4.8, *30*	11.4, *30*
	5	0.025		**2.7**, *28*	3.9, *26*	12.3, *28*
	6	0.019		**2.2**, *25*	3.3, *24*	12.5, *24*
	7	0.016		**2.0**, *22*	2.9, *21*	12.6, *21*
	8	0.012		**1.6**, *20*	2.5, *19*	11.1, *20*
	9	0.007		**1.5**, *18*	2.2, *18*	11.1, *18*
	10	0.004		**1.4**, *17*	2.0, *16*	11.2, *16*
	11	0.001		**1.2**, *15*	1.8, *15*	11.2, *15*
St	2	0.291	12.8, *26*	**9.1**, *26*	13.1, *26*	14.9, *27*
	3	0.207		12.8, 17	**11.2**, *16*	20.3, *16*
	4	0.169		**12.6**, *20*	15.3, *19*	26.7, *19*
	5	0.146		**8.3**, 17	18.4, *16*	27.9, *16*
	6	0.134		**14.6**, *14*	**15.0**, *15*	30.3, *14*
	7	0.127		**9.0**, *13*	18.6, 14	30.2, *13*
	8	0.122		**8.5**, *12*	17.9, *12*	30.1, *12*
	9	0.120		**8.0**, *11*	17.5, *11*	30.1, *11*
	10	0.118		**7.4**, 11	17.1, 11	28.9, *10*
	11	0.116		**6.7**, *10*	11.8, *10*	29.2, *10*
	12	0.116		**6.2**, *9*	16.0, 9	29.2, *9*
	13	0.115		**5.6**, 9	15.2, 9	29.4, *8*

Table 3: Stereotypes related to three embeddings of social groups (dimensions in rows, model weights in parentheses). The most important features for each dimension are in bold.

$m = 3$	$m = 4$	$m = 5$
wealthy (0.28)	**wealthy** (0.26)	**wealthy** (0.22)
scientific (0.06)		power (0.05)
diversity (0.05)	diversity (0.06)	diversity (0.01)
traditional (0.17)	traditional (0.04)	**traditional** (0.08)
religious (0.01)	**religious** (0.15)	**religious** (0.09)
	comfort (0.04)	comfort (0.04)
	prevention (0.02)	prevention (0.04)
conventional (0.15)	**conventional** (0.22)	**conventional** (0.14)
loyalty (0.05)	loyalty (0.07)	loyalty (0.01)
familiarity (0.01)		individualistic (0.01)
not smart (0.16)	**not smart** (0.13)	**not smart** (0.16)
egoistic (0.07)	egoistic (0.05)	egoistic (0.01)
masculine (0.06)	masculine (0.09)	masculine (0.04)
competitive (0.06)		competitive (0.05)
typical (0.04)	typical (0.03)	typical (0.03)
	intolerant (0.02)	
	familiarity (0.01)	
		conservative (0.14)
		masculine (0.03)
		preservation (0.03)

In order to study Stereotypes with BIOT, two choices must be made: (i) the number of dimensions m for the MDS embedding and (ii) the value of the hyperparameter λ in BIOT. For our case study, we choose the 3D, 4D and 5D embeddings for $\lambda = 0.04$. It is common practice to choose a λ resulting in the sparsest, most interpretable model possible while maintaining a low MSE. In order to remain objective, λ is chosen as follows. For the λ value with the smallest average test MSE, a 95% confidence interval is calculated. Then, the largest λ value with an average test MSE within this confidence interval is selected. The chosen value is highlighted in Fig. 1k.

For the first embedding, BIOT explains the three dimensions with the stereotypes wealthy, traditional/conventional and not smart. The details of the selected stereotypes are in column one of Table 3. For the 4D embedding, BIOT provides an explanation of the fourth dimension by roughly separating traditional and conventional into two dimensions (the details are in column two of Table 3). Finally, for the 5D embedding, BIOT explains the new fifth dimension as a political dimension through the conservative-liberal stereotype (more details in the third column of Table 3).

The advantage of analyzing more than two dimensions is that higher-dimensional embeddings have a small reconstruction error, which is quantified by Kruskal's stress (see Section 3.2). Moreover, with BIOT, it is possible to observe how low dimensional embeddings approximate trends from higher-dimensional embeddings. For example, when changing from 4D

to 3D (0.169 to 0.207 in stress), BIOT associates the traditional/religious and conventional stereotypes with a single dimension, rather than two. This combination may explain the increase in stress in the 3D embedding, as two orthogonal trends are approximated by a single trend.

