Technical report, University of Ljubljana (FRI) and University of Ioannina (DCS), 2007

An Efficient Method for Explaining the Decisions of the Probabilistic RBF Classification Network

Marko Robnik-Šikonja[1], Aristidis Likas[2],
Constantinos Constantinopoulos[2], Igor Kononenko[1]

[1]University of Ljubljana,
Faculty of Computer and Information Science,
Tržaška 25, 1001 Ljubljana, Slovenia
{marko.robnik,igor.kononenko}@fri.uni-lj.si

[2]University of Ioannina, Department of Computer Science,
GR 45110 Ioannina, Greece
{arly,ccostas}@cs.uoi.gr

Abstract

Transparency of the classification is an important requirement for many application fields. Robnik-Šikonja and Kononenko have recently presented a general method for explaining predictions for individual instances based on decomposition of a model's predictions on individual contributions of each attribute. In this work we apply the general methodology in the context of the probabilistic radial basis function (PRBF) network which constitutes an effective black box classifier. Nevertheless, like most other neural network models it is not straightforward to obtain explanations for the PRBF decisions. By exploiting the marginalization property of the Gaussian distribution, we demonstrate how this explanation technique can be implanted into PRBF to explain the model's decisions for new unlabeled cases. We also present a visualization technique that works both for single instances as well as for the attributes and their values, thus providing a valuable tool for inspection of the otherwise opaque model.

Keywords: model explanation, comprehensibility, probabilistic RBF network, model visualization, classification

1 Introduction

In many areas where machine learning and data mining models are applied, their transparency is of crucial importance. For example, in medicine the practitioners are far more interested in comprehension of the decision process, explanation of the model's

behavior for a given new case, and importance of the diagnostic features, than in classification accuracy the model may offer. The same is true for other areas where knowledge discovery dominates prediction accuracy.

Research in statistics, data mining, pattern recognition and machine learning is mostly focused in the other direction, and as a result we have many excellent prediction methods, which are approaching the theoretically achievable prediction accuracy. Some of the most successful and popular approaches are based on support vector machines (SVM) and artificial neural networks (ANN) as well as on ensemble methods such as boosting. Regrettably none of these approaches is offering an intrinsic introspection into their decision process, or offering an explanation for labeling the new case.

Recently, in [24] a general explanation method has been presented that is in principle independent of the model - the model can be either transparent (e.g., decision trees and rules) or a black box (such as SVM, ANN and classifier ensembles) and can be used with all classification models that output probabilities. It relies on the decomposition of model's predictions based on individual contributions of each attribute. The generated explanations closely follow the learned model and enable its visualization for each instance separately. The decompositions simulate the lack of knowledge about the value of decomposed attribute within the model. To achieve this for the general case, in [24] a sort of averaging over all possible feature's values is recommended, acknowledging that there may be better solutions for specific models. We present one such solution for the probabilistic radial basis function network (PRBF), which is an effective black box classifier [28, 5]. By implanting this explanation technique into PRBF, we demonstrate how we can explain the model's decisions for new unlabeled cases. In addition, using this explanation technique a graphical description of the otherwise opaque model is presented.

The probabilistic RBF classifier [28, 29] is a special case of the RBF network [3] that computes at each output unit the density function of a class. It adopts a cluster interpretation of the basis functions, where each cluster can generate observations of any class. This is a generalization of a Gaussian mixture model [19, 3], where each cluster generates observations of only one class. In [5] an incremental learning method based on Expectation-Maximization (EM) for supervised learning is proposed that provides classification performance comparable to SVM classifiers.

There exist a large corpora of techniques for extracting knowledge from ANN [10, 30, 6, 25, 27, 1, 8, 21, 14]. These techniques are mostly specific to ANN and do not allow comparison between different learning models. They try to extract knowledge from the ANN and in the form of rules or finite state machines which may become incomprehensible for large networks and complicated problems. Most of those techniques are based on assumptions about the ANN architecture (a small pruned ANN architecture is usually preferable) and the activation of the ANN units (discrete activation values are usually assumed). Our approach is simpler: we provide explanations for the classifications of individual instances and form explanations which can be easily visualized in a graphical form. In this work we apply the explanation method to the PRBF network and we show that the decompositions required for the explanations can be computed in a simple and efficient way.

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 representing 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.

In Sect. 2 we present the explanation principle, and give relevant implementation details. In Sect. 3 we describe PRBF and the marginalization property of the Gaussian distribution which is exploited for explanation. In Sect. 4 we use an artificial data set to illustrate that explanations are close to the "true explanations" for the PRBF model and present two visualizations: one for individual instances and one which exploits average explanations to demonstrate the effects that the attributes and their values have in a given model. Section 5 places the presented explanation into the context of rule extraction taxonomy from ANN, used by [1, 14] and reviews the related work. Section 6 summarizes and provides some ideas for further work.

