Learning Valued Preference Structures for Solving Classification Problems

Eyke Hüllermeier and Klaus Brinker
Philipps-Universität Marburg
Department of Mathematics and Computer Science
{eyke, brinker}@mathematik.uni-marburg.de

Abstract

This paper introduces a new approach to classification which combines pairwise decomposition techniques with ideas and tools from fuzzy preference modeling. More specifically, our approach first decomposes a polychotomous classification problem involving m classes into an ensemble of binary problems, one for each ordered pair of classes. The corresponding classifiers are trained on the relevant subsets of the (transformed) original training data. In the classification phase, a new query is submitted to every binary learner. The output of each classifier is interpreted as a fuzzy degree of preference for the first in comparison with the second class. By combining the outputs of all classifiers, one thus obtains a fuzzy preference relation which is taken as a point of departure for the final classification decision. This way, the problem of classification is effectively reduced to a problem of decision making based on a fuzzy preference relation. Corresponding techniques, which have been investigated quite intensively in the field of fuzzy set theory, hence become amenable to the task of classification. In particular, by decomposing a preference relation into a strict preference, an indifference, and an incomparability relation, this approach allows one to quantify different types of uncertainty in classification and thereby supports sophisticated classification and postprocessing strategies.

1 Introduction

As one of the standard problems of supervised learning, the performance task of classification has been studied intensively in the field of machine learning. In classification, the prediction to be made consists of a discrete class label. Thus, the problem is to learn a model that establishes an $\mathcal{X} \longrightarrow \mathcal{L}$ mapping from an instance space $\mathcal{X}$ to a finite set $\mathcal{L}$ of class labels. For the representation of such a mapping, various model classes have been proposed in machine learning and related fields, such as neural networks, kernel machines, or classification trees [13].

The arguably simplest type of classification problems are *dichotomous* (*binary*, *two-class*) problems for which $|\mathcal{L}| = 2$. For such problems, a multitude of efficient and theoretically well-founded classification methods exists. In fact, the representation of models is often geared toward the binary case, and sometimes even restricted to this problem class. For example, several popular machine learning techniques, such as support vector machines [24], learn a decision boundary that can only divide the instance space into two parts, one for each class. Similarly, *concept learning*, i.e., the task of learning a description of a target concept from examples and counter-examples of the concept, may be considered as a two-class classification task.

Needless to say, practically relevant problems are not always restricted to the binary case. One approach for tackling *polychotomous* (*multi-class*) problems is to use model classes that are able to represent an $\mathcal{X} \longrightarrow \mathcal{L}$ mapping for $|\mathcal{L}| > 2$ directly, such as classification trees. An alternative strategy is to transform the original

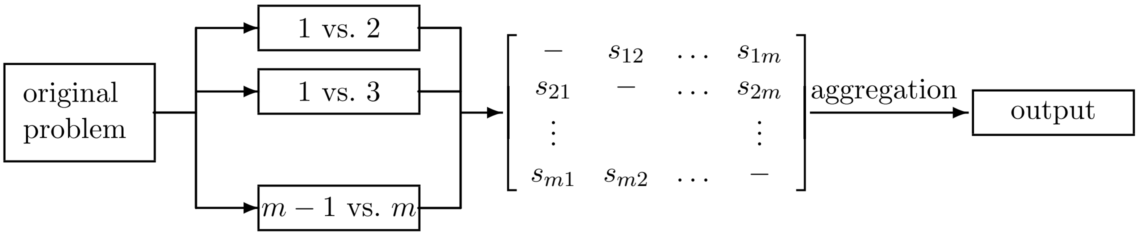
original problem
1 vs. 2
1 vs. 3
$m-1$ vs. m
$\begin{bmatrix} - & s_{12} & \cdots & s_{1m} \\ s_{21} & - & \cdots & s_{2m} \\ \vdots & & & \vdots \\ s_{m1} & s_{m2} & \cdots & - \end{bmatrix}$
aggregation
output

Figure 1: Basic structure of the learning by pairwise comparison approach.

problem into several binary problems via a *class binarization* technique. One very general approach to binarization is based on the idea of representing the original classes in terms of binary codes with error-correcting properties [4, 1]. Two special cases of this approach, the one-vs-rest and the all-pairs technique, have received special attention in the literature.

The unordered or one-vs-rest binarization is perhaps the most popular class binarization technique. It takes each class in turn and learns a binary concept that discriminates this class from all other classes. At prediction time, each binary classifier decides whether the query input belongs to its concept or not. Tie breaking techniques (typically based on confidence estimates for the individual predictions) are used in case of conflicts, which may arise when more than one concept is predicted or all classifiers abstain.

The key idea of the alternative *learning by pairwise comparison* (LPC) approach (aka pairwise classification, round robin learning, all-pairs) is to transform an m-class problem into $m(m-1)/2$ binary problems, one for each pair of classes (Fig. 1).[1] At classification time, a query instance is submitted to all binary models, and the predictions of these models (the s_{ij} in Fig. 1) are combined into an overall classification. In [9, 10], it was shown that pairwise classification is not only more accurate than the one-vs-rest technique but that, despite the fact that the number of models

[1] Alternatively, one can consider a binary problem for every *ordered* pair of classes, in which case the total number of such problems is doubled. We shall come back to this point later on.

that have to be learned is quadratic in the number of classes, pairwise classification is also more efficient (at least in the training phase) than one-vs-rest classification. The reason is that the binary decision problems not only contain fewer training examples (because all examples that do not belong to either of the classes are ignored), but that the decision boundaries of each binary problem may also be considerably simpler than for the problems generated by the one-vs-rest transformation (which in turn can further increase computational efficiency in the evaluation phase).

This paper elaborates on another interesting aspect of the LPC approach: Assuming that every binary learner outputs a score in the unit interval (or, more generally, an ordered scale), and that this score can reasonably be interpreted as a "fuzzy preference" for the first in comparison with the second class, the complete ensemble of pairwise learners produces a *valued* or *fuzzy preference relation*. The final classification decision is then made on the basis of this relation. In other words, the problem of classification has been reduced, in a first step, to a problem of decision making based on a valued preference relation.

Of course, the problem of deriving a final classification decision from the predictions of the whole ensemble of pairwise learners is an integral part of the LPC approach and might therefore not be regarded as a novel aspect by itself. What is new, however, is to look at the ensemble of predictions as a fuzzy preference relation. This perspective establishes a close connection between (pairwise) learning and fuzzy preference modeling, and therefore allows for applying techniques from the former field in the context of machine learning. For example, since the binary learners are trained independently of each other and, moreover, predictions can be incorrect, the preference relation produced by an ensemble of such learners is not guaranteed to have any "reasonable" properties, such as transitivity. Thus, a first step could consist of post-processing this relation, using techniques from fuzzy preferences, so as to resolve inconsistencies between the pairwise predictions.

In this paper, we are especially interested in exploiting techniques for decomposing a fuzzy (weak) preference relation into a preference structure consisting of a strict preference, an indifference, and an incomparability relation. As will be argued in more detail later on, the latter two relations have a quite interesting interpretation and important meaning in the context of classification, where they represent two types of uncertainty/ambiguity, namely conflict and ignorance. Consequently, these relations can support sophisticated classification strategies, including those that allow for partial reject options.

The remainder of the paper is organized as follows. Section 2 details the LPC approach to classification, and Section 3 recalls the basics of valued preference structures. Section 4 is devoted to the idea of classification based on valued preference relations, including the important element of learning weak preferences between class labels. Potential applications of the approach are discussed in Section 5, and first empirical results are presented in Section 6. Finally, Section 7 provides a brief discussion of related work and Section 8 concludes the paper.

