On Loss Functions in Label Ranking and Risk Minimization by Pairwise Learning*

Eyke Hüllermeier
Philipps-Universität Marburg
eyke@mathematik.uni-marburg.de

Johannes Fürnkranz
TU Darmstadt
juffi@ke.informatik.tu-darmstadt.de

Abstract

We study the problem of *label ranking*, a machine learning task that consists of inducing a mapping from instances to rankings over a finite number of labels. Our learning method, referred to as *ranking by pairwise comparison* (RPC), first induces pairwise order relations (preferences) from suitable training data, using a natural extension of so-called pairwise classification. A ranking is then derived from a set of such relations by means of a ranking procedure. In this paper, we first elaborate on a key advantage of such a decomposition, namely the fact that it allows the learner to adapt to different loss functions without re-training, by using different ranking procedures on the same predicted order relations. In this regard, we distinguish between two types of errors, called, respectively, *ranking error* and *position error*. Focusing on the position error, which has received less attention so far, we then propose a ranking procedure called *ranking through iterated choice* as well as an efficient pairwise implementation thereof. Apart from a theoretical justification of this procedure, we offer empirical evidence in favor of its superior performance as a risk minimizer for the position error.

*This research has been funded by the German Research Foundation (DFG). It summarizes and extends results presented in [13, 14, 15].

1 Introduction

An important development in contemporary machine learning research concerns the extension of learning problems from basic settings to more complex and expressive ones. In particular, learning problems with *structured output spaces* have recently received a great deal of attention [21]. In this paper, we study a problem belonging to this category, namely *label ranking*. Roughly speaking, label ranking can be seen as an extension of conventional classification learning, where the problem is to predict a *total order* of a finite number of class labels, instead of only guessing a single, presumably most likely candidate label.

One approach to the label ranking problem is offered by pairwise decomposition techniques [10]. The key idea of this approach is to learn an ensemble of simple models, where each model is trained to compare a *pair* of candidate labels. A ranking is then derived from the pairwise comparisons thus obtained. Not only is *ranking by pairwise comparison* (RPC) intuitively appealing and in line with common techniques from decision analysis, it also proved to be computationally efficient and quite effective in terms of generalization performance.

The purpose of this paper is to elaborate more closely on issues related to risk minimization in RPC, thereby contributing to the theoretical foundations of this approach. More specifically, the paper makes two main contributions: First, it broaches the issue of loss functions in label ranking and proposes a distinction between two types of error, called, respectively, *ranking error* and *position error*. In this regard, we elaborate on a key advantage of RPC, namely the fact that it can be adapted to different loss functions without the need to change the underlying models, simply by using different ranking procedures on the same underlying order relations. As a particular result, it will be shown that, by using suitable ranking procedures, RPC can minimize the risk for two important ranking errors, namely Spearman's rank correlation and Kendall's tau.

The paper then focuses on the position error, which has received less attention so far. The problem of minimizing this error will be investigated not only from a theoretical but also from a practical point of view. More specifically, we propose a ranking procedure called *ranking through iterated choice* (RIC) and offer empirical evidence in favor of its superior performance as a risk minimizer for the position error.

The remainder of the paper is organized as follows: Section 2 introduces the label ranking problem and elaborates on the aforementioned two types of prediction errors.

The idea of learning by pairwise comparison and its application to label raking are then discussed in Section 4. The problems of minimizing the ranking error and position error are studied in Sections 5 and 6, respectively. As mentioned above, the latter section introduces the RIC procedure, which is evaluated empirically in Section 7. We close the paper with some concluding remarks in Section 8.

2 Label Ranking

2.1 Problem Setting

We consider a formal setting that can be considered as an extension of the conventional setting of classification learning. Roughly speaking, the former is obtained from the latter through replacing single class labels by complete label rankings: Instead of associating every instance $\mathbf{x}$ from an instance space $\mathcal{X}$ with one among a finite set of class labels $\mathcal{L} = \{\lambda_1 \dots \lambda_m\}$, we now associate $\mathbf{x}$ with a total order of the class labels, i.e., a complete, transitive, and asymmetric relation $\succ_{\mathbf{x}}$ on $\mathcal{L}$; the meaning of $\lambda_i \succ_{\mathbf{x}} \lambda_j$ is that λ_i precedes λ_j in the ranking associated with $\mathbf{x}$; considering a ranking (metaphorically) as a special type of preference relation, we shall also say that λ_i is *preferred* to λ_j (in the context $\mathbf{x}$). To illustrate, suppose that instances are students (characterized by attributes such as sex, age, and major subjects in secondary school) and $\succ$ is a preference relation on a fixed set of study courses (such as math, CS, medicine, physics).

Formally, a ranking $\succ_{\mathbf{x}}$ can be identified with a permutation $\tau_{\mathbf{x}}$ of $\{1 \dots m\}$, e.g., the permutation $\tau_{\mathbf{x}}$ such that $\tau_{\mathbf{x}}(i) = \tau_{\mathbf{x}}(\lambda_i)$ is the position of label λ_i. This permutation encodes the ranking

$$\lambda_{\tau_{\mathbf{x}}^{-1}(1)} \succ_{\mathbf{x}} \lambda_{\tau_{\mathbf{x}}^{-1}(2)} \succ_{\mathbf{x}} \dots \succ_{\mathbf{x}} \lambda_{\tau_{\mathbf{x}}^{-1}(m)}. \tag{1}$$

We denote the class of permutations of $\{1 \dots m\}$ by $\mathcal{S}_m$. By abuse of terminology, though justified in light of the above one-to-one correspondence, we shall refer to elements $\tau \in \mathcal{S}_m$ as both permutations and rankings.

More specifically, and again in analogy with the classification setting, every instance is associated with a *probability distribution* over $\mathcal{S}_m$. That is, for every instance $\mathbf{x}$, there exists a probability distribution $\mathbb{P}(\cdot \,|\, \mathbf{x})$ such that, for every $\tau \in \mathcal{S}_m$,

$$\mathbb{P}(\tau \,|\, \mathbf{x}) \tag{2}$$

is the probability to observe the ranking τ as an output, given the instance $\mathbf{x}$ as an input.

The goal in label ranking is to learn a "label ranker" in the form of an $\mathcal{X} \longrightarrow \mathcal{S}_m$ mapping. As training data, a label ranker has access to a set $\mathcal{D}$ of example instances $\mathbf{x}_k$, $k = 1 \ldots n$, together with associated (pairwise) preferences $\lambda_i \succ_{\mathbf{x}_k} \lambda_j$. As a special case, this includes information about the complete ranking $\tau_{\mathbf{x}_k}$ associated with an instance $\mathbf{x}_k$. We explicitly mention, however, that this is not required. In practice, information about the order of labels will most often be incomplete and may, in the extreme, reduce to a single pairwise preference.

2.2 Label Ranking and Classification

As mentioned above, the setting of label ranking generalizes the one of conventional classification, and the former actually includes the latter as a special case. In classification, the training data consists of tuples $(\mathbf{x}_k, \lambda_{\mathbf{x}_k})$, which are assumed to be produced according a probability distribution on $\mathcal{X} \times \mathcal{L}$. That is, a vector

$$p_{\mathbf{x}} \;=\; (\, \mathbb{P}(\lambda_1 \,|\, \mathbf{x}) \ldots \mathbb{P}(\lambda_m \,|\, \mathbf{x}) \,) \tag{3}$$

of conditional class probabilities can be associated with every instance $\mathbf{x} \in \mathcal{X}$, where $\mathbb{P}(\lambda_i \,|\, \mathbf{x})$ denotes the probability of observing $\mathbf{x}$ together with the label λ_i. Now, by sorting the labels λ_i in decreasing order according to their probability, $\mathbf{x}$ can again be associated with a ranking $\tau_{\mathbf{x}}$. In this ranking, λ_i precedes λ_j, $\lambda_i \succ_{\mathbf{x}} \lambda_j$, if $\mathbb{P}(\lambda_i \,|\, \mathbf{x}) > \mathbb{P}(\lambda_j \,|\, \mathbf{x})$. Note that, in contrast to the more general setting of label ranking, every $\mathbf{x}$ is associated with a single, unique ranking. In other words, the probability measure in (2) is given by $\mathbb{P}(\tau_{\mathbf{x}} \,|\, \mathbf{x}) = 1$ and $\mathbb{P}(\tau \,|\, \mathbf{x}) = 0$ for all $\tau \neq \tau_{\mathbf{x}}$.

