IEEE Transactions on Knowledge and Data Engineering, 20:589-600, 2008

Explaining Classifications for Individual Instances

Marko Robnik-Šikonja, Igor Kononenko
University of Ljubljana, Faculty of Computer and Information Science,
Ljubljana, Slovenia
{Marko.Robnik, Igor.Kononenko}@fri.uni-lj.si

Abstract

We present a method for explaining predictions for individual instances. The presented approach is general and can be used with all classification models that output probabilities. It is based on decomposition of a model's predictions on individual contributions of each attribute. Our method works for so called black box models such as support vector machines, neural networks, and nearest neighbor algorithms as well as for ensemble methods, such as boosting and random forests. We demonstrate that the generated explanations closely follow the learned models and present a visualization technique which shows the utility of our approach and enables the comparison of different prediction methods.

Keywords: machine learning, decision support, knowledge modeling, information visualization, model explanation, model comprehensibility, decision visualization, prediction models, classification, nearest neighbor, neural nets, support vector machines

1 Introduction

One of important requirements for predictors, both in classification and regression, is the transparency of the prediction process. The user of the prediction model is often interested not only in the prediction accuracy but also in the explanation of the prediction for a given new case. For example, in our recent work on medical data physicians were mostly interested in explanation capabilities of learned models. Expectations of a study were to get for each new patient a prognosis and its explanation. Such an explanation of a model's decision on the level of individual instance is therefore main motivation of the presented work. We propose a general explanation method that is in principle independent of the model. The model can be generated manually or learned automatically, it can be transparent or black box (such as support vector machines (SVM) and artificial neural networks (ANN)), and it can be a single model or an ensemble of models (such as boosting and random forests).

We distinguish between two levels of explanation: the *domain level* and the *model level*. The domain level tries to find the true causal relationship between the dependent and independent variables. Typically this level is unreachable unless we are dealing

with artificial domains where all the relations as well as the probability distributions are known in advance. On the other hand, the model level explanation aims to make transparent the prediction process of a particular model. The prediction accuracy and the correctness of explanation at the model level are orthogonal: the correctness of the explanation is independent of the correctness of the prediction. However, we may assume that better models (with higher prediction accuracy) enable in principle better explanation at the domain level. However, this work is interested in the explanation at the model level and leave to the developer of the model the responsibility for its prediction accuracy. In this paper we talk about

- *instance explanation*: explanation of a classification of a single instance at the model level,
- *model explanation*: averages of explanations over many training instances at the model level, which provide more general explanations of features and features' values relevances,
- *domain explanation*: still unknown, although, if the accuracy of the model is high, it should be quite similar to the model explanation.

1.1 Notation and Organization

Throughout the paper we use a notation where each of the n learning instances is represented by an ordered pair (x,y); each vector of attribute values x consists of individual values of attributes A_i, $i = 1,...,a$ (a is the number of attributes), and is labeled with y. In case of classification, y is one of the discrete class values y_j, $j = 1,...,c$ (c is the number of class values). We write $p(y_j)$ for the probability of the class value y_j. Each discrete attribute A_i has values $a_1,...,a_{m_i}$ (m_i is the number of values of the attribute A_i). $p(a_j)$ is a probability of value a_j.

The paper is organized into 6 sections. In Section 2 we formally introduce our explanation principle, define several interpretations and give implementation details. In Section 3 we demonstrate the visualization method for individual instances and for the whole model. In Section 4 we use several artificial data sets to show that explanations are close to the models and give some advice for the use of our approach. Section 5 presents the related work and Section 6 summarizes and gives some ideas for further work.

2 Decomposition of the Prediction

Assume for a moment that we can observe the inner workings of the decision process which forms the relationship between the features and the predicted value. In other words, assume that we can observe a causal effect the change of an attribute's value has on the predicted value. By measuring such an effect we could reason about the importance of the attribute's values, and we could determine which values are the thresholds for the change of prediction. In practice, this is usually impossible, but we can use our model and data sample and try to approximate this reasoning.

We consider the model as a function mapping instances into numerical values $f : x \mapsto f(x)$. For classification these numerical values are the probabilities of the class values. An instance x has a known value for each attribute A_i. To see the effect the attribute values have on prediction of the instance we decompose the prediction on individual attributes' values and observe the model's prediction $f(x\backslash A_i)$. In other words, we observe the model's prediction for x without the knowledge of event $A_i = a_k$ (marginal prediction), where a_k is the value of A_i for our instance x. By comparing the values $f(x)$ and $f(x\backslash A_i)$ we get insight into the importance of event $A_i = a_k$. If the difference between $f(x)$ and $f(x\backslash A_i)$ is large, the fact $A_i = a_k$ plays an important role in the model; if this difference is small, the influence of $A_i = a_k$ in the model is minor. The source of our explanations are therefore the decompositions

$$\text{predDiff}_i(x) = f(x) - f(x\backslash A_i) \tag{1}$$

By restricting our reasoning to the model we provide explanations also for events where the change in more than one attribute at once affects the predicted value. For such events each of dependent attributes A_i affects the prediction and so also the score $\text{predDiff}_i(x)$, therefore the explanations for all A_i are nonzero. In this way the generated explanations not only provide information about simple one-attribute-at-a-time dependencies (as it may look at first sight) but also about complex multi-attribute dependencies, as long they are expressed in a given model. We provide a worked example at the end of this Section.

In evaluation of prediction difference (1) we have several options. For classification we present the information difference, the weight of evidence, and the difference between the probabilities. Normally one of these evaluations is sufficient, but each has its favorable properties and weaknesses which we explain below and analyze also in Section 4.

2.1 Evaluation of Prediction Differences in Classification

In classification the model is a mapping from the instance space to probabilities of the class values. The difference can be evaluated in (at least) three different ways. The first is based on the notion of information [1]. The second one is defined with the log odds or equivalently the weight of evidence [2]. The third one is a direct difference between the probabilities. Below we define these interpretations for an instance x and its value of attribute A_i.

2.1.1 Information Difference

The information difference for the class value y is defined as the difference between the amount of information, necessary to find out that y is true for the given instance with the knowledge about the value of A_i, and the amount of information, necessary to find out that y is true for the given instance without the knowledge about the value of A_i:

$$\text{infDiff}_i(y|x) = \log_2 p(y|x) - \log_2 p(y|x\backslash A_i) \;\; [bit] \tag{2}$$

The notion of information is frequently used, so this interpretation of the difference in information between two events is comprehensible. Because logarithm is asymmetric in the range of probabilities $[0,1]$, we get asymmetric results for complementary probabilities (p and $1-p$) in e.g. two class problems.

2.1.2 Weight of Evidence

The odds of event z is defined as the ratio of the probability of event z and its negation:

$$\text{odds}(z) = \frac{p(z)}{p(\bar{z})} = \frac{p(z)}{1-p(z)}$$

The weight of evidence for class value y is defined as the log odds of the model's probability with the knowledge about the value of A_i and without it:

$$\text{WE}_i(y|x) = \log_2(\text{odds}(y|x)) - \log_2(\text{odds}(y|x\backslash A_i)) \;\; [bit] \tag{3}$$

The weight of evidence is an alternative view on information [2] with similar properties (sometimes favorable, e.g. symmetry). In logistic regression where it is commonly used [3] it is referred to as log odds-ratio.