2 Learning by Pairwise Comparison

As mentioned earlier, learning by pairwise comparison (LPC) transforms a multi-class classification problem, i.e., a problem involving $m > 2$ classes (labels) $\mathcal{L} = \{\lambda_1 \ldots \lambda_m\}$, into a number of *binary* problems. To this end, a separate model (base learner) $\mathcal{M}_{i,j}$ is trained for each *pair* of labels $(\lambda_i, \lambda_j) \in \mathcal{L}$. $\mathcal{M}_{i,j}$ is intended to separate the objects with label λ_i from those having label λ_j. If $(x, \lambda_a) \in \mathcal{X} \times \mathcal{L}$ is an original training example (revealing that instance x has label λ_a), then x is considered as a *positive* example for all learners $\mathcal{M}_{a,j}$ and as a *negative* example for the learners $\mathcal{M}_{j,a}$ $(j \neq a)$; those models $\mathcal{M}_{i,j}$ with $a \notin \{i, j\}$ simply ignore this example.

At classification time, a query x is submitted to all learners, and each prediction $\mathcal{M}_{i,j}(x)$ is interpreted as a vote for a label. In particular, if $\mathcal{M}_{i,j}$ is a $\{0,1\}$-valued classifier, $\mathcal{M}_{i,j}(x) = 1$ is counted as a vote for λ_i, while $\mathcal{M}_{i,j}(x) = 0$ would be considered as a vote for λ_j. Given these outputs, the simplest classification strategy is to predict the class label with the highest number of votes. A straightforward extension of the above voting scheme to the case of $[0,1]$-valued (scoring) classifiers yields a *weighted voting procedure*: The score for label λ_i is computed by

$$s_i \overset{\mathrm{df}}{=} \sum_{1 \leq j \neq i \leq m} s_{i,j}, \tag{1}$$

where $s_{i,j} = \mathcal{M}_{i,j}(x)$, and again the label with the highest score is predicted.

The votes $s_{i,j}$ in (1) and, hence, the learners $\mathcal{M}_{i,j}$ are usually assumed to be (additively) *reciprocal*, that is,

$$s_{j,i} \equiv 1 - s_{i,j} \tag{2}$$

and correspondingly $\mathcal{M}_{i,j}(x) \equiv 1 - \mathcal{M}_{j,i}(x)$. Practically, this means that only one half of the $m(m-1)$ classifiers $\mathcal{M}_{i,j}$ needs to be trained, for example those for $i < j$. As will be seen later on, this restriction is not very useful in our approach. Therefore, we will train the whole set of classifiers $\mathcal{M}_{i,j}$, $1 \leq i \neq j \leq m$, which means that no particular relation between $s_{i,j}$ and $s_{j,i}$ will be assumed.

3 Fuzzy Preference Structures

Considering the classification problem as a *decision problem*, namely a problem of deciding on a class label for a query input x, an output $r_{i,j} = \mathcal{M}_{i,j}(x)$ can be interpreted as a *preference* for label λ_i in comparison with label λ_j: The higher $r_{i,j}$, the more preferred is λ_i as a classification for x, i.e., the more likely λ_i appears in comparison with label λ_j. Correspondingly, the matrix

$$\mathcal{R} = \begin{bmatrix} 1 & r_{1,2} & \dots & r_{1,m} \\ r_{2,1} & 1 & \dots & r_{2,m} \\ \vdots & & & \vdots \\ r_{m,1} & r_{m,2} & \dots & 1 \end{bmatrix} \qquad (3)$$

obtained by collecting the outputs of the whole classifier ensemble can be interpreted as a *fuzzy* or *valued preference relation*. More specifically, suppose that $\mathcal{R}$ can be considered as a *weak preference relation*, which means that $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$ is interpreted as $\lambda_i \succeq \lambda_j$, that is, "label λ_i is at least as likely as label λ_j". In this case, $\mathcal{R}$ is reflexive, which is the reason for setting the diagonal elements in (3) to 1. It is important to note that, from a learning point of view, equating classifier scores $s_{i,j}$ with weak preferences $r_{i,j}$ raises the crucial question of how to train models that are able to deliver predictions having this special semantics; we shall come back to this issue in Section 4.4.

A classification decision can then be made on the basis of the relation (3). To this end, one can resort to corresponding techniques that have been developed and investigated quite thoroughly in fuzzy preference modeling and decision making [7]. In principle, the simple voting scheme (1) outlined in Section 2 can be seen as a special case of such a decision making technique.

In this paper, our interest concerns the application of techniques for decomposing the relation $\mathcal{R}$ into three associated relations with different meaning. Recall that $\mathcal{R}$ is considered as a *weak preference relation*. In the non-fuzzy case where $r_{i,j} \in \{0, 1\}$, a weak preference relation induces a *strict preference relation* $\mathcal{P}$, an *indifference relation* $\mathcal{I}$, and an *incomparability relation* $\mathcal{J}$ in a straightforward way. Denoting strict preference by $\succ$, indifference by $\sim$, and incomparability by $\perp$, we get:

$$
\begin{aligned}
\lambda_i \succ \lambda_j &\overset{\mathrm{df}}{\Leftrightarrow} (\lambda_i \succeq \lambda_j) \wedge (\lambda_i \not\succeq \lambda_j) \\
\lambda_i \sim \lambda_j &\overset{\mathrm{df}}{\Leftrightarrow} (\lambda_i \succeq \lambda_j) \wedge (\lambda_i \succeq \lambda_j) \\
\lambda_i \perp \lambda_j &\overset{\mathrm{df}}{\Leftrightarrow} (\lambda_i \not\succeq \lambda_j) \wedge (\lambda_i \not\succeq \lambda_j)
\end{aligned}
\tag{4}
$$

The other way round, a triplet $(\mathcal{P},\mathcal{I},\mathcal{J})$ of binary relations in $\mathcal{L}$ is called a (Boolean) *preference structure* on $\mathcal{L}$ if is has the following properties:

- $\mathcal{P}$ and $\mathcal{J}$ are irreflexive, $\mathcal{I}$ is reflexive;
- $\mathcal{P}$ is asymmetrical, $\mathcal{I}$ and $\mathcal{J}$ are symmetrical;
- $\mathcal{P} \cap \mathcal{I} = \emptyset$, $\mathcal{P} \cap \mathcal{J} = \emptyset$, $\mathcal{I} \cap \mathcal{J} = \emptyset$;
- $\mathcal{P} \cup \mathcal{P}^t \cup \mathcal{I} \cup \mathcal{J} = \mathcal{L} \times \mathcal{L}$.

In the fuzzy case, preference degrees can be expressed on the continuous scale $[0,1]$, and a binary relation becomes an $\mathcal{L} \times \mathcal{L} \longrightarrow [0,1]$ mapping [27]. Referring to the class of t-norms [17] to operate on fuzzy preference degrees, a *fuzzy preference structure* can be defined as follows: Let (T,S,N) be a continuous De Morgan triplet consisting of a strong negation N, a t-norm T, and its N-dual t-conorm S; moreover, denote the T-intersection of two sets A and B by $A \cap_T B$ and the S-union by $A \cup_S B$. A fuzzy preference structure on $\mathcal{L}$ is a triplet $(\mathcal{P},\mathcal{I},\mathcal{J})$ of fuzzy relations satisfying

- $\mathcal{P}$ and $\mathcal{J}$ are irreflexive, $\mathcal{I}$ is reflexive;
- $\mathcal{P}$ is T-asymmetrical ($\mathcal{P} \cap_T \mathcal{P}^t = \emptyset$), $\mathcal{I}$ and $\mathcal{J}$ are symmetrical;
- $\mathcal{P} \cap_T \mathcal{I} = \emptyset$, $\mathcal{P} \cap_T \mathcal{J} = \emptyset$, $\mathcal{I} \cap_T \mathcal{J} = \emptyset$;
- $\mathcal{P} \cup_S \mathcal{P}^t \cup_S \mathcal{I} \cup_S \mathcal{J} = \mathcal{L} \times \mathcal{L}$.