Another connection between classification and label ranking is established by going from the latter to the former, namely by projecting rankings to their top-label. This is motivated by scenarios in which, even though there is a ranking $\tau_{\mathbf{x}}$ for every instance $\mathbf{x}$, only the top-label can be observed. Thus, like in classification, observations are of the form $(\mathbf{x}, \lambda_{\mathbf{x}})$, where $\lambda_{\mathbf{x}} = \tau_{\mathbf{x}}^{-1}(1)$. For example, one may assume that a student's preferences in principle give rise to a ranking of subjects, but only the maximally preferred subject, the one which is eventually chosen, can be observed. Again, it is possible to associate an instance $\mathbf{x}$ with a probability vector (3), namely the image of the measure in (2) under the mapping $\tau \mapsto \tau^{-1}(1)$. In this case, $\mathbb{P}(\lambda_i \,|\, \mathbf{x})$ corresponds to the probability that λ_i occurs as a top-label in a ranking $\tau_{\mathbf{x}}$.

Even though the following observation is quite obvious, it is important to realize and also essential to our later discussion of two types of loss functions on rankings: A low probability $\mathbb{P}(\lambda_i \,|\, \mathbf{x}) = \mathbb{P}(\lambda_i = \tau_{\mathbf{x}}^{-1}(1))$ does not imply that λ_i is likely to have a low position in the ranking $\tau_{\mathbf{x}}$, it only means that it is unlikely to have the first position. Conversely, ordering the class labels according to their probability of being the top-label does not usually yield a good prediction of the ranking $\tau_{\mathbf{x}}$.

To illustrate, suppose that $\mathbb{P}(\lambda_1 \succ \lambda_2 \succ \lambda_3) = 0.5$, $\mathbb{P}(\lambda_3 \succ \lambda_2 \succ \lambda_1) = 0.3$, $\mathbb{P}(\lambda_2 \succ \lambda_1 \succ \lambda_3) = 0.2$, while the probability of all other rankings $\mathbb{P}(\lambda_1 \succ \lambda_3 \succ \lambda_2) = \mathbb{P}(\lambda_2 \succ \lambda_3 \succ \lambda_1) = \mathbb{P}(\lambda_3 \succ \lambda_1 \succ \lambda_2) = 0$. From this, it follows that the probabilities for the three labels λ_1, λ_2, and λ_3 being the top label are, respectively, 0.5, 0.2, 0.3. Sorting the labels according to these probabilities gives the ranking $\lambda_1 \succ \lambda_3 \succ \lambda_2$, which by itself has a probability of 0 and, as will be seen later on, is also a suboptimal prediction in terms of risk minimization for common loss functions on rankings. This result is not astonishing in light of the fact that, by only looking at the top-labels, one completely ignores the information about the rest of the rankings.

3 Semantics of a Label Ranking

So far, we introduced the problem of predicting a label ranking in a formal way, though without speaking about the semantics of a predicted ranking. In fact, one should realize that a ranking can serve different purposes. Needless to say, this point is of major importance for the evaluation of a label ranker and its predictions.

In this paper, we are especially interested in two types of practically motivated performance tasks. In the first setting, which is probably the most obvious one, the complete ranking is relevant, i.e., the positions assigned to all of the labels. As an example, consider the problem of learning to predict the best order in which to supply a certain set of stores (route of a truck), depending on external conditions like traffic, weather, purchase order quantities, etc. In case the complete ranking is relevant, the quality of a prediction should be quantified in terms of a distance measure between the predicted and the true ranking. We shall refer to any deviation of the predicted ranking from the true one as a *ranking error* (see Section 3.1).

To motivate the second setting, consider a fault detection problem which consists of identifying the cause for the malfunctioning of a technical system. If it turned out that

a predicted cause is not correct, an alternative candidate must be tried. A ranking then suggests a simple (trial and error) search process, which successively tests the candidates, one by one, until the correct cause is found [3]. In this scenario, where labels correspond to causes, the existence of a single target label (instead of a target ranking) is assumed. Hence, an obvious measure of the quality of a predicted ranking is the number of futile trials made before that label is found. A deviation of the predicted target label's position from the top-rank will subsequently be called a *position error* and discussed in more detail in Section 3.2. Note that, while a ranking error relates to the comparison of two complete label rankings, $\tau_{\mathbf{x}}$ and its prediction $\hat{\tau}_{\mathbf{x}}$, the position error refers to the comparison of a label ranking $\hat{\tau}_{\mathbf{x}}$ and a true class $\lambda_{\mathbf{x}}$.

3.1 The Ranking Error

Distance metrics or, alternatively, similarity or correlation measures for rankings have been studied intensively in diverse research areas, notably in statistics. An important and frequently applied similarity measure for rankings is the *Spearman rank correlation* [19]. It was originally proposed as a non-parametric rank statistic to measure the strength of association between two variables [17]. Formally, it is defined as a linear transformation of the sum of squared rank distances

$$D(\tau', \tau) \stackrel{\text{df}}{=} \sum_{i=1}^{m} \big(\, \tau'(i) - \tau(i) \,\big)^2 , \tag{4}$$

namely $1 - 6D(\tau, \tau')/(m(m^2 - 1))$, which is a normalized quantity assuming values in $[-1, 1]$. A related measure, *Spearman's footrule*, is similarly defined, except that the squares of rank distances are replaced by absolute values:

$$D(\tau', \tau) \stackrel{\text{df}}{=} \sum_{i=1}^{m} \big|\tau'(i) - \tau(i)\,\big| . \tag{5}$$

Another well-known distance metric for rankings is the *Kendall tau measure* [16]. This measure essentially calculates the number of pairwise rank inversions on labels to measure the ordinal correlation of two rankings; more formally, with

$$D(\tau', \tau) \stackrel{\text{df}}{=} \#\{(i, j) \mid i < j, \tau(i) > \tau(j) \wedge \tau'(i) < \tau'(j)\} \tag{6}$$

denoting the number of *discordant* pairs of items (labels), the Kendall tau coefficient is given by $1 - 4D(\tau', \tau)/(m(m - 1))$, which is again a linear scaling of $D(\tau', \tau)$ to the interval $[-1, +1]$.

Kendall's tau counts the number of transpositions of *adjacent* pairs of labels needed to transform a ranking τ into a ranking τ'. Thus, it is a special type of edit distance. A related measure, the *Cayley distance*, is given by the minimum number of transpositions of *any* pair of labels (not necessarily adjacent). Yet another edit distance is the *Ulam measure* [22], which allows as a basic edit operation the transposition of a single label, that is, removing a label from a ranking and inserting it at another position.

Finally, the *Hamming distance* simply counts the number of labels that are put at different positions in the two rankings:

$$D(\tau', \tau) \stackrel{\text{df}}{=} \#\{i \mid \tau'(i) \neq \tau(i)\} \tag{7}$$

3.2 The Position Error

We define the *position error* of a prediction $\tau_{\mathbf{x}}$ as

$$\text{PE}(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) \stackrel{\text{df}}{=} \tau_{\mathbf{x}}(\lambda_{\mathbf{x}}),$$

i.e., by the position of the target label $\lambda_{\mathbf{x}}$ in the ranking $\tau_{\mathbf{x}}$. To compare the quality of rankings of different problems, it is useful to normalize the position error for the number of labels. This *normalized position error* is defined as

$$\text{NPE}(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) \stackrel{\text{df}}{=} \frac{\tau_{\mathbf{x}}(\lambda_{\mathbf{x}}) - 1}{m - 1} \in \{0, 1/(m-1) \ldots 1\}. \tag{8}$$

The position error of a label ranker is the *expected* position error of its predictions, where the expectation is taken with respect to the underlying probability measure on $\mathcal{X} \times \mathcal{L}$.

Compared with the conventional misclassification rate, i.e., the 0/1-loss yielding 0 for a correct and 1 for an incorrect prediction, the position error differentiates between "bad" predictions in a more subtle way: In the case of a correct classification, both measures coincide. In the case of a wrong prediction (top-label), however, the misclassification rate is 1, while the position error assumes values between 1 and m, depending on how "far away" the true target label actually is.

