XMAP: eXplainable mapping analytical process

Su Nguyen¹ · Binh Tran¹

Received: 17 December 2020 / Accepted: 31 October 2021 / Published online: 22 November 2021
© The Author(s) 2021

Abstract

As the number of artificial intelligence (AI) applications increases rapidly and more people will be affected by AI's decisions, there are real needs for novel AI systems that can deliver both accuracy and explanations. To address these needs, this paper proposes a new approach called eXplainable Mapping Analytical Process (XMAP). Different from existing works in explainable AI, XMAP is highly modularised and the interpretability for each step can be easily obtained and visualised. A number of core algorithms are developed in XMAP to capture the distributions and topological structures of data, define contexts that emerged from data, and build effective representations for classification tasks. The experiments show that XMAP can provide useful and interpretable insights across analytical steps. For the binary classification task, its predictive performance is very competitive as compared to advanced machine learning algorithms in the literature. In some large datasets, XMAP can even outperform black-box algorithms without losing its interpretability.

Keywords Explainable artificial intelligence · Context-aware machine learning · Classification

Introduction

In the last decade, artificial intelligence (AI) has risen as a key factor for success in the big data era [32,48]. With the recent advances in the field, AI has been widely adopted in various application domains from medicine [59], energy [12], to supply chain management [7]. For certain cognitive tasks such as image classification or diagnostic, AI-based tools have shown to be more powerful as compared to conventional solutions, and gradually outmatched human performance [44]. These achievements have motivated researchers and practitioners to improve and extend AI technologies in order to solve more complex problems such as self-driving cars, digital marketing, and medical diagnosis. To cope with these challenging problems, more sophisticated techniques such as reinforcement learning [55], transfer learning [34], and generative modelling [20] must be considered in modern AI. While the preliminary outcomes are promising, the increasing levels of sophistication and automation of AI applications have raised a number of concerns related to trust, reliability, and fairness [17]. Given that AI or the decisions made by AI will impact more and more users, it is more important

than ever to understand why AI makes a decision and how a decision is made. The first question may come from human curiosity or the genuine need of the users to solve their problems (e.g., the doctors from intensive care units must know what factors can increase the risk of mortality to come up with prevention or treatment plans). Meanwhile, answering the second question is useful to gain social acceptance, detect biases, and serve as a means to debug and interact with the AI systems. eXplainable AI (XAI) [1] is an emerging field that focuses on building AI systems that are understandable or interpretable by human users through the use of effective explanation. Interpretability and explainability are desirable properties of an AI system. However, both terms do not have a consensus definition. While interpretability is often referred to as algorithmic transparency, decomposability, or simulatability, explainability refers to the ability to explain the system decisions, which can be achieved in different ways such as model simplification or model translation [5]. XAI has been studied extensively since the early years of AI to address the above issues. Interpretable machine learning (ML) and people-centric AI are the current trend in the AI and ML research community to gain the values of these technologies by interpreting the representations learned and decisions made by these AI/ML models, and building new technologies to enhance human interaction with AI/ML.

✉ Binh Tran
b.tran@latrobe.edu.au

¹ Centre for Data Analytics and Cognition, La Trobe University, Melbourne, Australia

There is an increasing number of studies on XAI in the last decade [43]. The algorithms or approaches proposed in these studies can be categorised by (1) how the interpretability is achieved, (2) what outcomes are expected, or (3) their scope/limitations [45]. The first categorisation is based on whether the interpretability is obtained by controlling the complexity of the ML models or by dissecting the ML models (usually complex such as deep neural networks) after training. The second categorisation focuses more on the forms of interpretability as the outputs of XAI algorithms. For instance, weights or structure of trained models (e.g., linear regression, decision trees) are the outputs of interpretable ML while feature contributions such as LIME [51], SHAP [36] are explainers for complex ML models (e.g., from random forest, XGBoost). The third category looks at the scope of interpretability which classifies XAI algorithms based on their comprehensiveness, i.e., global vs local interpretability.

Most existing studies on XAI focused on algorithmic aspects to improve the interpretability either by (1) optimising the structures of the models to achieve a balance between interpretability and accuracy [60,62], or (2) developing explainers or surrogate models for the complex ML models [36,51,58]. These two approaches have their own advantages and disadvantages. While the first approach can produce interpretable models, it usually relies on a number of assumptions about the input data and the curse of dimensionality, which may restrict its applications in complex problem domains. A good representation learned from the input data is needed to overcome this limitation. Disentangle representation is a promising solution for this problem as it can provide interpretable high-level features to the ML algorithms. Meanwhile, the second approach is more suitable for the cases in which high prediction accuracy is important, the problem is highly complex, and some sacrifices in terms of interpretability is acceptable.

This paper attempts to overcome the limitation of the first approach by proposing eXplainable Mapping Analytical Process (XMAP). The novelty of this paper is the development of a new interpretable representation based on a combination of dimensional reduction and topological data analysis techniques. The new representation can provide the interpretability for the contexts in which data is generated and can be used to enhance the performance of interpretable ML algorithms. In this paper, XMAP is treated as a process rather than an algorithm as it presents systematic steps to explore data, capture patterns that emerged from the data, and perform prediction. XMAP can be used in both unsupervised (if the labels are not available) and supervised learning tasks. Also, data visualisation is incorporated in each step of XMAP, which allows the users to conveniently examine and validate the outputs from each step of the analytical process, rather than only the prediction step. The main contributions of this paper are:

- A new unsupervised learning algorithm to capture the topological relations of the input data
- A new and efficient technique to extract and interpret emerged patterns
- A new interpretable representation of high-dimensional data for binary classification tasks

The rest of this paper is organised as follows. In “Related works”, a brief review of related work in interpretable ML, dimensional reduction, learning representation, and context-aware machine learning is presented. “eXplainable mapping analytical process” shows how XMAP works and its key components. “Datasets and experimental settings” shows the datasets used in our experiments and the parameter settings of XMAP. Detailed results and analyses are presented in “Results and analyses”. Further discussions and conclusions are shown in “Further discussions” and “Conclusions”.

Related works

This section briefly reviews methods used to achieve interpretability in XAI such as interpretable ML, black-box AI, and visualisation. We also discuss some studies related to context-aware ML and its link to our work. It is noted that this section is not intended to provide a comprehensive review of XAI in the literature. For a comprehensive discussion of explainability, interpretability, and XAI algorithms, the readers can refer to [43,45].

Interpretable machine learning

The literature has presented many definitions of explanation, and it is still a debatable topic. From the practical perspective, researchers are more interested understand what makes a “good” explanation. Miller [42] provided a summary of good explanations, including (1) explanations are contrastive, (2) explanations are selected, (3) explanations are social, (4) explanations focus on the abnormal, (5) explanations are truthful, (6) explanations are consistent with prior beliefs of the explainees, and (7) explanations are general and probable. Machine learning techniques considered explainable or interpretable usually need to produce models that satisfy some of these criteria.

Interpretable ML (IML) includes a set of models and algorithms which are interpretable by design. Representatives of this class are linear/logistic regression, decision trees, Bayesian rule list, RiskSlim. Linear regression and logistic regression [21] relies on the weighted summation of feature inputs to provide a prediction. Because of the linear relationship, it is fairly easy to interpret the learned models when the number of features is small. As the number of features

increases, regularised versions of these techniques are needed to restrict the complexity of the final model. Lasso is one of the most popular techniques to train sparse linear models, i.e., perform feature selection to improve generalisation and interpretability of the trained models. Recently, more advanced techniques aiming at controlling the complexity and enhancing the interpretability at the feature level were also proposed. For example, RiskSlim [60] was developed for the optimised risk score problem in which the number of features and the coefficients of features are explicitly constrained.