2 Explanation Method

In the explanation method proposed in [24] two levels of explanation are identified: 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. 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.

The idea of explanations proposed in [24] is to observe the relationship between the features and the predicted value by monitoring the causal effect on the predicted values due to the lack of knowledge of a feature's value. By measuring this effect one can reason about the importance of the attribute's values. To see the effect the attributes' values have on the prediction of an instance, they decompose the prediction on individual attributes' values and define the model's probability $p(y|x\backslash A_i)$, as the

model's probability for instance x and class y 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 $p(y|x)$ and $p(y|x\backslash A_i)$ we get insight into the importance of event $A_i = a_k$. If the difference between $p(y|x)$ and $p(y|x\backslash A_i)$ is large, the event $A_i = a_k$ 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 explanations is therefore the decomposition

$$\text{probDiff}_i(y|x) = f(p(y|x), p(y|x\backslash A_i)). \tag{1}$$

For example, one possible function f could be simply the probability difference:

$$\text{probDiff}_i(y|x) = p(y|x) - p(y|x\backslash A_i) \tag{2}$$

By restricting this reasoning to the model, the explanations are provided also for multi-attribute dependencies. In such events each of the dependent attributes A_i affects the prediction and so also the score $\text{probDiff}_i(y|x)$, 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. For example, let's have a binary domain with three important (A_1, A_2, and A_3) and one irrelevant attribute (A_4). Let the learned model correctly expresses parity (xor) relation of three attributes $C = A_1 \oplus A_2 \oplus A_3$. It then classifies an instance x ($A_1 = 1$, $A_2 = 0$, $A_3 = 1$, $A_4 = 1$) as class $C = 0$, and assigns, $p(C = 0|x) = 1$. When explaining classification for this particular instance $\text{probDiff}_i(C = 0|x)$ we have to estimate $p(C = 0|x\backslash A_1)$, $p(C = 0|x\backslash A_2)$, $p(C = 0|x\backslash A_3)$, and $p(C = 0|x\backslash A_4)$. Without the knowledge about the values of each of the attributes A_1, A_2, and A_3, the model cannot determine the class value and let us assume that it correctly returns $p(C = 0|x\backslash A_1) = p(C = 0|x\backslash A_2) = p(C = 0|x\backslash A_3) = 0.5$. Using 2 the values $\text{probDiff}_1(C = 0|x) = \text{probDiff}_2(C = 0|x) = \text{probDiff}_3(C = 0|x) = 0.5$ so the explanations are positive for each of the three important attributes. Since the lack of knowledge about the value of irrelevant attribute does not change the classification $p(C = 0|x\backslash A_4) = 1$), and $\text{probDiff}_4(C = 0|x) = 0$ i.e., the explanation of the irrelevant attribute is zero.

The main weaknesses of this methodology is that it 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. For example, in the model expressing $C = A_1 \vee A_2$, for instance x ($A_1 = 1$, $A_2 = 1$), probability of class value 1 is $p(C = 1|x) = 1$. Decompositions on single attribute would make no difference in probability $p(C = 1|x\backslash A_1) = p(C = 1|x\backslash A_2) = 1$ and $\text{probDiff}_1(C = 0|x) = \text{probDiff}_2(C = 0|x) = 0$, and both A_1 and A_2 are incorrectly considered irrelevant for classification of this instance. However, for this model other combinations of values for A_1 and A_2 are explained correctly. 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. This is one of the key issues for further work as described in [24].

For the evaluation of the prediction difference (1) there are several options for function f: information gain [26], weight of evidence [11], or using difference of proba-

bilities directly as in Eq. (2). Due to its favorable properties (symmetry) weight of evidence seems the most promising and we use it in this work.

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 of attribute A_i for class value y is defined as the log odds of the class value y with the knowledge about the value of A_i and without it:

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

With respect to information gain, the weight of evidence provides an alternative view on information with similar properties (sometimes favorable, e.g. symmetry). In logistic regression where it is commonly used [13] it is referred to as log odds-ratio. To get the explanation factors we have to evaluate (3). To compute factor $p(y|x)$ we just classify the instance x with the model. To compute the factors $p(y|x\backslash A_i)$ 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.) which indeed does not contain any information about A_i . This method is appropriate only for modeling techniques which handle unknown values naturally (e.g., the Naive Bayesian classifier 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, each method has its own internal mechanism for handling unknown values. 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 model dependent treatment of unknown values, a technique is suggested in [24] that simulates the lack of information about A_i with several predictions. For nominal attributes the actual value $A_i = a_k$ is replaced with all possible values of A_i, and each prediction is weighed by the prior probability of the value

$$p(y|x\backslash A_i) = \sum_{s=1}^{m_i} p(A_i = a_s|x\backslash A_i) p(y|x \leftarrow A_i = a_s) \tag{4}$$

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

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. The simplification is used for the prior probability $p(A_i = a_s)$ which implies that (5) is only an approximation. For numerical attributes the procedure is similar. A discretization splits the values of A_i into sub-intervals. The middle points of these sub-intervals are taken as the representative replacement values in (5) 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. 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 this work, we show that in the case of PRBF networks, approximation (5) is not needed and the lack of information about attribute values can be handled in a way which is computationally efficient (does not require relearning of the model or several predictions) and sound (we get exactly the same results by learning without the knowledge of the given attribute). A description of PRBF networks and how they can be marginalized to treat missing attributes is presented in the next section.