As most performance measures, the position error is a simple scalar index. To characterize a label ranking algorithm in a more elaborate way, an interesting alternative is to look at the mapping

$$C : \{1 \ldots m\} \longrightarrow \mathbb{R},\ k \mapsto \mathbb{P}\left(\tau_{\mathbf{x}}(\lambda_{\mathbf{x}}) \leq k\right). \tag{9}$$

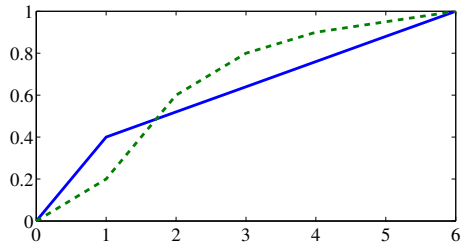

Figure 1: Exemplary C-distributions for two label ranker.

According to its definition, $C(k)$ is the probability that the target label is among the top k labels in the predicted ranking. In particular, $C(1)$ corresponds to the conventional classification rate. Moreover, $C(\cdot)$ is monotone increasing, and $C(m) = 1$. Formally, the mapping (9) is nothing else than the cumulative probability function of a random variable, namely the position of the target label, and the position error is the corresponding expected value. Of course, on the basis of the C-distribution (9), only a partial order can be defined on a class of learning algorithms: Two learners are incomparable in the case of intersecting C-distributions. Fig. 1 shows an example of that kind. The first learner (solid curve) is a good classifier, as it has a high classification rate. Compared with the second learner (dashed curve), however, it is not a good ranker, as its C-distribution has a rather flat slope.

3.3 Extensions of the Position Error

Depending on the concrete application scenario, various extensions and generalizations of the position error are conceivable. For example, imagine that, for whatever reason, it is essential to have the target label among the top k candidates in a predicted ranking. This goal might be captured most easily by means of a simple 0/1-loss function which yields 0 if $\mathrm{PE}(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) \leq k$ and 1 otherwise. More generally, one can use a (non-decreasing) weight function $w : \{1 \ldots m\} \longrightarrow \mathbb{R}$ and define the following weighted (transformed) version of the position error:

$$\mathrm{PE}(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) \stackrel{\mathrm{df}}{=} w\left(\tau_{\mathbf{x}}(\lambda_{\mathbf{x}})\right).$$

Obviously, the standard position error is obtained for $w(i) \equiv i$, while the above 0/1-loss is recovered for the special case where $w(\cdot)$ is given by $w(i) = 0$ for $i \leq k$ and $w(i) = 1$

for $i > k$. Moreover, the normalized position error (8) can be modeled by the weighting scheme $w(i) = (i-1)/m$.

Another interesting extension is related to the idea of *cost-sensitive* classification. In this respect, one usually associates a cost-value with each pair of class labels (λ_i, λ_j), reflecting the cost incurred when erroneously predicting λ_i instead of λ_j. In the context of our scenario, it makes sense to associate a cost value $c(i)$ with each individual label λ_i, reflecting the cost for *verifying* whether or not λ_i is the correct label. Then, the cost induced by a predicted label ranking $\tau_{\mathbf{x}}$ is given by the cost for testing the target label plus the labels preceding the target in the ranking:

$$\sum_{i=1}^{\tau_{\mathbf{x}}(\lambda_{\mathbf{x}})} c(\tau_{\mathbf{x}}^{-1}(i)).$$

Finally, we note that the idea of the position error can of course be generalized to multi-label (classification) problems which assume several instead of a single target label for each instance. In this connection, it makes a great difference whether one is interested in having at least one of the targets on a top rank (e.g., since finding one solution is enough), or whether all of them should have high positions (resp. none of them should be ranked low). In the latter case, it makes sense to count the number of non-target labels ranked above the lowest target label (an application of that type has recently been studied in [6]), while in the former case, one will look at the number of non-target labels placed before the highest-ranked target label.

4 Learning by Pairwise Comparison

The key idea of pairwise learning is well-known in the context of classification [8], where it allows one to transform a multi-class classification problem, i.e., a problem involving $m > 2$ classes $\mathcal{L} = \{\lambda_1 \dots \lambda_m\}$, into a number of *binary* problems. To this end, a separate model (base learner) $\mathcal{M}_{ij}$ is trained for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L}$, $1 \leq i < j \leq m$; thus, a total number of $m(m-1)/2$ models is needed. $\mathcal{M}_{ij}$ is intended to separate the objects with label λ_i from those having label λ_j. At classification time, a query instance is submitted to all models $\mathcal{M}_{ij}$, and their predictions are combined into an overall prediction. In the simplest case, each prediction of a model $\mathcal{M}_{ij}$ is interpreted as a vote for either λ_i or λ_j, and the label with the highest number of votes is proposed as a final prediction. It should be noted that although the number of

binary classifiers grows quadratically with the number labels m, the training time for this ensemble will only grow linearly with m [8], and, in practice, one is able to derive a prediction by querying only a linear number of classifiers [18].

The above procedure can be extended to the case of label ranking or, more generally, preference learning in a natural way [10]. A preference (order) information of the form $\lambda_a \succ_{\mathbf{x}} \lambda_b$ is turned into a training example $(\mathbf{x}, y)$ for the learner $\mathcal{M}_{ij}$, where $i = \min(a, b)$ and $j = \max(a, b)$. Moreover, $y = 1$ if $a < b$ and $y = 0$ otherwise. Thus, $\mathcal{M}_{ij}$ is intended to learn the mapping that outputs 1 if $\lambda_i \succ_{\mathbf{x}} \lambda_j$ and 0 if $\lambda_j \succ_{\mathbf{x}} \lambda_i$:

$$x \mapsto \begin{cases} 1 & \text{if} \quad \lambda_i \succ_{\mathbf{x}} \lambda_j \\ 0 & \text{if} \quad \lambda_j \succ_{\mathbf{x}} \lambda_i \end{cases} . \tag{10}$$

The model is trained with all examples $\mathbf{x}_k$ for which either $\lambda_i \succ_{\mathbf{x}_k} \lambda_j$ or $\lambda_j \succ_{\mathbf{x}_k} \lambda_i$ is known. Examples for which nothing is known about the preference between λ_i and λ_j are ignored.

The mapping (10) can be realized by any binary classifier. Alternatively, one can of course employ a classifier that maps into the unit interval $[0, 1]$ 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: the closer the output of $\mathcal{M}_{ij}$ to 1, the stronger the preference $\lambda_i \succ_{\mathbf{x}} \lambda_j$ is supported.

A preference learner composed of an ensemble of soft binary classifiers assigns a *valued preference relation* $\mathcal{R}_{\mathbf{x}}$ to every (query) instance $\mathbf{x} \in \mathcal{X}$:

$$\mathcal{R}_{\mathbf{x}}(\lambda_i, \lambda_j) = \begin{cases} \mathcal{M}_{ij}(\mathbf{x}) & \text{if} \quad i < j \\ 1 - \mathcal{M}_{ij}(\mathbf{x}) & \text{if} \quad i > j \end{cases} \tag{11}$$

for all $\lambda_i \neq \lambda_j \in \mathcal{L}$. Given such a preference relation $\mathcal{R}_{\mathbf{x}}$ for an instance $\mathbf{x}$, the next question is how to derive an associated ranking $\tau_{\mathbf{x}}$. This question is non-trivial, since a relation $\mathcal{R}_{\mathbf{x}}$, derived from potentially erroneous predictions, can contain inconsistencies and 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 [7]. In the context of pairwise classification and preference learning, several studies have empirically compared different ways of combining the predictions of individual classifiers [23, 1, 12, 9].

The perhaps simplest approach is a straightforward extension of the aforementioned

voting strategy: The alternatives λ_i are evaluated by the sum of (weighted) votes

$$S(\lambda_i) = \sum_{\lambda_j \neq \lambda_i} \mathcal{R}_{\mathbf{x}}(\lambda_i, \lambda_j) \tag{12}$$

and then ordered according to these evaluations, i.e.,

$$(\lambda_i \succeq \lambda_j) \Leftrightarrow (S(\lambda_i) \geq S(\lambda_j)). \tag{13}$$

This is a particular type of "ranking by scoring" strategy, where the scoring function is given by (12).[1] Even though this ranking procedure may appear rather ad-hoc at first sight, we shall give a theoretical justification in Section 5.

