Comparison-based Inverse Classification for Interpretability in Machine Learning

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, Marcin Detyniecki

Comparison-based Inverse Classification for Interpretability in Machine Learning

Thibault Laugel[1], Marie-Jeanne Lesot[1], Christophe Marsala[1], Xavier Renard[2],
Marcin Detyniecki[1,2,3]

[1]Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005
Paris, France
[2]AXA – Data Innovation Lab, 48 rue Carnot 92150 Suresnes, France
[3]Polish Academy of Science, IBS PAN, Warsaw, Poland
thibault.laugel@lip6.fr

Abstract. In the context of post-hoc interpretability, this paper addresses the task of explaining the prediction of a classifier, considering the case where no information is available, neither on the classifier itself, nor on the processed data (neither the training nor the test data). It proposes an inverse classification approach whose principle consists in determining the minimal changes needed to alter a prediction: in an instance-based framework, given a data point whose classification must be explained, the proposed method consists in identifying a close neighbor classified differently, where the closeness definition integrates a sparsity constraint. This principle is implemented using observation generation in the *Growing Spheres* algorithm. Experimental results on two datasets illustrate the relevance of the proposed approach that can be used to gain knowledge about the classifier.

Keywords: post-hoc interpretability, comparison-based, inverse classification, local explanation.

1 Introduction

Making machine learning systems interpretable, i.e. explaining to the user the decision made by a classifier, can take multiple forms [6, 7], depending on the intuition of what 'interpretable' means and the way the explanation is expressed. A basic characterisation distinguishes between *in-model* and *post-hoc* approaches: the former modifies the learning process so as to obtain, by design, understandable classifiers. Among these, many methods have been proposed in the framework of fuzzy systems, see e.g. [2, 3]: the use of fuzzy logic favors a fluid interface to human beings, although raising many challenges.

Post-hoc approaches build a posteriori explainer systems, using the results of a classifier to interpret its predictions for particular observations.

They can be further distinguished depending on the inputs they require and on the forms of explanation they provide: some methods exploit the classifier type [5, 12] or the training set [13, 1]. Regarding the outputs, some methods offer visual [15] or linguistic [9, 12] explanations, others use observations as explanations, in an instance-based framework [22, 19, 13]. Some other differences relate to the very definition of interpretability: for instance, feature importance analyzes [4, 21] identify the attributes that play a major role on the classifier prediction, inverse classification [18, 5] identify the minimal change that would change the prediction.

In this context, this paper proposes a method that can be characterised as (i) a post-hoc approach, i.e. explaining individual predictions of a classifier, (ii) in a model- and data-agnostic framework, i.e. considering that no information about the classifier to be explained nor about the training data is made available to the user, (iii) within the instance-based paradigm, i.e. explaining through comparison, (iv) applying an inverse classification principle.

More precisely, the principle of the proposed approach to explain the prediction for a given observation consists in exhibiting a close point classified differently: the reasons for the obtained prediction are characterised through the production of this neighbor counter-example. The closeness constraint integrates a sparsity constraint, to match the interpretability requirement that the explanation need to be simple and easy to understand for the user.

The paper is organised as follows: Section 2 presents related works in the framework of post-hoc interpretability, comparison-based approaches and inverse classification. Section 3 details the principle and formalisation of the proposed approach, as well as the *Growing Spheres* algorithm that implements this principle. Section 4 illustrates the results it obtains in two real-world applications.

2 Related Works

Post-hoc interpretability approaches aim at explaining the behavior of a classifier around particular observations to let the user understand their associated predictions, generally disregarding the actual learning process. They have received a lot of interest recently (see e.g. [14]), especially as black-box models such as deep neural networks and ensemble models are being more and more used for classification despite their complexity.

The variety of existing methods comes from the lack of consensus regarding the definition, and a fortiori the formalization, of the very notion of interpretability. Depending on the task performed by the classifier and the needs of the end-user, explaining a result can take multiple forms. Interpretability approaches rely on the meeting the following objectives to design explanations:

1. The explanations should be an accurate representation of what the classifier is doing.
2. The explanations should be understandably read by the user.

This section briefly discusses the hypotheses that are made about available inputs and details two categories especially related to the proposed method: instance-based approaches and inverse classification.