Decision tree (DT) [21] is another popular interpretable ML technique. DT can capture the interactions between features better as compared to linear models. DT is reasonably easy to interpret and has a number of useful tricks to measure feature importance or to deal with missing data. However, DT often has trouble with linear relationships or continuous variables. In those cases, DT can be unstable or produce an unintuitive prediction.

If-Then rules are one of the most acceptable solutions if the learned rules provide a decent accuracy. Most rule learning algorithms depend on some heuristics to extract frequent patterns or effective rules. Letham et al. [31] proposed the Bayesian Rule List algorithm that shows promising results. In the first stage, it uses a frequent set mining technique to identify patterns with strong support from the dataset. Then, the selected patterns are used to build the If-Then rules based on Bayesian statistics. Apart from their interpretability, another advantage of If-Then rules is that they can provide fast predictions and are easy to implement. Fuzzy rule-based systems which combine if-then rules with fuzzy sets are also part of this family. These systems allow verbally formulated rules to be defined for imprecise domains, creating readable and understandable models. However, designing interpretable fuzzy models is still challenging because fuzzy rules are defined using linguistic terms drawn from natural language [41]. In addition, similar to DT, learnt rules also have difficulty dealing with linear relationships.

RuleFit [19] tried to combine the strength of both DT and linear models. RuleFit first extracts decision rules from decision trees and uses them to generate high-level features. Then RuleFit uses Lasso [21] to train a sparse linear model based on both original features and new features (i.e., rules extracted from ensemble trees). The advantage of RuleFit is its ability to add feature interactions to linear models. However, there is no systematic guideline to generate and select rules in the first step. As a result, there is a high chance that RuleFit may end up with a lot of rules, which may be redundant or useless.

Interpretability of black-box algorithms

While the models obtained from the IML techniques discussed in the previous section have some nice properties to support interpretability, they are usually criticised for their

predictive performance. On the other hand, black-box algorithms have recently received a lot of attention due to their exceptional performance, even though their interpretability is very limited. As a result, researchers have made a lot of efforts to improve the interpretability of these black-box algorithms either by enforcing a structure to the representation of the ML models or developing explainers (i.e., model-agnostic methods) or surrogate models to help the users make sense of the predictive results.

Recently many methods have been proposed to create summaries of features (e.g., evaluate the feature importance) based on the learned ML models, and they are widely applied in different applications. The main advantage of these methods is their flexibility, allowing them to cope with different complex ML algorithms and representations. The existing methods range from calculating the local effects of each feature, permuting the feature values, or building a surrogate model (using IML techniques). Interpretability methods such as local interpretable model-agnostic explanations (LIME) [51] and SHapley Additive exPlanations (SHAP) [36] have been used successfully to explain complex models such as XGBoost and deep neural network (DNN).

Also belonging to the model-agnostic methods are example-based explanations. This group of methods focus on selecting examples or instances to explain the behaviour of machine learning methods. They are especially useful when the instances can be presented in a meaningful way, such as images. Depending on the explanation goal, the instances can be selected to support counterfactual explanations (i.e., find a counterfactual instance is closest to the instance of interest) [61] or to serve as the prototypes (or criticisms) representing the distribution of (or representing instances which prototypes have not well represented) input data [27].

Interpreting DNN is one of the most popular areas in XAI. In the image domain, special structures of DNN such as convolution layers can be used to explain how abstract features are extracted and which features contribute to the prediction. Some studies also focus on bringing structures into the latent representation of DNN. For example, InfoGAN [9] uses mutual information to help DNN learns disentangle (interpretable) representations. Many researchers have also tried to improve the interpretability of deep autoencoder by forcing the algorithms to learn sparse latent representations [10,38].

As IML and black-box algorithms have their strengths and weaknesses, combining and integrating both in a system have gained much attention from XAI researchers in the last decades [5].

Visualisation and mapping

Visualisation has long been an effective approach to interpret ML models. Even for interpretable ML methods such

as linear regression and decision trees, visualisation can still be useful to highlight what impact the prediction outputs. Most well-established visualisation software allows the users to visualise the outputs of ML models as a mean to understand how they work and communicate the model outputs. Although visualisation is commonly used for supervised learning tasks, some promising works have been done for unsupervised learning. For example, many researchers have been interested in visual clustering, i.e., using visualisation to explain the clustering results [57].

Mapping from high-dimensional space to lower-dimensional space is also a powerful approach to explore complex datasets. In this paper, we refer to mapping as a set of methods that perform both dimension reduction and topological preservation. Principal component analysis (PCA) [21] and multidimensional scaling (MDS) [13] are good examples of mapping techniques in which high-dimensional data can be transformed into few-dimensional space which can be used for supervised learning or visualisation. Many mapping methods have their roots in topological data analysis (TDA) such as MAPPER [56], t-SNE [37], and UMAP [40] or unsupervised artificial neural network such as self-organizing map (SOM) [29], and Self-Organizing Incremental Neural Network (SOINN) [54]. The strength of mapping methods is their ability to capture both the local relationships and global structure of the datasets. This property makes mapping a suitable method for a number of interpretability approaches such as counterfactual explanations or prototypes and criticisms. However, mapping methods are mainly treated as dimensional reductions or visualisation, and their interpretability has not been fully explored.

Context-aware machine learning

In real-world applications, understanding contexts, i.e., the situations in which data are generated, plays an important role in prediction tasks. For example, contextual features such as seasonality can be used to construct better models for predicting air quality [47]. In human activity recognition, a priori knowledge and context information such as person location, scene description can be incorporated to guide the recognition process [49]. In previous studies, context information can be used to learn independent ML models [47] or can be treated as contextual features [35]. Contexts are usually known as a priori or defined by users. However, some researchers also tried to discover contextual features during the learning process automatically. A good example of this approach is the grammar guided genetic programming (G3P) algorithm to mine context-aware association rules [35]. In their algorithm, a feature is defined as a contextual feature if it does not correlate with a target feature, but it is useful when using it together with other predictive features. Based on that definition, the authors proposed an evaluation process

to determine whether a rule generated by G3P comprises contextual features.

Contexts are also useful to improve the interpretability of the ML models. Zou et al. [63] proposed the interactive contextual interaction explanation (ICIE) framework that allows users to view explanations of each instance under different contexts. In ICIE, a context can be defined as a set of constraints to describe a subspace of the feature space. In this case, the users can define the contexts using an interactive user interface. The demonstrations on a number of datasets showed that the use of contexts makes local feature interactions visible and helps inspect wrong predictions of a classifier.

Previous studies show a wide range of approaches to achieve interpretability in ML; each has its own strengths and weaknesses. As the AI/ML outputs will be used by a diverse group of users with different backgrounds and interests, it is desirable that the AI/ML systems can provide the users with some flexibility to obtain interpretability. Running multiple interpretability methods is a simple solution, but there is a risk that they may result in conflicting explanations. Meanwhile, combining existing methods at an algorithmic level is difficult, if not impossible, as they are developed based on different paradigms. In this paper, we propose XMAP, a process-based approach, to cope with this issue. Within XMAP, a new mapping algorithm and a new interpretable representation are developed to enhance the interpretability and the predictive performance. In addition, interactive data visualisation is attached to each step to facilitate the explanation process.

eXplainable mapping analytical process

Figure 1 shows an overview of XMAP. The inputs for XMAP is a dataset $\mathcal{D}$ with N instances. Each instance includes a feature vector $x_i = (x_{i,1}, \dots, x_{i,d})$ where d is the number of features, and optionally a label y_i (e.g., class for classification task or a target value for a regression task). In this paper, we only focus on binary features. Different from existing interpretable ML algorithms in the literature, XMAP is highly modularised, and the users can review the outputs of each step via the user interface. XMAP starts with four fundamental steps:

- *Pre-processing raw data*: with discretisation, scaling, normalising, and handling missing data.
- *Applying mapping techniques*: to transform the pre-processed data from high dimensional space to low dimensional space. The output of this step is the transformed data that preserves the topological structure of the raw dataset.