Fuzzy preference structures, especially their axiomatic construction, have been studied extensively in the literature (e.g. [7, 2, 8, 20, 21, 22]). The same is true for the question of how to decompose a weak (valued) preference relation $\mathcal{R} \in [0,1]^{m \times m}$ into a strict preference relation $\mathcal{P}$, and indifference relation $\mathcal{I}$, and an incomparability relation $\mathcal{J}$ such that $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ is a fuzzy preference structure. Without going into technical detail, we only give an example of a commonly employed decomposition scheme which is simply obtained by replacing, respectively, the conjunction and negation in (4) by the product t-norm and the $1 - (\cdot)$ mapping (again, we denote $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$):

$$\begin{aligned} \mathcal{P}(\lambda_i, \lambda_j) &= r_{i,j} \times (1 - r_{j,i}) \\ \mathcal{I}(\lambda_i, \lambda_j) &= r_{i,j} \times r_{j,i} \\ \mathcal{J}(\lambda_i, \lambda_j) &= (1 - r_{i,j}) \times (1 - r_{j,i}) \end{aligned} \tag{5}$$

A related decomposition scheme will also be used in the experimental part below. More generally, one could of course ask for an "optimal" choice of the decomposition, i.e., for the decomposition that performs best in our context of classification learning. This question, however, is beyond the scope of this paper and will be left for future work. Besides, as will be seen later on, it cannot be addressed in separation, since the suitability of a decomposition also depends on the way in which the weak preference degrees $\mathcal{R}(\lambda_i, \lambda_j)$ are learned.

4 Fuzzy Modeling of Classification Knowledge

Regardless of the particular decomposition scheme employed, the crucial point is that the relations $\mathcal{I}$ and $\mathcal{J}$ do have a very interesting meaning in the context of classification: Indifference corresponds to the *conflict* involved in a classification situation, while incomparability reflects the corresponding degree of *ignorance*.

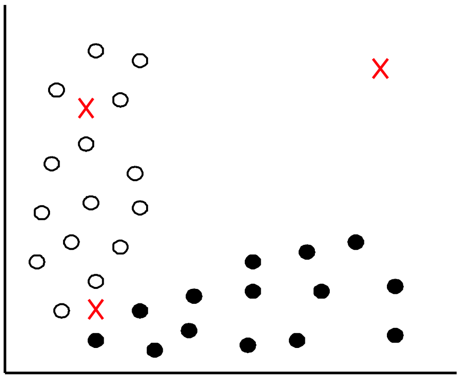

Figure 2: Classification scenario: Observations from two classes (points) and new query instances (crosses).

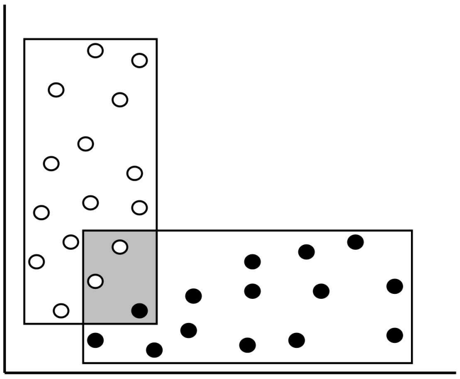

Figure 3: Regions of conflict (gray area) and ignorance (area not covered by any rule) in case of a rule-based model (rules indicated as rectangles).

4.1 Conflict and Ignorance in Classification

To illustrate what we mean, respectively, by conflict and ignorance, consider the simple classification scenario shown in Fig. 2: Given observations from two classes, `black` and `white`, three new instances marked by a cross need to be classified. Obviously, given the current observations, the upper left instance can quite safely be classified as `white`. The case of the lower left instance, however, involves a high level of conflict, since both classes, `black` and `white`, appear plausible. The third situation is an example of ignorance: The upper right instance is located in a region of the instance space in which no observations have been made so far. Consequently, there is neither evidence in favor of class `black` nor in favor of class `white`.

More generally, one may speak of a conflict if there is evidence in favor of two or more classes simultaneously, while a situation of ignorance occurs if none of the classes is supported. For instance, when characterizing each class in the above example in terms of (exception-tolerant) interval-based rules, that is, axis-parallel rectangles, we may obtain the model shown in Fig. 3. Here, a conflict occurs for the points in the intersection, where both rules apply, while the region of ignorance is given by the points that are not covered by any rule.

4.2 Model Assumptions

In the above example, the meaning of and difference between conflict and ignorance is intuitively quite obvious. Upon closer examination, however, these concepts turn out to be more intricate. In particular, one should realize that ignorance is not immediately linked with sparseness of the input space. This is due to the fact that generalization in machine learning is not only based on the observed data but also involves a model class with associated model assumptions.[2] In fact, a direct connection between ignorance and sparely populated regions of the input space can only be established for instance-based (prototype-based) classifiers, since these classifiers are explicitly based on the assumption that closely neighbored instances belong to the same class.

The situation is different, however, for other types of models. For example, Fig. 4 shows a scenario in which a query point in a sparse input region can be classified quite safely, given the observed data *in conjunction with the assumption of a linear model.* In other words, given the correctness of the inductive bias of the learner (linearity assumption), the current observations allow for quite confident conclusions about the label of the query, even though the latter does not have any close neighbors.

[2] It is well-known that learning from data is impossible without an (appropriate) inductive bias [19].

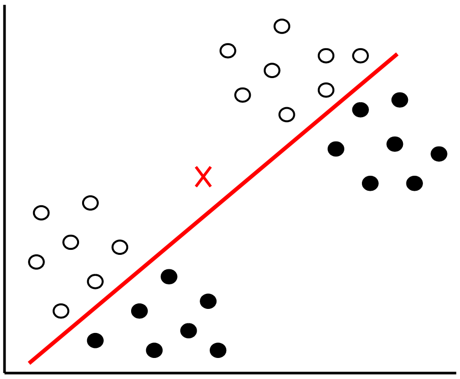

Figure 4: Given the assumption of linear separability, the query instance can be classified quite safely, even though it is spatially isolated from all other examples.

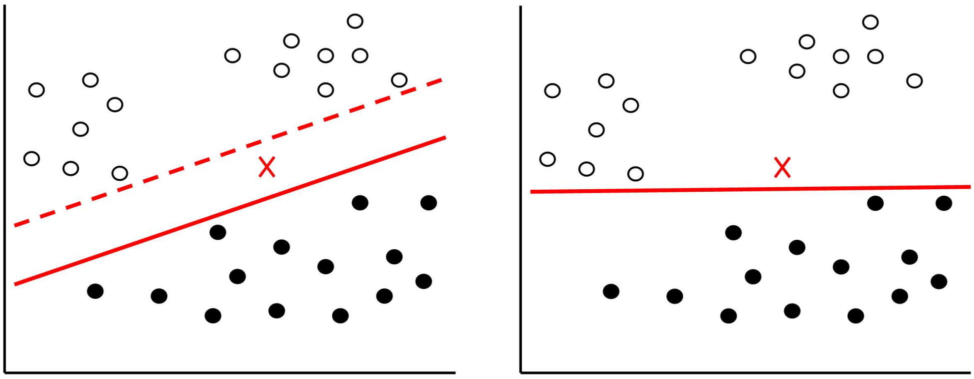

Figure 5: Depending on the model assumptions (linear decision boundary vs. axis-parallel boundary) the classification of the query (cross) is ambiguous (left) or not (right).

As suggested by this example, the ambiguity of a classification scenario “lies in the eye of the beholder”, that is, it depends on the “biased” view of the underlying model class: The same classification scenario may appear ambiguous for one model class but unambiguous for another one. To illustrate, consider the example in Fig. 5. Given the assumption of a linear model (left picture), the situation appears to be ambiguous, as there are consistent models that classify the query point as white but also others that will predict black. Given the more restrictive model class of *decision stumps* (decision trees with only one inner node or, stated differently, a decision boundary which is parallel to one of the axes), the situation is unambiguous (right picture).