Available Inputs To illustrate this discussion, let us consider the case of a physician using a diagnostic tool. It is natural to speculate that (s)he does not have any information about the machine learning model used to make disease predictions, neither may (s)he have any idea about what patients were used to train it. This raises the question of what knowledge (about the machine learning model and the training or other data) an end-user has, and hence what inputs a post-hoc explainer should use.

Several approaches rely specifically on the knowledge of the algorithm used to make predictions, taking advantage of the classifier structure to generate explanations [5, 12]. However, in other cases, no information about the classifier is available (the model might be only accessible as an oracle for instance): model-agnostic interpretability methods that can explain predictions without making any hypotheses on the classifier are then required [4, 1, 21]. These approaches, sometimes called *sensitivity analyzes*, generally try to analyze how the classifier locally reacts to small perturbations: they for instance perform local approximation of the classifier decision boundary, e.g. using linear functions (LIME [21]) or Parzen window-based gradients [4].

Instance-based Approaches Instance-based approaches constitute a family of post-hoc methods that bring interpretability by comparing an observation to relevant neighbors [22, 13, 1]. They use other observations, from the train set, from the test set or generated ones as explanations to bring transparency to a prediction of a black-box classifier.

One of the motivations for instance-based approaches lies in the fact that in some cases the two aforementioned objectives 1 and 2 are contradictory and cannot be both reached in a satisfying way. In these complex situations, finding examples is an easier and more accurate way to describe the classifier behavior than trying to force a specific inappropriate explanation representation, which would result in incomplete, useless or misleading explanations for the user.

As an illustration, the Parzen window-based approach [4] is shown to not succeed well in providing explanations for individual predictions that are at the boundaries of the training data, giving explanation vectors (gradients) actually pointing in the (wrong) opposite direction from the decision boundary. In such a case, seeing this problem as an instance-based one, and more particularly using comparisons with observations from the other class, would probably make more sense and give more useful insights.

Inverse Classification Inverse classification (see e.g. [16]) is a machine learning task that aims at identifying the minimal changes that can be applied to an observation to as to modify its associated prediction: it has been introduced as an approach to perform sensitivity analysis [18] and later formulated as an interpretability approach [5]. In this view, it belongs to the post-hoc framework and approaches can be categorised using the same characteristics, in particular regarding the assumptions about the available inputs: they can for instance use specific knowledge of the model [5, 16] or the training data [18].

Existing approaches for inverse classification consider modifications on the data features, making them related to the feature importance family of methods. In the specific case of text classification, where texts are represented by possibly weighted bags of words, a related approach studies the terms whose removal would lead to modify the observation classification [19], whereas removing features cannot be considered in a general setting.

It can be underlined that inverse classification can also be related to the task of adversarial learning [23], which aims at 'fooling' a classifier by generating close variations of observations to change their predictions. However, adversarial learning focuses on the classifier robustness and exploits its sensitivity.

3 Proposed Approach

This section presents the principles and formalisation of the proposed approach, as well as its implementation in the *Growing Spheres* algorithm.

3.1 Motivations and Characteristics of the Proposed Approach

Motivations In the light of the axes of discussion presented in the previous section, the justifications for the proposed approach are the following:

First regarding the available inputs, we consider a model- and data-agnostic approach, not requiring any knowledge from the user about the model nor processed data. The only hypotheses we make is that the numerical representation of the data as a feature vector is known, as well as the meaning of the attributes, and that the user can use the classifier to make predictions at will. These weak assumptions seem realistic: for instance, in the aforementioned example of a physician using a diagnostic tool, (s)he is supposed to know what features the system requires to be run, regardless whether the system used performs attribute rescaling or combinations (e.g. through PCA).

Secondly, we consider the instance-based framework and add to the motivations detailed in the previous section the strong justification provided by cognitive sciences of learning through examples [24, 20, 10]. For instance in [24], it is shown through experiments that generated examples help students 'see' abstract concepts that they had trouble understanding with more formal explanations.

Finally, we propose to apply this paradigm to the task of inverse classification in a hybrid approach to take advantage of their respective benefits.