3 Probabilistic RBF Networks

Consider a classification problem with c classes y_k $(k = 1,\ldots,c)$ and input instances $x = (A_1,\ldots,A_a)$. For this problem, the corresponding PRBF classifier has a inputs and c outputs, one for each class. Each output provides and estimate of the probability density $p(x|y_k)$ of the corresponding class y_k.

Assume that we have M components (hidden units), each one computing a probability density value $f_j(x)$ of the input x. In the PRBF network all component density functions $f_j(x)$ are utilized for estimating the conditional densities of all classes by considering the components as a common pool [28]. Thus, for each class a conditional density function $p(x|y_k)$ is modeled as a mixture model of the form:

$$p(x|y_k) = \sum_{j=1}^{M} \pi_{jk} f_j(x), \quad k = 1,\ldots,c, \tag{6}$$

where the mixing coefficients π_{jk} are probability vectors; they take positive values and satisfy the following constraint:

$$\sum_{j=1}^{M} \pi_{jk} = 1, \quad k = 1,\ldots,c. \tag{7}$$

Once the outputs $p(x|y_k)$ have been computed, the class of data point x is determined using the Bayes rule, i.e. x is assigned to the class with maximum posterior $p(y_k|x)$ computed by

$$p(y_k|x) = \frac{p(x|y_k)P_k}{\sum_{\ell=1}^{c} p(x|y_\ell)P_\ell} \tag{8}$$

The class priors P_k are usually computed as the percentage of training instances belonging to class y_k.

In the following, we assume the Gaussian component densities of the general form:

$$f_j(x) = \frac{1}{(2\pi)^{a/2}|\Sigma_j|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right\} \tag{9}$$

where $\mu_j \in \Re^a$ represents the mean of component j, while Σ_j represents the corresponding $a \times a$ covariance matrix. The whole adjustable parameter vector of the model consists of the mixing coefficients π_{jk} and the component parameters (means μ_j and covariances Σ_j).

It is apparent that the PRBF model is a special case of the RBF network, where the outputs correspond to probability density functions and the output layer weights are

constrained to represent the prior probabilities π_{jk}. Furthermore, the separate mixtures model [19] can be derived as a special case of PRBF by setting $\pi_{jk} = 0$ for all classes k, except for the class that the component j belongs to. Training of the PRBF network is based on the EM algorithm for likelihood maximization [28, 5].

3.1 Marginalization of PRBF

A notable convenient characteristic of the Gaussian distribution is the *marginalization property*: if the joint distribution of a set of random variables $S = \{A_1, \ldots, A_a\}$ is Gaussian with mean μ and covariance matrix Σ, then for any subset A of these variables, the joint distribution of the subset $F = S - A$ of the remaining variables is also a Gaussian. The mean $\mu_{\setminus A}$ of this Gaussian is obtained by removing from μ the components corresponding to the variables in subset A and covariance matrix $\Sigma_{\setminus A}$ obtained by removing the rows and columns of Σ also corresponding to the variables in subset A. Therefore, if we know the mean and covariance of the joint distribution of a set of variables, we can immediately obtain the distribution of any subset of these variables.

For an input $x = (A_1 = v_1, \ldots, A_a = v_a)$ each output $p(x|y_k)$, $k = 1, \ldots, c$ of the PRBF is computed as a mixture of Gaussians:

$$p(x|y_k) = \sum_{j=1}^{M} \pi_{jk} N(x; \mu_j, \Sigma_j) \tag{10}$$

Consequently, based on the marginalization property of the Gaussian distribution, it is straightforward to analytically compute $p(x\setminus A|y_k)$ obtained by excluding any subset A of the attributes:

$$p(x\setminus A|y_k) = \sum_{j=1}^{M} \pi_{jk} N(x\setminus A; \mu_{j\setminus A}, \Sigma_{j\setminus A}) \tag{11}$$

where $\mu_{j\setminus A}$ and $\Sigma_{j\setminus A}$ are obtained by removing the corresponding elements from μ_j and Σ_j. Then we can directly obtain $p(y_k|x\setminus A)$ as

$$p(y_k|x\setminus A) = \frac{p(x\setminus A|y_k)P_k}{\sum_{\ell=1}^{c} p(x\setminus A|y_\ell)P_\ell} \tag{12}$$

In our Matlab implementation this takes $O(1)$ steps.