Fig. 1 Overview of XMAP

- *Learning topology*: to capture the topological relationships and distributions of the transformed data. The outputs of this step is an abstract representation of the transformed data, or indirectly the raw dataset.
- *Extracting interpretable contexts*: to identify the interpretable contexts emerging from the input data by analysing the topological representation. The outputs of this step are context descriptions to determine the contexts that cover each instance.

Depending on the predictive tasks and the availability of labelled data, XMAP can flexibly utilise the contexts obtained from the previous steps. If the labels are available, XMAP can be used to improve the prediction performance of the task. Theoretically, although XMAP can be used for both regression and classification, we only focus on classification tasks in this paper to avoid confusion when discussing technical details. For classification tasks, the following steps can be performed:

- *Determining the interpretable representation*: to determine contextual feature vectors based on the context descriptions obtained from the previous steps.
- *Estimating problem difficulty*: to determine how difficult it is to discriminate between different classes.
- *Learning interpretable ML models*: to learn interpretable ML models based on the newly discovered contextual features and the original features.
- *Analysing prediction outputs and interpretability*: to calculate and explain prediction outputs to the users.

If the labels are not available, the outputs from the fundamental steps can still be utilised to identify the clusters from the input data or to detect outliers/anomalies in the dataset. In the upcoming sections, the details of each step will be presented. To facilitate the discussion, we will use the IBM-HR Employee Attrition dataset [24] with 1470 instances and 121 (discretised) features (e.g., age, business travel, daily rate, department in company) to illustrate the outputs of each step.

Mapping

For the mapping step, we use the Uniform Manifold Approximation and Projection (UMAP) algorithm, a recently proposed technique for dimension reduction [40]. UMAP is a graph-based technique (similar to t-SNE [37] or Isomap [11]) to obtain a low dimensional representation of the raw dataset. Compared to the conventional PCA algorithm, UMAP can better characterise the data distribution as it does not follow the linearity assumptions. Compared to other graph-based dimension reduction algorithms such as Isomap and t-SNE, UMAP is more robust and has better scalability [40]. Another advantage of UMAP is that its parameters are intuitive and easy to select. When experimenting with single-cell data, Becht et al. [2] showed that UMAP performs exceptionally well across all investigated aspects from preserving the global structure, robustness, and running times as compared to PCA (with different numbers of principal components), t-SNE, Fit-SNE [33], and Autoencoder SCVIS [16].

However, UMAP is mainly used for dimensional reduction and visualisation, and its interpretability has not been

Fig. 2 IBM HR dataset embedded in two-dimensional space. The X-axis and Y-axis are respectively the first and second embedding dimension. Red points are for instances where Attrition is True (Class #1), and blue points are for instances where Attrition is False (Class #0)

fully explored. By combining UMAP with the proposed topological learning (ATL) and context description approximation (CDA) described in Sect. 3.3, XMAP reveals the meaning of the data clusters shown in the low-dimensional UMAP embedding, providing users with interpretable insights about the data. Furthermore, with a novel context description approximation method, XMAP connects the mapping methods with the latter stage of supervised learning in a transparent manner, as shown in 3.4, which has never been investigated in the literature. By utilising the extracted context knowledge, XMAP can enhance the performance of the prediction method.

The UMAP algorithm is used in XMAP to transform the raw data $X \in \mathbb{R}^{N \times d}$ into low dimensional space $Z \in \mathbb{R}^{N \times d'}$ where $d' \ll d$. In this study, we fixed $d' = 2$ to visualise the results conveniently. The map generated for the IBM-HR dataset is shown in Fig. 2. Based on this map, we can visually inspect the data distribution and identify where attrition (red points) is likely to occur. In this case, it seems that more attrition occurs in the right part of the map.

Topological learning

The map obtained by UMAP is an excellent visualisation tool to establish a general understanding of the dataset. However, it is impossible to examine each point/instance to understand why it ends up in that position. Moreover, although we can recognise certain clusters in the map, those are not easily picked up because of noisy points. Therefore, a high-level summarisation is important to avoid information overloading. Ideally, we would like to capture both a set of prototypes and their topology structure of the distribution of transformed data Z . To achieve the above goal, we propose Adaptive

Algorithm 1 Adaptive Topological Learning (ATL)

Input: transformed data $Z = \{z_1, \dots, z_N\}$
Output: ATL network $\mathcal{N}$

- 1: set $epoch \leftarrow 0, t \leftarrow 0, a_{max} \leftarrow \infty$
- 2: randomly initialise $\mathcal{N}$ with two nodes
- 3: **repeat**
- 4: randomly shuffle Z
- 5: **for** each input $z \in Z$ **do**
- 6: calculate the distance $d_{z,i}$ between z and node i
- 7: identify the nearest (winning) node s_1 and the second-nearest node s_2 for input z , and their respective weights w_{s_1} and w_{s_2}
- 8: update the similarity thresholds for T_{s_1} and T_{s_2}
- 9: **if** $\|z - w_{s_1}\| > T_{s_1}$ or $\|z - w_{s_2}\| > T_{s_2}$ **then**
- 10: insert a new node s into $\mathcal{N}$ with $w_s \leftarrow z$
- 11: create an edge between s_1 and s_2 and set its age to zero if the edge does not exist; otherwise set the edge's age to zero
- 12: $NB_{s_1} \leftarrow$ get the neighbours of s_1
- 13: $\bar{d}_{s_1} \leftarrow \left(\sum_{n \in NB_{s_1}} d_{z,n} \right) / |NB_{s_1}|$
- 14: **for** each $n \in NB_{s_1}$ **do**
- 15: $age_n \leftarrow age_n + 1 + |NB_n| \frac{d_{z,n}}{d_{s_1}}$
- 16: updating node s_1 : $w_{s_1} \leftarrow w_{s_1} + \epsilon_1(z - w_{s_1})$
- 17: updating neighbour n of s_1 : $w_n \leftarrow w_n + \epsilon_2(z - w_n)$
- 18: removing edges with age larger than age_{max} and nodes with no emanating edges
- 19: **if** t is an integer multiple of parameter λ **then**
- 20: delete nodes with no neighbour or only one neighbour
- 21: $age_{max} \leftarrow (\lambda + |\mathcal{N}|)/2$
- 22: $t \leftarrow t + 1$
- 23: **until** $epoch = maximum_epoch$

Topological Learning (ATL) algorithm, a variant of adjusted self-organising incremental neural network (ASOINN) [53]. The pseudo-code of ATL is presented in Algorithm 1.