The type of model also plays an important role when it comes to formalizing the concepts of conflict and ignorance in a rigorous way. For example, it makes a great difference whether or not an individual model by itself is already able to represent conflict and/or ignorance. For example, a rule-based classifier is in principle able to express ignorance, while this is impossible for standard discriminative learners such as support vector machines (that are forced to make a decision). Likewise, probabilistic classifiers are unable to represent ignorance, even though they can express a conflict, namely in terms of a distribution that assigns a positive probability to more than one class. In the following, we develop a formal conception of conflict and ignorance that assumes this kind of probabilistic or, more generally, scoring classifier as a base learner.

4.3 A Formal Conception of Conflict and Ignorance

Let $\mathfrak{M}$ denote the model class underlying the classification problem, where each model is a scoring classifier in the form of an $\mathcal{X} \longrightarrow [0,1]$ mapping. As mentioned before, we assume that each individual model is able to represent a conflict: The stronger a prediction deviates from the two extremes 1 (evidence in favor of the first class) and 0 (support of the second class) or, in other words, the closer it is to 1/2, the stronger the conflict. Moreover, let $\mathcal{V} = \mathcal{V}(\mathcal{D})$ be the set of models which are compatible with the examples given, i.e., the set of models which can still be regarded as possible candidates given the data $\mathcal{D}$; in the machine learning literature, $\mathcal{V}$ is called the *version space.*[3] Then, given a query $x_0 \in \mathcal{X}$, the set of possible predictions is

$$Y_0 = \{\mathcal{M}(x) \,|\, \mathcal{M} \in \mathcal{V}(\mathcal{D}) \subseteq \mathfrak{M}\} \tag{6}$$

[3] The concept of a version space essentially assumes noise-free data. We come back to this problem in Section 4.4.

It seems reasonable to define the degree of ignorance of a prediction in terms of the *diversity* of Y_0: The more predictions appear possible, i.e., the higher the diversity of predictions, the higher is the degree of ignorance.

According to this view, ignorance corresponds to that part of the (total) uncertainty about a prediction that can potentially be reduced by gathering more examples. In fact, the more examples have been observed, the smaller becomes the version space $\mathcal{V}$, and therefore the more precise the set of possible predictions Y_0. To illustrate, consider two extreme scenarios: Firstly, imagine that no example has been observed so far. This is a situation of *complete ignorance*, since every model and therefore every output for a new query x_0 appears to be possible. Secondly, suppose a large amount of data to be given, so that a single correct model $\mathcal{M}^*$ can be identified (at least approximately and with high probability). In this case, there is no ignorance left, since only one prediction $\mathcal{M}^*(x_0)$ remains. Note, however, that the prediction could still be uncertain or, say, *conflicting*, as $\mathcal{M}^*$ may support more than one class. To summarize on this score,

- the degree of ignorance (incomparability) corresponds to that part of the uncertainty of a prediction which is due to limited empirical data and which can in principle be reduced by gathering additional examples;
- the degree of conflict (indifference) corresponds to that part of the uncertainty which is due to a known conflict and which cannot be reduced any further.

4.4 Learning Valued Preferences for Classification

The general idea of our method, subsequently referred to as LVPC (Learning Valued Preferences for Classification), is the following: First, the weak preference relation (3) is learned using an LPC approach. Then, this relation is decomposed into a preference structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$, that is, into three corresponding relations

$$\begin{bmatrix} 0 & p_{1,2} & \dots & p_{1,m} \\ p_{2,1} & 0 & \dots & p_{2,m} \\ \vdots & & & \vdots \\ p_{m,1} & p_{m,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} 1 & i_{1,2} & \dots & i_{1,m} \\ i_{2,1} & 1 & \dots & i_{2,m} \\ \vdots & & & \vdots \\ i_{m,1} & i_{m,2} & \dots & 1 \end{bmatrix} \begin{bmatrix} 0 & j_{1,2} & \dots & j_{1,m} \\ j_{2,1} & 0 & \dots & j_{2,m} \\ \vdots & & & \vdots \\ j_{m,1} & j_{m,2} & \dots & 0 \end{bmatrix}$$

such that $\mathcal{J}$ characterizes the ignorance involved in a prediction, in the sense as outlined above, and $\mathcal{I}$ the degree of conflict. In this context, two crucial problems have to be solved: Firstly, how to learn a suitable weak preference relation $\mathcal{R}$, and secondly, how to decompose $\mathcal{R}$ into a structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$. As the decomposition problem has already been studied thoroughly in the literature and, moreover, the suitability of a particular scheme will strongly depend on the way in which $\mathcal{R}$ has been induced, we shall subsequently focus on the first problem.

Learning a weak preference relation $\mathcal{R}$ means that, for every pair of labels (λ_i, λ_j), we have to induce models $\mathcal{M}_{i,j}$ and $\mathcal{M}_{j,i}$ such that, for a given query input x, $\mathcal{M}_{i,j}(x)$ corresponds to the degree of weak preference $\lambda_i \succeq \lambda_j$ and, vice versa, $\mathcal{M}_{j,i}(x)$ to the degree of weak preference $\lambda_j \succeq \lambda_i$. The models $\mathcal{M}_{i,j}$ are of special importance as they directly determine the degrees of conflict and ignorance associated with a comparison between λ_i and λ_j. This fact is also crucial for the properties that the models $\mathcal{M}_{i,j}$ should obey.

According to the idea outlined above, a weak preference in favor of a class label should be derived from the set (6) of possible predictions. As this set in turn depends on the version space $\mathcal{V}$, the problem comes down to computing or at least approximating this space. In this connection, it deserves mentioning that an exact representation of the version space will usually not be possible for reasons of complexity. Apart from that, however, a representation of that kind would not be very useful either. In fact, despite the theoretical appeal of the version space concept, a

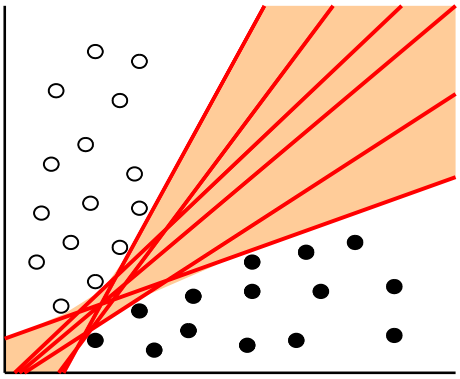

Figure 6: Illustration of the version space (class of hyperplanes that classify the training data correctly) and the "region of ignorance" (shaded in light color).

considerable practical drawback concerns its extreme sensitivity toward noise and inconsistencies in the data.

To overcome these problems, our idea is to approximate a version space in terms of a finite number of representative models; to some extent, these models should also be robust toward noise. More specifically, consider the problem of learning a binary model $\mathcal{M}_{i,j}$ from an underlying model class $\mathfrak{M}$. To approximate the version space associated with $\mathcal{M}_{i,j}$, we induce a finite set of models

$$\mathbb{M}_{i,j} = \left\{ \mathcal{M}_{i,j}^{(1)}, \mathcal{M}_{i,j}^{(2)} \ldots \mathcal{M}_{i,j}^{(K)} \right\} \subseteq \mathfrak{M} \tag{7}$$

The set of possible predictions (6) is approximated correspondingly by

$$\widehat{Y}_0 \ = \mathbb{M}_{i,j}(x) = \bigcup_{k=1\ldots K} \mathcal{M}_{i,j}^{(k)}(x).$$

An illustration is given in Fig. 6. Assuming that the two classes `black` and `white` can be separated in terms of a linear hyperplane, the version space consists of all those hyperplanes that classify the training data correctly. Given a new query instance, a unique class label can be assigned only if that instance lies on the same side of *all* hyperplanes (this situation is sometimes called "unanimous voting" [26]). Otherwise, both predictions are possible; the corresponding set of instances constitutes the "region of ignorance" which is shaded in light color.

