Comparison of Ranking Procedures in Pairwise Preference Learning

Eyke Hüllermeier
Marburg University, Germany
eyke@informatik.uni-marburg.de

Johannes Fürnkranz
Darmstadt University of Technology
fuernkranz@informatik.tu-darmstadt.de

Abstract

Computational methods for discovering the preferences of individuals are useful in many applications. In this paper, we propose a method for learning *valued preference structures*, using a natural extension of so-called pairwise classification. A valued preference structure can then be used in order to induce a *ranking*, that is a linear ordering of a given set of alternatives. This step is realized by means of a so-called ranking procedure. In the second part of the paper, we compare the performance of alternative ranking procedures in an experimental way.

Keywords: Machine learning, valued preference structures, ranking.

1 Introduction

The increasing trend toward *personalization* of products and services in e-commerce and various other fields requires computational methods for discovering the *preferences* of individuals. And indeed, methods for learning and predicting preferences in an automatic way are among the very recent research topics in disciplines such as machine learning and recommender systems.

The term *preference elicitation* usually refers to the problem of estimating the preferences of a *single* individual. In this paper, we take a slightly different view on preference learning. Our goal is to predict the preferences of an individual on the basis of certain properties of that individual and known preferences of other individuals, i.e. to establish a relationship between features describing individuals and preference models. To illustrate, consider a salesman who knows from experience that "Middle-aged, working women without children usually prefer product A to product B to product C". The salesman has learned from experience to predict preferences of his clients, and this is what we want a machine learning or recommender system to do.

The above can be seen as an extension of supervised machine learning, where examples are labeled with preference relations over possible categorizations. For machine learning, this type of problem is particularly challenging as it goes beyond the prediction of single values (such as real numbers in regression analysis and class labels in pattern recognition). Instead, it involves the prediction of preference models, such as relational structures or value functions.

The problem of preference learning, as roughly outlined above, is formally introduced in Section 2. In Section 3, we propose a method for learning a special type of preference model, namely valued (fuzzy) preference structures. The problem of predicting a ranking is then addressed in Section 4, where several ranking procedures are introduced. Section 5 presents results of experimental studies investigating the performance of these ranking methods.

2 Preference Learning

Consider a set $\mathcal{X}$ of individuals/instances characterized in terms of an attribute-value representation: $\mathcal{X} = X_1 \times X_2 \times \ldots \times X_l$, where $X_\imath$ is the domain of the $\imath$-th attribute. Thus, an instance is represented as a vector $x = (x_1 \ldots x_l) \in \mathcal{X}$. Let $\mathcal{L} = \{\lambda_1 \ldots \lambda_m\}$ be a set of alternatives/labels. We assume that each individual has preferences concerning the alternatives $\lambda \in \mathcal{L}$. Formally, this can be expressed in terms of a *preference function*

$$\mathcal{X} \to \mathfrak{P} \tag{1}$$

that maps individuals to preference models; $\mathfrak{P}$ denotes the class of potential models. In this paper, we are especially interested in two types of models: *valued preference* relations and total orders aka *rankings*. Both models are closely related, as will be discussed in Section 4 below. Roughly, one can imagine that the "true" preferences of an individual are represented by a valued relation. A ranking, then, can be seen as "revealed" preferences: If it comes to acting in situations where a definite choice between alternatives must be made, the individual is forced to map his fuzzy preferences to non-fuzzy ones. Doing this in a consistent (rational) way, he should finally be able to come up with a ranking of all alternatives. For convenience, we shall subsequently assume that this ranking is not only a weak but even a total order (i.e. no ties are allowed in the ordering).

The ranking $\succ_x$ of individual x can be expressed in terms of a permutation π_x of $\{1 \ldots m\}$ such that

$$\lambda_{\pi_x(1)} \succ_x \lambda_{\pi_x(2)} \succ_x \ldots \succ_x \lambda_{\pi_x(m)}. \tag{2}$$

Here, $\lambda_\imath \succ_x \lambda_\jmath$ means that x (strictly) prefers $\lambda_\imath$ to $\lambda_\jmath$.