In this algorithm, the similarity thresholds are used as the condition to decide when the network needs to grow. The thresholds can be calculated as follows:

Fig. 3 Topological structure of IBM-HR dataset

- If the node i has a non-empty set of neighbours NB_i :

$$T_i = \max_{c \in NB_i} ||w_i - w_c|| \quad (1)$$

- If node i has no neighbour:

$$T_i = \min_{c \in \mathcal{N} \setminus \{i\}} ||w_i - w_c|| \quad (2)$$

The adaptive learning rates ϵ_1 and ϵ_2 for a node i can be calculated as follows:

$$\epsilon_1 = \frac{1}{M_i} \quad (3)$$

$$\epsilon_2 = \frac{1}{100M_i} \quad (4)$$

where M_i is the times for node i to be the winner.

As compared to ASOINN and SOINN, ATL adopts a new adaptive heuristic to update the edges between nodes in $\mathcal{N}$. First, besides increasing the ages of all edges emanating from s_1 by 1, ATL also adds the adaptive term $|NB_n| \frac{d_{z,n}}{d_{s_1}}$ (line 15). By using this new updating rule, the edge connecting s_1 and a node n with more neighbours or with a further distance $d_{z,n}$ will age faster. This trick will help to reduce the complexity of the learned network by eliminating useless edges and nodes. Furthermore, we make age_{max} adaptive to the need to represent the distribution and topological relations of input data (line 21). As a result, ATL only relies on one user-designed parameter λ .

Because the output of this step is a network, we can easily identify the network communities, i.e., clusters of nodes, which help us determine the clusters of the transformed (embedding) data and the raw data. In this research, we use

the Louvain algorithm [3] to determine the partition of the graph nodes, which maximises the modularity. The community detection and clustering results are shown in Fig. 3 and Fig. 4, respectively. The advantage of this approach is that the user does not need to explicitly define the number of clusters (such as in k-means [21]).

Extracting interpretable contexts

The clusters $\mathcal{C} = \{C_1, \dots, C_K\}$ obtained in the previous step include instances with similar patterns, which we refer to as contexts in this paper. Different from the contexts defined in the supervised learning studies [35] which define contexts or contextual features based on their impact on the importance of other predictive features, contexts in this research are defined in a purely unsupervised manner. Similar to [63], we define a context T_k as a set of constraints to describe a subspace of the feature space. Based on those constraints, we can decide whether an instance is covered by this context or not. For example, the context description for the IBM-HR dataset can be $T_k = \{[\text{Sales_Department}] = \text{True}, [\text{Years_Since_Last_Promotion} < 3] = \text{True}\}$. By this description, any employee (instance) x_i that works in the Sales Department and has a promotion in the last 3 years will be covered by context T_k .

Ideally, we would like to have a context description T_k that exactly covers all instances $x_i \in C_k$. This task is equivalent to a binary classification task (to classify whether an instance is covered by a context) in which we need to make a trade-off between precision and recall. It is likely that a short description will have a high recall but a low precision, and vice versa. Building such a classifier is as hard as building the classifier for the original problem. Fortunately, we do not need to determine perfect context descriptions as they can be too complex

Fig. 4 Learned clusters of IBM-HR dataset

Algorithm 2 Context Description Approximation (CDA)

Input: $x_i \in C_k$, max description size S , threshold θ
Output: context description T_k

```

1:  $T_k \leftarrow \{\}$ 
2: for  $j \in \{1, \dots, d\}$  do
3:    $\text{impurity}_j \leftarrow \min \left\{ \frac{\sum_{i \in C_k} x_{i,j}}{|C_k|}, \frac{1 - \sum_{i \in C_k} x_{i,j}}{|C_k|} \right\}$ 
4:  $\text{sorted\_features} \leftarrow$  sort features in the ascending order of  $\text{impurity}_j$ 
5: for  $j \in \text{sorted\_features}$  do
6:   if  $\text{impurity}_j \leq \theta$  then
7:     if value with the highest frequency in  $x_{\cdot,j} = 1$  then
8:        $T_k \leftarrow T_k \cup \{x_{\cdot,j} = \text{True}\}$ 
9:     else
10:       $T_k \leftarrow T_k \cup \{x_{\cdot,j} = \text{False}\}$ 
11:   if  $|T_k| \geq S$  then
12:     break

```

and prevent us from interpreting the contexts. Moreover, the fact that we try to find the description for instances in the same cluster makes the problem easier. As all instances in the same cluster should have some features with similar values, we can easily come up with a decent approximation of a context description T_k by identifying features with dominating values in C_k . Algorithm 2 shows the context description approximation (CDA) algorithm to approximate the context description T_k (noted that all features are binary).

Figure 5 shows the data matching the description of context #2 (based on data in cluster #2). In this example, the description size $S = 5$ and the threshold $\theta = 0.1$. Applying the Algorithm 2, we obtained the context description $T_2 = \{[\text{Sales_Department}] = \text{True}, [\text{Is_Sales_Executive}] = \text{True}, [\text{Job_Level_3}] = \text{False}\}$. As compared to the full description in Fig. 5, we have removed the last two conditions in the description because they are

redundant (as an employee cannot be in the Sales Department and holds research positions). Such redundancy can be easily eliminated as we know what features are mutually exclusive based on the encoding rules used in the pre-processing step. It is also noted that T_2 covers both instances in cluster #2 and cluster #7 rather than just cluster #2. This is due to the approximation nature of Algorithm 2. Figure 6 shows the data points that match the description of context #7, which nicely covers the instances in cluster #7. As can be seen in these examples, one instance can be assigned to more than one context. However, as the goal here is to identify the interpretable contexts, it is more desirable if instances can be assigned to a few interpretable contexts rather than a complex and uninterpretable context.

Interpretable prediction

By using the context description set $\mathcal{T} = \{T_1, \dots, T_K\}$, we can easily determine a binary contextual feature vector $Z_i^C = \{t_{i,1}, \dots, t_{i,K}\}$ for each instance i . If $t_{i,k} = 1$, it means that instance i is covered by the context description T_k . Because Z_i^C is an interpretable latent representation of instance i , we can use it to enhance the predictive performance of supervised learning algorithms. In this paper, we propose two ways to utilise the contextual feature vectors:

- *Merging representation*: by merging the original features and the contextual features to create a new training set $X' = [X \ Z^C]$ where $Z^C \in \mathbb{R}^{N \times K}$ is the contextual feature matrix.
- *Context-aware representation*: by generating interactive terms between the original features and contextual features to create a new training set $X' = [X \ X_1^C \dots X_K^C]$

Fig. 5 Example of a context description T_2 for IBM-HR dataset

Fig. 6 Example of a context description T_7 for IBM-HR dataset

where $X_k^C \in \mathbb{R}^{N \times d}$ is the context-aware feature matrix $(t_{1,k}X_{1,\cdot} \ t_{2,k}X_{2,\cdot} \ \dots \ t_{N,k}X_{N,\cdot})^T$.

The merging representation attempts to use contextual features to predict the outputs directly. Meanwhile, the context-aware representation tries to build specialised predictors for each context. Figure 7 illustrates how these two representations work for a classification task. The main advantage of using contextual features as compared to other latent representations (e.g., auto-encoders) is that the contextual features are interpretable, and the encoding process (i.e., from X to Z^C) can be done efficiently with the context description set $\mathcal{T}$. This property makes $\mathcal{T}$ portable and

easily incorporated directly into the existing database. When feeding X' into any IML algorithms, the interpretability of those algorithms is still preserved, and the contexts can help to enhance the predictive performance of IML algorithms. For complex algorithms, the interpretable representation can also help to improve the prediction performance, and model-agnostic interpretability methods can be used to determine the importance of contextual features.