In summary, our approach, referred to as *ranking by pairwise comparison* (RPC), consists of the following two steps:

- the derivation of a valued preference relation (11) by training an ensemble of (soft) binary classifiers, and
- the subsequent ranking of labels, using a ranking procedure such as (12–13).

We like to emphasize the *modularity* of RPC thus defined as a particular advantage of the approach. This modularity allows, for example, to adapt the ranking procedure in the second step to the problem at hand. In fact, as will be seen in the following sections, this allows one to adjust RPC to minimize different loss functions on label rankings without the need for re-training the pairwise classifiers.

5 Minimizing the Ranking Error

The quality of a model $\mathcal{M}$ (induced by a learning algorithm) is commonly measured in terms of its *expected loss* or *risk*

$$\mathbb{E}\left(D(y, \mathcal{M}(\mathbf{x})) \right), \tag{14}$$

where $D(\cdot)$ is a loss or distance function, $\mathcal{M}(\mathbf{x})$ denotes the prediction made by the model for the instance $\mathbf{x}$, and y is the true outcome. The expectation $\mathbb{E}$ is taken

[1] Strictly speaking, (12) does not necessarily define a ranking, as it does not exclude the case of indifference between labels. In such cases, a ranking can be enforced by any tie braking strategy.

with respect to a probability measure over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ is the output space. In this section, we are interested in the case where $\mathcal{Y}$ is given by a set of rankings, $\mathcal{S}_m$, of a label set $\mathcal{L}$, and $D(\cdot)$ is a ranking error. More specifically, we show that, by using suitable ranking procedures, RPC as outlined in Section 4 can minimize the risk with respect to two important ranking errors, namely Spearman's rank correlation and Kendall's tau.

5.1 Spearman's Rank Correlation and Kendall's Tau

As a first result, we show that, if the binary models $\mathcal{M}_{ij}$ provide correct probability estimates, i.e.,

$$\mathcal{R}_{\mathbf{x}}(\lambda_i, \lambda_j) = \mathcal{M}_{ij}(\mathbf{x}) = \mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j), \tag{15}$$

then RPC using (12–13) as a ranking procedure yields a risk minimizing prediction

$$\hat{\tau}_{\mathbf{x}} = \arg\min_{\tau \in \mathcal{S}_m} \sum_{\tau' \in \mathcal{S}_m} D(\tau, \tau') \cdot \mathbb{P}(\tau' \,|\, \mathbf{x}) \tag{16}$$

with respect to (4) as a loss function, i.e., it maximizes the expected Spearman rank correlation between a true ranking $\tau_{\mathbf{x}}$ and the prediction thereof.

Theorem 1 *Suppose that (15) holds and let $D(\cdot)$ be given by (4). The expected distance*

$$\mathbb{E}(D(\hat{\tau}, \tau) \mid \mathbf{x}) = \sum_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau \mid \mathbf{x}) \cdot D(\hat{\tau}, \tau) = \sum_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau \mid \mathbf{x}) \sum_{i=1}^{m} (\hat{\tau}(i) - \tau(i))^2$$

becomes minimal by choosing $\hat{\tau}$ such that $\hat{\tau}(i) \leq \hat{\tau}(j)$ whenever $S(\lambda_i) \geq S(\lambda_j)$, with $S(\lambda_i)$ given by (12).

A proof of this theorem is given in Appendix A.

Admittedly, (15) is a relatively strong assumption, as it requires the pairwise preference probabilities to be perfectly learnable. Yet, the important point is that the above result sheds light on the aggregation properties of our technique, albeit under ideal conditions. In fact, recalling that RPC consists of two steps, namely *pairwise learning* and *ranking*, it is clear that in order to study properties of the latter, some assumptions about the result of the former step have to be made. And even though (15) might at best hold approximately in practice, it seems to be at least as natural as any other assumption about the output of the ensemble of pairwise learners.

Next, we consider another important ranking error, namely Kendall's tau.

Theorem 2 *Given that (15) holds, the risk with respect and to the Kendall tau measure (6) as a loss function can be minimized by RPC.*

Proof: For every ranking $\hat{\tau}$,

$$\begin{aligned}
\mathbb{E}(D(\hat{\tau},\tau) \mid \mathbf{x}) &= \sum_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau) \cdot D(\hat{\tau},\tau) \qquad (17)\\
&= \sum_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau \mid \mathbf{x}) \cdot \sum_{i<j \,|\, \hat{\tau}(i)<\hat{\tau}(j)} \left\{ \begin{array}{ll} 1 & \text{if} \quad \tau(i) > \tau(j) \\ 0 & \text{if} \quad \tau(i) < \tau(j) \end{array} \right. \\
&= \sum_{i<j \,|\, \hat{\tau}(i)<\hat{\tau}(j)} \sum_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau \mid \mathbf{x}) \cdot \left\{ \begin{array}{ll} 1 & \text{if} \quad \tau(i) > \tau(j) \\ 0 & \text{if} \quad \tau(i) < \tau(j) \end{array} \right. \\
&= \sum_{i<j \,|\, \hat{\tau}(i)<\hat{\tau}(j)} \mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j)
\end{aligned}$$

Thus, knowing the pairwise probabilities $\mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j)$, the expected loss can be derived for every ranking $\hat{\tau}$ and, hence, a risk minimizing ranking can be found. □

Finding the ranking that minimizes (17) is formally equivalent to solving the graph-theoretical *feedback arc set* problem (for *weighted* tournaments) which is known to be NP complete [2]. Of course, in the context of label ranking, this result should be put into perspective, because the set of class labels is typically of small to moderate size. Nevertheless, from a computational point of view, the ranking procedure that minimizes Kendall's tau is definitely more complex than the procedure for minimizing Spearman's rank correlation.

5.2 Limitations of RPC

Despite the results of the previous section, it is important to realize that RPC is indeed not able to minimize the risk with respect to *every* loss function. The simple 0/1–loss, i.e., $D(\hat{\tau},\tau) = 0$ if $\hat{\tau} = \tau$ and $= 1$ otherwise, is a concrete counter-example. The prediction minimizing the expected 0/1–loss is obviously given by the (Bayes) decision $\hat{\tau} = \arg\max_{\tau \in \mathcal{S}_m} \mathbb{P}(\tau \,|\, \mathbf{x})$. Now, consider the following distributions $\mathbb{P}(\cdot \,|\, \mathbf{x}) \neq \mathbb{P}'(\cdot \,|\, \mathbf{x})$ for an instance $\mathbf{x}$:

$$\mathbb{P}(\tau \,|\, \mathbf{x}) = \left\{ \begin{array}{ll} 1/2 & \text{if } \tau = (1\,2\,3\,4) \\ 1/2 & \text{if } \tau = (4\,3\,2\,1) \\ 0 & \text{otherwise} \end{array} \right. , \; \mathbb{P}'(\tau \,|\, \mathbf{x}) = \left\{ \begin{array}{ll} 1/2 & \text{if } \tau = (2\,1\,4\,3) \\ 1/2 & \text{if } \tau = (3\,4\,1\,2) \\ 0 & \text{otherwise} \end{array} \right. \qquad (18)$$

Then, if (15) holds, one derives the same probabilities $\mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j)$ and, therefore, the same preference relations from both distributions $\mathbb{P}$ and $\mathbb{P}'$:

$$\mathcal{R}_{\mathbf{x}} = \begin{bmatrix} - & 1/2 & 1/2 & 1/2 \\ 1/2 & - & 1/2 & 1/2 \\ 1/2 & 1/2 & - & 1/2 \\ 1/2 & 1/2 & 1/2 & - \end{bmatrix}$$

Consequently, RPC cannot distinguish between the original distributions on $\mathcal{S}_m$ and, hence, cannot minimize the 0/1–loss.

The same example can also be used to show that RPC can not minimize risk with respect to Spearman's footrule (5): For the distribution $\mathbb{P}$ in (18), this measure yields an expected distance of 4 for the rankings (1 2 3 4) and (4 3 2 1), which is the minimal risk, while (2 1 4 3) and (3 4 1 2) are both suboptimal with a risk of 6; for $\mathbb{P}'$, the situation is just the reverse. Again, since RPC cannot distinguish these two cases, it cannot deliver a risk minimizing prediction. In the same way, one verifies that RPC cannot minimize risk with respect to the Hamming distance (minimal risk values 2 versus suboptimal 4), the Cayley distance (1 versus 2), and the Ulam distance (1.5 versus 2).