The problem of *preference learning* now consists of learning (approximating) the mapping (1) on the basis of empirical data. Again, one can think of different types of observations. Here, we assume that a single piece of information corresponds to *comparative* preference information of the form $\lambda_\imath \succ_x \lambda_\jmath$, i.e. "individual x prefers $\lambda_\imath$ to $\lambda_\jmath$". This type of information is often easier to obtain than absolute ratings of single alternatives in terms of utility degrees. Note that knowledge about the complete ranking (2) can be expanded into $m(m-1)/2$ binary preferences $\lambda_{\pi_x(\imath)} \succ \lambda_{\pi_x(\jmath)}$ for $1 \leq \imath < \jmath \leq m$.

3 Pairwise Preference Learning

The idea of pairwise learning is well-known in the context of classification [2], where it allows one to transform an m-class classification problem, i.e., a problem involving $m > 2$ classes $\mathcal{L} = \{\lambda_1 \ldots \lambda_m\}$, into a number of *binary* problems. To this end, a separate model (base learner) $\mathcal{M}_{\imath\jmath}$ is trained for each *pair* of labels $(\lambda_\imath, \lambda_\jmath) \in \mathcal{L}$, $1 \leq \imath < \jmath \leq m$. Thus, a total number of $m(m-1)/2$ models is needed. $\mathcal{M}_{\imath\jmath}$ is intended to separate the classes $C_\imath$ (objects with label $\lambda_\imath$) and $C_\jmath$.

At classification time, a query is submitted to all learners, and each prediction is interpreted as a vote for a label: If classifier $\mathcal{M}_{\imath\jmath}$ predicts $\lambda_\imath$, this is counted as a vote for $\lambda_\imath$. Conversely, the prediction $\lambda_\jmath$ would be considered as a vote for $\lambda_\jmath$. The label with the highest number of votes is then proposed as a prediction.

The above procedure can be extended to the case of preference learning or, more precisely, the learning of rankings in a natural way [3]. A preference information of the form $\lambda_\imath \succ_x \lambda_\jmath$ is turned into a training example (x, y) for the learner $\mathcal{M}_{ab}$, where $a = \min(\imath, \jmath)$ and $b = \max(\imath, \jmath)$. Moreover, $y = 1$ if $\imath < \jmath$ and $= 0$ otherwise. Thus, $\mathcal{M}_{ab}$ is intended to learn the mapping

$$x \mapsto \begin{cases} 1 & \text{if} \quad \lambda_a \succ_x \lambda_b \\ 0 & \text{if} \quad \lambda_b \succ_x \lambda_a \end{cases}. \tag{3}$$

In other words, given an instance x as an input, $\mathcal{M}_{ab}$ is assumed to output 1 if $\lambda_a \succ_x \lambda_b$ and 0 if $\lambda_b \succ_x \lambda_a$.

The mapping (3) can be realized by any binary classifier. Alternatively, one might of course also employ a classifier that maps into $[0, 1]$ instead of $\{0, 1\}$. The output of such a "soft" binary classifier can usually be interpreted as a probability or, more generally, a kind of confidence in the classification. Thus,

the closer the output of $\mathcal{M}_{ab}$ to 1, the stronger the preference $\lambda_a \succ_x \lambda_b$ is supported.

A soft classifier naturally leads to a valued (fuzzy) preference relation $\mathcal{R}_x$ associated with an instance x:

$$\mathcal{R}_x(\lambda_i, \lambda_j) = \begin{cases} \mathcal{M}_{ij}(x) & \text{if} \quad i < j \\ 1 - \mathcal{M}_{ij}(x) & \text{if} \quad i > j \end{cases}$$

for all $\lambda_i \neq \lambda_j \in \mathcal{L}$. Thus, we have obtained a preference learner, composed of an ensemble of (soft) binary classifiers, which can be constructed on the basis of training data in the form of individuals with associated (partial) preferences. This preference learner assigns a valued preference relation to any (query) instance $x \in \mathcal{X}$.