It is important to note that our primary goal here is to give insights about the classifier, not the reality it is approximating. This approach thus aims at understanding a prediction regardless of whether the classifier is right or wrong, or of the distribution of the original data.

Principle Given a black-box classifier and an observation, the explanation we propose to provide is based on a data point, in the light of instance-based interpretability; furthermore, in the light of inverse classification, this data point must belong to the other class. The final explanation is expressed in the form of the displacement vector between the observation and the identified data point.

Following the dual objective of interpretability mentioned in Section 2, the explaining data point must additionally be close to the considered observation. The closeness definition, discussed in the next section, is a key factor for the relevance of the proposed method.

3.2 Formalisation: Proposed Cost Function

We use the following classical notation: we consider a binary classifier f mapping the input space $\mathcal{X}$ of dimension d to an output space $\mathcal{Y} = \{0, 1\}$ (extension to multiclass classification is straightforward), and suppose that no information is available about this classifier. Let $x = (x_i)_i \in \mathcal{X}$ be the observation to be interpreted and $f(x) \in \mathcal{Y}$ its associated prediction. The goal of the proposed approach is to explain x through another observation $e \in \mathcal{X}$, belonging to another class, i.e. such that $f(e) \neq f(x)$. The final form of explanation is the difference vector $e - x$.

For simplification purposes, in the following we call *ally* an observation belonging to the same class as x by the classifier, and *enemy* if it is classified differently.

Recalling objective 1 mentioned earlier, the explanation $e - x$ should be an accurate representation of what the classifier is doing. This is why we decide to transform this problem into a minimization problem by defining the function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ such that $c(x, e)$ is the cost of moving from observation x to enemy e.

Using this notation, we focus on solving the following minimization problem:

$$e^* = \underset{e \in \mathcal{X}}{\arg\min}\{c(x, e) \quad | \quad f(e) \neq f(x)\} \tag{1}$$

$$\text{with } c(x, e) = ||x - e||_2 + \gamma ||x - e||_0 \tag{2}$$

where $||.||_2$ the Euclidean norm, $||.||_0$ the l_0 norm defined as the number of non-zero coordinates, $||e - x||_0 = \sum_{i \leq d} 1_{x_i \neq e_i}$, and $\gamma \in \mathbb{R}^+$ a hyperparameter weighting the two terms.

Indeed, looking up to [22], we choose to use the l_2 norm of the vector $e - x$ as a component of the cost function to measure the proximity between e and x. However, recalling objective 2, we need to make sure that this cost function guarantees a final explanation that can be easily read by the user. In this regard, we consider that users intuitively find explanations of small dimension to be simpler. Hence, we decide to integrate vector sparsity, measured by the l_0 norm, as another component of the cost function c and combine it with the l_2 norm as a weighted average.

3.3 The Growing Spheres Algorithm

Due to the cost function c being discontinuous and the hypotheses made (black-box classifier and no existing data) solving problem defined in Equation (1)

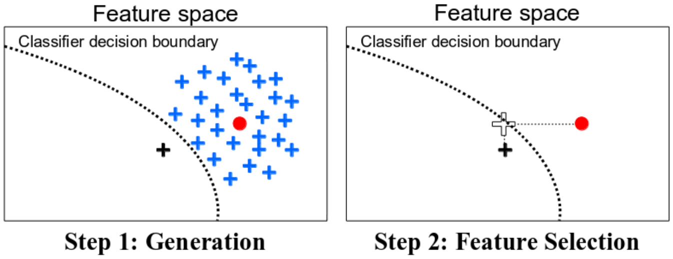
Feature space
Classifier decision boundary
Step 1: Generation
Feature space
Classifier decision boundary
Step 2: Feature Selection

Fig. 1. Illustration of Growing Spheres: The red circle represents the observation to interprete, the plus signs the generated observations (blue for allies, black for ennemies). The white plus is the final enemy e^* used to generate explanations.

is difficult. Hence, we choose to solve sequentially the two components of the cost function and propose *Growing Spheres*, a two-step heuristic approach that approximates the solution of this problem. These two steps, namely Generation and Feature Selection, are described in turn below and illustrated in Figure 1.