4 Instance Level and Model Level Explanation of PRBF

To show the practical utility of the presented marginalization we demonstrate its use in a visualization method called explainVis, on a 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 passenger's status during travel (first, second, or third class, or crew), age (adult or child), and gender of the passenger. Fig. 1 shows explanations for the decision of the PRBF network for an instance concerning a first class adult male passenger.

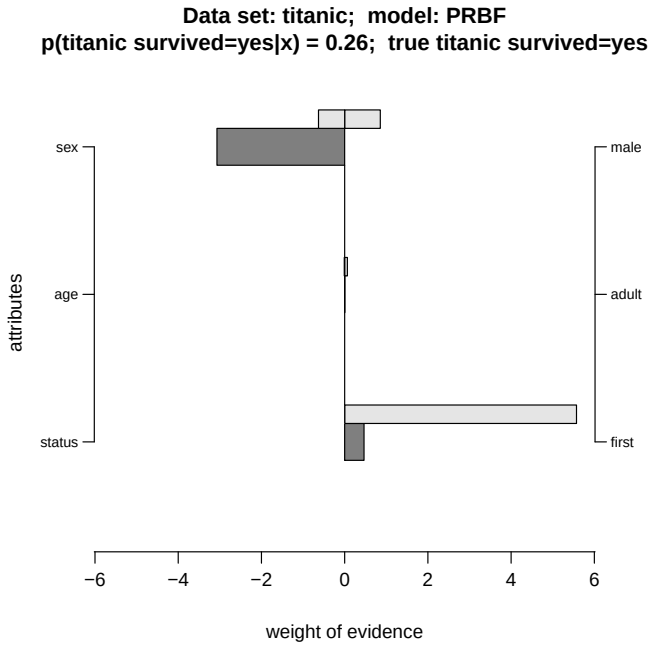

Figure 1: Instance explanation of the PRBF model for one of the instances in the Titanic data set.

The weight of evidence is shown on the horizontal axis. 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 probability $p(y|x)$ for class "survived" returned by the method for the given instance x is reported on the top (0.26). The lengths of the thicker bars correspond to the influences of the given attribute values in the model, expressed by (3). The positive weight of evidence is given on the right-hand side and the negative one is on the left-hand side. Thinner bars above the explanation bars indicate the average value of the weight of evidence over all training instances for the corresponding attribute value.

For the given instance we observe that "sex = male" speaks strongly against the survival and "status=first class" is favorable for survival, while being adult has no influence. Thinner average bars mostly agree with the effects expressed for the particular instance (being male is on average less dangerous than in the selected case, and being in the first class is even more beneficial).

While in our simple example we present all the attributes, for domains with many attributes a threshold parameter can select only attributes with sufficient impact for presentation. In some application domains (for example in medicine) experts interpret similar graphs as points in favor/against the decision and expect that the total sum of points will be 1 (or 100). Such normalization is also an option in the explainVis.

To get a more general view of the model we can use explanations for the training data and visualize them in a summary form, which shows average importance of each feature and its values. An example of such visualization for titanic data set is presented in Fig. 2.

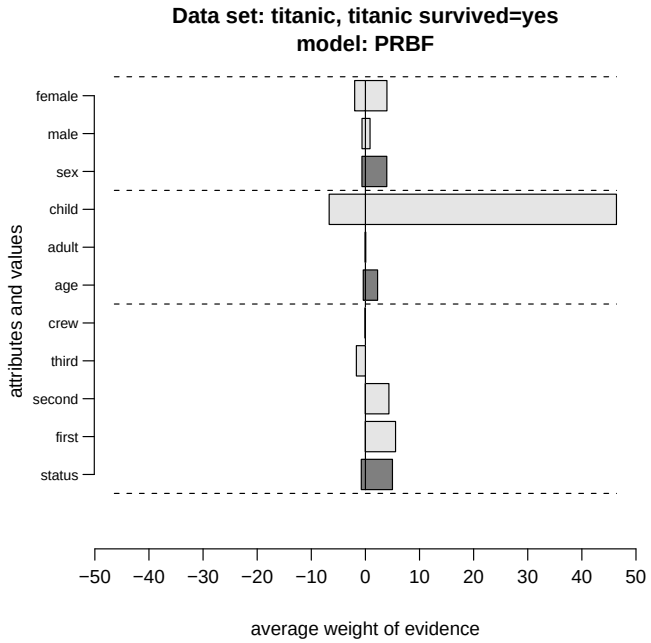

Figure 2: Model explanation of the PRBF model in the Titanic data set.