2.1.3 Difference of Probabilities

Another possibility is to evaluate the difference between probabilities directly. The probability difference is the difference in prediction of the model having the knowledge about the value of A_i and without it.

$$\text{probDiff}_i(y|x) = p(y|x) - p(y|x\backslash A_i) \tag{4}$$

It may not be wise to use the difference between probabilities directly, without normalization, as humans are known not to be very good at comprehension and evaluation of probabilities [4]. This is especially true for probabilities close to 0 and 1. The good thing about this method is its simplicity and the fact that we do not need any corrections for probabilities 0 and 1.

2.2 Implementation

To get the explanation factors we have to evaluate either (2), (3), or (4). To compute factor $p(y|x)$ we just classify the instance x with the model. The only condition the model has to satisfy is that it outputs class probabilities. The majority of statistical and machine learning modeling techniques satisfy this condition directly or with appropriate post-modeling calibration.

The factors $p(y|x\backslash A_i)$ (or $f(x\backslash A_i)$) are a bit more tricky. The simplest, but not always the best option is to replace the value of attribute A_i with a special unknown value (NA, don't know, don't care, etc.). This special value does not contain any information of A_i, indeed. However, this method is appropriate only for modeling techniques which handle unknown values naturally, e.g., naive Bayesian classificator (NB) just omits the attribute with unknown value from the computation. For other models we have to bear

in mind that while this approach is simple and seemingly correct, we are left to the mercy of each method's internal mechanism for handling these special values[1]. The techniques for handling unknown values are very different: from replacement with the most frequent value for nominal attributes and with median for numerical attributes to complex model-based implantations. To avoid the dependence on the implementation of the model, we propose an approach which simulates the lack of information about A_i with several predictions.

For nominal attributes we replace the actual value $A_i = a_k$ with all possible values of A_i, and weight each prediction by the prior probability of the value[2]:

$$p(y|x\backslash A_i) = \sum_{s=1}^{m_i} p(A_i = a_s|x\backslash A_i)p(y|x \leftarrow A_i = a_s) \tag{5}$$

$$p(y|x\backslash A_i) \doteq \sum_{s=1}^{m_i} p(A_i = a_s)p(y|x \leftarrow A_i = a_s) \tag{6}$$

Here the term $p(y|x \leftarrow A_i = a_s)$ represents the probability we get for y when in x we replace the value of A_i with a_s. Note our simplification of the prior probability $p(A_i = a_s)$ which implies that (6) is only an approximation.

This method as it may seem ad-hoc at first sight is actually exactly what NB does. As we mentioned, for NB to compute $p(y|x\backslash A_i)$, all we have to do, is to ignore the value of attribute A_i in computation. In Appendix we prove that for NB the method (6) is equivalent to excluding the attribute A_i from computation.

For numerical attributes the procedure is similar; we use a discretization method to split the values of A_i into sub-intervals. The middle points of these sub-intervals are taken as the representative replacement values in (6) for which we compute predictions $p(y|x \leftarrow A_i = a_s)$. Instead of prior probabilities of single values $p(A_i = a_s)$, we use probabilities of the sub-intervals for weighting the predictions.

To avoid division by zero and logarithm of zero in evaluation of (2) and (3) we use the Laplace correction [5]; instead of each probability p, we use the factor $\frac{(pn+1)}{(n+c)}$, where c is the number of class values and n is inversely proportional to the strength of belief in uniform prior probability. For n we use the number of training instances.

The generation of explanations does not affect the learning phase; for a single instance and for a particular model we need $O(a)$ model evaluations (at least one prediction for each attribute). In reality the time to generate explanations is negligible on today's computers.

2.3 An Example of Computing Explanations

To illustrate how the generation of explanations works we define a simple Boolean problem with three important attributes (A_1, A_2, and A_3) and one irrelevant one (A_4). The class value 1 is defined as $C = A_1 \wedge (A_2 \equiv A_3)$. The whole truth table is listed in Table 1.

[1] In the R system, which we used as our testing environment, the default behavior of many learning models is to fail when predicting an input with NA values, but of course many methods exist how to handle them.

[2] The use of this method therefore assumes that we have access to the prior probabilities of the values also during explanation.

Table 1: The data set for the problem $C = A_1 \wedge (A_2 \equiv A_3)$.

A_1	A_2	A_3	A_4	C
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

A decision tree model built from this data is presented in Fig. 1. In the leaves, below the classification, the numbers of instances for each class are given. We have selected decision trees for our example because the decision process is transparent with these models and we can follow it from the root to the leaves.

The first instance we want an explanation for is $x = (1,0,0,1)$, meaning $A_1 = 1$, $A_2 = 0$, $A_3 = 0$, and $A_4 = 1$. Suppose we want explanation for class value $c = 1$ with weight of evidence (3), then for each attribute A_i we have to compute

$$\mathrm{WE}_i(c = 1|x) = \log_2 \mathrm{odds}(c = 1|x) - \log_2 \mathrm{odds}(c = 1|x\backslash A_i)$$

When we classify this instance with the model from Fig. 1, it ends in the bottom left-hand leaf classified to class 1 with $(0,4)$ as the distribution of the instances. If we estimate probabilities with relative frequency, then $p(c = 1|x) = 1$. To get explanations we have to compute also $p(c = 1|x\backslash A_i)$ for each attribute i using (6). For A_1 its values 0 and 1 have equal probabilities 0.5 and we get $p(c = 1|x\backslash A_1) =$

$$\begin{aligned} &= 0.5 \cdot p(c = 1|(0,0,0,1)) + 0.5 \cdot p(c = 1|(1,0,0,1)) \\ &= 0.5 \cdot 0 + 0.5 \cdot 1 = 0.5 \end{aligned}$$

For this we had to classify also the instance $(0,0,0,1)$ (A_1 with value 0), which ended in the top left-hand side leaf with distribution $(8,0)$, giving $p(c = 1|(0,0,0,1)) = 0$. Using the Laplace correction with $n = 16$ we finally get explanation for A_1:

$$\begin{aligned} \mathrm{WE}_1(c = 1|x) &= \log_2 \mathrm{odds} \frac{1 \cdot 16 + 1}{16 + 2} - \log_2 \mathrm{odds} \frac{0.5 \cdot 16 + 1}{16 + 2} \\ &= \log_2 \frac{\frac{17}{18}}{\frac{1}{18}} - \log_2 \frac{\frac{9}{18}}{\frac{9}{18}} = 4.09 - 0 = 4.09 \end{aligned}$$

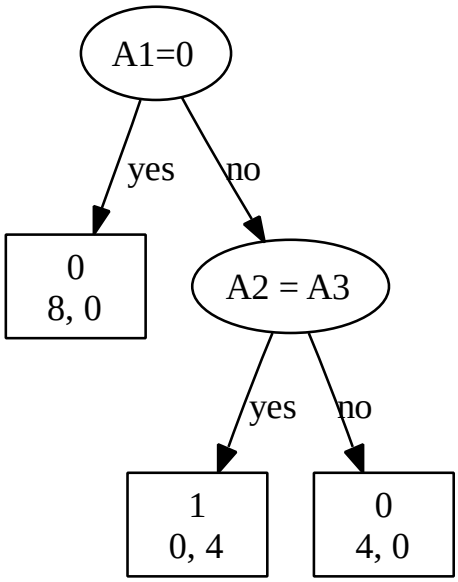

Figure 1: A decision tree for the problem $C = A_1 \wedge (A_2 \equiv A_3)$. In the leaves (square nodes) below the predicted class values the distributions of class values are given.

meaning that in the given model A_1 positively influences the class 1. If we compute the explanation for class 0, we get the complementary result:

$$\begin{aligned} \mathrm{WE}_1(c=0|x) &= \log_2 \mathrm{odds}\frac{0\cdot 16+1}{16+2} - \log_2 \mathrm{odds}\frac{0.5\cdot 16+1}{16+2} \\ &= -4.09, \end{aligned}$$

indicating that A_1 negatively affects the class 0. For explanation with information difference (2) we get 0.92 and -3.17 for classes 1 and 0, respectively (note the asymmetry). For difference of probabilities (4) we get 0.5 and -0.5, respectively.