4 Ranking Procedures

Let us now consider the problem of predicting the revealed preferences of an instance x, characterized in terms of a ranking π_x. One possibility is to induce a preference relation $\mathcal{R}_x$ as outlined above and to derive a ranking from that relation. Unfortunately, a relation $\mathcal{R}_x$ does not always suggest a unique ranking in an unequivocal way. In fact, the problem of inducing a ranking from a (valued) preference relation has received a lot of attention in several research fields, e.g., in fuzzy preference modeling and (multi-attribute) decision making [1].

4.1 Simple Voting

The most common approach to ranking on the basis of a preference relation $\mathcal{R}$ makes use of a so-called *scoring function* S. This function assigns a score $S(\lambda_i) = S(\lambda_i \,|\, \mathcal{R})$ to any alternative λ_i, and a ranking is then derived on the basis of these scores:

$$(\lambda_i \succeq \lambda_j) \Leftrightarrow (S(\lambda_i) \geq S(\lambda_j)).$$

The simplest scoring function is defined by the sum of (weighted) votes

$$S(\lambda_i) = \sum_{\lambda_j \neq \lambda_i} \mathcal{R}(\lambda_i, \lambda_j). \tag{4}$$

The corresponding voting procedure is commonly used in pairwise classification and ranking [3].

4.2 Ranking Through Iterated Choice

An alternative approach to ranking makes use of a so-called *choice function*, which is a function that selects one or several maximally preferred elements from a set of candidates. Obviously, a ranking can be obtained by applying such a function in a repeated way: First, the top-label is chosen from the complete set, then the second best label is selected from the remaining alternatives, and so on.

In our context, a natural choice function is based on the *probability* that a particular label is maximally preferred among all candidates. Thus, let $\mathsf{Pr}(E_i)$ denote the event that λ_i is maximally preferred. In the classification setting, $\mathsf{Pr}(E_i)$ is nothing else than the probability that λ_i is the correct class.

Note that

$$\begin{aligned}(m-1)\,\mathsf{Pr}(E_i) &= \sum_{j \neq i} \mathsf{Pr}(E_i) \\ &= \sum_{j \neq i} \mathsf{Pr}(E_i \,|\, E_{ij}) \mathsf{Pr}(E_{ij}),\end{aligned} \tag{5}$$

where E_{ij} denotes the event that either λ_i or λ_j is selected and m is the number of labels. Since the (pairwise) estimates $\mathcal{R}(\lambda_i, \lambda_j)$ can be considered as probabilities $\mathsf{Pr}(E_i \,|\, E_{ij})$, we have

$$\mathsf{Pr}(E_i) = \frac{1}{m-1} \sum_{j \neq i} \mathcal{R}(\lambda_i, \lambda_j) \mathsf{Pr}(E_{ij}).$$

Replacing E_{ij} in the last equation by $\mathsf{Pr}(E_i) + \mathsf{Pr}(E_j)$ leads to a system of linear equations for the probabilities $\mathsf{Pr}(E_i)$. In conjunction with the constraint $\sum_{i=1}^{m} \mathsf{Pr}(E_i) = 1$, this system has a unique solution provided that $\mathcal{R}(\lambda_i, \lambda_j) > 0$ for all $1 \leq i, j \leq m$ [6].

The above results suggest the following ranking procedure: First, the label λ_i with maximal $\mathsf{Pr}(\lambda_i)$ is chosen as the top-label. This label is then removed, i.e., the corresponding row and column of the relation $\mathcal{R}$ is deleted. The same procedure is then applied to the reduced relation in order to find the second best label, and so on.

4.3 Slater-Optimal Rankings

Another approach is to look for a ranking that is maximally consistent with a given preference relation $\mathcal{R}$. Since the distinguishing feature of a ranking is transitivity, this mainly comes down to finding the smallest possible modification of $\mathcal{R}$ that makes it transitive. This approach appears especially appealing in our context of learning: Since the values $\mathcal{R}(\lambda_\imath, \lambda_\jmath)$ are predicted by non-perfect learners $\mathcal{M}_{\imath\jmath}$, they are not necessarily correct.