On the left-hand side on the vertical axis, all the attributes and their values are listed (each attribute and its values are separated by dashed lines). For each of them the average negative and the average positive weights of evidence are presented with the horizontal bar. For each attribute (a darker shade) the average positive and negative influences of its values are given. For titanic problem status and sex have approximately the same effect, and age plays less important role in PRBF model. Particular values give even more precise picture: first and second class is perceived as undoubtedly advantageous, being child or female has greater positive than negative effect, traveling in third class or being male is considered as a disadvantage, while a status of crew or being adult plays only a minor negative role.

4.1 True Explanations

In this section we demonstrate that explanations generated from PRBF closely follow the model: if the model achieves high classification accuracy, which is an indication that it captures the actual properties of the problem, than the obtained explanations are close to the ”true explanations”. We adapted the methodology from [24].

Our evaluation scenario includes five different learning algorithms besides PRBF: NB, decision trees (DT), nearest neighbor classifier (kNN), support vector machines, and neural networks[1]. We wish to demonstrate that the explanations of the predictions 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 the correct explanation and the explanation given by the model. We use Euclidean distance over weight of evidence (3) for all the attributes:

$$d_{exp}(x) = \left(\sum_{i=1}^{a} \left(\frac{1}{2} \left(\frac{\text{trueExpl}_i(x)}{\sum_{i=1}^{a} |\text{trueExpl}_i(x)|} - \frac{\text{WE}_i(x)}{\sum_{i=1}^{a} |\text{WE}_i(x)|} \right) \right)^2 \right)^{\frac{1}{2}} \tag{13}$$

Here $\text{trueExpl}_i(x)$ represents the "true explanation" weight of the contribution of the attribute A_i for the prediction of the instance x. For the attribute A_i the $\text{trueExpl}_i(x)$ and $\text{WE}_i(x)$ 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, therefore the absolute values of the model's explanations are normalized to 1 for each instance: we divide the $\text{trueExpl}_i(x)$ and $\text{WE}_i(x)$ with the sum of their respective absolute values, and multiply them by $\frac{1}{2}$. Let $\overline{d_{exp}}$ be an average of (13) over all testing instances. Therefore, $\overline{d_{exp}}$ is an indicator of how close the model's explanations are to the true explanations.

We used an artificial problem which is tailored to the abilities of the PRBF (and kNN) and rather difficult for the others. The labels of the instances in this problem can be explained by the attribute values in a clear and unambiguous way and we can therefore talk about correct explanation for each instance. The *groups* data set is defined with two important attributes I_1 and I_2 and two unrelated to the class value R_1 and R_2, all being real numbers in $[0,1]$. The data set is plotted in Fig. 3 (instances are scattered around group centers, which define class values). Class value is computed as $C = 1 + (\lfloor 3 \cdot I_1 \rfloor + \lfloor 3 \cdot I_2 \rfloor) \bmod 3$. Instances with class labels 1, 2, and 3 are represented with circles, triangles and squares, respectively. We used 1000 instances for training of the model and another 1000 for testing of explanations. The true explanation assigns 0.5 to each important attribute and 0 to unrelated ones.

Table 1 presents for each method its accuracy (acc) and the average distance to the true explanation ($\overline{d_{exp}}$).

Note that the most suitable algorithms for this data set (PRBF and kNN) achieve the highest accuracy, and also have the lowest average distance to the true explanation. The reason why PRBF is slightly closer to the ideal explanation is due to its better robustness to noise in the form of irrelevant attributes. This is a confirmation that explanations for PRBF are closely following the model: if the model captures the underlying structure of the problem, the explanations reflect that. Note also the last row of Table 1 which shows results for PRBF with approximation given by Eq. (5). The average distance to true explanations is the same as for PRBF with marginalization. The approach

[1] All the compared learning algorithms are from the R Project (http://www.r-project.org/): packages e1071, nnet, kknn, RWeka, and CORElearn. PRBF is coded in Matlab: the code for the PRBF, its marginalization, generation of probabilities, the data sets, as well as the visualization module can be obtained from the authors.

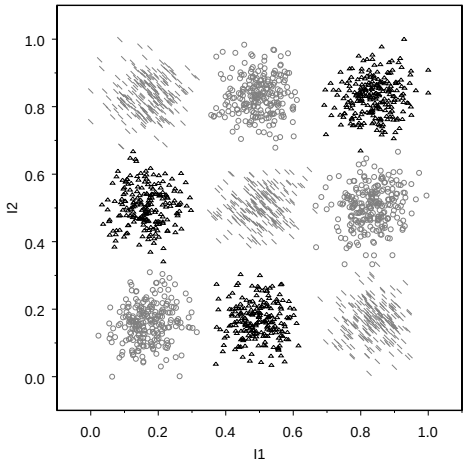

Figure 3: The *groups* data set plotted with respect to the two relevant attributes.

with marginalization is therefore as successful as the general approach presented in [24], but is preferable due to its better efficiency and theoretical foundation.