This section describes the main steps in XMAP and shows how the outputs of each step can provide different insights into the datasets. Depending on the availability of the data and the analysis requirements, the users can focus on some specific tasks in XMAP rather than the entire process. Because

Fig. 7 Representations for interpretable prediction

of this advantage, XMAP is useful for multiple stages of data science or ML projects. To demonstrate various applications of XMAP, we will conduct experiments using a set of well-known benchmark datasets.

Datasets and experimental settings

This section describes the datasets and the parameters settings for XMAP used in our experiments.

Datasets

Table 1 shows 12 datasets used in our experiments with different numbers of examples and numbers of features. These datasets are selected as they have been widely used in the literature for both conventional ML algorithms and IML. As noted in the table, the number of features d reported here are those after the discretisation step. For continuous features such as monthly income in the IBM-HR dataset or the customer age in the German Credit Risk dataset, we apply k -bins discretiser and one-hot encoding [4] with $k = 4$.

Parameter settings and performance metrics

Parameters settings for algorithms used in XMAP are presented in Table 2. For UMAP, we used the default parameters, which produce robust performance across different datasets. For ATL, we use $\lambda = 200$, as it gives a good balance between speed and performance. For CDA, we do not see a big difference when changing the description size; therefore, we keep the size $S = 5$ for better explanations. The very small $\theta = 0.01$ is selected to ensure that the context description can efficiently cover the corresponding cluster. In the interpretable prediction step, we only focus on logistic regression with $L1$ regularisation as it is fairly easy to interpret. We also compare the prediction performance of XMAP with other popular algorithms in the literature, such as decision trees (DT), artificial neural network (ANN) [21], and XGBoost [8]. The parameters settings for DT, ANN, and XGBoost in Table 2 are selected as these parameters provide a good generalisation in most datasets investigated in this paper based on our pilot experiments. To compare

the performance in the binary classification task, we use the area under the curve (AUC) and the accuracy (ACC) with the ten-fold cross-validation. XMAP is implemented as an interactive and visualisation website using Python.

Results and analyses

This section shows the results of XMAP in each main step. For the first fundamental steps, we will use qualitative evaluations by visually examining the outputs of XMAP. For the classification tasks, we will compare XMAP classifiers with DT, ANN, and XGBoost.

Mapping outputs and problem difficulty

As the performance of UMAP, compared to other dimensional reduction techniques, has been well investigated in the previous studies [2], we will not discuss their performance in this section. Instead, we will emphasise the correlation between the mapping outputs and problem difficulty. Figure 8 shows the mapping outputs and the AUC (based on logistic regression) obtained for six out of the twelve datasets. German Credit Risk (GC) and Australian Credit Approval (AC) are two popular datasets in the credit risk literature, but we can see that the distribution of default instances on the maps in these two datasets are different. While default instances of GC are distributed across the map, those of AC are mainly located in the lower part of their map. Intuitively, we can anticipate that it is much harder to build a good classifier for GC, as compared to AC. The empirical results show that it is, in fact, true. We can easily see that the AUC for AC is much higher than AUC for GC given that the same classification algorithm is used. For Bank and Adult data, we see that default instances are highly concentrated in a part of the map, which makes the classification slightly easier. However, because instances from the two classes are not separable in those clusters, it is hard to achieve perfect prediction performance for these two datasets. For the last two datasets, i.e., Breast Cancer and Mushroom, there are clear separations between the classes. It means that the features used in these two datasets have strong discrimination power.

Table 1 Datasets used in the experiments

Dataset	N	d	Conditions for $y_i = 1$	$Pr(y_i = 1)$	Ref.
Heart Disease	303	42	Patient has heart disease	54.50%	[15]
Breast Cancer	683	36	Diagnosis is malignant	35.00%	[39]
Australian Credit	690	56	Credit is approved	44.50%	[50]
Mammo	961	14	Person has breast cancer	46.30%	[18]
German Credit Risk	1000	77	Customer has bad credit	30.00%	[26]
Human Resource-IBM	1470	121	Employee attrites	16.10%	[24]
Spambase	4601	228	e-mail is spam	39.40%	[14]
Customer Churn	7032	49	Customer churns	26.60%	[25]
Mushroom	8124	113	Mushroom is poisonous	48.20%	[52]
Intensive Care Unit	11,773	53	Patient is dead	10.70%	[23]
Adult	32,561	36	Person in 1994 US census earns over \$50	24.10%	[28]
Bank	41,188	57	Person opens bank account after marketing call	11.30%	[46]

Table 2 Parameter settings

Step	Algorithm	Parameters	Value
Mapping	UMAP	Neighbour size	15
		Negative sample rate	5
		Distance metric	Euclidean
Topological Learning	ATL	λ	200
Extracting interpretable contexts	CDA	Description size	5
		θ	0.01
Interpretable prediction	Logistic Regression (LR)	Inverse regularization	0.1
Other ML algorithms	Decision Tree (DT)	Max depth	5
		Min split	2
	Artificial Neural Network (ANN)	Hidden layer size	15
		Max iteration	100
		Number of estimators	100

The AUC values show that the classifiers learned for the two datasets achieve almost perfect prediction performance. Based on these results, it is safe to say that mapping is a useful technique to help us estimate the problem difficulty.

Topological structures and clusters

Figure 9 shows the topological structures and clusters obtained by XMAP. The figure shows that ATL algorithm has successfully captured the topological structure from the complex data distributions shown in Fig. 8. Compared to the maps obtained by UMAP, the topological structure captured by ATL provides a much cleaner overview of the dataset. The clusters are also effectively captured via the ATL networks, even for complex datasets. For example, it is not straightforward to determine the clusters based on the map of the Bank dataset in Fig. 8c (even by visually inspecting the map) as this dataset shows a complex distribution. Existing clus-

tering algorithms such as k-means [21] and HDBSCAN [6] cannot capture these clusters effectively. However, as seen in Fig. 9c, interesting clusters of the Bank dataset are effectively captured by XMAP. Another good example to demonstrate the clustering effectiveness of XMAP is the Adult dataset. For this dataset, it is challenging to identify the appropriate clusters (or the number of clusters) because of noises. In this case, XMAP simultaneously identifies main clusters in the centre of the map as well as the clusters of more isolated instances. Another advantage of XMAP clustering is that the hierarchical representation of clusters can be easily obtained since the connectivity of clusters is determined with the ATL networks.

Prediction performance

Tables 3 and 4 shows the prediction results, i.e., AUC, and ACC, of XMAP and other ML algorithms. XMAP-M-LR and

Fig. 8 Mapping outputs and prediction performance

XMAP-C-LR are the XMAP classifiers with merging representation and context-aware representation, respectively. The results are listed from small to large datasets. For each dataset in these tables, the top three algorithms are shaded, bold, and italicised, respectively. Regarding AUC, the most popular metric for binary classification [22], both XMAP versions perform well as compared to LR and DT. XMAP-M-LR is better than DT in all datasets and produces similar results to those of LR (using only the original features). Meanwhile, XMAP-C-LR is one of the most competitive algorithms in our comparison. XMAP-C-LR is in the top three algorithms in 10 out of 12 datasets. As expected, the two black-box

algorithms, ANN and XGBoost, produce the best AUCs in most datasets. However, XMAP-C-LR, as an IML algorithm, is very competitive as compared to ANN and XGBoost. Interestingly XMAP-C-LR even outperforms the black-box algorithms when the number of examples increases. XMAP-C-LR is the best algorithm in four out of the five largest datasets. It is also noted that the three largest datasets are also among the most unbalanced datasets. These results show that XMAP-C-LR is an effective and scalable algorithm to deal with challenging binary classification problems. Similar patterns are observed in Table 4, even though XMAP-C-LR is not as dominating as in Table 3. This observation suggests

Fig. 9 Topological structures and clusters

that XMAP-C-LR tries to learn more discriminating classifiers rather than just focusing on accuracy.