By adding dimensions, it is also possible to identify trends that are not apparent in lower-dimensional embeddings. While the original study did not identify smartness as a relevant stereotype, the 3D analysis with BIOT identifies it as important for explaining a third dimension in the data. Indeed, social groups such as criminals and red necks are identified as egoistic, masculine and not smart by the participants. At the same time, the two other dimensions explained by BIOT correspond to the findings in the original paper, where MDS embeddings were explained by two quasi-orthogonal trends: the socio-economical status (represented here by wealthy) and the type of beliefs (represented here by the stereotypes conventional and traditional) [8].

This case study shows how insightful it is to use BIOT to analyze MDS embeddings with more than two dimensions. Given BIOT's sparsity and MSE performance, increasing the number of dimensions (and therefore reducing the stress) does not make the new embedding much more difficult to understand. Indeed, each new dimension is explained by small, generally disjoint sets of external features. The results of this case study were presented to the main investigator of the original study [8], who found them coherent with current theory in psychology, while providing interesting insights.

7. Conclusion

In this paper, we proposed a method, called BIOT, that makes it possible to explain MDS embeddings of any number of dimensions. BIOT is based on an iterative optimization of two parameter matrices: a weight matrix $\mathbf{W}$ and an orthogonal transformation matrix $\mathbf{R}$.

BIOT was evaluated on datasets corresponding to real-world problems. We demonstrated that BIOT outperforms competitive methods with respect to the interpretability of solutions. The analysis of MSE-sparsity curves revealed that, for the same level of MSE, BIOT provides models that are more sparse, and thus easier to interpret. In order to demonstrate BIOT's ease of use, a case study based on a dataset from a psychological experiment on stereotypes was presented.

In future work, a grouping-penalty could be added to BIOT to encourage groups, rather than individual features, to be selected for each embedding dimension. By grouping features in a meaningful way, even models with many features could be easily interpreted. This would be advantageous for datasets where mostly non-sparse models have the best test MSE or for applications where feature grouping is desired.

Acknowledgements

The authors would like to thank Alex Koch, assistant professor at the University of Chicago, for his feedback on the application of BIOT to his dataset. The work of R. Marion was supported by the Belgian Fund for Scientific Research (F.R.S.-FNRS, FRIA grant).

References

[1] A. Bibal, B. Frénay, Interpretability of machine learning models and representations: an introduction, in: Proceedings of the European Symposium on Artificial Neural Networks, Bruges, Belgium, 2016, pp. 77–82.

[2] Z. C. Lipton, The mythos of model interpretability, in: ICML Workshop on Human Interpretability of Machine Learning, New York, USA, 2016.

[3] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Computing Surveys 51 (5) (2018) 1–42.

[4] R. E. Bellman, Adaptive control processes: a guided tour, Princeton university press, 1961.

[5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag New York, 2009.

[6] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

[7] J. B. Kruskal, M. Wish, Multidimensional Scaling, Sage, 1978.

[8] A. Koch, R. Imhoff, R. Dotsch, C. Unkelbach, H. Alves, The ABC of stereotypes about groups: Agency/socioeconomic success, conservative–progressive beliefs, and communion, Journal of Personality and Social Psychology 110 (5) (2016) 675–709.

[9] T. Næs, P. B. Brockhoff, O. Tomic, Statistics for sensory and consumer science, John Wiley & Sons, 2011.

[10] J. A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Springer, 2007.

[11] J. J. Chang, J. D. Carroll, How to use PROFIT, a computer program for property fitting by optimizing nonlinear or linear correlation, Unpublished Manuscript, Bell Laboratories (1968).

[12] R. Marion, A. Bibal, B. Frénay, BIR: A method for selecting the best interpretable multidimensional scaling rotation using external variables, Neurocomputing 342 (2019) 83–96.

[13] A. Bibal, R. Marion, B. Frénay, Finding the most interpretable MDS rotation for sparse linear models based on external features, in: Proceedings of the European Symposium on Artificial Neural Networks, Bruges, Belgium, 2018, pp. 537–542.