In the binary case where $\mathcal{R}(\lambda_\imath, \lambda_\jmath) \in \{0, 1\}$, modifying $\mathcal{R}$ comes down to inverting edges in an associated directed graph (each label $\lambda_\imath$ corresponds to a node $n_\imath$, and a directed edge $n_\imath \to n_\jmath$ indicates that $\lambda_\imath$ is preferred to $\lambda_\jmath$). The problem of making a complete directed graph acyclic with a minimal number of such modifications is known as the *feedback arc set problem* in graph theory, and the number itself is called the *Slater order* of the graph [5]. Thus, the problem is to find a ranking (permutation) π that minimizes the number of feedback arcs, i.e., arcs from $\lambda_\jmath$ to $\lambda_\imath$ for $\pi(\imath) < \pi(\jmath)$.

In our context, inverting an edge can be interpreted as outvoting the learner $\mathcal{M}_{\imath\jmath}$. A minimal modification is then also a most probable one, at least under the assumption that learners are independent and make correct decisions with probability $> 1/2$. Obviously, the feedback arc set problem can be extended to the case of valued relations in a relatively straightforward way: Instead of simply counting the feedback arcs, the goal is to find a ranking (permutation) π that minimizes

$$\sum_{\imath<\jmath} \mathcal{R}(\lambda_{\pi(\jmath)}, \lambda_{\pi(\imath)}). \tag{6}$$

We shall say that such a ranking is Slater-optimal. It deserves mentioning that the problem of finding Slater-optimal rankings is known to be NP complete [4]. This is hardly relevant for small enough label sets $\mathcal{L}$ but of course becomes a disadvantage in the case where $\mathcal{L}$ comprises many labels.

4.4 Example

In order to illustrate the difference between the above ranking methods, we consider a simple example with $m = 4$ labels. Suppose that the true ranking is given by $\lambda_1 \succ \lambda_2 \succ \lambda_3 \succ \lambda_4$ and let the relation $\mathcal{R}$ be given by

$$\mathcal{R} = \begin{pmatrix} - & .9 & .8 & .1 \\ .1 & - & .9 & .9 \\ .2 & .1 & - & .9 \\ .9 & .1 & .1 & - \end{pmatrix}.$$

Obviously, only the learner $\mathcal{M}_{14}$ has made an error, since it strongly prefers λ_4 to λ_1.

The simple voting function (4) yields the scores 1.8, 1.9, 1.2, 1.1 and, hence, the corresponding ranking $\lambda_2 \succ \lambda_1 \succ \lambda_3 \succ \lambda_4$.

The Slater-approach recognizes and repairs the mis-classification by $\mathcal{M}_{14}$. Indeed, this mis-classification is the simplest explanation for the fact that the binary graph associated with $\mathcal{R}$ (containing an edge $n_\imath \to n_\jmath$ if $\mathcal{R}(\lambda_\imath, \lambda_\jmath) > .5$) is not transitive. More generally, the correct ranking $\lambda_1 \succ \lambda_2 \succ \lambda_3 \succ \lambda_4$ is the minimizer of (6) and is hence suggested by this method.

The iterated choice method introduced in Section 4.2 leads to the following equations for the probabilities $\mathsf{Pr}(E_\imath)$:

$$\begin{aligned} 1.2\,\mathsf{Pr}(E_1) &= .9\,\mathsf{Pr}(E_2) + .8\,\mathsf{Pr}(E_3) + .1\,\mathsf{Pr}(E_4) \\ 1.1\,\mathsf{Pr}(E_2) &= .1\,\mathsf{Pr}(E_1) + .9\,\mathsf{Pr}(E_3) + .9\,\mathsf{Pr}(E_4) \\ 1.8\,\mathsf{Pr}(E_3) &= .2\,\mathsf{Pr}(E_1) + .1\,\mathsf{Pr}(E_2) + .9\,\mathsf{Pr}(E_4) \\ 1.9\,\mathsf{Pr}(E_4) &= .9\,\mathsf{Pr}(E_1) + .1\,\mathsf{Pr}(E_2) + .1\,\mathsf{Pr}(E_3) \end{aligned}$$