The fact that XMAP-C-LR is competitive to or better than ANN and XGBoost in the large datasets is encouraging because XMAP-C-LR, different from ANN and XGBoost, is interpretable. From the algorithm perspective, XMAP-C-LR is not too different from ANN as they both try to obtain some kinds of latent representations of the input data and use them to enhance classification performance. The main difference here is that XMAP-C-LR enforces some constraints/structures to its representation to achieve interpretability. Although the interpretable representation is

obtained via an approximation process, the results show that it is still useful. Also, the results in Table 3 shows that using contexts to adapt the weights of original features (i.e., with XMAP-C-LR) is better than using contexts as high-level features (i.e. with XMAP-M-LR). Also, it seems that XMAP-C-LR performs well when the clusters/contexts are easily captured from the mapping and topological learning steps. As observed in Figs. 8 and 9, clusters/contexts can be easily identified for the Bank, Adult, Breast Cancer, and Mushroom datasets, in which XMAP-C-LR shows good performance. It means that contexts, if properly supported by the data, can be useful for classification.

Table 3 AUC with 10-fold CV

Dataset	LR		XMAP-M-LR		XMAP-C-LR		DT		ANN		XGBoost	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Heart Disease	0.902	0.067	0.904	0.066	0.896	0.061	0.795	0.054	0.903	0.073	0.896	0.054
Breast Cancer	0.992	0.007	0.991	0.009	0.993	0.008	0.963	0.026	0.992	0.009	0.994	0.007
Australian Credit	0.910	0.039	0.916	0.041	0.915	0.039	0.872	0.045	0.917	0.050	0.919	0.048
Mammo	0.851	0.049	0.851	0.049	0.850	0.048	0.838	0.051	0.854	0.052	0.848	0.048
German Credit Risk	0.761	0.038	0.763	0.037	0.770	0.038	0.710	0.044	0.780	0.047	0.765	0.042
Human Resource-IBM	0.808	0.062	0.807	0.061	0.809	0.058	0.686	0.065	0.833	0.068	0.812	0.062
Spambase	0.947	0.008	0.949	0.007	0.949	0.008	0.876	0.015	0.958	0.007	0.953	0.007
Customer Churn	0.842	0.018	0.842	0.018	0.842	0.019	0.830	0.020	0.840	0.020	0.841	0.020
Mushroom	1.000	0.000	1.000	0.000	1.000	0.000	0.997	0.003	1.000	0.000	1.000	0.000
Intensive Care Unit	0.857	0.014	0.857	0.014	0.862	0.012	0.842	0.017	0.884	0.013	0.888	0.010
Adult	0.890	0.006	0.890	0.006	0.894	0.006	0.861	0.006	0.894	0.006	0.892	0.006
Bank	0.785	0.012	0.788	0.012	0.793	0.012	0.778	0.012	0.785	0.010	0.792	0.012

Table 4 ACC with 10-fold CV

Dataset	LR		XMAP-M-LR		XMAP-C-LR		DT		ANN		XGBoost	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Heart Disease	0.839	0.074	0.835	0.080	0.816	0.079	0.767	0.066	0.823	0.078	0.826	0.058
Breast Cancer	0.959	0.024	0.958	0.022	0.958	0.028	0.962	0.019	0.972	0.021	0.966	0.025
Australian Credit	0.855	0.059	0.855	0.059	0.849	0.057	0.845	0.049	0.845	0.060	0.870	0.049
Mammo	0.802	0.038	0.802	0.038	0.802	0.038	0.804	0.045	0.800	0.046	0.797	0.053
German Credit Risk	0.721	0.034	0.731	0.033	0.744	0.023	0.709	0.043	0.735	0.020	0.749	0.027
Human Resource-IBM	0.857	0.010	0.858	0.009	0.864	0.017	0.835	0.021	0.870	0.022	0.865	0.012
Spambase	0.885	0.013	0.886	0.013	0.887	0.013	0.833	0.011	0.903	0.013	0.893	0.011
Customer Churn	0.799	0.013	0.800	0.012	0.799	0.012	0.787	0.013	0.801	0.012	0.802	0.013
Mushroom	0.998	0.002	0.998	0.002	0.998	0.002	0.997	0.003	1.000	0.000	1.000	0.000
Intensive Care Unit	0.908	0.006	0.908	0.006	0.911	0.004	0.911	0.005	0.915	0.006	0.915	0.006
Adult	0.837	0.006	0.838	0.005	0.839	0.005	0.822	0.005	0.839	0.006	0.838	0.005
Bank	0.900	0.003	0.900	0.003	0.900	0.003	0.899	0.004	0.897	0.003	0.899	0.004

Interpretability of XMAP classifiers

Although XMAP-C-LR representation seems to be complex, as shown in Fig. 7b, it can be interpreted as easily as the logistic regression models. To further analyse the interpretability of XMAP-C-LR, we will reuse the HR-IBM dataset. Figure 10 shows the odds ratio for the LR and XMAP-C-LR classifiers. The odds ratio of a feature j is defined as the ratio of the odds when $x_{.,j} = 1$ and the odds when $x_{.,j} = 0$, which can be determined as $OR_j = \frac{odds[x_{.,j}=1]}{odds[x_{.,j}=0]} = \exp(\beta_j)$ where β_j is the weight of feature j from the logistic regression model, and $odds = \frac{Pr(y=1)}{1-Pr(y=1)}$. In general, it is easy to see that the odds ratios are consistent between LR and XMAP-C-LR. For both classifiers, features such as [Overtime] and [JobLevel = 1] increase the odds of attrition and features such as

[Department_Research_Development], [StockOptionLevel = 1], and [JobSatisfaction = 4] decrease the odds of attrition. For XMAP-C-LR, the odds ratio are adapted to the context. In Fig. 10b, we show the odds ratio of all features when instances in Context #9:

$$T_9 = \{ \begin{array}{l} [\text{JobLevel} = 1] = \text{True} \\ [1000 \leq \text{Monthly_Income} < 4330] = \text{True} \\ [\text{Years_Since_Last_Promotion} < 3] = \text{True} \\ [\text{Years_With_Current_Manager} < 4.3] = \text{True} \\ [\text{Years_In_Current_Role} < 4.5] = \text{True} \end{array} \} \quad (5)$$

Fig. 10 Odds ratios for the HR-IBM dataset

In this context, XMAP-C-LR increases the impact of overtime, stock option, job satisfaction, and incorporates a number of other features (shown in red in Fig. 10b) into the model, such as business travel frequency and marital status. Specifically, given that the context describes employees who are relatively new to the role/manager and with a low income, overtime can increase the odds of attrition by a factor of $2.68 \times 1.6 = 4.3$ rather than 3.2 with the standard logistic regression model. In addition, frequent travel patterns will increase the odds of attrition by the factor of 1.27. This example shows that XMAP-C-LR develops a more specialised classification model based on the matched context. It should be noted that contexts, obtained based on the unsupervised learning algorithms introduced in this paper, are not always useful for the classification task. When the contexts are not useful for the classification task, their existence will not affect the prediction performance since XMAP-C-LR with $L1$ regularisation will be able to exclude the context-aware features (i.e., in X_k^C) from the models.