[14] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: the Lasso and Generalizations, Chapman and Hall/CRC, 2015.

[15] J. C. Gower, G. B. Dijksterhuis, Procrustes Problems, Oxford University Press, 2004.

[16] G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, 2013.

[17] L. Chen, J. Z. Huang, Sparse reduced-rank regression for simultaneous dimension reduction and variable selection, Journal of the American Statistical Association 107 (500) (2012) 1533–1545.
[18] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.
[19] G. Obozinski, B. Taskar, M. Jordan, Multi-task feature selection, Statistics Department, UC Berkeley, Tech. Rep 2 (2.2).
[20] S. Lee, J. Zhu, E. P. Xing, Adaptive multi-task lasso: with application to eqtl detection, in: Advances in neural information processing systems, 2010, pp. 1306–1314.
[21] F. Nie, H. Huang, X. Cai, C. H. Ding, Efficient and robust feature selection via joint 2, 1-norms minimization, in: Advances in neural information processing systems, 2010, pp. 1813–1821.
[22] F. Bunea, Y. She, M. H. Wegkamp, et al., Joint variable and rank selection for parsimonious estimation of high-dimensional matrices, The Annals of Statistics 40 (5) (2012) 2359–2388.
[23] K. Varmuza, P. Filzmoser, Introduction to Multivariate Statistical Analysis in Chemometrics, CRC press, 2016.
[24] H. Abdi, Partial least squares regression and projection on latent structure regression (PLS regression), Wiley Interdisciplinary Reviews: Computational Statistics 2 (1) (2010) 97–106.
[25] D. M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics 10 (3) (2009) 515–534.
[26] I. Wilms, C. Croux, Sparse canonical correlation analysis from a predictive point of view, Biometrical Journal 57 (5) (2015) 834–851.
[27] X. Suo, V. Minden, B. Nelson, R. Tibshirani, M. Saunders, Sparse canonical correlation analysis, arXiv preprint arXiv:1705.10865.
[28] Q. Mai, X. Zhang, An iterative penalized least squares approach to sparse canonical correlation analysis, Biometrics.
[29] H. Chun, S. Keleş, Sparse partial least squares regression for simultaneous dimension reduction and variable selection, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72 (1) (2010) 3–25.
[30] J. Verneaux, Cours d'eau de franche-comté (massif du jura). recherches écologiques sur le réseau hydrographique du doubs. essai de biotypologie., Ph.D. thesis, Université de Besançon (1973).
[31] D. Borcard, P. Legendre, P. Drapeau, Partialling out the spatial component of ecological variation, Ecology 73 (3) (1992) 1045–1055.
[32] D. Borcard, P. Legendre, Environmental control and spatial structure in ecological communities: an example using oribatid mites (acari, oribatei), Environmental and Ecological Statistics 1 (1) (1994) 37–61.
[33] N. Smeenk-Enserink, P. Van Der Aart, Correlations between distributions of hunting spiders (lycosidae, ctenidae) and environmental characteristics in a dune area, Netherlands Journal of Zoology 25 (1) (1974) 1–45.

CHAPTER 24

EXPLAINING T-SNE EMBEDDINGS LOCALLY BY ADAPTING LIME

The article presented in this chapter is accepted and will be published in the proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) in 2020.

Explaining t-SNE Embeddings Locally by Adapting LIME

Adrien Bibal*, Viet Minh Vu*, Géraldin Nanfack* and Benoît Frénay

University of Namur - NADI - Faculty of Computer Science - PReCISE
rue Grandgagnage 21, B-5000 Namur - Belgium

Abstract. Non-linear dimensionality reduction techniques, such as t-SNE, are widely used to visualize and analyze high-dimensional datasets. While non-linear projections can be of high quality, it is hard, or even impossible, to interpret the dimensions of the obtained embeddings. This paper adapts LIME to locally explain t-SNE embeddings. More precisely, the sampling and black-box-querying steps of LIME are modified so that they can be used to explain t-SNE locally. The result of the proposal is to provide, for a particular instance $\mathbf{x}$ and a particular t-SNE embedding $\mathbf{Y}$, an interpretable model that locally explains the projection of $\mathbf{x}$ on $\mathbf{Y}$.