For explanation of A_2 we have to compute

$$\mathrm{WE}_2(c=1|x) = \log_2 \mathrm{odds}(c=1|x) - \log_2 \mathrm{odds}(c=1|x\backslash A_2)$$

To get $p(c=1|x\backslash A_2)$ we classify $(1,1,0,1)$, getting $p(c=1|(1,1,0,1)) = 0$ and $p(c=1|x\backslash A_2) =$

$$\begin{aligned} &= 0.5\cdot p(c=1|(1,0,0,1)) + 0.5\cdot p(c=1|(1,1,0,1)) \\ &= 0.5\cdot 1 + 0.5\cdot 0 = 0.5 \end{aligned}$$

We get $\mathrm{WE}_2(c=1|x) = 4.09$, indicating positive (and equivalent to A_1) influence of A_2 on class 1 in this model.

While A_3 is identical to A_2, for A_4 which is left out from the model, we get the same classification for its values 0 and 1, giving $p(c=1|x) = p(c=1|x\backslash A_4) = 1$. Finally we get $\mathrm{WE}_4(c=1|x) = 0$, an indication of irrelevance.

For another instance we would get different explanation. For example, if x=(0,1,1,0), the explanations for class 1 for (A_1, A_2, A_3, A_4) are $(-4.09, 0.0, 0.0, 0.0)$. The complete list of explanations for decision tree from Fig. 1, the weight of evidence and class 1, is provided in Table 2.

It is interesting that for some instances with $A_1 = 0$ and $A_2 \neq A_3$, e.g., $(0,0,1,0)$, the explanations for all the attributes are 0. The reason for this is, that if we change the value of any attribute, the classification will remain the same, either in the top left-hand side, or the bottom right-hand side leaf. Note that this is due to the following concept: "if $A1 = 0$ or $A2 \neq A3$ then C = 0". This is the main weaknesses of our approach. Namely, the proposed methodology is not able to correctly evaluate the utility of attributes' values in instances where the change in more than one attribute value at once is needed to affect the predicted value. This problem can be overcome only by an extensive search of pairs, triples, etc. of attribute values. The exhaustive search is of course unfeasible and a heuristic search reduction is necessary (see the further work in Section 6).

Table 2: The explanations for class 1 for the decision tree from Fig. 1 using the weight of evidence.

instance x				explanations $WE_i(c = 1\|x)$			
A_1	A_2	A_3	A_4	WE_1	WE_2	WE_3	WE_4
0	0	0	0	-4.09	0.00	0.00	0.00
0	0	0	1	-4.09	0.00	0.00	0.00
0	0	1	0	0.00	0.00	0.00	0.00
0	0	1	1	0.00	0.00	0.00	0.00
0	1	0	0	0.00	0.00	0.00	0.00
0	1	0	1	0.00	0.00	0.00	0.00
0	1	1	0	-4.09	0.00	0.00	0.00
0	1	1	1	-4.09	0.00	0.00	0.00
1	0	0	0	4.09	4.09	4.09	0.00
1	0	0	1	4.09	4.09	4.09	0.00
1	0	1	0	0.00	-4.09	-4.09	0.00
1	0	1	1	0.00	-4.09	-4.09	0.00
1	1	0	0	0.00	-4.09	-4.09	0.00
1	1	0	1	0.00	-4.09	-4.09	0.00
1	1	1	0	4.09	4.09	4.09	0.00
1	1	1	1	4.09	4.09	4.09	0.00

From our example we see some practical properties of the explanations:

1. model dependency: explanations express decision process taking place inside the model, so if the model is wrong for a given problem, explanation will reflect that and will be correct for the model, therefore wrong for the problem;

2. instance dependency: different instances are predicted differently, so the explanations will also be different;

3. class dependency: explanations for different classes are different, different attributes may have different influence on different classes (for two class problems, the effect is complementary);

4. capability to detect strong conditional dependencies: if the model captures strong conditional dependency (e.g., equivalence relation in Fig. 1), the explanations will also reflect that;

5. inability to detect and correctly evaluate the utility of attributes' values in instances where the change in more than one attribute value at once is needed to affect the predicted value. To overcome this problem one can perform a search over the combinations of attribute values but this is a matter of further work.

6. dependency on the type of the difference evaluation: we get different scores for (2), (3), and (4). While the sign of the explanations is equal for all three types, the size and the ratio between them is not. There is no single best way, each has some advantages and some disadvantages.

3 Visualization

To make our explanation method practical we have developed a visualization method called explainVis, which we demonstrate on the well-known Titanic data set. The learning task is to predict the survival of a passenger in the disaster of the Titanic ship. The three attributes report the traveling class (first, second, third, crew), age (adult or child), and gender of the passenger. Altogether there are 2201 instances, of which we randomly selected 50% for learning. Fig. 2 shows explanations for NB and SVM methods for one of the first class, adult, male passengers.

We show information differences (2) on the horizontal axis. Weight of evidence and difference of probabilities produce similar graphs and explanations, but on a different scale. The vertical axis contains names of the attributes on the left-hand side and their values for the chosen instance on the right-hand side. The class probability $p(y|x)$ returned by the method for the given instance x is reported on the top. The length of the thicker bars correspond to the influence of the given attribute values in the model expressed as (2). Positive information difference is given on the right-hand side and negative information difference is on the left-hand side. Thinner bars above the explanation bars indicate the average value of the information difference over all the training instances for the corresponding attribute value. Their purpose is to show trends of particular attributes' values. By comparing the full- and half-height bars the user gets an impression what is the "usual effect" of particular instance values.