Generation The instance generation, detailed in Algorithm 1, is performed without relying on existing data. Thus, for the considered observation x, we ignore in which direction the closest classifier boundary might be. A greedy approach to find the closest enemy is to explore the input space $\mathcal{X}$ by generating instances in all possible directions further and further until the decision boundary of the classifier is crossed, thus minimizing the l_2-component of function c. More precisely, the algorithm generates observations in the feature space in l_2-spherical layers around x until an enemy is found.

Formally, given two positive numbers a_0 and a_1, we define a (a_0, a_1)-spherical layer SL around x as: $SL(x, a_0, a_1) = \{z \in \mathcal{X} \;\; : \;\; a_0 \leq ||x - z||_2 \leq a_1\}$. To generate observations following a uniform distribution over these subspaces, we use the YPHL algorithm [11] which generates observations uniformly distributed over the surface of the unit sphere. We then draw $\mathcal{U}_{[a_0, a_1]}$-distributed values and use them to rescale the distances between the generated observations and x. As a result, we obtain observations that are uniformly distributed over $SL(x, a_0, a_1)$.

The first step of the algorithm consists in generating uniformly n observations in the l_2-ball of radius η and center x, which corresponds to $SL(x, 0, \eta)$ (line 1 of Algorithm 1), with n and η hyperparameters of the algorithm.

In case this initial generation step already contains ennemies, we need to make sure that the algorithm did not miss the closest decision boundary. This is done by updating the value of the initial radius: $\eta \leftarrow \eta/2$ and repeating the initial step until no enemy is found in the initial ball $SL(x, 0, \eta)$ (lines 2 to 5).

However, if no enemy is found in $SL(x, 0, \eta)$, we update a_0 and a_1 using η, generate over $SL(x, a_0, a_1)$ and repeat this process until the first enemy is found (as detailed in lines 6 to 11).

Algorithm 1 Growing Spheres Generation

```
Require: $f : \mathcal{X} \rightarrow \{-1; 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ an observation to be interpreted
Require: Hyperparameters: $\eta, n$
Ensure: enemy $e$
Generate $(z_i)_{i \leq n}$ in $SL(x, 0, \eta)$ following a uniform distribution
while $\exists\ e \in (z_i)_{i \leq n} \mid f(e) \neq f(x)$ do
    $\eta = \eta/2$
    Generate $(z_i)_{i \leq n}$ in $SL(x, 0, \eta)$ following a uniform distribution
end while
Set $a_0 = \eta$, $a_1 = 2\eta$
while $\nexists\ e \in (z_i)_{i \leq n} \mid f(e) \neq f(x)$ do
    Generate $(z_i)_{i \leq n}$ uniformly in $SL(x, a_0, a_1)$
    $a_0 = a1$
    $a_1 = a1 + \eta$
end while
Return $e$, the $l_2$-closest generated enemy from $x$
```

In the end, Algorithm 1 returns the l_2-closest generated enemy e from the observation to be interpreted x (as represented by the black plus in Figure 1). Once this is done, we focus on making the associated explanation as easy to understand as possible through feature selection.

Feature Selection In the second step, in order to make the difference vector of the closest enemy sparse, we simplify it by reducing the number of features used when moving from x to e (thus minimizing the l_0 component of the cost function $c(x, e)$ and generating the final solution e^*), as explained in the Feature Selection part. To do so, we consider again a naive heuristic based on the idea that the smallest coordinates of $e - x$ might be less relevant locally regarding the classifier decision boundary and should thus be the first ones to be ignored. Thus, the algorithm tries to align as many coordinates of e with x as possible, as long as the predicted class does not change. The proposed feature selection algorithm we use is detailed in Algorithm 2.

The final explanation provided to interprete the observation x and its associated prediction is the vector $x - e^*$, with e^* the final enemy identified by the algorithms (represented by the white plus in Figure 1).