One obtains

$$\begin{aligned} \mathsf{Pr}(E_1) \approx .35, \quad \mathsf{Pr}(E_2) \approx .31, \\ \mathsf{Pr}(E_3) \approx .15, \quad \mathsf{Pr}(E_4) \approx .19 \end{aligned} \tag{7}$$

and, hence, λ_1 as the top-label. The first column and first row of the relation $\mathcal{R}$ are then deleted, and the reduced set of three linear equations is solved. The solution suggests λ_2 as the maximally preferred label, hence λ_2 is placed second. In the third step, λ_3 is selected, which means that the last rank is assigned to the remaining alternative λ_4.

Therefore, the ranking predicted by this method is again the correct ranking $\lambda_1 \succ$

$\lambda_2 \succ \lambda_3 \succ \lambda_4$. Note that this result is different from the ranking $\lambda_1 \succ \lambda_2 \succ \lambda_4 \succ \lambda_3$ that would have been obtained by simply using the probabilities (7) of the standard classification setting for ordering the labels.

5 Comparison of Ranking Procedures

In this section, we present some empirical results for pairwise learning of valued preference relations and rankings. Our special interest concerns the comparison of the ranking procedures discussed above, as well as the comparison between ranking on the basis of valued and binary preference relations.

5.1 Base Learners and Data Model

Our experiments are based on synthetic data that comes from an ensemble of "idealized" base learners. This way, it becomes possible to conduct experiments in a controlled way. More specifically, we consider the learners $\mathcal{M}_{ab}$ as independent and identically distributed random variables. That is, each learner yields outputs according to fixed probability distributions f_0 and f_1: For each $0 \leq x \leq 1$, $f_0(x)$ is the probability (density) that $\mathcal{M}_{ab} = x$ given that the correct output is 0, which means that $\lambda_b \succ_x \lambda_a$. Likewise, $f_1(x)$ is the probability (density) that $\mathcal{M}_{ab} = x$ given that $\lambda_a \succ_x \lambda_b$. It is of course reasonable to assume $f_0(x) = f_1(1-x)$. Without loss of generality, we can also assume that the correct output is always 1, i.e., that the true ranking is $\lambda_1 \succ_x \lambda_2 \succ_x \ldots \succ_x \lambda_m$. Consequently, we only need one probability distribution $f = f_1$, where $f(x)$ corresponds to the probability (density) that the distance between the given and the correct output is $1-x$.

For our experiments, we specified f by truncating the normal distribution with mode μ and standard deviation σ (see Fig. 1). These two parameters can be used in order to control the performance of a learner. The condition $\mu > 0.5$ is necessary to guarantee that a learner is better than random guessing. Note that under this condition the *expected* outcome is smaller than μ. Moreover, the larger

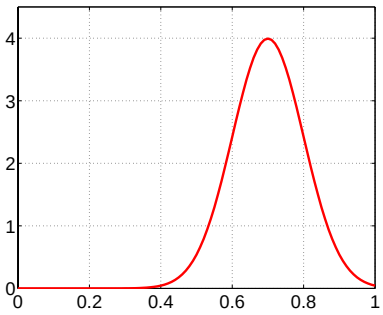

Figure 1: Example of a probability distribution f characterizing the performance of a base learner ($\mu = 0.7$, $\sigma = 0.1$).

the standard deviation σ, the smaller the expected outcome will be. In other words, the performance of a learner, measured in terms of the expected distance to the correct output, increases with μ and decreases with σ.

In order to allow for a reasonable comparison between valued and binary learners, we simply derive the latter from the former:

$$\mathcal{M}_{ab}^{bin}(x) = \begin{cases} 1 & \text{if} \quad \mathcal{M}_{ab}(x) \geq .5 \\ 0 & \text{if} \quad \mathcal{M}_{ab}(x) < .5 \end{cases} .$$

This corresponds to the usual way of deriving a definite classification from a soft classifier.