For the given instance we observe that in the NB model "status = first" speaks strongly in favor of survival and being male strongly against it, while being adult has a tiny negative influence. Exact probabilities are displayed next to each bar. Thinner average bars mostly agree with that (being male is on average even more dangerous than in this case). The SVM model is more pessimistic about survival of the particular person, giving only 22% chances of survival (NB: 50%). In classification of this case it uses only sex and age attributes which both speak against survival. On average the first

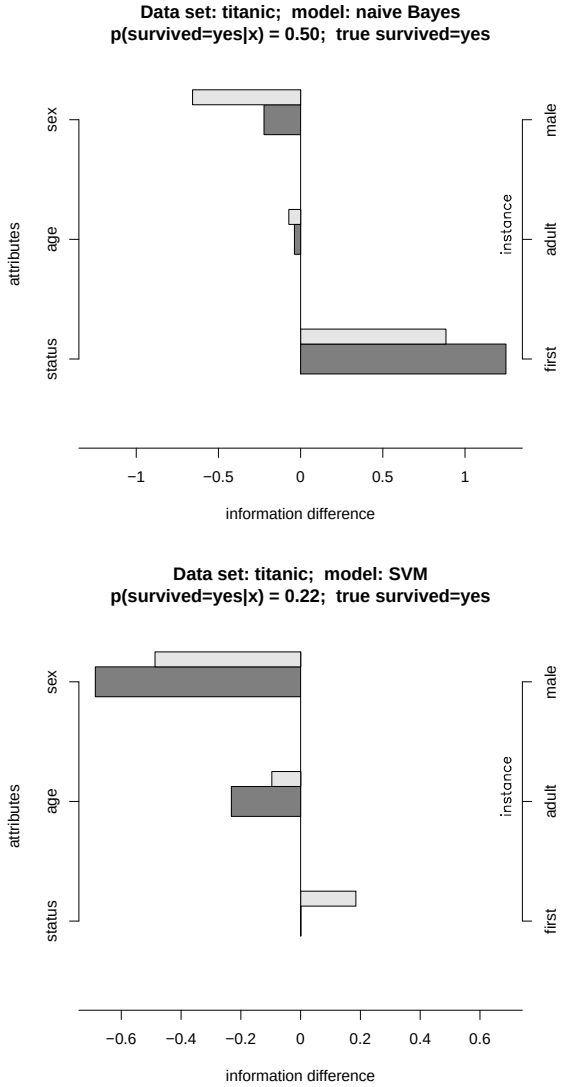

Figure 2: Explanation of NB and SVM models for one of the first class, adult, male passengers in Titanic data set. Explanations for particular instance are depicted with dark bars. Average positive and negative explanations for given attributes' values are presented with light shaded half-height bars above them.

class status has positive impact, adult a tiny negative one, and male a strong negative one.

While in our simple example we present all the attributes, for domains with many attributes we introduce a threshold parameter and only attributes with sufficient impact are displayed. In some application domains (for example in medicine) experts inter-

pret similar graphs as points in favor/against the decision. There it is expected that a total sum of points will be 1 (or 100). Such normalization is also an option in our visualization tool.

To get a more general view of the model we can use explanations for the training data and visualize the averages in a summary form, which shows importance of each feature and its value. One has to be aware that there is no single best knowledge representation and there are many different opinions of which approach and visualization is better. We believe that averaging the explanations over the training data and visualization with a sort of evidence for/against is useful and informative. An example of such visualizations for titanic data set are presented in Fig. 3. On top we see model explanation for NB and on bottom there is explanation of ANN.

In each model explanation graph on the vertical axis all the attributes and their values are listed (each attribute and its values separated by dashed lines). An average negative and positive information difference is presented with the horizontal bar. For attributes (a darker shade bar) an average effect of all its values is given. For both NB and ANN sex plays the most important role, following by status and age. The attributes' values give more precise picture. Both models roughly agree that male, adult, and third class have a negative effect on survival. Also they agree that female, child and first class have a positive effect, but they disagree for second class and crew. NB sees crew as a negative indication of survival, while in neural network it has a slightly positive effect. Second class is a positive indicator for NB, while in ANN it plays double role: a slight positive or a much stronger negative one.

4 Closeness to the model

How can we evaluate explanations when there is no objective measure defined for it? While our approach is practically useful and "makes sense" for the algorithms and data we have investigated, in this Section we also show that the explanations are close to the model: the better the model captures the properties of the problem, the closer are the obtained explanations to the "correct explanations".

Our evaluation scenario includes five different learning algorithms: NB, decision trees (DT), nearest neighbor (kNN), SVM and ANN. We have included decision trees in this evaluation because the learned structure of the tree already provides explanation, and we want to compare our method with it. We constructed several artificial problems. Each problem is tailored to the abilities of one of the learning algorithms and rather difficult for the others. The labels of the instances in these problems can be explained by their attribute values in a clear and unambiguous way and we can therefore talk about correct explanation for each instance. We want to show that the explanations of the best model for the given problem are closer to the correct explanations than the explanations of other models. For this purpose we define the distance between correct explanation and the explanation given by the model. We use Euclidean distance over all the attributes of the explanations of prediction differences (1):

$$d_{exp}(x) = \left(\sum_{i=1}^{a} \left(\frac{1}{2} \left(\frac{\text{trueExpl}_i(x)}{\sum_{i=1}^{a} |\text{trueExpl}_i(x)|} - \frac{\text{predDiff}_i(x)}{\sum_{i=1}^{a} |\text{predDiff}_i(x)|} \right) \right)^2 \right)^{\frac{1}{2}} \tag{7}$$

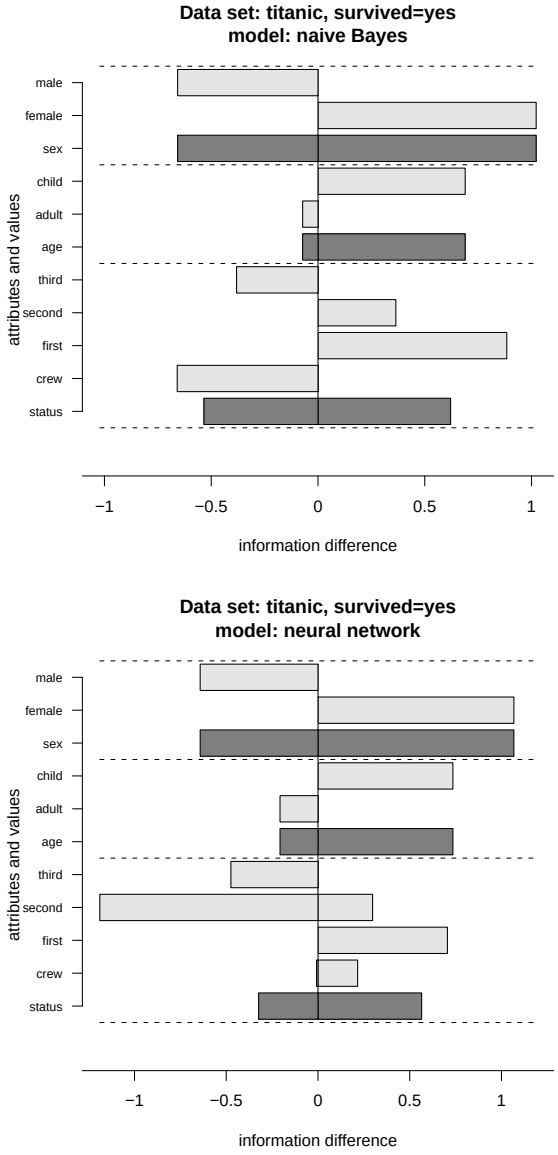

Figure 3: Model explanation of the NB (left-hand side) and ANN (right-hand side) on the Titanic data set. Light bars are average explanation for attributes' values and dark bars are averages (separately for positive and negative scores) over all values of each attribute.