Figures 4 and 5 show instance explanations for two instances from the groups data set (note that in both cases the target class values for explanations are circles). The values of the important attributes for the instance in Fig. 4 are positively related with the circles class. The average weight of evidence for values in their range can be either positive or negative. The influence of unrelated features is clearly weak. The instance in Fig. 5 is not from the circle class, as the model has correctly predicted, and the values of important features speak against circles. Thinner bars on both figures presenting averages are computed for subintervals of feature's values. To get those intervals we can use one of the discretization techniques. For the presented data set we used equal-width discretization into three intervals.

Table 1: Performance and average distance to the true explanation for six classification methods on groups problem.

method	acc	$\overline{d_{exp}}$
NB	0.35	0.46
DT	0.33	0.35
kNN	0.99	0.08
SVM	0.66	0.22
ANN	0.90	0.09
PRBF	0.97	0.06
PRBF (5)	0.97	0.06

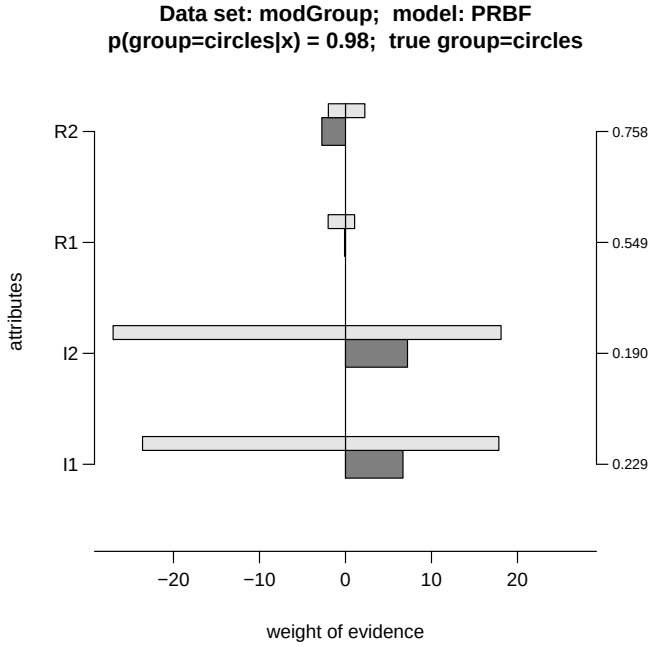

Figure 4: Instance explanations of PRBF for a case from circles class in the groups data set.

Figure 6 presents an overall picture of the groups data set (model explanation). I_1 and I_2 and all their subintervals show strong positive and negative influence on the class value, while random features R_1 and R_2 and their subintervals are relatively unimportant compared to them.

5 Discussion

Andrews et al. [1] introduced taxonomy and criteria for evaluation of rule extraction methods from ANN. Our approach is graphical, does not produce rules, and is oriented towards explaining effects for individual learning instances. It is therefore not directly comparable with these methods, but some aspects of the taxonomy are relevant to it. According to [1, 14] the rule extraction techniques from neural networks can be classified according to:

1. *expressive power* describing the language of extracted knowledge: propositional logic (i.e.. if-then rules), nonconventional logic (e.g., fuzzy logic), first-order logic, and finite state machines (deterministic, nondeterministic, stochastic, e.g., Moore and Mealy types).

2. *translucency* describing the degree to which the explanation method looks inside the ANN: decompositional (combining rules for individual neurons), ped-

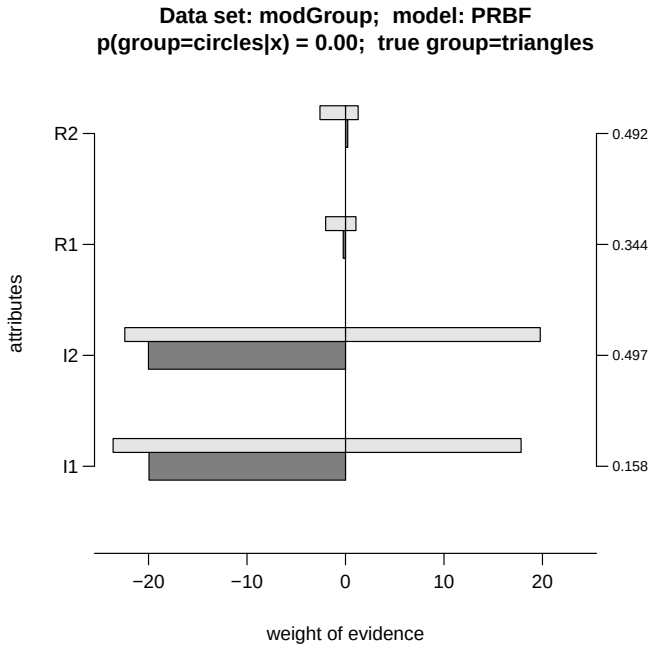

Figure 5: Instance explanations of PRBF for a case from triangles class in the groups data set.

agogical (treating ANN as a black box), and eclectic methods (combining both compositional and pedagogical types),

3. *portability* describes how well the technique covers the set of available ANN architectures,

4. *quality* of the extracted rules, which can be further split into several criteria: accuracy (generalization ability), fidelity (how well the rule reflects behavior of ANN), consistency (similarity of behavior for different ANN trained on the same task), and comprehensibility (readability and size of the rules).