These negative results are a direct consequence of an information loss which is inherent to learning by pairwise comparison: In this approach, the original probability distribution on the set of rankings, $\mathcal{S}_m$, is replaced by pairwise probabilities $\mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j)$. From these pairwise probabilities alone, however, it is not possible to recover the original distribution on $\mathcal{S}_m$.

Nevertheless, we like to emphasize that this information loss should not be taken as a serious deficiency of RPC. In fact, one should realize that learning the original distribution on $\mathcal{S}_m$ is practically infeasible, mainly due to the large number of rankings and conditional probabilities that need to be estimated. Besides, complete rankings will often not be available for training anyway. Therefore, the question rather becomes how to utilize the given information in an optimal way, and how to learn "suitable" condensed models. In this regard, we have seen above that the pairwise approach is especially tailored to the sum of squared rank distances as a loss function, i.e., to the Spearman rank correlation, and that RPC is also able to minimize risk for another important and frequently used loss function, namely for Kendall's tau.

Finally, we note that, in practice, pairwise preferences will often capture much more information about a distribution on $\mathcal{S}_m$ than might be suggested by the above exam-

ples. In fact, these are rather extreme in the sense that, in both cases, a ranking and its complete reversal are considered as equally probable. Roughly speaking, both $\mathbb{P}$ and $\mathbb{P}'$ are bimodal distributions with two maximally distant modes. In practice, one may expect distributions on $\mathcal{S}_m$ that are more "centered" in the sense of allocating probability mass to closely neighbored rankings. Regardless of the concrete distance measure $D(\cdot)$, this means that highly probable rankings are unlikely to show a strong disagreement on the level of pairwise preferences.

Indeed, as all distance (similarity) measures on rankings are more or less closely related, minimizing risk with respect to one measure usually prevents from poor solutions for another measure. In some cases, it is even possible to prove exact approximation bounds. For example, an interesting connection between the two ranking error measures discussed above has recently been established in [5], where it has been shown that optimizing rank correlation yields a 5-approximation to the ranking which is optimal for the Kendall measure.

6 Minimizing the Position Error

What kind of ranking procedure should be used in order to minimize the risk of a predicted ranking with respect to the position error as a loss function? As mentioned before, an intuitively plausible idea is to order the candidate labels λ according to their probability $\mathbb{P}(\lambda = \lambda_{\mathbf{x}})$ of being the target label. In fact, this idea is not only plausible but also provably correct. Even though the result is quite obvious, we state its formally as a theorem.

Theorem 3 *Given a query instance $\mathbf{x} \in \mathcal{X}$, ranking the labels $\lambda \in \mathcal{L}$ according to their (conditional) probabilities of being the target class $\lambda_{\mathbf{x}}$ yields a risk minimizing prediction with respect to the position error (8) as a loss function. That is, the expected loss*

$$\mathbb{E}(\tau_{\mathbf{x}}) = \frac{1}{m-1} \sum_{i=1}^{m} (i-1) \cdot \mathbb{P}\left(\tau_{\mathbf{x}}(\lambda_{\mathbf{x}}) = i\right)$$

becomes minimal for any ranking $\succ_{\mathbf{x}}$ such that $\mathbb{P}(\lambda_i = \lambda_{\mathbf{x}}) > \mathbb{P}(\lambda_j = \lambda_{\mathbf{x}})$ implies $\lambda_i \succ_{\mathbf{x}} \lambda_j$.

Proof: This result follows almost by definition. In fact, note that we have

$$\mathbb{E}(\tau_{\mathbf{x}}) \propto \sum_{i=1}^{m} \mathbb{P}\left(\tau_{\mathbf{x}}(\lambda_{\mathbf{x}}) > i\right)$$

and that, for each position i, the probability to excess this position when searching for the target $\lambda_{\mathbf{x}}$ is obviously minimized when ordering the labels according to their (conditional) probabilities. □

6.1 Conventional Conditioning

According to the above result, the top rank (first position) should be given to the label $\lambda_\top$ for which the estimated probability is maximal. Regarding the second rank, recall the fault detection metaphor, where the second hypothesis for the cause of the fault is only tested in case the first one turned out to be wrong. Thus, when having to make the next choice, one principally has additional information at hand, namely that $\lambda_\top$ *was not the correct label*. Taking this information into account, the second rank should not simply be given to the label with the second highest probability according to the original probability measure, say, $\mathbb{P}_1(\cdot) = \mathbb{P}(\cdot)$, but instead to the label that maximizes the *conditional* probability $\mathbb{P}_2(\cdot) = \mathbb{P}(\cdot \mid \lambda_{\mathbf{x}} \neq \lambda_\top)$ of being the target label given that the first proposal was incorrect.

At first sight, passing from $\mathbb{P}_1(\cdot)$ to $\mathbb{P}_2(\cdot)$ may appear meaningless from a ranking point of view, since standard probabilistic conditioning yields

$$\mathbb{P}_2(\lambda) = \frac{\mathbb{P}_1(\lambda)}{1 - \mathbb{P}_1(\lambda_\top)} \propto \mathbb{P}_1(\lambda) \tag{19}$$

for $\lambda \neq \lambda_\top$, and therefore does not change the order of the remaining labels. And indeed, in case the original $\mathbb{P}(\cdot)$ is a proper probability measure and conditioning is performed according to (19), the predicted ranking will not be changed at all.

6.2 Empirical Conditioning

One should realize, however, that standard conditioning is not an incontestable updating procedure in our context, simply because $\mathbb{P}_1(\cdot)$ is not a "true" probability measure over the class labels. Rather, it is only an estimated measure coming from a learning algorithm, perhaps one which is not a good probability estimator. In fact, it is

well-known that most machine learning algorithms for classification, especially in the multi-class case, perform rather poorly in probability estimation, even though they may produce good classifiers. Thus, it seems sensible to perform "conditioning" not on the measure itself, but rather on the learner that produced the measure. What we mean by this is retraining the learner on the original data without the $\lambda_\top$-examples, an idea that could be paraphrased as "empirical conditioning".

This type of conditioning depends on the data $\mathcal{D}$ and the model assumptions, that is, the hypothesis space $\mathcal{H}$ from which the classifier is taken. To emphasize this dependence and, moreover, to indicate that it concerns an *estimated* ("hat") probability, the conditional measure $\mathbb{P}_2(\cdot)$ could be written more explicitly as

$$\mathbb{P}_2(\cdot) = \widehat{\mathbb{P}}(\cdot \,|\, \lambda_{\mathbf{x}} \neq \lambda_\top, \mathcal{D}, \mathcal{H}).$$

To motivate the idea of empirical conditioning, suppose that the estimated probabilities come from a classification tree. Of course, the original tree trained with the complete data will be highly influenced by $\lambda_\top$-examples, and the probabilities assigned by that tree to the alternatives $\lambda \neq \lambda_\top$ might be inaccurate. Retraining a classification tree on a reduced set of data might then lead to more accurate probabilities for the remaining labels, especially since the multi-class problem to be solved has now become simpler (as it involves fewer classes).