4 Experimental Results

Although many numerical criteria for interpretability have been proposed (see e.g. [9]), there is no consensus about a global measure for the quality of an explanation. Evaluations based on user satisfaction [4, 21, 7], although ideal, also depend on the global task the explanations are integrated to and require difficult definitions of experimental protocol. Moreover, the variety of interpretability

Algorithm 2 Growing Spheres Feature Selection

```
Require: $f : \mathcal{X} \rightarrow \{-1; 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ the observation to be interpreted
Require: $e \in \mathcal{X} \mid f(e) \neq f(x)$ the solution of Algorithm 1
Ensure: enemy $e^*$
    Set $e' = e$
    while $f(e') \neq f(x)$ do
        $e^* = e'$
        $i = \underset{j \in [1:d],\ e'_j \neq x_j}{\arg\min} |e'_j - x_j|$
        Update $e'_i = x_i$
    end while
    Return $e^*$
```

methods (see Sections 1 and 2), both in terms of required inputs and of result forms, makes it difficult to compare them.

As a preliminary experiment, this section presents the results obtained when applying the proposed approach to news and image classification. It illustrates the explanations provided by *Growing Spheres* and shows, at a higher level, how a user can exploit them to derive knowledge about the characteristics of the considered classifier, including its possible weaknesses. It also examines whether the explanations can be easily read by a user by measuring the sparsity.

4.1 Prediction of News Popularity

Experimental Protocol The news popularity dataset [8] is made of 39644 online articles from website Mashable, described by 58 numerical features. The latter encode information about the format and content of the articles, they for instance include the number of words in the title, a measure of the content subjectivity or the popularity of the used keywords. The binary classification task aims at predicting whether an article is popular or not, where popularity is defined as having been shared more than 1400 times.

We apply *Growing Spheres*, to explain the predictions of a classifier. Although it is of no importance for the proposed approach, the experimental protocol consisted in training a random forest (RF) on 70% of the data, after applying a grid search to select the best hyperparameters of RF (number of trees). Tested on the rest of the data, the RF achieved 0.7 AUC and 0.69 accuracy.

Regarding *Growing Spheres*, we use $\gamma = 1$ to define the cost function c (see Eq. 2) and set the hyperparameters of Algorithm 1 to $\eta = 0.001$ and $n = 10000$.

The hypothesis that no information is available about the classifier can be illustrated considering an online journalist writing for Mashable, who would like to predict whether the articles (s)he wrote are going to be popular or not and understand why. The journalist uses a black-box machine learning tool to make the prediction, and has hence no idea about what algorithm was used nor what data was used to train it. The user thus decides to use *Growing Spheres* to generate explanations for the prediction.

Article/class	Feature	Move
A1 Not Popular	Min. shares of referenced articles in Mashable	+2016
	Avg. keyword (max. shares)	+913
A2 Popular	Avg. keyword (max. shares)	-911
	Min. shares of referenced articles in Mashable	-3557
	Rate of positive words (content)	-0.01

Table 1. Output example of *Growing Spheres*

Illustrative Examples We consider two observations from the test set: Article A1, entitled 'The White House is Looking for a Few Good Coders', that is predicted to be not popular by the considered classifier, and article A2, entitled '8 Vendors You Didn't Know Accepted Bitcoins', predicted to be popular. The explanation vector given by *Growing Spheres* are shown in Table 1.

For article A1, among the articles it refers, the least popular of them would need to have 2016 more shares and the most popular article associated to its keywords would need to have 913 more shares in order to change the prediction. In other words, article A1 would be predicted to be popular by the considered classifier if the references and the keywords it uses were more popular themselves.

As for article A2, its associated prediction can be explained by three characteristics: to be predicted as unpopular, the same features relevant for A1 would need to be reduced; moreover, the feature 'rate of positive words in the content' would need to be reduced by 0.01. This means that a slightly less positive writing angle would contribute to have article A2 predicted as being not popular.

Sparsity Evaluation In order to check whether the proposed approach fulfills its goal of finding explanations that can be easily understood by the user, we evaluate the global sparsity of the generated explanations. We measure sparsity as the number of non-zero coordinates of the explanation vector $||x - e^*||_0$.

Figure 2 shows the smoothed cumulative distribution of this value for all 11893 test data points. We observe that the maximum value over the whole test set is 20, meaning that each observation of the test dataset only needs to change 20 or less among the 58 available coordinates to cross the decision boundary. Moreover, 80% of them only need to move in 10 directions or less, that is 17% of the features only. This shows that the proposed method indeed achieves sparsity in order to make explanations more readable. It is important to note that this does not mean that only 20 features are needed to explain all the observations, since nothing guarantees different explanations use the same features.