5. *algorithmic complexity.*

By imbedding our approach into the taxonomy above, we can classify it as follows:

1. Expressive power: extracted explanations are in the form of contributions of individual attributes (see Figure 2) and possibly their combinations, which speak in favor/against the given class; these contributions can be seen as a limited form of propositional logic.

2. Translucency: while the explanation approach based on [24] can be classified as pedagogical (the model is treated as a black-box, only the causal relationship be-

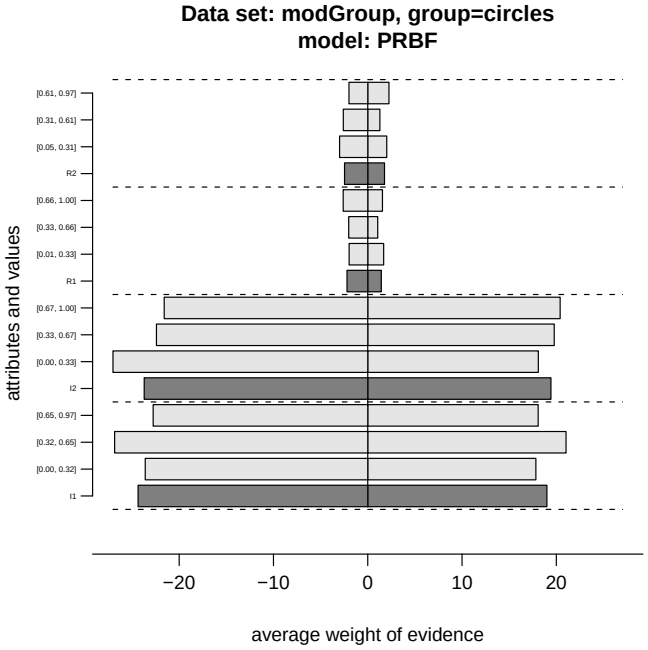

Figure 6: Model explanation of PRBF in the groups data set.

tween input and output are considered), the particular contribution of this paper is in exploiting specifics of PRBF network, so we can classify it as eclectic.

3. Portability: similarly to translucency the basic explanation approach [24] is very general and applicable to arbitrary ANN architecture, but the contribution of this paper is on the specific of PRBF networks.

4. Quality of extracted knowledge: while accuracy and consistency are not an appropriate criteria for evaluation of our method, we have shown that explanations are fidel to the learned model. It is also good concerning the comprehensibility due to its simplicity and graphical visualization.

5. Algorithmic complexity: the marginalization (12) of PRBF inputs which is needed for our explanation method can be efficiently computed by removing appropriate elements from μ_j and Σ_j. In our Matlab implementation this takes $O(1)$ steps. 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 practice, the time required to generate individual explanations is negligible.

5.1 Related work

We discuss some representative references relevant to our work, while for a more complete review we refer the reader to [14], which concentrates mostly on finite-state machines extracted from recurrent ANN but covers also other approaches.

Explanation of prediction is straightforward for symbolic models such as decision trees, decision rules and inductive logic programs. Consequently, if such models are extracted from trained ANNs, they represent the knowledge stored in the ANN in a symbolic form. Therefore, to obtain the explanation of a prediction, one simply has to read the corresponding rule in the extracted symbolic model. Whether such explanation is comprehensive in the case of large trees and rule sets is questionable.

Techniques for extracting explicit (if-then) rules from black box models (such as ANN) are described in [30, 6, 25, 27]. In [7] 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 ANNs using interval propagation is suggetsed in [21], while an approach which can provably extract sound and also nonmonotonic rules was presented in [8]. In [30] a search-based method has been proposed suitable for ANNs with binary units only, while application of the method proposed in [25] requires discretization of the hidden unit activations and pruning of the ANN architecture.

Främling [9] 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.

The works summarized below are not aimed directly at ANN but are relevant to our approach as they aim to provide explanation and visualization principles and applications to specific methods.

Some less complex 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 [17]. In [15] nomograms were developed for SVM, but they work only for a restricted class of kernels and cannot be used for general non-linear kernels. SVMs can also be visualized using projections into lower dimensional subspaces [4, 22] or using self-organizing maps from unsupervised learning [12] . These methods concentrate on visualization of the separating hyperplane and decision surface and do not provide explanations for individual decisions.