In the above example, $\{0,1\}$-valued classifiers were used to ease exposition. In the context of fuzzy classification, however, scoring classifiers with outputs in the unit interval are more reasonable. Suppose that each ensemble member $\mathcal{M}_{i,j}^{(k)}$ in (7) outputs a score $s_{i,j}^{(k)} \in [0,1]$. The minimum of these scores would in principle be suitable as a degree of (weak) preference for λ_i in comparison with λ_j:

$$r_{i,j} = \min_{k=1\ldots K} s_{i,j}^{(k)} .$$

As this order statistic is quite sensitive toward noise and outliers, however, we propose to replace it by the empirical α-quantile of the distribution of the $s_{i,j}^{(k)}$ (a reasonable choice is $\alpha = 0.1$).

Note that, in case the models in $\mathfrak{M}$ are reciprocal, only $\mathbb{M}_{i,j}$ or $\mathbb{M}_{j,i}$ needs to be trained, but not both. We then have $s_{i,j}^{(k)} = 1 - s_{j,i}^{(k)}$, and the α-quantile for $\mathcal{M}_{i,j}$ is given by 1 minus the $(1-\alpha)$-quantile for $\mathcal{M}_{j,i}$. In other words, the degree of ignorance is directly reflected by the distribution of the scores $s_{i,j}^{(k)} = 1 - s_{j,i}^{(k)}$ and corresponds to the length of the interval between the α-quantile and the $(1-\alpha)$-quantile of this distribution. Thus, the more precise this distribution, the smaller the degree of ignorance. In particular, if all models $\mathcal{M}_{i,j}^{(k)}$ output the same score s, the ignorance component shrinks to 0. An illustration is given in Fig. 7.

Regarding the practical implementation of the approach outlined above, there are of course different ways to approximate the version space. In fact, the way in which the models in (7) are obtained strongly depends on the model class $\mathfrak{M}$. The basic idea is to apply randomization techniques as they are typically employed in ensemble learning methods. In the experiments below, we shall use ensembles of linear perceptrons, each of which is trained on a random permutation of the whole data.

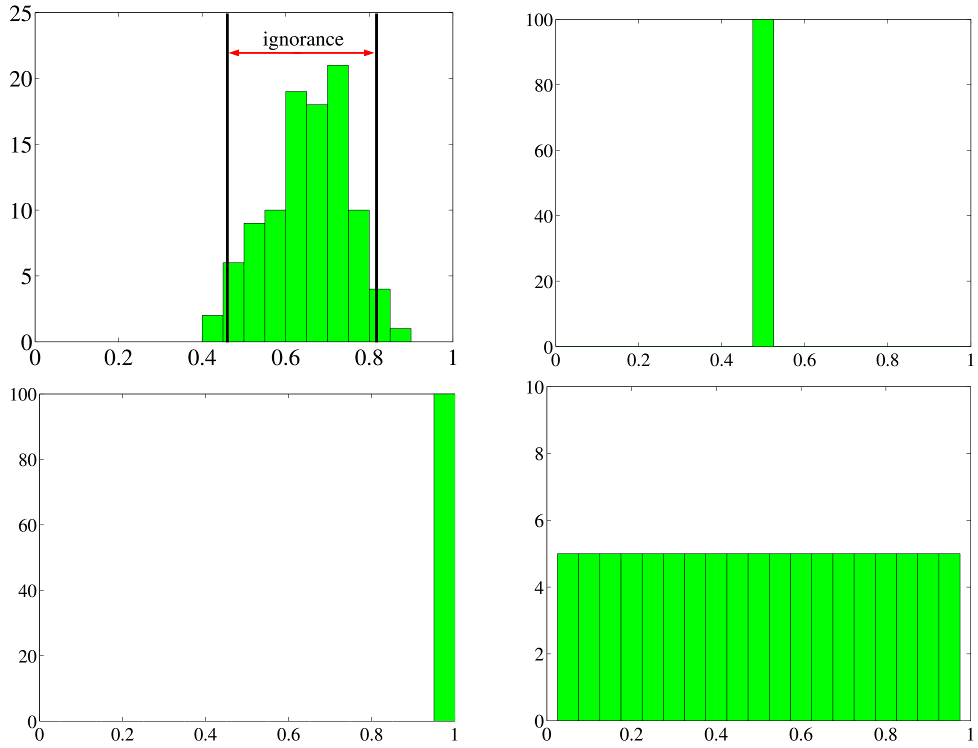

Figure 7: Distribution of the scores output by an ensemble $\mathbb{M}_{i,j}$. The degree of ignorance corresponds to the imprecision (width) of the distribution (here measured in a robust way in terms of the distance between the α- and $(1-\alpha)$-quantile), see upper left figure. Special cases include high conflict by no ignorance (upper right), complete certainty (lower left), and complete ignorance (lower right).

We conclude this section with a remark on complexity issues. Of course, the more detailed information that LVPC provides in comparison with simple multi-class classifiers is not obtained for free. Instead, it comes along with an increased computational complexity for training, mainly for two reasons: First, due to the decomposition, a quadratic number of binary models is trained instead of only a single m-class model. Note, however, that this does not imply an increase in complexity that is quadratic in m. In fact, since each binary problem involves only a fraction of the whole training data, the total number of examples used in the training phase increases only linearly, that is, by the factor m [10]. The second increase in complexity is due to the training of an ensemble (7) consisting of K models; obviously, the total complexity thus increases also linearly with K.

5 Potential Applications

LVPC as outlined above can be seen as a technique for deriving a condensed representation of the classification-relevant information as reflected by the version space. In fact, a simple visualization of the relations $\mathcal{P}$, $\mathcal{I}$, and $\mathcal{J}$ as shown, e.g., in Fig. 8, where preference degrees are indicated as levels of gray, may already be useful in order to get a rough idea of the current state of affairs. For example, to quality as a top-class, the associated row in the strict preference relation $\mathcal{P}$ should be rather dark (except the diagonal element), suggesting that the class label is (more or less) strictly preferred to all other ones. Likewise, by inspecting the matrices $\mathcal{I}$ and $\mathcal{J}$, one gets a first idea about the uncertainty of the pairwise comparisons.

However, once a preference structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ has been induced, it can also be taken as a point of departure for sophisticated decision strategies which go beyond simple voting procedures. This approach becomes especially interesting in extended classification scenarios, that is, generalizations of the conventional setting in which

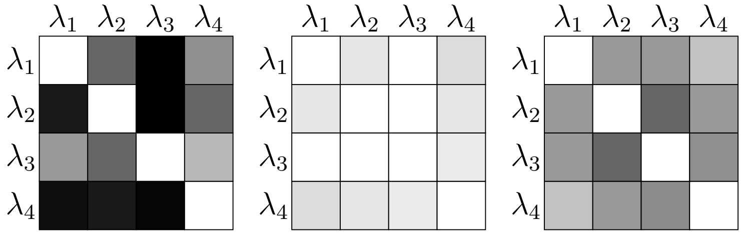

Figure 8: Visual representation of the three relations $\mathcal{P}$, $\mathcal{I}$, $\mathcal{J}$. Preference degrees are indicated as levels of gray. (Instead of strict preference degrees $p_{i,j}$ we plotted $1 - p_{i,j}$ for $i \neq j$, so that, for all three relations, dark fields are "critical" for the class label associated with the corresponding row; the rows of an ideal top-class are white throughout.)

a single decision in favor of a unique class label is requested; more generally,

- it might be allowed to predict several class labels instead of only a single one in cases of conflict, or
- to defer an immediate decision in cases of ignorance (or conflict).

The second scenario is known as *classification with reject option* in the literature, where one often distinguishes between *ambiguity rejection* [3, 12] and *distance rejection* [6]. Interestingly, this corresponds roughly to our distinction between conflict and ignorance. As we explained above, however, our conception of ignorance is more general and arguably more faithful, as it takes the underlying model assumptions into account: equating distance (between the query and observed examples) with ignorance does make sense for instance-based classifiers but not necessarily for other approaches with different model assumptions.