As mentioned earlier in the paper, XMAP is a process rather than an algorithm to enable transparency across the analytical steps and encourage human involvement in the data quality control and analyses. The results from this section demonstrate that XMAP can produce various insights and predictive outputs from the datasets. Apart from its predictive

abilities, the outputs from each step of XMAP also facilitate visualisation, allowing users to conveniently explore their data and examine the predictive outputs. The contexts obtained by XMAP not only provide meaningful descriptions of the data, but also enhance the prediction performance via interpretable representation. Because XMAP is highly modularised, it can easily benefit from upcoming breakthroughs in AI and ML to enhance its effectiveness.

Further discussions

XMAP proposed in this paper can be related to a number of interpretability methods in the literature, especially in the classification step. The use of context features in XMAP-M-LR is similar to the high-level features used in the RuleFit algorithm. The main difference is that the context features or descriptions are obtained using the CDA algorithm (in an unsupervised manner) rather than rule extraction heuristics used in RuleFit. However, similar to RuleFit, XMAP-M-LR does not show a clear advantage in terms of prediction performance. From our experiments, contexts are more powerful when they are used to adapt the classifier behaviour (as in XMAP-C-LR) rather than when they are used as high-level

Table 5 Explanations for the Breast Cancer Dataset**Top-left cluster (malignant)**[Mitoses $\in$ [5.36, 10.0)] = False[UniformityOfCellShape $\in$ [1.0, 2.49)] = False[UniformityOfCellSize $\in$ [1.0, 2.46)] = False[BareNuclei $\in$ [1.0, 2.55)] = False**Top-right cluster**[Mitoses $\in$ [1.0, 2.26)] = True[BlandChromatin $\in$ [1.0, 3.24)] = True[ClumpThickness $\in$ [1.0, 3.25)] = False**Bottom-right cluster**[ClumpThickness $\in$ [1.0, 3.25)] = True[MarginalAdhesion $\in$ [1.0, 2.41)] = True[UniformityOfCellShape $\in$ [1.0, 2.49)] = True[BareNuclei $\in$ [1.0, 2.55)] = True[BlandChromatin $\in$ [1.0, 3.24)] = True

features together with the original features (as in XMAP-M-LR or RuleFit).

The outputs of mapping and topological learning steps themselves can directly be used to build an interpretable classifier if the instances for each class are well clustered. The Breast Cancer dataset, as shown in Fig. 8e is a good example to demonstrate how simple explanations can be derived. In this case, instances can be categorised into three big groups or clusters, in which most default instances are in the clusters on the right-hand side. The descriptions of contexts obtained by CDA (with some simplifications) on this dataset are shown in Table 5. Although the descriptions here are not sufficient to build a perfect classifier, they are straightforward to understand and have strong discrimination power.

XMAP outputs can potentially be used for prototype-criticism explanations. As ATL network can effectively capture both the distributions and topological structures of the datasets, identifying prototypes and criticisms can be done in an efficient and intuitive way. This is an interesting point that can be investigated in future research.

Conclusions

This paper presents XMAP, a novel approach to exploring and building predictive models for challenging high-dimensional datasets. By treating XMAP as a process rather than an algorithm, each analytical step can be made transparent, and its outputs can be easily examined or explored by the users. Within XMAP, a number of new algorithms have been proposed to efficiently capture distributions and topological structures of the high-dimensional data, identify clusters and interpretable contexts, and generate context-aware rep-

resentations for binary classification tasks. XMAP is tested on a wide range of datasets, and the experimental results show that XMAP successfully produces valuable insights across analytical steps. The mapping outputs and the topological structures provide the users with a convenient way to understand how the data are distributed and estimate the difficulty of the predictive tasks. Then these outputs can be used to define contexts and build an effective interpretable representation for the classifiers. In our experiments, XMAP classifiers are very competitive as compared to popular classifiers in the literature. When dealing with large datasets, XMAP classifiers perform even better than some black-box algorithms such as ANN and XGBoost. Further analyses show that these performance gains are obtained without deteriorating XMAP's interpretability.

Our future studies will further investigate a number of interesting points. First, although the mapping techniques used in this work is effective, their efficiency may decrease as the problem size and the number of features significantly increase. Incremental or distributed mapping algorithms are required to overcome this problem. Second, a more powerful algorithm to obtain accurate context descriptions is needed to enhance the effectiveness of the interpretable representations. Third, it would be interesting to explore how XMAP can be used for other difficult analytics tasks such as regression, anomaly detection, and reinforcement learning. Fourth, as the mapping outputs and the ATL networks can be used as an efficient way to capture the distributions of data, a promising direction is to investigate how XMAP can be used to obtain counterfactual explanations or prototype-criticism explanations. Finally, investigating human evaluation, a good approach to measure interpretability of machine learning tools and is currently a hot topic in XAI [30], is one of our future studies to improve and extend XMAP.

Availability of data and material Some datasets will be available online when the paper is accepted.

Declarations

Conflicts of interest/Competing interests On behalf of all authors, the corresponding author states that there is no conflict of interest.

Code availability All source codes will be available online when the paper is accepted.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your

intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Darpa's explainable artificial intelligence (xai) program 40. <https://doi.org/10.1609/aimag.v40i2.2850>
2. Becht E, McInnes L, Healy J, Dutertre CA, Kwok IWH, Ng LG, Ginhoux F, Newell E (2019) Dimensionality reduction for visualizing single-cell data using umap. *Nat Biotechnol* 37:38–44
3. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E et al (2008) Fast unfolding of communities in large networks. *J Stat Mech: Theory Exp* 10:2
4. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G (2013) API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122
5. Calegari R, Ciatto G, Omicini A (2020) On the integration of symbolic and sub-symbolic techniques for xai: A survey. *Intell Artif* 14(1):7–32
6. Campello RJGB, Moulavi D, Zimek A, Sander J (2015) Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans Knowl Discov Data* 10(1):1–51
7. Carbonneau R, Laframboise K, Vahidov R (2008) Application of machine learning techniques for supply chain demand forecasting. *Eur J Oper Res* 184(3):1140–1154. <https://doi.org/10.1016/j.ejor.2006.12.004>
8. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pp. 785–794. ACM, New York, NY, USA. <https://doi.org/10.1145/2939672.2939785>
9. Chen X, Duan Y, Houthoofd R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: *Advances in Neural Information Processing Systems* 29, pp. 2172–2180. Curran Associates, Inc
10. Chen Y, Zaki MJ (2017) Kate: K-competitive autoencoder for text. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, pp. 85–94. ACM, New York, NY, USA. <https://doi.org/10.1145/3097983.3098017>
11. Choi H, Choi S (2007) Robust kernel isomap. *Pattern Recogn* 40(3):853–862. <https://doi.org/10.1016/j.patcog.2006.04.025>
12. Chou JS, Bui DK (2014) Modeling heating and cooling loads by artificial intelligence for energy-efficient building design. *Energy Build* 82:437–446. <https://doi.org/10.1016/j.enbuild.2014.07.036>
13. Cox TF, Cox M (2000) *Multidimensional scaling*, 2nd edn. Chapman and Hall/CRC, New York
14. Cranor LF, LaMacchia BA (1998) Spam. *Commun ACM* 41(8):74–83. <https://doi.org/10.1145/280324.280336>
15. Detrano R, János A, Steinbrunn W, Pfisterer M, Schmid J, Sandhu SK, Guppy KH, Lee SW, Froelicher V (1989) International application of a new probability algorithm for the diagnosis of coronary artery disease. *Am J Cardiol* 64(5):304–310
16. Ding J, Condon A, Shah SP (2018) Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. In: *Nature Communications*
17. Doshi-Velez Finale Kim B (2017) Towards a rigorous science of interpretable machine learning. In: eprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608)
18. Elter M, Schulz-Wendtland R, Wittenberg T (2007) The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Med Phys* 34(11):4164–4172
19. Friedman JH, Popescu BE (2008) Predictive learning via rule ensembles. *Ann Appl Stat* 2(3):916–954
20. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: *Advances in Neural Information Processing Systems* 27, pp. 2672–2680. Curran Associates, Inc
21. Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learning: data mining, inference and prediction*, 2nd edn. Springer, Berlin
22. Huang J, Ling CX (2005) Using auc and accuracy in evaluating learning algorithms. *IEEE Trans Knowl Data Eng* 17(3):299–310. <https://doi.org/10.1109/TKDE.2005.50>
23. Johnson AE, Aboab J, Raffa J, Pollard T, Deliberato R, Celi L, Stone D (2018) A comparative analysis of sepsis identification methods in an electronic database. *Crit Care Med* 46(4):494–499
24. Kaggle: IBM HR Analytics Employee Attrition & Performance (2019 (accessed November 11, 2019)). <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>
25. Kaggle: Telco Customer Churn (2019 (accessed November 11, 2019)). <https://www.kaggle.com/blatchar/telco-customer-churn>
26. Khashman A (2010) Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Syst Appl* 37(9):6233–6239. <https://doi.org/10.1016/j.eswa.2010.02.101>
27. Kim B, Khanna R, Koyejo OO (2016) Examples are not enough, learn to criticize! criticism for interpretability. In: *Advances in Neural Information Processing Systems* 29, pp. 2280–2288. Curran Associates, Inc
28. Kohavi R (1996) Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD '96*, pp. 202–207. AAAI Press
29. Kohonen T, Schroeder MR, Huang TS (eds) (2001) *Self-Organizing Maps*, 3rd edn. Springer-Verlag, Berlin, Heidelberg
30. Lage I, Chen E, He J, Narayanan M, Kim B, Gershman S, Doshi-Velez F (2019) An evaluation of the human-interpretability of explanation
31. Letham B, Rudin C, McCormick TH, Madigan D (2013) An interpretable stroke prediction model using rules and bayesian analysis. In: *Proceedings of the 17th AAAI Conference on Late-Breaking Developments in the Field of Artificial Intelligence, AAAIWS'13-17*, pp. 65–67. AAAI Press
32. Lina Z, Shimei P, Jianwu W, Athanasios V (2017) Machine learning on big data, Opportunities and challenges. *Neurocomputing* 237:350–361. <https://doi.org/10.1016/j.neucom.2017.01.026>
33. Linderman GC, Rachh M, Hoskins JG, Steinerberger S, Kluger Y (2018) Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. In: *Nature Methods*
34. Long M, Zhu H, Wang J, Jordan MI (2017) Deep transfer learning with joint adaptation networks. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, pp. 2208–2217. JMLR.org
35. Luna JM, Pechenizkiy M, del Jesus MJ, Ventura S (2018) Mining context-aware association rules using grammar-based genetic programming. *IEEE Trans Cybern* 48(11):3030–3044. <https://doi.org/10.1109/TCYB.2017.2750919>
36. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pp. 4768–4777. Curran Associates Inc., USA
37. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9:2579–2605

38. Makhzani A, Frey BJ (2014) k-sparse autoencoders. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings
39. Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. *Oper Res* 43(4):570–577
40. McInnes L, Healy J, Saul N, Großberger L (2018) UMAP: uniform manifold approximation and projection. *J Open Source Software* 3(29):861
41. Mencar C, Alonso JM (2018) Paving the way to explainable artificial intelligence with fuzzy modeling. In: International Workshop on Fuzzy Logic and Applications, pp. 215–227. Springer
42. Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. *Artif Intell* 267:1–38. <https://doi.org/10.1016/j.artint.2018.07.007>. <https://www.sciencedirect.com/science/article/pii/S0004370218305988>
43. Mittelstadt B, Russell C, Wachter S (2019) Explaining explanations in ai. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19, pp. 279–288. ACM, New York, NY, USA. <https://doi.org/10.1145/3287560.3287574>
44. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533
45. Molnar C (2020) Interpretable Machine Learning. Lulu. com
46. Moro S, Cortez P, Rita P (2014) A data-driven approach to predict the success of bank telemarketing. *Decis Support Syst* 62:22–31
47. Nascimento N, Alencar P, Lucena C, Cowan D (2018) A context-aware machine learning-based approach. In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, CASCON '18, pp. 40–47. IBM Corp., Riverton, NJ, USA
48. O'Leary DE (2013) Artificial intelligence and big data. *IEEE Intell Syst* 28(2):96–99. <https://doi.org/10.1109/MIS.2013.39>
49. Onofri L, Soda P, Pechenizkiy M, Iannello G (2016) A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Syst Appl* 63:97–111. <https://doi.org/10.1016/j.eswa.2016.06.011>
50. Quinlan JR (1987) Simplifying decision trees. *Int J Man-Mach Stud* 27(3):221–234. [https://doi.org/10.1016/S0020-7373\(87\)80053-6](https://doi.org/10.1016/S0020-7373(87)80053-6)
51. Ribeiro MT, Singh S, Guestrin C (2016) “why should i trust you?”: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pp. 1135–1144. ACM, New York, NY, USA. <https://doi.org/10.1145/2939672.2939778>
52. Schlimmer JC (1987) Concept acquisition through representational adjustment. Ph.D. thesis, University of California, Irvine. AAI8724747
53. Shen F, Hasegawa O (2008) A fast nearest neighbor classifier based on self-organizing incremental neural network. *Neural Netw* 21(10):1537–1547. <https://doi.org/10.1016/j.neunet.2008.07.001>
54. Shen F, Hasegawa O (2010) Self-organizing incremental neural network and its application. In: Proceedings of the 20th International Conference on Artificial Neural Networks: Part III, ICANN' 10, pp. 535–540. Springer-Verlag, Berlin, Heidelberg. <http://dl.acm.org/citation.cfm?id=1886436.1886516>
55. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484
56. Singh G, Memoli F, Carlsson G (2007) Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In: Eurographics Symposium on Point-Based Graphics. The Eurographics Association. <https://doi.org/10.2312/SPBG/SPBG07/091-100>
57. Sips M (2009) Visual clustering. Springer, Boston
58. Sokol K, Flach P (2018) Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI' 18, pp. 5868–5870. AAAI Press
59. Topol EJ (2019) High-performance medicine: the convergence of human and artificial intelligence. *Nat Med* 25(1):44–56. <https://doi.org/10.1038/s41591-018-0300-7>
60. Ustun B, Rudin C (2017) Optimized risk scores. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, pp. 1125–1134. ACM, New York, NY, USA. <https://doi.org/10.1145/3097983.3098161>
61. Wachter S, Mittelstadt B, Russell C (2018) Counterfactual explanations without opening the black box: automated decisions and the gdpr. *Harvard J Law Technol* 31(2):841–887
62. Yang H, Rudin C, Seltzer M (2017) Scalable bayesian rule lists. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML' 17, pp. 3921–3930. JMLR.org. <http://dl.acm.org/citation.cfm?id=3305890.3306086>
63. van der Zon SB, Duivesteijn W, van Ipenburg W, Veldsink J, Pechenizkiy M (2019) Icie 1.0: A novel tool for interactive contextual interaction explanations. In: ECML PKDD 2018 Workshops, pp. 81–94. Springer International Publishing, Cham

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.