4.2 Applications to Digit Classification

Experimental Protocol We now use the MNIST handwritten digits database [17] and apply *Growing Spheres* to the binary classification problem of recognizing the digits 8 and 9 vs each other. The dataset contains 60000 instances of 784 features (28 by 28 pictures of digits). We use a support vector machine

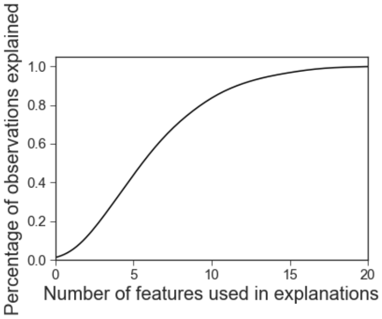

Fig. 2. Sparsity distribution over the news test dataset. Reading: '40% of the observations of the test dataset have explanations that use 5 features or less'.

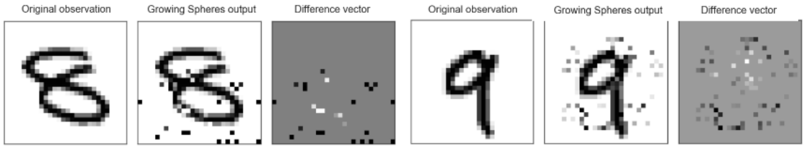

Fig. 3. *Growing Spheres* output examples. From left to right: example of the original instance x, closest enemy found e^*, explanation vector $x - e^*$. First for an 8, then for a 9. A white pixel indicates a 0 value, black a 1.

classifier with a RBF kernel and parameter $C = 15$. Once again, the choice of this model is arbitrary, since *Growing Spheres* is model-agnostic. We train the model on 50% of the data and test it on the rest (0.98 AUC score). As in the first experiment, we use $\gamma = 1$, $\eta = 0.001$ and $n = 10000$.

Illustrative Example Given a picture of an 8, our goal is to understand, according to the classifier, why it is predicted to be an 8 (and reciprocally). Our intuition would be that closing the bottom loop of a 9 should be the most influential change needed to make it become an 8, and hence features provoking a class change should include pixels found in the bottom-left area of the digits. Output examples to interprete an 8 and a 9 predictions are shown in Figure 3.

The first thing we observe is that the closest ennemies found by *Growing Spheres* in both cases are not proper 9 and 8 digits respectively. In fact, a human observer would probably still identify the generated enemies as noised versions of the original digits: (i) the pixels involved in the move vector are not all located around the digit but all accross the picture, and (ii) the pixels located in the bottom-left area of the digits do not form a line, and are not 'dark enough'.

This is consistent with the principle of our proposed approach: as mentioned in Section 3, *Growing Spheres* is trying to understand the classifier decision, not the reality it is approximating. In this case, the fact that the classifier apparently considers these pixels to be influential the classification of these digits could be an evidence of the learned boundary inaccuracy. Contrary to feature importances, these pixels are not an indication of the contribution of each pixel to the prediction, but rather of the shape of the local decision bourder.

5 Conclusion and Future Works

The proposed post-hoc interpretability approach *Growing Spheres* provides explanations of a single prediction through the comparison of its considered observation with its closest enemy. In the case where no information is available, neither about the classifier nor about the data, it offers an instance-based inverse classification tool taking into account the objective of sparse explanations. Preliminary experiments illustrate its relevance for explaining predictions and providing insights about the classifier.

Ongoing works aim at experimentally studying the influence of the algorithm hyperarameters and validating the relevance of the explanations it provides in the framework of real-world applications. Another perspective consists in relaxing some of the strong constraints *Growing Spheres* relies on, in particular so as to cases where some information about the data is available: any knowledge about the data distribution might help to guide the generation process, thus for instance minimizing the risk of exploring irrelevant areas of the input space.

6 Aknowledgements

This work has been done as part of the Joint Research Initiative (JRI) project ”Interpretability for human-friendly machine learning models” funded by the AXA Reseach Fund.