Fig. 2 shows a simpler example, where the hypothesis space $\mathcal{H}$ is given by the class of decision stumps (univariate decision trees with only one inner node, i.e., axis-parallel splits in the case of numerical attributes). Given the examples from three classes (represented, respectively, by the symbols $\diamond$, $\star$, and $\bullet$), the best model corresponds to the split shown in the left picture. By estimating probabilities through relative frequencies in the leaf nodes of the decision stump, one derives the following estimates for the query instance, which is marked by a $\oplus$ symbol: $\widehat{\mathbb{P}}(\diamond \,|\, \oplus) = 12/15$, $\widehat{\mathbb{P}}(\star \,|\, \oplus) = 2/15$, $\widehat{\mathbb{P}}(\bullet \,|\, \oplus) = 1/15$; thus, the induced ranking is given by $\diamond \succ \star \succ \bullet$. Now, suppose that the top label $\diamond$ turned out to be an incorrect prediction. According to the above ranking (and probabilistic conditioning), the next label to be tested would be $\star$. However, when fitting a new model to the training data without the $\diamond$-examples, the preference between $\star$ and $\bullet$ is reversed, because the query instance is now located "on the $\bullet$-side" of the decision boundary, and $\widehat{\mathbb{P}}(\bullet \,|\, \oplus, \lambda_\oplus \neq \diamond) = 1$ (as shown on the right-hand side of Fig. 2). Roughly speaking, conditioning by "taking a different view" on the data (namely a view that suppresses the $\diamond$ examples) gives a quite different picture of the

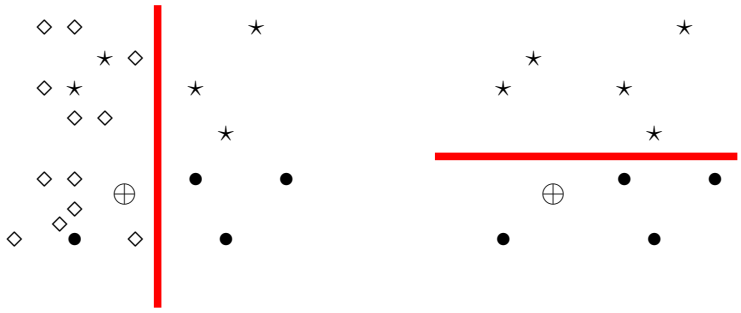

Figure 2: Example of empirical conditioning: The optimal model (decision stump) for the complete training data (left) and the data omitting the examples of the top label ($\diamond$).

situation. In fact, one should realize that, in the first model, the preference between $\star$ and $\bullet$ is strongly biased by the $\diamond$-examples: The first decision boundary is optimal only because it classifies all $\diamond$-examples correctly, a property that looses importance once it turns out that $\diamond$ is not the true label of the query.

6.3 Ranking through Iterated Choice

According to the above idea, a classifier is used as a *choice function*: Given a set $\mathcal{C} \subseteq \mathcal{L}$ of potential labels with corresponding training data (and a new query instance $\mathbf{x}$), it selects the most likely candidate among these labels. We refer to the process of successively selecting alternatives by estimating top labels from (conditional) probability measures $\mathbb{P}_1(\cdot), \mathbb{P}_2(\cdot) \ldots \mathbb{P}_m(\cdot)$ as *ranking through iterated choice* (RIC).

As an important advantage, note that this approach can be used to turn any multi-class classifier into a label ranker, as shown in Algorithm 1. In principle, it is not required that a corresponding classifier outputs a score, or even a real probability, for every label. In fact, since only a simple decision in favor of a single label has to be made in each iteration, any classifier is good enough. In this regard, let us note that, for the ease of exposition, the term "probability" will subsequently be used in a rather informal manner.

Regarding its effect on label ranking accuracy, one may expect the idea of RIC to produce two opposite effects:

Algorithm 1: Ranking through Iterated Choice (RIC)

$i \leftarrow 0$
$\mathcal{C} \leftarrow \{\lambda_1, \lambda_2 \dots \lambda_m\}$
repeat
  $i \leftarrow i + 1$
  train a classifier $\mathcal{M}^{(i)}$ on all training examples with labels in $\mathcal{C}$
  let $\lambda^{(i)} = \mathcal{M}^{(i)}(\mathbf{x})$ be the prediction of the classifier for instance $\mathbf{x}$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus \lambda^{(i)}$
until $i = m$;
return the ranking $\lambda^{(1)} \succ \lambda^{(2)} \succ \cdots \succ \lambda^{(m)}$ for instance $\mathbf{x}$

- *Information loss:* In each iteration, the size of the data set to learn from becomes smaller.
- *Simplification:* Due to the reduced number of classes, the learning problems become simpler in each iteration.

The first effect will clearly have a negative influence on generalization performance, as a reduction of data comes along with a loss of information. In contrast to this, the second effect will have a positive influence: The classifiers will become increasingly simple, because it can be expected that the decision boundary for separating m classes is more complex than the decision boundary for separating $m' < m$ classes of the same problem. The hope is that, in practice, the second (positive) effect will dominate the first one.

6.4 Efficient Implementation

An obvious disadvantage of RIC concerns its computational complexity. In fact, since empirical conditioning essentially means classifying on a subset of $\mathcal{L}$, the number of models needed to classify a set of examples is (potentially) of the order $2^{|\mathcal{L}|}$. To overcome this problem, we resort to the idea of pairwise learning as outlined in Section 4. More specifically, the idea is to train, in a first step, pairwise models $\mathcal{M}_{ij}$, the outputs $\mathcal{M}_{ij}(\mathbf{x})$ of which can be interpreted as (approximate) conditional probabilities

$$p_{ij} \;=\; \mathbb{P}\left(\lambda_{\mathbf{x}} = \lambda_i \,|\, \lambda_{\mathbf{x}} \in \{\lambda_i, \lambda_j\}, \mathbf{x}\right)$$

Then, in a second step, an estimation of the probability vector (3), i.e., of the individual probabilities $p_i = \mathbb{P}(\lambda_{\mathbf{x}} = \lambda_i \,|\, \mathbf{x})$, is derived from these pairwise probabilities. To this end, different techniques have been developed (e.g., [11]). Here, we resorted to the approach proposed in [23], which derives the p_i as a solution of a system of linear equations, S, that includes one equation for every label.

Let E_i denote the event that $\lambda_i = \lambda_{\mathbf{x}}$, i.e., that λ_i is the target label, and let $E_{ij} = E_i \vee E_j$ (either λ_i or λ_j is the target). Then,

$$(m-1)\,\mathbb{P}(E_i) = \sum_{j \neq i} \mathbb{P}(E_i) = \sum_{j \neq i} \mathbb{P}(E_i \,|\, E_{ij}) \cdot \mathbb{P}(E_{ij}), \tag{20}$$

where m is the number of labels. Considering the (pairwise) estimates $\mathcal{R}(\lambda_i, \lambda_j)$ as conditional probabilities $\mathbb{P}(E_i \,|\, E_{ij})$, we obtain a system of linear equations for the (unconditional) probabilities $\mathbb{P}(E_i)$:

$$\begin{aligned} \mathbb{P}(E_i) &= \frac{1}{m-1} \sum_{j \neq i} \mathcal{R}(\lambda_i, \lambda_j) \cdot \mathbb{P}(E_{ij}) \\ &= \frac{1}{m-1} \sum_{j \neq i} \mathcal{R}(\lambda_i, \lambda_j) \cdot (\mathbb{P}(E_i) + \mathbb{P}(E_j)) \end{aligned} \tag{21}$$

In conjunction with the constraint $\sum_{i=1}^{m} \mathbb{P}(E_i) = 1$, this system has a unique solution provided that $\mathcal{R}(\lambda_i, \lambda_j) > 0$ for all $1 \leq i, j \leq m$ [23].

RIC can then be realized as follows: First, the aforementioned system of linear equations is solved, and the label λ_i with maximal probability p_i is chosen as the top-label $\lambda_\top$. This label is then removed, i.e., the corresponding variable p_i and its associated equation are deleted from S. To find the second best label, the same procedure is then applied to the reduced system S' thus obtained, i.e., by solving a system of $m-1$ linear equations and $m-1$ variables. This process is iterated until a full ranking has been constructed.