1 Introduction

An important step in data analysis is to look at the data at hand with the use of dimensionality reduction (DR) techniques. If the dimensionality is reduced to two, the embedding can be presented in a scatter plot and the data can be visually explored. One of the most effective DR techniques is t-SNE [1]. t-SNE is a non-linear DR (NLDR) technique, whose objective is to preserve high-dimensional (HD) neighborhood in the low-dimensional (LD) embedding. While t-SNE is effective to grasp HD patterns visually, the non-parametric mapping is hard to interpret. Moreover, the two dimensions of the embedding may not have a particular meaning [2]. However, as t-SNE preserves neighborhoods, it can be expected that these two dimensions can be analyzed *locally*.

A popular technique for studying black-box models locally through the use of interpretable models is LIME [3]. However, LIME is designed to explain supervised learning models, and is unfortunately not suitable for t-SNE.

In this paper, we propose to drastically change some steps of the LIME algorithm to explain t-SNE and other non-parametric NLDR techniques that need local explanations. The interpretation of t-SNE is discussed in Section 2. LIME is presented and explained in Section 3. The adaption of LIME to locally explain t-SNE embeddings is proposed in Section 4. Results using this new algorithm are shown in Section 5 and the paper is concluded in Section 6.

2 The Interpretability of t-SNE

t-SNE is a non-parametric dimensionality reduction (DR) method that learns embeddings of high-dimensional (HD) data [1]. t-SNE computes pairwise similarities between instances, which are then converted into neighborhood probabilities. Given instances $\mathbf{x}_i$ and $\mathbf{x}_j$ in HD, the probability that they are neighbors is

*The first three authors have contributed equally. G. Nanfack is funded by the EOS VeriLearn project n. 30992574 of the Fonds de la Recherche Scientifique (F.R.S-FNRS) in Belgium.

$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$, where n is the number of instances, $p_{j|i} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||\mathbf{x}_k - \mathbf{x}_i||^2/2\sigma_i^2)}$ and σ_i is set by the perplexity. In LD, t-SNE similarly computes pairwise similarities with the Student t-distribution $q_{ij} = \frac{(1+||\mathbf{y}_i - \mathbf{y}_j)||^2)^{-1}}{\sum_{k \neq l}(1+||\mathbf{y}_k - \mathbf{y}_1||^2)^{-1}}$, where $\mathbf{y}_i$ is the projection of $\mathbf{x}_i$ in LD. The projections $\mathbf{y}_i$, $i = 1..n$ are learned by minimizing the Kullback-Leibler (KL) divergence between $\mathbf{P}$ and $\mathbf{Q}$ through gradient descent.

t-SNE achieves state-of-the-art results for dimensionality reduction. Yet, unlike PCA, interpreting the embedding dimensions is difficult or even impossible. Earlier work proposes versions of t-SNE that can be to some extent interpretable. A parametric t-SNE is proposed in [4, 5] as a generalized linear model with an explicit mapping $\mathbf{y}_i = \sum_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$, with K being any kernel. The authors use a Gaussian kernel, which makes the embedding difficult to interpret. Using a linear kernel would make it possible to interpret the mapping, at the expense of losing the non-linear projection quality of t-SNE. More generally, Bunte et al. propose a general framework for DR mappings, that makes it possible to extend DR methods to obtain a (potentially interpretable) explicit mapping [6].

In this paper, we aim to directly explain the non-linear mapping of t-SNE (also called post-hoc interpretability), instead of modifying the method. The challenge is that t-SNE is known for breaking the relationship between instances whose distance is large in HD [2]. Therefore, explaining the embedding globally does not always make sense. As neighborhoods are preserved, we hypothesize that explanations can be made in those very neighborhoods. Using the neighborhood of an instance for a local explanation of a black-box model can be performed by LIME in supervised learning. This paper adapts LIME for t-SNE.

3 LIME for Explaining Models

Local interpretable model-agnostic explanations (LIME) is an algorithm designed to locally explain black-box classifiers and regressors [3]. LIME addresses the question *how a black-box model f behaves in the neighborhood of* $\mathbf{x}$. To this end, LIME globally samples new samples $\mathbf{z}_j$ around $\mathbf{x}$ and captures the *locality* by weighting the samples $\mathbf{z}_j$ w.r.t. their distance from $\mathbf{x}$. The black-box model is then queried to get the predictions $f(\mathbf{z}_j)$. An interpretable model, such as a weighted sparse linear model, is used to approximate the behavior of f near $\mathbf{x}$. While LIME is very popular in supervised learning, very little has been done to use it in unsupervised learning. Because LIME explains locally, it is a good candidate to explain t-SNE embeddings where neighborhoods are preserved.