References

1. Adler, P., Falk, C., Friedler, S.A., Rybeck, G., Scheidegger, C., Smith, B., Venkatasubramanian, S.: Auditing Black-box Models for Indirect Influence. 2016 IEEE 16th Int. Conf. on Data Mining (ICDM) pp. 1–10 (2016)
2. Alonso, J., Magdalena, L.: Special issue on interpretable fuzzy systems. Information sciences 181(20) (2011)
3. Alonso, J.M., Castiello, C., Mencar, C.: Interpretability of fuzzy systems: Current research trends and prospects. In: Kacprzyk, J., Pedrycz, W. (eds.) Springer Handbook of Computational Intelligence, pp. 219–237. Springer (2015)
4. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Mueller, K.R.: How to Explain Individual Classification Decisions. Journal of Machine Learning Research 11, 1803–1831 (2009)
5. Barbella, D., Benzaid, S., Christensen, J., Jackson, B., Qin, X.V., Musicant, D.: Understanding Support Vector Machine Classifications via a Recommender System-Like Approach. In: Proc. of the Int. Conf. on Data Mining. pp. 305–11 (2009)
6. Biran, O., Cotton, C.: Explanation and Justification in Machine Learning : A Survey. International Joint Conference on Artificial Intelligence Workshop on Explainable Artificial Intelligence (IJCAI-XAI) (2017)

7. Doshi-Velez, F., Kim, B.: Towards A Rigorous Science of Interpretable Machine Learning. arXiv preprint:1702.08608 (2017)
8. Fernandes, K., Vinagre, P., Cortez, P.: A proactive intelligent decision support system for predicting the popularity of online news. In: Proc. of the 17th EPIA Conference. pp. 535–546 (2015)
9. Gacto, M., Alcal, R., Herrera, F.: Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Information Sciences 181(20), 4340 – 4360 (2011), special Issue on Interpretable Fuzzy Systems
10. van Gog, T., Kester, L., Paas, F.: Effects of worked examples, example-problem, and problem-example pairs on novices' learning. Contemporary Educational Psychology 36(3), 212–218 (2011)
11. Harman, R., Lacko, V.: On decompositional algorithms for uniform sampling from n-spheres and n-balls. Journal of Multivariate Analysis 101(10), 2297 – 2304 (2010)
12. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. Proc. of the European Conf. on Computer Vision EECV 2016 pp. 3–19 (2016)
13. Kabra, M., Robie, A., Branson, K.: Understanding classifier errors by examining influential neighbors. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. pp. 3917–3925 (2015)
14. Kim, B., Doshi-Velez, F.: Interpretable machine learning : The fuss , the concrete and the questions. In: ICML Tutorial on interpretable machine learning (2017)
15. Krause, J., Perer, A., Bertini, E.: Using visual analytics to interpret predictive machine learning models. In: ICML Workshop on Human Interpretability in Machine Learning. pp. 106–110 (2016)
16. Lash, M.T., Lin, Q., Street, W.N., Robinson, J.G.: A budget-constrained inverse classification framework for smooth classifiers. 2017 IEEE Int. Conf. on Data Mining Workshops (ICDMW17) (2017)
17. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
18. Mannino, M.V., Koushik, M.V.: The Cost Minimizing Inverse Classification Problem : a Genetic Algorithm Approach. Decision Support Systems 29(3), 283–300 (2000)
19. Martens, D., Provost, F.: Explaining data-driven document classifications. Mis Quarterly 38(1), 73–99 (2014)
20. Mvududu, N., Kanyongo, G.Y.: Using real life examples to teach abstract statistical concepts. Teaching Statistics 33(1), 12–16 (2011)
21. Ribeiro, M.T., Singh, S., Guestrin, C.: Why Should I Trust You? In: Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining - KDD'16. pp. 1135–1144 (2016)
22. Štrumbelj, E., Kononenko, I., Robnik Šikonja, M.: Explaining instance classifications with interactions of subsets of feature values. Data and Knowledge Engineering 68(10), 886–904 (2009)
23. Tygar, J.D.: Adversarial machine learning. IEEE Internet Computing 15(5), 4–6 (2011)
24. Watson, A., Shipman, S.: Using learner generated examples to introduce new concepts. Educational Studies in Mathematics 69(2), 97–109 (2008)