Lemma 4 *In each iteration of the RIC procedure, the correct conditional probabilities are derived.*

Proof: Without loss of generality, assume that λ_m has obtained the highest rank in the first iteration. The information that this label is incorrect, $\lambda_m \neq \lambda_{\mathbf{x}}$, is equivalent to $\mathbb{P}(E_m) = 0$, $\mathbb{P}(E_m \,|\, E_{jm}) = 0$, and $\mathbb{P}(E_j \,|\, E_{jm}) = 1$ for all $j \neq m$. Incorporating

these probabilities in (21) yields, for all $i < m$,

$$\begin{aligned}(m-1)\mathbb{P}(E_i) &= \sum_{j=1\ldots m, j\neq i} \mathbb{P}(E_i \mid E_{ij}) \cdot \mathbb{P}(E_{ij}) \\ &= \sum_{j=1..m-1, j\neq i} \mathbb{P}(E_i \mid E_{ij}) \cdot \mathbb{P}(E_{ij}) + 1 \cdot \mathbb{P}(E_{im})\end{aligned}$$

and as $\mathbb{P}(E_{im}) = \mathbb{P}(E_i) + \mathbb{P}(E_m) = \mathbb{P}(E_i)$,

$$(m-2)\mathbb{P}(E_i) = \sum_{j=1..m-1, j\neq i} \mathbb{P}(E_i \mid E_{ij}) \cdot \mathbb{P}(E_{ij}).$$

Obviously, the last equation is equivalent to (21) for a system with $m-1$ labels, namely the system obtained by removing the m-th row and column of $\mathcal{R}$. □

This approach reduces the training effort from an exponential to a quadratic number of models. Roughly speaking, a classifier on a subset $\mathcal{C} \subseteq \mathcal{L}$ of classes is efficiently assembled "on the fly" from the corresponding subset of pairwise models $\{\mathcal{M}_{ij} \mid \lambda_i, \lambda_j \in \mathcal{C}\}$. Or, stated differently, the *training* of classifiers is replaced by the *combination* of associated binary classifiers.

7 Empirical Evaluation

In this section, we intend to empirically validate two conjectures:

(i) Empirical conditioning (RIC) is an empirically better way to minimize the position error than conventional probabilistic conditioning, and

(ii) the increased efficiency of the pairwise implementation, RIC-P, is achieved without sacrificing this gain in accuracy.

The hope that empirical conditioning improves accuracy in comparison with conventional probabilistic conditioning is essentially justified by the aforementioned *simplification effect* of RIC. Note that this simplification effect is also inherently present in pairwise learning. Here, the simplification due to a reduction of class labels is already achieved at the very beginning and, by decomposing the original problem into *binary* problems, carried to the extreme. Thus, if the simplification effect is indeed beneficial

in the original version of RIC, it should also have a positive influence in the pairwise implementation (RIC-P).

To validate these hypotheses, we compare the RIC strategy to the most obvious alternative, namely ordering the class labels right away according to the respective probabilities produced by a multi-class classifier (probabilistic ranking, PR). So, given any multi-class classifier, capable of producing such probabilities, as a base learner, we consider the following three learning strategies:

- **PR:** A ranking is produced by applying the base learner to the complete data set only once and ordering the class labels according to their probabilities.
- **RIC:** This version refers to the *ranking through iterated choice* procedure outlined in Section 6.2, using the multi-class classifier as a base learner.
- **RIC-P:** This is the pairwise implementation of RIC as introduced in Section 6.4 (again using as base learners the same classifiers as RIC and PR).

In connection with selecting the top-label or ordering the labels according to their probability, ties are always broken through coin flipping.

Table 1 shows the results that we obtained for a number of benchmark data sets from the UCI repository and the StatLib archive[2], using two widely known machine learning algorithms as base learners: C4.5 and Ripper. For comparison purpose, we also derived results for the naive Bayes (NB) classifier, as this is one of the most commonly used "true" probabilistic classifiers. Note that, since conditional probabilities in NB are estimated individually for each class, empirical conditioning is essentially the same as conventional conditioning, i.e., RIC is equivalent to PR [20]; this is why the results for RIC and RIC-P are omitted.

For each data set and each method we estimated the mean (absolute) position error using leave-one-out cross validation, except for the data set `letter`, for which we used the predefined separation into training and test data. The results are summarized in Table 1.

From the win-loss statistics for NB in comparison with PR using, respectively, C4.5 (10/8) and Ripper (10/8), there is no visible difference between these multi-class classifiers in terms of label ranking accuracy. Important are the win-loss statistics sum-

[2] `http://www.ics.uci.edu/~mlearn`, `http://stat.cmu.edu/`

Table 1: Position error for conventional probabilistic ranking (PR), ranking through iterated choice (RIC), and its pairwise implementation (RIC-P), using C4.5, Ripper, and naive Bayes as base learners.

		C4.5			Ripper			NB
data	m	PR	RIC	RIC-P	PR	RIC	RIC-P	PR
abalone	28	4,650	4,004	3,552	4,667	4,358	3,500	4,346
anneal	6	1,023	1,028	1,024	1,031	1,028	1,017	1,150
audiology	24	2,310	2,186	3,190	2,394	3,274	3,270	3,102
autos	7	1,273	1,293	1,502	1,449	1,376	1,449	1,771
balance-scale	3	1,397	1,326	1,294	1,406	1,325	1,256	1,170
glass	7	1,547	1,486	1,449	1,612	1,486	1,463	1,855
heart-c	5	1,231	1,231	1,224	1,218	1,218	1,218	1,165
heart-h	5	1,197	1,197	1,197	1,187	1,187	1,187	1,16
hypothyroid	4	1,005	1,007	1,008	1,012	1,011	1,007	1,054
iris	3	1,073	1,053	1,053	1,067	1,073	1,073	1,047
lymph	4	1,270	1,250	1,236	1,284	1,277	1,297	1,189
primary-tumor	22	4,254	3,764	3,531	4,478	4,316	3,472	3,248
segment	7	1,135	1,042	1,042	1,131	1,075	1,060	1,258
soybean	19	1,205	1,113	1,085	1,220	1,123	1,073	1,136
vehicle	4	1,411	1,309	1,313	1,489	1,449	1,343	1,831
vowel	11	2,314	1,274	1,309	2,501	1,516	1,423	1,555
zoo	7	1,238	1,099	1,149	1,307	1,327	1,188	1,069
letter	26	2,407	1,279	1,202	2,168	1,375	1,188	2,515

Table 2: Win/loss statistics for each pair of methods, using C4.5 (left) and Ripper (right) as base learners.

	PR	RIC	RIC-P	PR	RIC	RIC-P
PR	—	3/13	4/13	—	3/13	3/12
RI	13/3	—	7/8	13/3	—	2/13
RIC-P	13/4	8/7	—	12/3	13/2	—

marized in Table 2. These results perfectly support the two conjectures raised above. First, RIC significantly outperforms PR: According to a simple sign test for the win-loss statistic, the results are significant at a level of 2%. Second, RIP-P is fully competitive with RIC (and actually shows a better performance in the case of Ripper as a base learner).

8 Concluding Remarks

In this paper, we have investigated aspects of predictive accuracy and risk minimization in ranking by pairwise comparison (RPC), the application of pairwise learning in the context of the label ranking problem. In this regard, we have proposed a distinction between two types of prediction errors, the ranking error and the position error.

It has been shown that even though RPC is not able to minimize risk with respect to every ranking error, it can still minimize the expected loss for two important and frequently used distance measures, namely Spearman's rank correlation and Kendall's tau. Empirical studies complementing these theoretical results have not been presented here, as these can be found in the companion paper [4].

To minimize the position error, we proposed *ranking through iterated choice* (RIC), a strategy that essentially reduces label ranking to repeated classification. In each iteration, RIC performs "empirical conditioning", which means that the predictions for higher ranks utilize the information that the lower ranks have already been predicted. In practice, this is achieved by re-training the classifier with progressively smaller sets of candidate labels. To implement this procedure efficiently, we again employed the pairwise learning approach. This way, empirical conditioning is performed implicitly, by combining a proper subset of pairwise models, thereby reducing the number of

required classifiers from exponential to quadratic.

In an experimental study, RIC was compared to standard probabilistic ranking, where the class labels are ranked according to the originally estimated probabilities. Our results suggest that retraining (empirical conditioning) does indeed reduce the expected loss when using standard multi-class classifiers as base learners, and that this gain in accuracy is preserved by the pairwise implementation.

Thus, in summary, we obtained a method that improves the ranking performance of classification algorithms at an acceptable increase in complexity. In some sense, RIC may be viewed as an implicit multi-class calibration technique, which has no effect in case of perfectly estimated probabilities (cf. equation (19)), but which yields better results in the practically more relevant case of inaccurate estimates. In fact, one should note that RIC principally allows the use of arbitrary multi-class classifiers as base learners, even pure classifiers that are not able to rank but only to make a single prediction.

The results of this paper can be extended in various directions, both theoretically and practically. For example, an open question regarding the ranking error is a complete characterization of the loss functions that RPC is able to minimize. As to the position error, one important aspect of future work is to generalize our framework to the variants outlined in Section 2.5.

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] N. Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20(1):137–142, 2000.