The naive Bayesian classifier (NB) is able to explain its decisions as the sum of information gains [16]. A straightforward visualization of NB was used in [2] while in [20] nomograms were developed for visualization of NB decisions. In [24] the NB list of information gains was generalized to a list of attribute weights for any prediction model.

Madigan et al. [18] consider the case of belief networks and 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. This approach cannot be applied to PRBF because for it the whole input vector (containing all attributes) is considered a single variable that is assumed to be sampled from multivariate Gaussian distributions. Also PRBF assumes continuous attributes as inputs. The negation of continuous attributes is meaningless, but would be needed to compute weights of evidence as defined by [18].

In ExplainD framework [23] weights of evidence and visualizations similar to ours are used, but the explanation approach is limited to (linear) additive models, while our approach can be used for all probabilistic models. As PRBF is not linear in the space of features, but in the space of Gaussian components, where each component presents multivariate Gaussian kernel, it cannot be visualized by the ExplainD. Also our method is not restricted to the weight of evidence, it can use also other interpretations such as information gain, or difference of probabilities.

6 Conclusions

We presented a decomposition for PRBF classifier by exploiting the marginalization property of the Gaussian distribution and applied this decomposition inside a general method for explaining predictions for individual instances, recently presented in [24].

Using an artificial data set we demonstrated that the generated explanations for PRBF are close to the true explanations if the accuracy of the model is high. We also showed how we can explain and visualize the classifications of unlabeled cases provided by the otherwise opaque PRBF model. Additionally we developed a visualization technique which uses explanations of the training instances to describe the effects of all the attributes and their values at the model level.

The explanation method exhibits the following properties:

- Model dependency: it expresses the 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. We also believe that our explanation method is not responsible for difficulties a learning algorithm is sensitive to, such as problem of redundant and/or highly correlated attributes.

- Instance dependency: different instances are predicted differently, so the explanations will also be different.

- 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).

- Capability to detect strong conditional dependencies: if the model captures strong conditional dependency (e.g., sum by modulo relation in Fig. 3), the explanations will also reflect that.

- 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 as described in [24]. In PRBF it is straightforward and at no extra cost to marginalize any number of features, which is useful property for heuristic methods searching for combinations of attribute values.

- Visualization ability: the generated explanations can be graphically presented in terms of the positive/negative effect each attribute and its value have on the selected class.

In future work we plan to use PRBF in practical applications where explanation capability is of crucial importance (e.g. medicine), and use the feedback provided by the experts to further improve both the method and the visualization tool.

Acknowledgements

This work was supported in the framework of the "Bilateral S+T cooperation between the Hellenic Republic and the Republic of Slovenia (2004-2006)".

References

[1] R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–384, 1995.

[2] 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.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] 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.*, pages 497– 500, 2003.

[5] C. Constantinopoulos and A. Likas. An incremental training method for the probabilistic RBF network. *IEEE Trans. Neural Networks*, 17(4):966–974, 2006.

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

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

[8] A. S. d'Avila Garcez, K. Broda, and D. M. Gabbay. Symbolic knowledge extraction from trained neural networks: a sound approach. *Artificial Intelligence*, 125 (1-2):155–207, 2001.

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

[10] S. I. Gallant. Connectionist expert systems. *Communications of the ACM*, 31(2): 152–169, 1988.

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

[12] 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.

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

[14] H. Jacobsson. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

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

[16] I. Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence*, 7(4):317–337, 1993.

[17] 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*, 17: 127–129, 1978.

[18] D. Madigan, K. Mosurski, and R. G. Almond. Graphical explanation in belief networks. *Journal of Computational and Graphical Statistics*, 6(2):160–181, 1997.

[19] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

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

[21] 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*, pages 152–161, London, UK, 2001. Springer-Verlag. ISBN 3-540-42732-5.

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

[23] 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.

[24] M. Robnik Šikonja and I. Kononenko. Explaining classifications for individual instances. Technical report, University of Ljubljana, Faculty of Computer and Information Science, Slovenia, 2007. http://lkm.fri.uni-lj.si/rmarko/papers/ (to be published).

[25] R. Setiono and H. Liu. Understanding neural networks via rule extraction. In *Proceedings of IJCAI'95*, pages 480–487, 1995.

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

[27] S. Thrun. Extracting rules from artifical neural networks with distributed representations. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 505–512. The MIT Press, 1995.

[28] M. K. Titsias and A. Likas. Shared kernel models for class conditional density estimation. *IEEE Trans. Neural Networks*, 12(5):987–997, 2001.

[29] M. K. Titsias and A. Likas. Class conditional density estimation using mixtures with constrained component sharing. *IEEE Trans. Pattern Anal. and Machine Intell.*, 25(7):924–928, 2003.

[30] G. G. Towell and J. W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13(1):71–101, 1993.