It is true that uncertainty of a classification can also be represented by a standard probabilistic classifier, or even by a more general type of scoring classifier: A classification appears uncertain as soon as at least one other class has a probability (score) which is almost as high as the probability (score) of the predicted class. One should recall, however, the well-known problem that probability cannot distinguish between conflict and ignorance, mainly due to the normalization constraint requiring that probability degrees must add to 1. In probability theory, for example, complete ignorance is usually modeled by the uniform probability distribution, as suggested by the principle of insufficient reason. In the third case of our example in Fig. 2, this means giving a probability of 1/2 to both classes. Now, the adequacy of the uniform distribution as a representation of ignorance has been called into question by several scholars [5]. In particular, the uniform distribution does not distinguish between complete ignorance and the situation where one can be quite sure that the class labels are indeed equi-probable, since the uniform distribution is strongly suggested by the examples given (case 2 in our example).

From a knowledge representational point of view, a distinction between conflict (e.g., exactly knowing that outcomes are equally likely) and ignorance (e.g., not knowing anything) clearly seems worthwhile. For example, telling a patient that your experience does not allow any statement concerning his prospect of survival is very different from telling him that his chance is fifty-fifty! Moreover, separating conflict from ignorance can also be useful from a decision making (classification) perspective, at least if there is a possibility to abstain from a decision as in classification with reject option. In fact, in a situation of high ignorance, it might be reasonable to refuse an immediate decision, and instead to gather additional information. For example, consider a physician who is looking for a diagnosis. If the current knowledge about the patient at hand is not sufficient for a reliable diagnosis, he will gather additional information, e.g., by making complementary medical tests or by searching the literature for similar cases. In machine learning terms, the first type of information procurement corresponds to adding features (attributes) to the instance representation, while the second one corresponds to gathering additional observations.

While abstaining from an immediate decision may be reasonable in a situation of ignorance, the same is not necessarily true in cases of conflict. In the second case

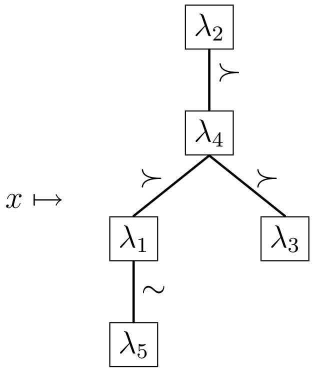

Figure 9: Illustration of a generalized prediction problem: An instance is mapped to a partial order over the complete set of class labels $\mathcal{L} = \{\lambda_1 \ldots \lambda_5\}$. A corresponding mapping can be seen as a generalization of a classification function.

of our example in Fig. 2, for instance, the usefulness of additional observations will be quite limited, i.e., corresponding examples will hardly resolve the conflict. As opposed to this, additional examples could be very helpful in the third case, where they may indeed reduce the level of conflict.

In this connection, another advantage of our *pairwise* learning scheme deserves mentioning: In order to characterize the current decision situation, the relation $\mathcal{J}$ does not only provide a single degree of ignorance but instead offers such degrees for every pair of classes separately. Thus, it becomes possible to focus the acquisition of additional information on particular classes or pairwise classifiers, thereby saving resources and reducing costs (see again Fig. 8).

Another interesting field of application is the prediction of *structured outputs.* The latter refers to an extension of standard classification learning in which, instead of predicting a single class label (the top-label), the problem is to predict a certain type of *preference relation* on the complete label set $\mathcal{L}$. As a special case, the problem of *label ranking* has recently attracted attention in the literature [11, 16]. Here, the sought preference structure is a total order (ranking): Given an input $x \in \mathcal{X}$, predict the ranking of the class labels $\mathcal{L}$ associated with this instance. To illustrate, imagine that an instance is a person, and that the ranking expresses his or her preferences with respect to a fixed set of movies (the class labels). An obvious extension of label ranking is the prediction of less restricted types of preference relations, such as weak or partial orders (see Fig. 9). For the time being, it is completely unclear how such types of prediction problems can be approached in a theoretically sound way, especially due to the problem of handling the incomparability relation. Our approach of LVPC may therefore provide an interesting basis for solving prediction problems of that kind.

6 Experimental Results

As pointed out in the previous section, our method LVPC is potentially useful in the context of various types of extended classification scenarios. Basically, it can be seen as a first step of the overall classification process: The preference structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ derived by LVPC serves as an input for a subsequent decision policy which is responsible for the final decision. As mentioned earlier, the design of suitable policies is highly application-specific and beyond the scope of this paper. In this section, we therefore restrict ourselves to a simple experimental setup which is suitable for testing a key feature of LVPC, namely its ability to represent the amount of uncertainty involved in a classification.

More specifically, we used LVPC as a means for implementing a reject option in the context of binary classification. To this end, we conducted an experimental study on 10 binary classification data sets from the Statlog and UCI repositories (cf. Fig. 10).[4] Each of the data sets was randomly split into a training and test set of (roughly) equal size. As model classes $\mathbb{M}_{i,j}$, we used ensembles of 100 perceptrons with linear kernels and the default additive diagonal constant 1 (to account for non-separable problems), which were induced on the training data. Each perceptron was provided with a random permutation of the training set in order to obtain a diverse ensemble

[4] These are preprocessed versions from the LIBSVM-website.

	name	# features	# classes	# examples
1	australian_scale	14	2	690
2	breast-cancer_scale	10	2	683
3	diabetes_scale	8	2	768
4	fourclass_scale	2	2	862
5	german	24	2	1000
6	heart_scale	13	2	270
7	ionosphere_scale	34	2	351
8	splice_scale	60	2	1000
9	sonar_scale	60	2	208
10	w1a	300	2	2477
11	vehicle	18	4	846
12	vovel	10	10	528

Figure 10: Data sets used in the experiments.

[14]. This process was repeated 10 times to reduce the bias induced by the random splitting procedure, and the results were averaged.

On the test sets, the real-valued classification outputs of the perceptrons were converted into normalized scores using a common logistic regression approach by Platt [23]. For a given test instance, the weak preference component $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$ was derived by the 0.1-quantile of the distribution of the scores from the ensemble $\mathbb{M}_{i,j}$ (see Section 4.4). Moreover, as a decomposition scheme we used a slight modification of (5):

$$\begin{aligned} \mathcal{P}(\lambda_i, \lambda_j) &= r_{i,j}\,(1 - r_{j,i}) \\ \mathcal{I}(\lambda_i, \lambda_j) &= 2\,r_{i,j}\,r_{j,i} \\ \mathcal{J}(\lambda_i, \lambda_j) &= 1 - (r_{i,j} + r_{j,i}) \end{aligned} \tag{8}$$

The reason for the modification is that in (8), the ignorance component nicely agrees with our derivation of weak preference degrees: It just corresponds to the width of the distribution of the scores generated by $\mathbb{M}_{i,j}$ (or, more precisely, the length of the interval between the quantiles of this distribution); therefore, it reflects the diversity of the predictions and becomes 0 if all ensemble members $\mathcal{M}_{i,j}^{(k)}$ agree on exactly the same score.

Finally, all test instances were ordered with respect to the associated degrees of conflict (ignorance), and corresponding accuracy-rejection diagrams were derived. These diagrams provide a visual representation of the accuracy levels α as a function of the rejection rate ρ: If the $\rho\%$ test instances with the highest degrees of conflict (ignorance) are refused, then the classification rate on the remaining test instances is α. In practice, an accuracy-rejection curve could be used, for example, in order to derive a reject rule that guarantees a certain reliability: Fixing a desired classification rate α^*, one finds a level of uncertainty γ (expressed, e.g., in terms of conflict and/or ignorance) such that, by rejecting instances for which the uncertainty is $> \gamma$, those instances that are not rejected are classified correctly with probability at least α.