Note that, for a learner $\mathcal{M}_{ab}$ modeled by a random variable X, the expected distance from the correct output is given by $1 - \mathbb{E}(X)$, where $\mathbb{E}(X)$ is the expected value of X. For the corresponding binary learner $\mathcal{M}_{ab}^{bin}$, the expected distance is given by $\mathsf{Pr}(X < .5) = \int_0^{.5} f(x)\,dx$. Therefore, the former performs better than the latter only if

$$\mathbb{E}(X) \geq \mathsf{Pr}(X \geq .5).$$

Strictly speaking, this comparison only applies to a classification setting with a single learner, not to the ranking setting including a complete ensemble of learners. Moreover, it assumes a particular performance measure, namely the distance from the correct output. Note, however, that this measure is quite reasonable, especially in connection with the scoring approach (4) where individual outputs $\mathcal{M}_{ab}$ are aggregated by summation. In any case, the above result suggests that using a

valued preference relation instead of its "binarization" will not necessarily give better results. At first sight, this might be surprising, since replacing $\mathcal{M}_{ab}$ by $\mathcal{M}_{ab}^{bin}$ resp. $\mathcal{R}$ by $\mathcal{R}^{bin}$ obviously involves a loss of information. Note, however, that binarization can be considered as a *reinforcement* of the learners' estimations, which might be reasonable if these estimations are reliable enough.

5.2 Ranking Procedures

In our experimental studies, we have compared the ranking procedures introduced in Section 4:

VOTE: The simple ranking procedure based on the scoring function (4);
CHOICE: The ranking procedure based on the choice function discussed in Section 4.2;
SLATER: The selection of a Slater-optimal ranking (6).

5.3 Distance Between Rankings

In order to quantify the accuracy of a predicted ranking it is necessary to measure its distance from the true ranking. For our experiments, we employed the normalized Euclidean distance

$$\left(\frac{1}{c_m}\sum_{\imath=1}^{m}(\pi(\imath)-\pi^*(\imath))^2\right)^{1/2}, \qquad (8)$$

where π and π^* denote, respectively, the permutations associated with the predicted and the true ranking. The normalizing constant c_m is given by $(m-1)^2+(m-3)^2+\ldots = m(m^2-1)/3$ and corresponds to the (squared) distance between two completely opposite rankings. Thus, (8) yields values in the unit interval, in particular 0 for identical rankings and 1 for opposite rankings.

5.4 Experiments

Our probabilistic setting is parameterized by the number of labels, m, and the parameters μ and σ characterizing a learner's performance.

For a particular experimental setup, i.e., a fixed set of parameters, the distance between the true ranking and the ranking predicted by a particular method is a random variable with a well-defined expectation value. The latter corresponds to the *expected distance* between the true ranking and the predicted one and defines a reasonable quality measure for a ranking method. We have approximated these expected distances by corresponding averages over 100,000 random experiments.

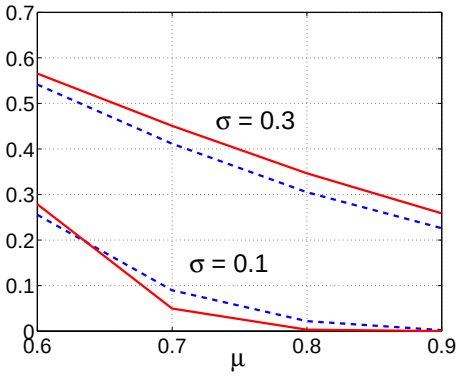

Figure 2: Expected distance between predicted and true ranking as a function of the parameter μ: VOTE (dashed, blue line) and VOTE-B (solid, red line).