Here $\text{trueExpl}_i(x)$ represents the "true explanation" weight of the contribution of the attribute A_i to the prediction of the instance x. For an attribute A_i the $\text{trueExpl}_i(x)$ and $\text{predDiff}_i(x)$ can be either positive (if the attribute's value supports the selected

class) or negative (if the attribute's value supports some other class). We want all the instances to be equally represented so the absolute values of the model's explanations are normalized to 1 for each instance: we divide the $\text{trueExpl}_i(x)$ and $\text{predDiff}_i(x)$ with the sum of their respective absolute values, and multiply them by $\frac{1}{2}$.

Let $\overline{d_{exp}}$ be an average of (7) over all the testing instances, than this is an indicator of how close the model's explanations are to the true explanations.

Below we report results for the weight of evidence ($\text{predDiff}_i(x)$ is replaced by the weight of evidence $\text{WE}_i(x)$ (3) in the distance (7)). Information difference (2) and difference of probabilities (4) produce similar results. All the explanations are computed for the class 1 in each problem domain. The data sets used are:

condInd The class value is a boolean variable with 50% probability of 1. The four important conditionally independent attributes have the same value as class in 90, 80, 70 and 60% of the cases. There are also 4 attributes unrelated to the class (random) in this data set. Because of conditionally independent attributes the problem suits NB method. The correct explanation (with sum normalized to 1) would assign 0.4, 0.3, 0.2, and 0.1 to important attributes and 0 to unrelated ones.

xor We generated a data set with three important attributes describing a parity problem of order 3. We added noise to the class value by reverting it in 10% of the cases. There are also 3 attributes unrelated to the class in this data set. The problem can be best captured by a decision tree, where each path from the root to the leaf contains test on both important attributes. The true explanation assigns 0.333 to each of the important attributes and 0 to unrelated ones.

groups Two important attributes I_1 and I_2 and class are visualized in Fig. 4 (values are scattered around group centers, which define class values). We added also two attributes unrelated to the class. Instances with class values 0, 1, and 2 are represented with circles, triangles and lines, respectively. The true explanation assigns 0.5 to each important attribute and 0 to unrelated ones. The kNN algorithm is the most appropriate for this problem.

cross This problem with two important attributes I_1 and I_2 is visualized in Fig. 5. The class value 1 (triangles) is assigned to instances where $(I_1 - 0.5)(I_2 - 0.5) > 0$. We also added 4 attributes unrelated to the class. The true explanation assigns 0.5 to each of the important attributes and 0 to unrelated ones. The SVM with polynomial kernel of order 2 is best suited for this problem as it can linearly separate the instances.

chess Two important attributes I_1 and I_2 and class are visualized in Fig. 6. We have a 4×4 chessboard, with circles and triangles representing class value 0 and 1, respectively. We added also two attributes unrelated to the class. The true explanation assigns 0.5 to each important attribute and 0 to unrelated ones. The ANN algorithm is the most appropriate for this problem.

For each data set we generated 1000 instances for training and another 1000 for testing of explanations. Cross-validation and other sampling techniques are not necessary

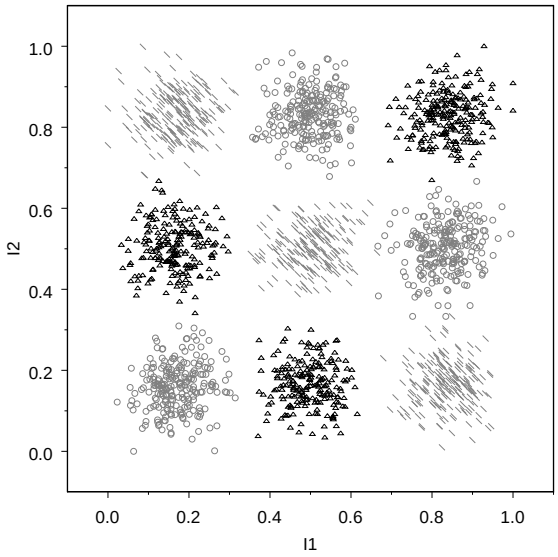

Figure 4: Visualization of two important attributes in the *groups* data set. Circles, triangles, and lines represent class values 0, 1, and 2.

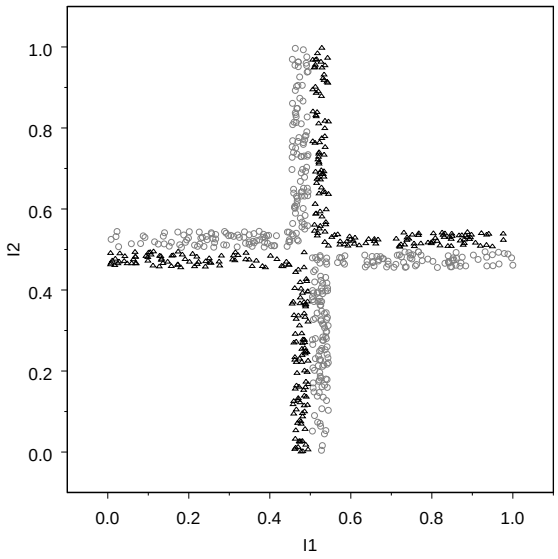

Figure 5: Visualization of two important attributes in *cross* data set. Class values 0 and 1 are visualized as circles and triangles, respectively.

in our case. For computation of $p(y|x\backslash A_i)$ in explanations we used (6)[3]. For numeri-

[3]For NB using (6) is not necessary as $p(y|x\backslash Ai)$ can be obtained by excluding A_i from the computation.

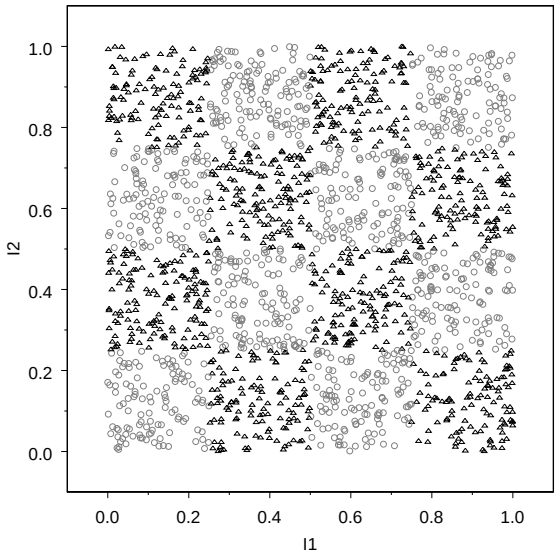

Figure 6: Visualization of two important attributes in *chess* data set. Class values 0 and 1 are visualized as circles and triangles, respectively.

cal attributes we used discretization to calculate (6). We used the prior knowledge of each toy problem and selected the right number of intervals. We used discretization with equal width of intervals (3 intervals for *groups*, 2 for *cross*, and 4 intervals for attributes in *chess* data set).

Table 3 presents for each of the data sets and each method its accuracy (acc), area under the ROC curve (AUC) and the average distance to the true explanation ($\overline{d_{exp}}$). Other settings of the parameters could give better accuracy and AUC score for some of the models (especially for SVM and ANN), but it is not our aim to compare the classifiers' accuracies but rather to test closeness of the explanations to the models.

Note that for each data set (in one column) the most suitable algorithm achieves the highest accuracy and AUC score. It also has the lowest average distance to the true explanation (in bold type). This is a clear confirmation that our explanations are closely following the models; if any of the above model captures the underlying structure of the problem, the explanations reflect that. It is also a confirmation of suitability for a wide variety of different models.