Obviously, the effectiveness of LVPC in representing uncertainty is in direct correspondence with the shape of the accuracy-rejection curve: If the degree of conflict (ignorance) produced by LVPC is a good indicator of the reliability of a classification, then the ordering of instances according to conflict (ignorance) is in agreement with their respective degree of reliability (chance of misclassification), which in turn means that the accuracy-rejection curve is monotone increasing. The presumption that LVPC is indeed effective in this sense is perfectly confirmed by the experimental results, as can be seen in Fig. 11–12.

Experiments of this kind can of course also be made for multi-class data sets, even though the rejection rule needs to be generalized in this case. One possibility, for example, is to base the reject decision on the degree of conflict (ignorance) of the best class (in terms of the sum of strict pairwise preference) in comparison with the second-best class. To illustrate, Fig. 13 shows corresponding results that we obtained, respectively, for a 10-class and a 4-class data set.

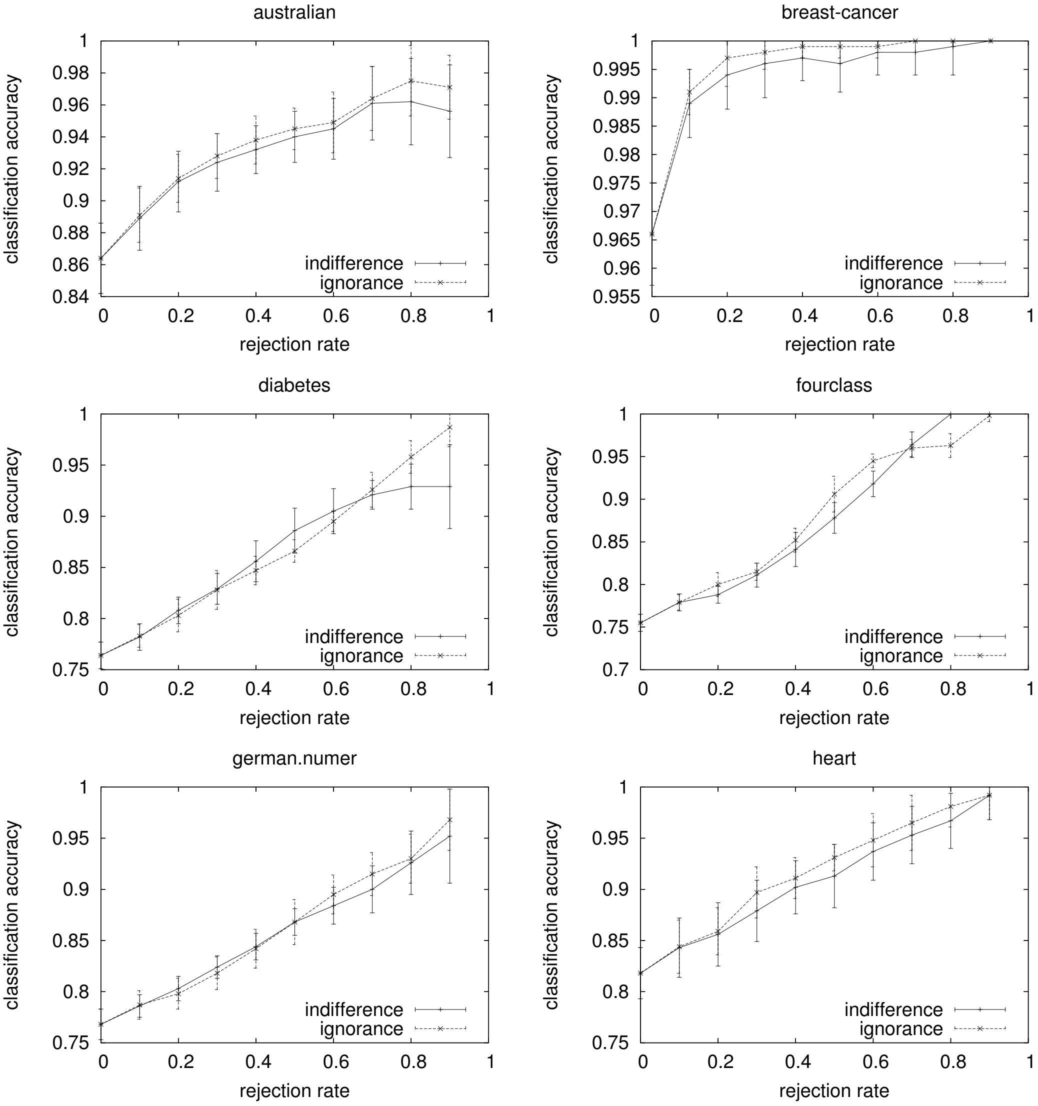

Figure 11: Accuracy-rejection curves for the data sets 1–6.

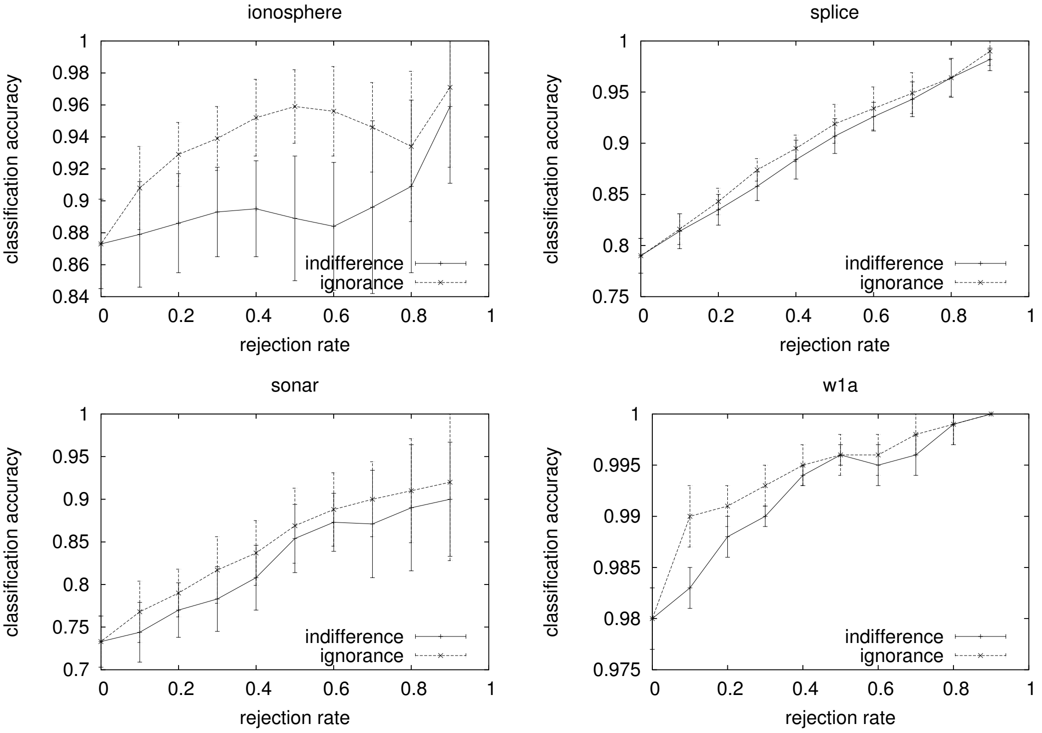

Figure 12: Accuracy-rejection curves for the data sets 7–10.