The LIME algorithm has two phases: sampling new samples $\mathbf{z}_j$ and query the black-box model for predictions $f(\mathbf{z}_j)$. With classical t-SNE, such a query is not possible because no explicit mapping $\mathbf{y}_j = f(\mathbf{z}_j)$ exists. Indeed, if new samples $\mathbf{z}_j$ have to be inserted in an already computed embedding, the embedding must be entirely re-calculated, which means that the whole HD-LD mapping will change. This is an issue because a particular embedding cannot be explained with new instances without being altered. Furthermore, contrarily to LIME, we need to find samples $\mathbf{z}_j$ for which their projection is close to the projection of $\mathbf{x}$.

In order to implement LIME for t-SNE, three issues must be tackled: (i) the

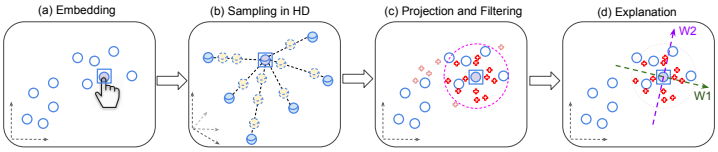


Fig. 1: The proposed workflow for adapting LIME to explain t-SNE embedding.

way new instances are sampled, (ii) the way t-SNE, as a black-box, is queried and (iii) the use of an interpretable model to locally explain t-SNE embeddings.

4 Adapting LIME to Explain t-SNE Embeddings

This section proposes an adaptation of LIME to locally explain t-SNE as illustrated in Fig. 1. Two important changes are introduced: how to sample new instances adequately for t-SNE (Section 4.1) and how to query t-SNE to know how new samples would have been projected (Section 4.2). Finally, an interpretable model for explaining t-SNE locally is presented in Section 4.3.

4.1 Adapting Sampling in LIME for t-SNE

The first contribution of this paper is a sampling strategy to generate samples $\mathbf{z}_j$ to explain a t-SNE embedding around a particular instance $\mathbf{x}$ (see Fig. 1b). The main issue related to the sampling is that the distance between instances that are far apart in HD are not necessarily preserved in LD. In order to solve this issue, several neighbors $\mathbf{x}_j$ of $\mathbf{x}$ in the original dataset are chosen according to the neighborhood size t-SNE used when building the embedding. The SMOTE oversampling [7] is used to produce new samples $\mathbf{z}_j = \mathbf{x} + \alpha * (\mathbf{x}_j - \mathbf{x})$, with $\alpha \in [0, 1]$. Considering new instances between the instance of interest $\mathbf{x}$ and one of its neighbor $\mathbf{x}_j$, the aim is to obtain a point that fits on the HD manifold.

4.2 Adapting Black-Box Querying in LIME for t-SNE

As explained in Section 3, projecting new samples on an already computed embedding is difficult: t-SNE is non-parametric and the HD-to-LD mapping is unknown. One could use a parametric version of t-SNE [8, 5]. However, this paper focuses on explaining classical t-SNE, instead of modifying to make it interpretable. In order to query t-SNE, we only optimize the projection of each new sample $\mathbf{z}_j$, while fixing the projection of the instances from the original dataset unchanged (see Fig. 1c). Samples that are projected far away from the projection of $\mathbf{x}$ are filtered out to focus on a local region of the embedding.

When the sampling procedure explained in Section 4.1 is performed and the sampled instances $\mathbf{z}_j$ are projected, the last step is to use an interpretable model to understand the projection of the samples $\mathbf{z}_j$ in the embedding (see Fig. 1d).

4.3 Explaining t-SNE Locally with BIR

t-SNE has particularities that must be taken into account to explain its embeddings with a sparse linear model. t-SNE produces embeddings that are invariant to rotation, as its only purpose is to preserve neighborhoods. Furthermore, clusters inside the embedding are also invariant to rotation to some extent.