4.1 Redundant attributes

Machine learning algorithms use different strategies when dealing with redundant and highly correlated attributes. This issue is mostly tackled through feature subset selection and feature weighting [6], which keeps only useful features and also tries to eliminate redundant features. If redundant features are not eliminated before learning,

However, as shown in the Appendix, the two approaches are equivalent

Table 3: Performance and average distances to the true explanation for five classification methods on five data sets.

method		condInd	xor	group	cross	chess
NB	acc	0.90	0.51	0.35	0.50	0.50
	AUC	0.96	0.51	0.50	0.50	0.50
	$\overline{d_{exp}}$	**0.06**	0.39	0.46	0.45	0.47
DT	acc	0.89	0.90	0.33	0.52	0.52
	AUC	0.95	0.90	0.50	0.56	0.50
	$\overline{d_{exp}}$	0.17	**0.01**	0.35	0.33	0.35
kNN	acc	0.86	0.90	0.99	0.55	0.71
	AUC	0.93	0.90	0.83	0.59	0.78
	$\overline{d_{exp}}$	0.16	0.10	**0.08**	0.40	0.33
SVM	acc	0.89	0.58	0.66	0.98	0.53
	AUC	0.95	0.52	0.76	0.99	0.52
	$\overline{d_{exp}}$	0.12	0.39	0.22	**0.04**	0.42
ANN	acc	0.89	0.90	0.98	0.95	0.84
	AUC	0.92	0.90	0.82	0.98	0.90
	$\overline{d_{exp}}$	0.27	0.09	0.09	0.08	**0.16**

they may have undesirable effects on the learned models. A particular strategy for dealing with redundant attributes used by a model is reflected in explanations. Below we discuss this for extreme case when we have two or more identical attributes.

The NB classification is strongly affected by identical attributes, as their contributions are repeated (multiplied). The explanation gives all copies of the same attribute equal utility. The decision tree selects one copy of the attribute in one path from root to the leaf and ignores the others. Here the explanation gives a positive utility to the former attribute and zero to the others. The kNN is sensitive to the distance, and as the attribute space is deformed by additional copies of attributes, the classification performance gradually deteriorates and this is also reflected in the explanations. SVM and ANN are also sensitive to redundant attributes, but to a lesser extent. How this affects explanations is instance dependent, as certain parts of problem space may be treated differently by the model. We illustrate this on *groups* problem (see Fig. 4) with two most suitable models for it, namely, kNN and ANN. We duplicated both important attributes, I_1 and I_2. For a particular instance from class 1 (triangles) we show explanations on Fig. 7.

Dark-shaded bars present explanations for the selected instance. Light-shaded bars illustrate average positive and negative explanations of instances from the same value interval. The kNN (on top) assigns both copies of important attributes the same utility. ANN (on bottom) also assigns the same utility to both copies of I_2, but uses almost no information from the second copy of I_1, which results in different instance explanation for I_1 and I_2. Average explanations show a more general picture of this data set: both copies of important attributes play important role in kNN and ANN. Unimportant attributes R_1 and R_2 affect kNN model much stronger than ANN.

The same thing is confirmed with visualization of model level explanations on Fig. 8. As attributes are numerical in this data set, the explanations are averaged over intervals of values. High scores are assigned to important attributes and their values, and much lower to unimportant ones. We see that kNN is affected by redundant attributes

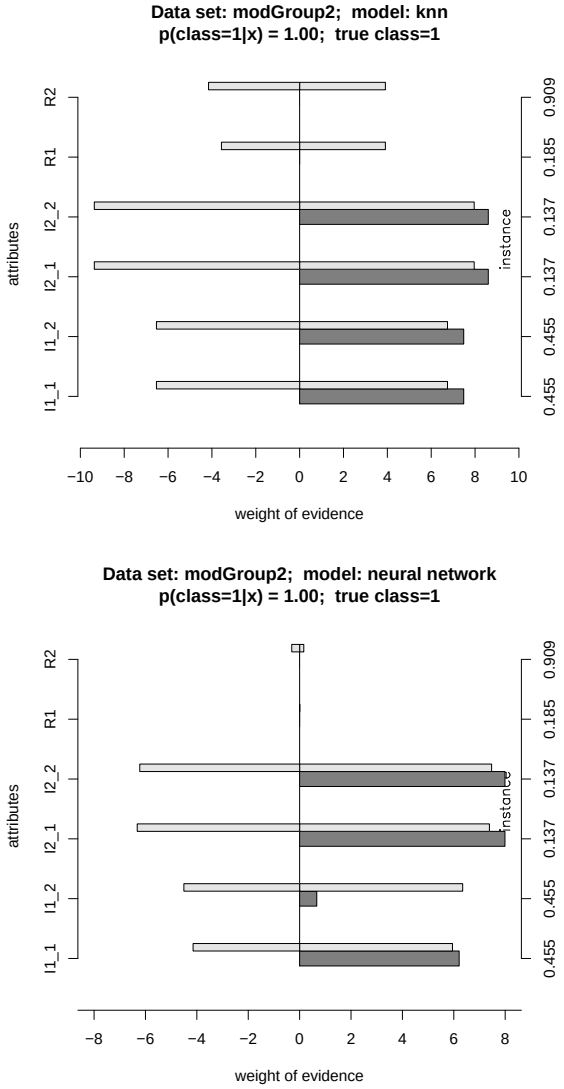

Figure 7: Explanation for one of the instances in the groups data set with duplicated important attributes. The instance is labeled with class 1 (triangle). On the top explanations are for kNN model and for ANN they are on the bottom.

much more than ANN, as the explanation scores for R_1 and R_2 are much larger. The reason for that is in distortion of distances caused by multiple copies of attributes.

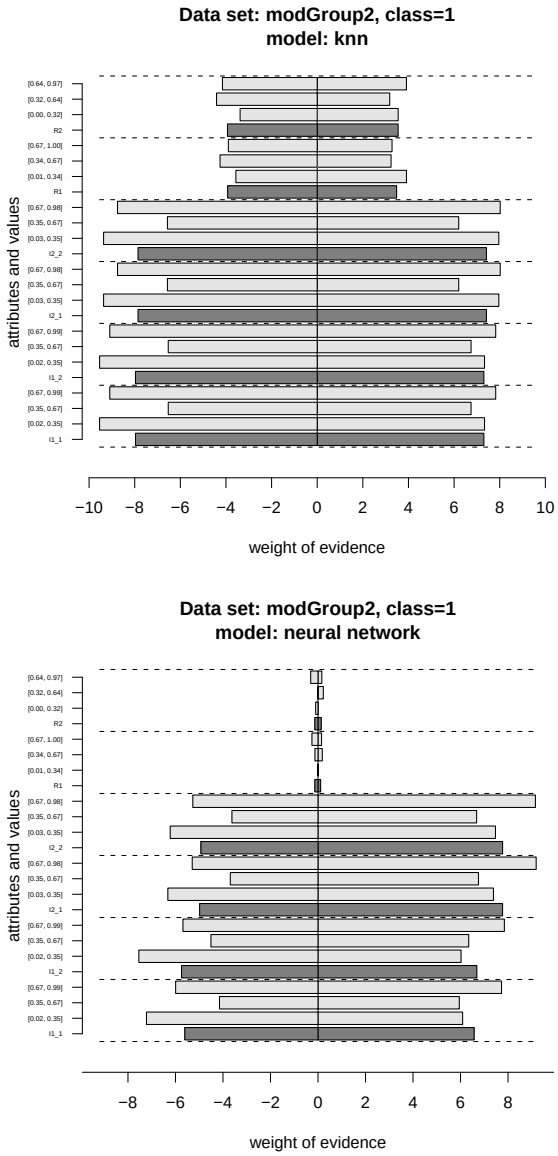

Figure 8: Model explanations for kNN (top) and ANN (bottom) on the groups data set with duplicated important attributes. As attributes are numerical, explanations are averaged and presented for intervals.