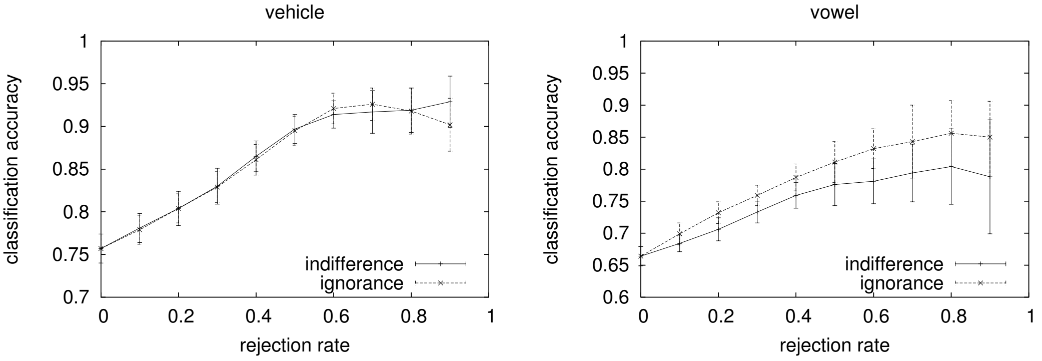

Figure 13: Accuracy-rejection curves for the data sets 11–12.

7 Related Work

The idea of learning valued preference relation for classification as introduced in this paper is, to the best of our knowledge, novel and has not been presented in this form before. Needless to say, however, there are several methods that are related to LVPC in one way or the other. In particular, regarding the methodological framework, the close connection of LVPC to pairwise learning in the field of machine learning and to preference modeling in fuzzy set theory has already been mentioned repeatedly. Moreover, regarding potential applications to classification, we already indicated the usefulness of LVPC for classification with reject option and reliable classification (e.g. [18]). Especially interesting in this connection is the approach proposed in [26], as it also makes use of the version space concept.

There is also an interesting relation to *Bayesian model averaging* [15]. Instead of picking a single model $\mathcal{M}^* \in \mathcal{H}$ from the hypothesis space, thereby ignoring uncertainty in model selection, the key idea of Bayesian model averaging is to combine the outputs of all models, weighing each model $\mathcal{M}$ by its posterior probability given the data $\mathcal{D}$. This way, Bayesian model averaging principally offers an alternative to characterizing uncertainty in prediction problems, even though the delicate question of whether or not ignorance can adequately be captured within a probabilistic framework remains an issue. Roughly speaking, the Bayesian approach, or at least an approximation thereof, would correspond to deriving the average of the classifier scores $s_{i,j}^{(k)}$, $k = 1 \ldots K$, thereby losing information about the (im)precision of the distribution.

Finally, let us note that a classifier using fuzzy relations was also developed in [25]. However, this approach is completely different from ours. In a first, explanatory step, it employs clustering techniques in order to structure the input data (disregarding class labels). In a second step, a fuzzy relation is created which associates the classes (rows) with the clusters (columns). Using a suitable fuzzy inference mechanism, new queries are then classified on the basis of this relation and its distance to the cluster centers.

8 Conclusions

In this paper, we have introduced a new approach to classification learning which refers to the concept of fuzzy preference structures. This approach is intimately related with learning by pairwise comparison (LPC), a well-known machine learning technique for reducing multi-class to binary problems. The key idea of our approach, called LVPC, is to use LPC in order to learn a fuzzy (weak) preference relation among the potential class labels. The original classification problem thus becomes a problem of decision making, namely of taking a course of action on the basis of this fuzzy preference relation. This way, our approach makes machine learning amenable to techniques and decision making strategies that have been studied intensively in the literature on fuzzy preferences.

An interesting example of corresponding techniques has been considered in more detail in this paper, namely the decomposition of a weak preference relation into a strict preference, an indifference, and an incomparability relation. We have argued that, in a classification context, indifference can be interpreted as the *conflict* involved in a prediction while indifference represents the level of *ignorance*. These concepts can be extremely useful, especially in extended classification scenarios which go beyond the prediction of a single label or do offer the option to abstain from an immediate classification decision.

First empirical studies have shown that LVPC is indeed able to represent the uncertainty related to a classification decision: The implementation of a reject option turned out to be highly effective, regardless of whether the decision to abstain is made on the basis of the degree of conflict or the degree of ignorance.

The main contribution of this paper is a basic conceptual framework of classification based on learning valued preference relations, including first empirical evidence in favor of its usefulness. Nevertheless, as we mentioned repeatedly, this framework is far from being fully elaborated and still leaves much scope for further developments. This concerns almost all steps of the approach and includes both aspects of learning and decision making. Just to give an example, our approach outlined in Section 4.4 is of course not the only way to induce a valued preference structure from given data. In particular, rule-based models may provide an interesting alternative. A distinguishing feature of such classifiers, that we already illustrated in Fig. 3, is their ability to represent conflict and ignorance in a direct way. In principle, this model class hence offers the possibility to learn the relations $\mathcal{P}$, $\mathcal{I}$, $\mathcal{J}$ directly instead of inducing them indirectly via the weak preference $\mathcal{R}$. We are currently exploring this alternative as part of ongoing work.

Acknowledgements. This research was supported by the German Research Foundation (DFG). We like to thank three anonymous reviewers for useful comments and suggestions on a previous version of this paper.

References

[1] E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

[2] B. De Baets, B. Van de Walle, and E. Kerre. Fuzzy preference structures and their characterization. *The Journal of Fuzzy Mathematics*, 3:373–381, 1995.

[3] CK. Chow. An optimum character recognition system using decision functions. *Trans. on Electronic Computers*, 6:247–253, 1957.

[4] TG. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[5] D. Dubois, H. Prade, and P. Smets. Representing partial ignorance. 26(3):361–377, 1996.

[6] B. Dubuisson and MH. Masson. A statistical decision rule with incomplete knowledge about classes. *Pattern Recognition*, 26(1):155–165, 1993.

[7] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support.* Kluwer, 1994.

[8] J. Fodor and M. Roubens. Valued preference structures. *European Journal of Operational Research*, 79:277–286, 1994.

[9] J. Fürnkranz. Round robin rule learning. In *ICML-2001, Proc. 18th International Conference on Machine Learning*, pages 146–153, Williamstown, MA, 2001.

[10] J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

[11] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Proc.* ECML–2003*, 13th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, September 2003.

[12] TM. Ha. The optimum class-selective decision rule. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):608–615, 1997.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2001.

[14] Ralf Herbrich. *Learning Kernel Classifiers.* MIT Press, 2002.

[15] JA. Hoeting, D. Madigan, AE. Raftery, and CT. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.

[16] E. Hüllermeier and J. Fürnkranz. Learning label preferences: Ranking error versus position error. In *Proceedings IDA–05, 6th International Symposium on Intelligent Data Analysis*, number 3646 in LNCS, pages 180–191, Madrid, 2005. Springer-Verlag.

[17] EP. Klement, R. Mesiar, and E. Pap. *Triangular Norms.* Kluwer Academic Publishers, 2002.

[18] M. Kukar and I. Kononenko. Reliable classifications with machine learning. In *Proc. European Conference on Machine Learning, ECML*, pages 219–231, 2002.

[19] T.M. Mitchell. The Need for Biases in Learning Generalizations. Technical report CBM–TR–117, Rutgers University, 1980.

[20] S. Orlovski. Decision-making with a fuzzy preference relation. *Fuzzy Sets and Systems*, 1:155–167, 1978.

[21] S. Ovchinnikov and V. Ozernoy. Using fuzzy binary relations for identifying noninferior decision alternatives. *Fuzzy Sets and Systems*, 25:21–32, 1988.

[22] P. Perny and B. Roy. The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems*, 49:33–53, 1992.

[23] John Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A.J. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, Cambridge, MA, 1999. MIT Press.

[24] B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[25] M. Setnes and R. Babuska. Fuzzy relational classifier trained by fuzzy clustering. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 29(5):619–625, 1999.

[26] E. Smirnov, I. Sprinkhuizen-Kuyper, G. Nalbantov, and S. Vanderlooy. Version space support vector machines. In *Proc. ECAI-06, 17th European Conference on Artificial Intelligence*, pages 809–810, Riva del Garda, Spain, 2006.

[27] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.