This local invariance to rotation means that a linear regression explaining the embedding dimensions locally must find the best orientation of these dimensions. Let $\mathbf{X}$ $(n \times d)$ be the original dataset and $\mathbf{Y}$ $(n \times 2)$ the embedding, the regression problem is $\mathbf{YR} = \mathbf{XW}$, where $\mathbf{R}$ is a two-dimensional rotation matrix and $\mathbf{W}$ corresponds to the weights of the linear regression model. This is a best interpretable rotation (BIR) problem [9, 10]. The objective of BIR is to find the angle θ^* of $\mathbf{R}$ that provide the best Lasso regression weights $\mathbf{W}$. Similarly to sparse linear models, BIR involves an hyper-parameter λ that balances the importance of the mean squared error (MSE) with respect to the sparsity.

BIR is run with the best hyper-parameter λ^* found by cross-validation on the sampled data. The result is an angle θ^* and sparse weights $\mathbf{W}$. The next section shows the interest of the proposed adaption of LIME for t-SNE.

5 Evaluation and Discussion

The proposed method is evaluated on the *Country* dataset [11], which contains 45 socio-economic indicators (e.g. GDP, women in the economy, healthcare, etc.) released in 2007 for 138 countries. The t-SNE visualization is built with a perplexity of 10. Three countries with very different socio-economic characteristics are chosen for the analysis: Spain (Fig. 2a), Bulgaria (Fig. 2b) and Tunisia (Fig. 2c). They are located in different zones of the embedding: Spain at the center of the occidental cluster (top-right of the embedding), Bulgaria and Tunisia at the edge and the center of the largest cluster. For each country, the left-most scatter plot represents the original embedding in blue and the projected samples instances in red. The transparency indicates the errors made by the linear model applied on the original instances, which gives an idea of the zone that can be explained. The scatter plot in the middle is a zoom on the region explained. The right-most figure represents the weights to explain the two local dimensions.

The quasi-horizontal trend centered on Spain (W1 in Fig. 2a) is mainly explained by the GDP PPP (purchasing power parity), the healthcare (e.g. babies immunized to measles) and the number of women in the parliament. On this axis, it can be observed that the country at the far right of W1 is Iceland, a small country that is known for having favored the number of women at the parliament. On the other side of the axis, big countries with an effective economy can be found, such as the USA and Japan. The quasi-vertical trend (W2 in Fig. 2a) is uniquely determined by the aid towards developing countries.

The first axis explaining the trend around Bulgaria (W1 in Fig. 2b) is characterized by economic and political features. Countries towards the right have higher GDP per capita than Bulgaria ($3109), e.g. Estonia($8331), Croatia ($7724) and Lithuania ($6480), while countries towards the left receive more refugees than Bulgaria (4k), e.g. Guyana (73k) and Malaysia (34k). The second axis (W2 in Fig. 2b) is a mix of demographic, health and economic features. Towards the top, we find countries with larger expenditure on public health and larger imports of good and services than Bulgaria (4.1% and 69% of GDP) like Malta (7.4% and 83%) and Slovakia (5.2% and 79%), while toward the bottom, countries with smaller population in 1975 than Bulgaria (8.7M) can be found,