[3] C. Alonso, J.J. Rodríguez, and B. Pulido. Enhancing consistency based diagnosis with machine learning techniques. In *Current Topics in Artificial Intelligence: 10th Conference of the Spanish Association for Artificial Intelligence*, pages 312–321. Springer, 2004.

[4] K. Brinker, J. Fürnkranz, and E. Hüllermeier. *Label ranking by learning pairwise preferences.* Technical Report TUD-KE-2007-01, TU Darmstadt, 2007.

[5] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 776–782, 2006.

[6] K. Crammer and Y. Singer. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, 3:1025–1058, 2003.

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

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

[9] J. Fürnkranz. Round robin ensembles. *Intelligent Data Analysis*, 7(5), 2003.

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

[11] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In M.I. Jordan, M.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[12] E. Hüllermeier and J. Fürnkranz. Comparison of ranking procedures in pairwise preference learning. In *Proceedings IPMU–04, 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Perugia, Italy, 2004.

[13] E. Hüllermeier and J. Fürnkranz. Ranking by pairwise comparison: A note on risk minimization. In *Proceedings* FUZZ-IEEE–04*,* IEEE *International Conference on Fuzzy Systems*, Budapest, Hungary, 2004.

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

[15] E. Hüllermeier and J. Fürnkranz. On minimizing the position error in label ranking. In *Proceedings* ECML–07*, 17th European Conference on Machine Learning*, pages 583–590, Warsaw, Poland, 2007. Springer-Verlag.

[16] M.G. Kendall. *Rank correlation methods.* Charles Griffin, London, 1955.

[17] E.L. Lehmann and H.J.M. D'Abrera. *Nonparametrics: Statistical Methods Based on Ranks.* Prentice-Hall, Englewood Cliffs, NJ, 1998.

[18] S.-H. Park and J. Fürnkranz. Efficient Pairwise Classification. In *Proceedings* ECML–07*, 17th European Conference on Machine Learning*, pages 658–665, Warsaw, Poland, September 2007. Springer-Verlag.

[19] C. Spearman. The proof and measurement for association between two things. *Amer. Journal of Psychology*, 15:72–101, 1904.

[20] J.-N. Sulzmann, J. Fürnkranz, and E. Hüllermeier. On pairwise Naive Bayes classifiers. In *Proceedings* ECML–07*, 17th European Conference on Machine Learning*, pages 371–381, Warsaw, Poland, September 2007. Springer-Verlag.

[21] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. ICML–2004*, Banff, Alberta, Canada, 2004.

[22] S.M. Ulam. Future applications of mathematics in the natural sciences. *American Mathematicsl Heritages: Algebra and Applied Mathematics, Texas Tech. University, Mathematics Series*, 13:101–114, 1981.

[23] T.F. Wu, C.J. Lin, and R.C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

A Proof of Theorem 1

Lemma 5 *Let s_i, $i = 1 \dots m$, be real numbers such that $0 \leq s_1 \leq s_2 \dots \leq s_m$. Then, for all permutations $\tau \in \mathcal{S}_m$,*

$$\sum_{i=1}^{m} (i - s_i)^2 \leq \sum_{i=1}^{m} (i - s_{\tau(i)})^2 \tag{22}$$

Proof: We have

$$\begin{aligned}\sum_{i=1}^{m}(i - s_{\tau(i)})^2 &= \sum_{i=1}^{m}(i - s_i + s_i - s_{\tau(i)})^2 \\ &= \sum_{i=1}^{m}(i - s_i)^2 + 2\sum_{i=1}^{m}(i - s_i)(s_i - s_{\tau(i)}) + \sum_{i=1}^{m}(s_i - s_{\tau(i)})^2.\end{aligned}$$

Expanding the last equation and exploiting that $\sum_{i=1}^{m} s_i^2 = \sum_{i=1}^{m} s_{\tau(i)}^2$ yields

$$\sum_{i=1}^{m}(i - s_{\tau(i)})^2 = \sum_{i=1}^{m}(i - s_i)^2 + 2\sum_{i=1}^{m} i\, s_i - 2\sum_{i=1}^{m} i\, s_{\tau(i)}.$$

On the right-hand side of the last equation, only the last term $\sum_{i=1}^{m} i\, s_{\tau(i)}$ depends on τ. This term is maximal for $\tau(i) = i$, because $s_i \leq s_j$ for $i < j$, and therefore $\max_{i=1\ldots m} m s_i = m s_m$, $\max_{i=1\ldots m-1}(m-1)s_i = (m-1)s_{m-1}$, etc. Thus, the difference of the two sums is always positive, and the right-hand side is larger than or equal to $\sum_{i=1}^{m}(i - s_i)^2$, which proves the lemma. □

Lemma 6 *Let $\mathbb{P}(\cdot \,|\, \mathbf{x})$ be a probability distribution over $\mathcal{S}_m$. Moreover, let*

$$s_i \stackrel{\text{df}}{=} m - \sum_{j \neq i} \mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j) \tag{23}$$

with

$$\mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j) = \sum_{\tau\,:\,\tau(i)<\tau(j)} \mathbb{P}(\tau \,|\, \mathbf{x}). \tag{24}$$

Then, $s_i = \sum_{j \neq i} \mathbb{P}(\tau \,|\, \mathbf{x})\, \tau(i)$.

Proof: We have

$$
\begin{aligned}
s_i &= m - \sum_{j \neq i} \mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j) \\
&= 1 + \sum_{j \neq i} (1 - \mathbb{P}(\lambda_i \succ_{\mathbf{x}} \lambda_j)) \\
&= 1 + \sum_{j \neq i} \mathbb{P}(\lambda_j \succ_{\mathbf{x}} \lambda_i) \\
&= 1 + \sum_{j \neq i} \sum_{\tau \,:\, \tau(j) < \tau(i)} \mathbb{P}(\tau \,|\, \mathbf{x}) \\
&= 1 + \sum_{\tau} \mathbb{P}(\tau \,|\, \mathbf{x}) \sum_{j \neq i} \begin{cases} 1 & \text{if} \quad \tau(i) > \tau(j) \\ 0 & \text{if} \quad \tau(i) < \tau(j) \end{cases} \\
&= 1 + \sum_{\tau} \mathbb{P}(\tau \,|\, \mathbf{x}) (\tau(i) - 1) \\
&= \sum_{\tau} \mathbb{P}(\tau \,|\, \mathbf{x})\, \tau(i)
\end{aligned}
$$

□

Note that $s_i \leq s_j$ is equivalent to $S(\lambda_i) \geq S(\lambda_j)$ (as defined in (12)) under the assumption (15). Thus, ranking the alternatives according to $S(\lambda_i)$ (in decreasing order) is equivalent to ranking them according to s_i (in increasing order).

Using the above results, the claim of Theorem 1 can be verified as follows: We have

$$\begin{aligned}
\mathbb{E}(D(\tau', \tau) \mid \mathbf{x}) &= \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) \sum_{i=1}^{m} (\tau'(i) - \tau(i))^2 \\
&= \sum_{i=1}^{m} \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) (\tau'(i) - \tau(i))^2 \\
&= \sum_{i=1}^{m} \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) (\tau'(i) - s_i + s_i - \tau(i))^2 \\
&= \sum_{i=1}^{m} \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) \left[(\tau(i) - s_i)^2 - 2(\tau(i) - s_i)(s_i - \tau'(i)) \right. \\
&\qquad \left. + (s_i - \tau'(i))^2 \right] \\
&= \sum_{i=1}^{m} \left[\sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) (\tau(i) - s_i)^2 - 2(s_i - \tau'(i)) \cdot \right. \\
&\qquad \left. \cdot \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) (\tau(i) - s_i) + \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x}) (s_i - \tau'(i))^2 \right]
\end{aligned}$$

In the last equation, the mid-term on the right-hand side becomes 0 according to Lemma 6. Moreover, the last term obviously simplifies to $(s_i - \tau'(i))$, and the first term is a constant $c = \sum_{\tau} \mathbb{P}(\tau \mid \mathbf{x})(\tau(i) - s_i)^2$ that does not depend on τ'. Thus, we obtain $\mathbb{E}(D(\tau', \tau) \mid \mathbf{x}) = c + \sum_{i=1}^{m} (s_i - \tau'(i))^2$ and the theorem follows from Lemma 5.