4.2 Notes on Use

Explanations for numerical attributes are somewhat sensitive to discretization used for computation of $p(y|x\backslash A_i)$ in (6). For example, when we used 3 intervals (instead of 2)

in the *cross* data set the average distance for SVM and ANN increased from 0.04 and 0.08 to 0.05 and 0.12, showing a slight decrease in the quality of explanation. NB, DT, and kNN have approximately the same (poor) performance with both discretizations. Indeed, if discretization is allowed to blur the information of an attribute, this is likely to reduce the quality of the explanation. If there is no other prior knowledge available, our advice is to use more fine grained discretization. With more intervals there is less opportunity to cover up the attribute's information. For example, with 15 intervals in the *cross* data set the average distance of SVM and ANN is again back at 0.04 and 0.08.

Note also that the computation of $p(y|x\backslash A_i)$ with (6) is not the only way. For NB it is better to drop A_i from the computation and for neural networks which usually encode discrete attributes as several binary inputs we can just simulate an empty input. Other models may have their particular solutions, simpler than (6). One such solution for probabilistic radial basis function networks which exploits a marginalization property of the Gaussian distribution is presented in [7].

It may be the case that for some poorly calibrated classifier the probabilities $p(y|x)$ and $p(y|x\backslash A_i)$ computed in classification are disproportionate to their true probability. Our explanation method which relies on this differences could be disproportionate in that case as well. We haven't observed that in any of our experiments, but this is a potential problem and can be prevented by calibration of the classifiers. As reported in [8] boosting and SVM are particularly prone to this defect, while neural networks are already perfectly calibrated.

Our experience in classification shows that all three interpretations: information difference (2), weight of evidence (3), and difference of probabilities (4) perform quite similarly across different problems, especially if we normalize the effects to 1 or 100 as explained in Section 3. A notable exception are the problems where the attributes are conditionally independent (as in condInd data set); for such problems the weight of evidence gives explanations which are proportional to their true effect, while information difference (which uses logarithms directly) gives nonlinear explanations. For this reason we recommend weight of evidence as a default choice.

All the learning algorithms used are from the R Project (http://www.r-project.org/), packages e1071, nnet, kknn, RWeka, and CORElearn. The sources code for the generation of probabilities in classification, the data sets, as well as the visualization module can be obtained from the authors.

5 Related Work

Explanation of prediction is straightforward for symbolic models such as decision trees, decision rules and inductive logic programming, where the model itself gives an overall transparent knowledge in a symbolic form. Therefore, to obtain the explanation of a prediction, one simply has to read the corresponding rule in the model. Whether such explanation is comprehensive in the case of large trees and rule sets is questionable.

For non-symbolic models there is no such possibility. Techniques for extracting explicit (if-then) rules from black box models (such as ANN) are described in [9–12]. In [13] a black box model is approximated with a symbolic model, such as decision tree, by sampling additional training instances from the model. Extraction of fuzzy

rules from trained ANN using interval propagation is suggested in [14], while an approach which can provably extract sound and also nonmonotonic rules was presented in [15]. In [9] a search-based method has been proposed suitable for ANN with binary units only, while application of the method proposed in [11] requires discretization of the hidden unit activations and pruning of the ANN architecture. Främling [16] explains neural network behavior by using contextual importance and utility of attributes which are defined as the range of changes of attribute and class values. Importance and utility can be efficiently computed only in a special INKA network. Our approach is based on marginalization and can be efficiently computed for any probabilistic classifier. For a more complete review of knowledge extraction methods from ANN we refer the reader to [17], where taxonomy and criteria for evaluation of rule extraction methods from ANN are introduced, and [18], which concentrates mostly on finite-state machines extracted from recurrent ANN but covers also other approaches.

Some non-symbolic models enable the explanation of their decisions in the form of weights associated with each attribute. A weight can be interpreted as the proportion of the information contributed by the corresponding attribute value to the final prediction. Such explanations can be easily visualized. For logistic regression a well known approach is to use nomograms, first proposed in [19]. In [20] nomograms were developed for SVM, but they work only for a restricted class of kernels and cannot be used for general non-linear kernels.

SVM can also be visualized using projections into lower dimensional subspaces [21, 22], or using self-organizing maps from unsupervised learning [23]. These methods concentrate on visualization of separating hyperplane and decision surface. The visualization of individual instances to lower dimensional projections and their position relative to decision surface can be considered a sort of explanation.

The naive Bayesian classifier is able to explain its decisions as the sum of information differences [24]. A straightforward visualization of NB was used in [25] while in [26] nomograms were developed for visualization of NB decisions. We generalize the NB list of information differences to a list of attribute weights for any prediction model. For classification we propose three variants of weights and elaborate in more detail the weights based on information differences and the weight of evidence.

In ExplainD framework [27] the weight of evidence and visualizations similar to ours are used, but the explanation approach is limited to (linear) additive models, while ours can be used for all probabilistic models and is not restricted to the weight of evidence.

In the tools accompanying his Random Forests algorithm [28], Breiman has used bootstrap sampling and random permutation of values to visualize the importance of features, outliers, and also the importance of features for prediction of individual instances. Due to specifics of the techniques used, the approach is limited to Random Forests.

Madigan et al. [29] in their belief networks use each (binary or multivalued discrete) attribute as a node in the graph. By computing ”evidence flows” in the network it is possible to explain its decisions.

A marginal effect of an attribute is defined as the partial derivative of the event probability with respect to the attribute of interest. A more direct measure is the change in predicted probability for a unit change in the attribute. Marginal effects are used

for explanation of specific models where the marginal distribution can be estimated e.g., in logistic regression [30]. They are used also to determine causal relationships, for example, in marginal structural models [31]. These models expect user's input of suspected dependencies (confounders) and do not deal with explanations on the instance level.

6 Conclusions

We present an approach to explanation of predictions, which generates explanations of predictions for individual instances. The presented approach is general and can be used with any classification method that outputs class probabilities. The method is based on the decomposition of model's predictions (marginal prediction). These decompositions can be interpreted as local gradients and are used to identify the individual contribution of each attribute. We have empirically shown that proposed explanations closely follow the models for five different classification methods. This is shown in an experiment confirming that explanations of better models correctly reflect the learned concept. With the presented explainVis visualization method we see effects of various features on instance prediction. The visualization of average explanations gives information about attributes and their values (intervals for numeric attributes) on the level of the model. Both types of graphs enables comparison of different prediction methods.