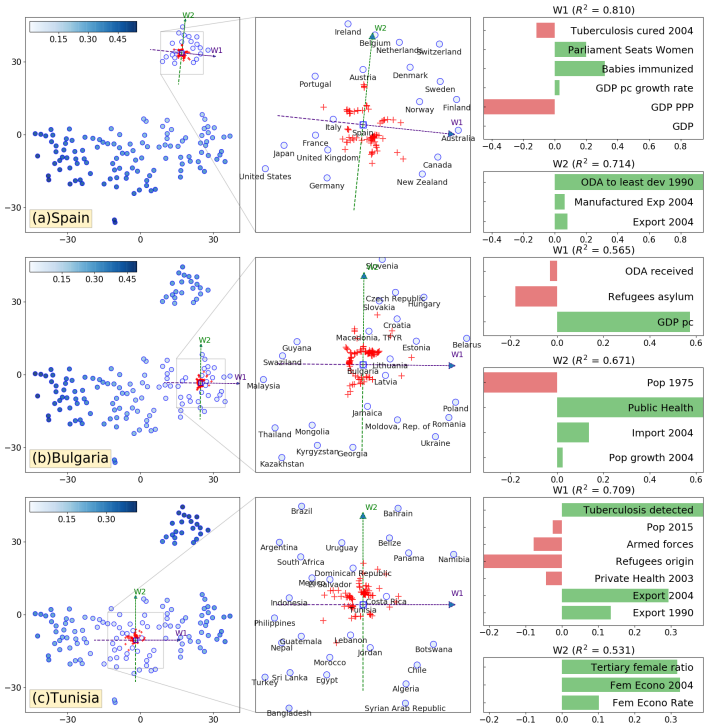


Fig. 2: Evaluation of the proposed method for explaining the local trends in the t-SNE embedding for three selected countries: Spain, Bulgaria and Tunisia. For each axis, R^2 measures how well the embedding is linearly and locally explained. The blue transparency corresponds to the errors of the local model.

like Jamaica (2M), Latvia (2.5M) or Moldova (3.8M).

For the local region around Tunisia on the embedding, the horizontal axis (W1 in Fig. 2c) broadly represents countries that have increased exports from 1990 to 2004, have a rather low population, a small armed force and a high rate of tuberculosis detected in 2004. The countries on the right have a greater rate of tuberculosis detected than Tunisia (96%), e.g. Chile (114%), Panama (133%) and Costa Rica (153%). They also export more than Tunisia (45 % of its GDP in 2004), e.g. Panama (63%) and Costa Rica (46%). In contrast, Indonesia on the left has a much lower export rate of only 31%. The vertical axis (W2 in Fig. 2c) represents the place of women in the economy measured by the ratio of male/female enrolled in the tertiary education, the female economic activity rate

in 2004 and the evolution of the female economic activity rate from 1990 to 2004. Considering the rate of female activity in 2004 and the evolution from 1990 to 2004, in the embedding below Tunisia (with 27.9% and 37%), we see Morocco (26.7% and 33%) and Egypt (20.1% and 28%). Above it, we see Dominican Republic (45.5% and 55%) and Uruguay (55.7% and 71%).

It should be noted that some local regions cannot be explained linearly. For instance, BIR does not found any solution for local regions around Denmark and Lithuania. This can be due to the fact that (i) t-SNE makes mistakes in its projection (i.e. it does not make sense to explain it) and (ii) the mapping can be highly non-linear, so much that it is impossible to explain the region linearly.

6 Conclusion

The main contribution of this paper is to adapt LIME, a method designed to explain any predictive model, in order to explain t-SNE embeddings. First, an oversampling method based on SMOTE is used to generate m relevant samples in HD for a selected instance of interest. Second, only the positions of these m newly created samples are computed by t-SNE. Third, a sparse linear model is used to explain the local orthogonal trends around the selected instance. In future works, the proposed approach will be extended to explain embeddings of image and text datasets. The complexity $O((m+n)^2)$ of the out-of-sample projection can also be improved, e.g. with interpolation [12]. Work can also be done to show where local models are relevant in the visualization.

References

[1] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008.

[2] M. Wattenberg et al. How to use t-SNE effectively. *Distill*, 1(10):e2, 2016.

[3] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proc. of SIGKDD*, pages 1135–1144, 2016.

[4] A. Gisbrecht, B. Mokbel, and B. Hammer. Linear basis-function t-SNE for fast nonlinear dimensionality reduction. In *Proc. of IJCNN*, pages 1–8, 2012.

[5] A. Gisbrecht, A. Schulz, and B. Hammer. Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing*, 147:71–82, 2015.

[6] K. Bunte, M. Biehl, and B. Hammer. A general framework for dimensionality-reducing data visualization mapping. *Neural Computation*, 24(3):771–804, 2012.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *JAIR*, 16:321–357, 2002.

[8] L. Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Proc. of AISTATS*, pages 384–391, 2009.

[9] A. Bibal, R. Marion, and B. Frénay. Finding the most interpretable MDS rotation for sparse linear models based on external features. In *Proc. of ESANN*, pages 537–542, 2018.

[10] R. Marion, A. Bibal, and B. Frénay. BIR: a method for selecting the best interpretable multidimensional scaling rotation using external variables. *Neurocomputing*, 342:83–96, 2019.

[11] United Nations Development Program. Human development report, 2006.

[12] Z. Yang, J. Peltonen, and S. Kaski. Scalable optimization of neighbor embedding for visualization. In *Proc. of ICML*, pages 127–135, 2013.