A single experiment basically consists of the following steps: (1) For each of the $m(m-1)/2$ learners $\mathcal{M}_{ab}$, the output is generated according to the probability distribution f which is specified by μ and σ. (2) For all ranking methods under consideration, the predicted ranking is derived from the resulting relation $\mathcal{R}$. (3) The distance between the predicted and the correct ranking is computed.

In our first experimental study, we compared ranking on the basis of a valued preference relation with ranking on the basis of a binary relation. More precisely, we compared the performance of VOTE and VOTE-B, where the latter stands for the VOTE method applied to the binary relation $\mathcal{R}^{bin}$ instead of $\mathcal{R}$. In VOTE-B there is obviously a non-negligible probability that labels have the same score. Such ties were simply broken by coin flipping, that is, labels having the same score were put in a random order.

The results that have been derived for $\mu \in$

μ	σ	VOTE	VOTE-B
.6	.1	**.2556**	.2790
.7	.1	.0897	**.0498**
.8	.1	.0224	**.0031**
.9	.1	.0023	.0001
.6	.3	**.5411**	.5652
.7	.3	**.4119**	.4510
.8	.3	**.3055**	.3468
.9	.3	**.2253**	.2578
.6	.5	**.6214**	.6316
.7	.5	**.5650**	.5837
.8	.5	**.5196**	.5353
.9	.5	**.4568**	.4895
.6	.7	**.6501**	.6531
.7	.7	**.6199**	.6296
.8	.7	**.5889**	.6023
.9	.7	**.5592**	.5778
.6	.9	**.6596**	.6625
.7	.9	**.6431**	.6483
.8	.9	**.6239**	.6321
.9	.9	**.6048**	.6159

Table 1: Performance of VOTE and VOTE-B for various parameter settings.

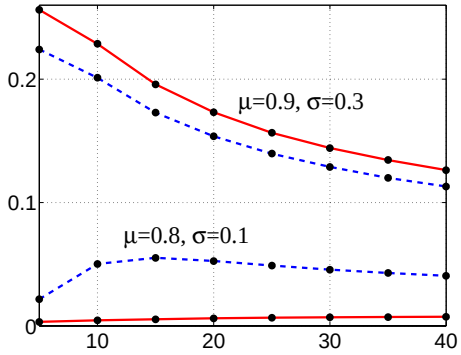

Figure 3: Expected distance for $(\mu, \sigma) = (.8, .1)$ and $(.9, .3)$ as a function of m: VOTE (dashed, blue line) and VOTE-B (solid, red).

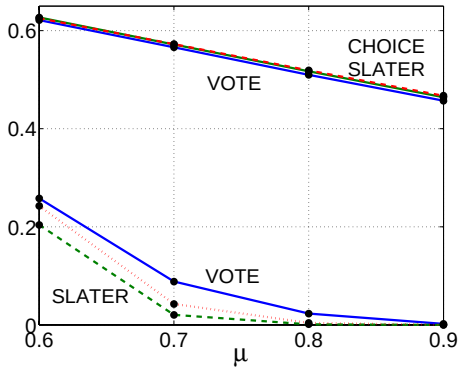

Figure 4: Comparison between VOTE (solid line), CHOICE (dotted), and SLATER (dashed) for $m = 5$, $\sigma = 0.1$ (lower curves) and $\sigma = 0.5$ (upper curves).

$\{.6, .7, .8, .9\}$, $\sigma \in \{.1, .3, .5, .7, .9\}$ and $m = 5$ are shown in Table 1. Apart from the case $(\mu, \sigma) = (.9, .1)$, all differences between the mean distance for VOTE and VOTE-B are statistically significant (at the 5% or even 1% level for a standard t-test). In these latter cases, the result of the superior method is set off in bold face.

As can be seen, VOTE-B gives indeed better results for a certain range of (μ, σ) values, namely $\sigma = 0.1$ and $\mu \in \{.7, .8, .9\}$ (cf. Fig. 2). Roughly, the results show that the stronger the learner (i.e. the larger μ and the smaller σ), the better VOTE-B performs in comparison with VOTE.