Currently we are using the presented techniques to generate explanations in regression and also to go beyond the model explanation to the domain explanation. In particular, we are interested in the decomposition of predictions through sequential generation of models. Another issue for further work is the problem with explanation of instances where the change in more than one attribute value at once is needed to affect the predicted value, as mentioned in Section 2.3. Namely, our methodology is not able to correctly evaluate the utility of attribute values that appear in such instances. This problem can be overcome only by an extensive search of pairs, triples, etc. of attribute values. The exhaustive search is of course unfeasible and a heuristic search reduction is necessary. The search shall probably be combined with the sequential generation of models, i.e. models that use different subsets of attributes.

Appendix

We prove that for NB classifier Eq. (6) is sound, i.e., it is equivalent to excluding attribute A_i from the computation.

NB calculates the probability of class y for a given instance $x = (A_1 = a_{k_1}, ..., A_a = a_{k_a})$ with the following formula, where it assumes the conditional independence of attributes given the class:

$$p(y|x) = p(y) \prod_{j=1}^{a} \frac{p(A_j = a_{k_j}|y)}{p(A_j = a_{k_j})} \tag{8}$$

When attribute A_i has unknown value, it is excluded from the computation:

$$p(y|x\backslash A_i) = p(y)\prod_{j\neq i}\frac{p(A_j = a_{k_j}|y)}{p(A_j = a_{k_j})} \tag{9}$$

On the other hand, if we replace the value a_{k_i} of attribute A_i with value a_s, we get:

$$p(y|x \leftarrow A_i = a_s) = p(y)\frac{p(A_i = a_s|y)}{p(A_i = a_s)}\prod_{j\neq i}\frac{p(A_j = a_{k_j}|y)}{p(A_j = a_{k_j})} \tag{10}$$

Therefore we have:

$$p(A_i = a_s)p(y|x \leftarrow A_i = a_s) =$$

$$\begin{aligned} &= p(y)p(A_i = a_s|y)\prod_{j\neq i}\frac{p(A_j = a_{k_j}|y)}{p(A_j = a_{k_j})} \\ &= p(A_i = a_s|y)p(y|x\backslash A_i) \end{aligned}$$

and finally we get:

$$\sum_{s=1}^{m_i} p(A_i = a_s)p(y|x \leftarrow A_i = a_s) = \tag{11}$$

$$\begin{aligned} &= p(y|x\backslash A_i)\sum_{s=1}^{m_i} p(A_i = a_s|y) \\ &= p(y|x\backslash A_i) \end{aligned}$$

□

Acknowledgement

The authors were supported by Slovenian Research Agency (ARRS) through the research programme P2-0209.

References

[1] C. E. Shannon, "A mathematical theory of communications," *The Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[2] I. J. Good, *Probability and the weighing of evidence*. C. Griffin London, 1950.

[3] D. W. Hosmer and S. Lemeshow, *Applied logistic regression, 2nd edition*. New York: Wiley, 2000.

[4] A. Tversky and D. Kahneman, "Judgement under uncertainty: heuristics and biases," *Science*, vol. 185, pp. 1124–1130, 1974.

[5] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[7] M. Robnik-Šikonja, A. Likas, C. Constantinopoulos, and I. Kononenko, "An efficient method for explaining the decisions of the probabilistic RBF classification network," University of Ljubljana (FRI) and University of Ioannina (DCS), Tech. Rep., 2007.

[8] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Machine Learning Conference*, L. De Raedt and S. Wrobel, Eds. ACM Press, 2005.

[9] G. G. Towell and J. W. Shavlik, "Extracting refined rules from knowledge-based neural networks," *Machine Learning*, vol. 13, no. 1, pp. 71–101, 1993.

[10] M. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *International Conference on Machine Learning*, 1994, pp. 37–45.

[11] R. Setiono and H. Liu, "Understanding neural networks via rule extraction," in *Proceedings of IJCAI'95*, 1995, pp. 480–487.

[12] S. Thrun, "Extracting rules from artifical neural networks with distributed representations," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. The MIT Press, 1995, pp. 505–512.

[13] M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., vol. 8. The MIT Press, 1996, pp. 24–30.

[14] V. Palade, C.-D. Neagu, and R. J. Patton, "Interpretation of trained neural networks by rule extraction," in *Proceedings of the International Conference, 7th Fuzzy Days on Computational Intelligence, Theory and Applications*. London, UK: Springer-Verlag, 2001, pp. 152–161.

[15] A. S. d'Avila Garcez, K. Broda, and D. M. Gabbay, "Symbolic knowledge extraction from trained neural networks: a sound approach," *Artificial Intelligence*, vol. 125, no. 1-2, pp. 155–207, 2001.

[16] K. Främling, "Explaining results of neural networks by contextual importance and utility," in *Proceedings of the AISB'96 conference*, 1996.

[17] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Systems*, vol. 8, no. 6, pp. 373–384, 1995.

[18] H. Jacobsson, "Rule extraction from recurrent neural networks: A taxonomy and review," *Neural Computation*, vol. 17, no. 6, pp. 1223–1263, 2005.

[19] J. Lubsen, J. Pool, and E. van der Does, "A practical device for the application of a diagnostic or prognostic function," *Methods of Information in Medicine*, vol. 17, pp. 127–129, 1978.

[20] A. Jakulin, M. Možina, J. Demšar, I. Bratko, and B. Zupan, "Nomograms for visualizing support vector machines." in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, R. Grossman, R. Bayardo, and K. P. Bennett, Eds. ACM, 2005, pp. 108–117.

[21] D. Caragea and V. Cook, Dianne Honavar, "Towards simple, easy-to-understand, yet accurate classifiers," in *Third IEEE International Conference on Data Mining, ICDM 2003.*, 2003, pp. 497– 500.

[22] F. Poulet, "Svm and graphical algorithms: A cooperative approach," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004, pp. 499–502.

[23] L. Hamel, "Visualization of support vector machines with unsupervised learning," in *Proceedings of 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2006.

[24] I. Kononenko, "Inductive and bayesian learning in medical diagnosis," *Applied Artificial Intelligence*, vol. 7, no. 4, pp. 317–337, 1993.

[25] B. Becker, R. Kohavi, and D. Sommereld, "Visualizing the simple bayesian classier," in *KDD Workshop on Issues in the Integration of Data Mining and Data Visualization*, 1997.

[26] M. Možina, J. Demšar, M. W. Kattan, and B. Zupan, "Nomograms for visualization of naive bayesian classifier." in *Knowledge Discovery in Databases: PKDD 2004*, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Springer, 2004, pp. 337–348.

[27] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. Macdonell, and J. Anvik, "Visual explanation of evidence with additive classifiers," in *Proceedings of AAAI'06*. AAAI Press, 2006.

[28] L. Breiman, "Random forests," *Machine Learning Journal*, vol. 45, pp. 5–32, 2001.

[29] D. Madigan, K. Mosurski, and R. G. Almond, "Graphical explanation in belief networks," *Journal of Computational and Graphical Statistics*, vol. 6, no. 2, pp. 160–181, 1997.

[30] T. F. Liao, *Interpreting Probability Models: Logit, Probit, and Other Generalized Linear Models*. Sage Publications Inc., 1994.

[31] J. M. Robins, M. A. Hernn, and B. Brumback, "Marginal structural models and causal inference in epidemiology," *Epidemiology*, vol. 11, no. 5, pp. 550–560, 2000.