First experiments with larger label sets indicate that the superiority of one method over another one is not reversed when changing the parameter m (hence only depends on (μ, σ)). However, the absolute value of a performance measure as well as the distance between two measures may thoroughly change (despite normalization), even in a non-monotone way. For example, Fig. 3 shows the expected distance for VOTE and VOTE-B as a function of m. As can be seen, the dependence on m is quite different for different parameters (μ, σ). A thorough investigation of the influence of the number of labels is beyond the scope of this paper.

In our second experimental study, we compared the performance of VOTE with the performance of CHOICE and SLATER. The complete results for $m = 5$ are shown in Table 2 (qualitatively similar results have been obtained for larger values of m). Again, the best performing method is set off in bold if the distance to the second best is statistically significant at the 5% level. As can be seen, the simple VOTE procedure is indeed outperformed by CHOICE and SLATER in the

μ	σ	VOTE	SLATER	CHOICE
.6	.1	.2558	**.2047**	.2404
.7	.1	.0898	**.0226**	.0452
.8	.1	.0220	**.0012**	.0039
.9	.1	.0023	**.0001**	.0002
.6	.3	**.5414**	.5501	.5513
.7	.3	.4115	.4125	.4181
.8	.3	**.3071**	.2872	.3011
.9	.3	**.2248**	.1857	.2037
.6	.5	**.6218**	.6267	.6265
.7	.5	**.5666**	.5737	.5736
.8	.5	**.5093**	.5162	.5176
.9	.5	**.4565**	.4614	.4646
.6	.7	**.6479**	.6508	.6505
.7	.7	**.6198**	.6236	.6240
.8	.7	**.5906**	.5962	.5961
.9	.7	**.5585**	.5650	.5656
.6	.9	.6596	.6614	.6609
.7	.9	**.6431**	.6452	.6450
.8	.9	**.6245**	.6291	.6282
.9	.9	**.6056**	.6104	.6107

Table 2: Performance of VOTE, CHOICE, and SLATER for various parameter settings.

case of rather strong base learners (cf. Fig. 4). Again, however, this result does not extend to other parameter settings. In fact, in most cases VOTE is the best method, even though the differences are marginal.

These findings suggest that the computationally more complex methods SLATER and CHOICE can indeed improve the predictive performance, but only in the case of strong base learners. In order to explain this phenomenon, recall that both methods somehow try to "correct" or "repair" the pairwise preferences estimated by the base learners. Thus, one might suppose that this repairing works well in the case where only a few of such preferences are erroneous, while it fails or at least becomes ineffective in the case where too many of them are distorted.

6 Concluding Remarks

We have proposed a method for learning valued preference structures and related rankings, using a quite natural extension of pairwise classification. Some procedures for inducing a ranking from such preference relations have then been investigated empirically. Our results suggest the following main conclusion: The commonly used voting method that simply counts the (weighted) base learners' votes in favor of each alternative gives a good account in comparison with more sophisticated (and computationally more complex) ranking methods. Still, such methods may improve the predictive performance in the case of sufficiently strong base learners. Likewise, the weighted voting procedure can be improved by means of binary voting, i.e., by using binary instead of soft base learners. Again, however, this requires these learners to be sufficiently strong. Intuitively, both findings might be explained by arguing that a correction or a reinforcement of the base learners' votes can be successful only if these votes are reliable enough. Stated differently, weighted voting seems to be a rather *robust* alternative, which is completely in agreement with the statistical properties of the simple arithmetic mean (of several random variables).

References

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

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

[3] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In *Proc.* ECML–2003, Croatia, September 2003. Springer-Verlag.

[4] RM Karp. Reducibility among combinatorial problems. In RE Miller and JW Thatcher, editors, *Complexity Of Computer Computations*, pages 85–103. New York: Plenum Press, 1972.

[5] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrika*, 48:303–312, 1961.

[6] TF Wu, CJ Lin, and RC Weng. Probability estimates for multi-class classification by pairwise coupling. In *Proceedings NIPS–2003*, Vancouver and Whistler, British Columbia, Canada, 2003.