Instance-Based Label Ranking using the Mallows Model

Weiwei Cheng and Eyke Hüllermeier

Mathematics and Computer Science
University of Marburg, Germany
{cheng,eyke}@mathematik.uni-marburg.de

Abstract. In this paper, we introduce a new instance-based approach to the label ranking problem. This approach is based on a probability model on rankings which is known as the Mallows model in statistics. Probabilistic modeling provides the basis for a theoretically sound prediction procedure in the form of maximum likelihood estimation. Moreover, it allows for complementing predictions by diverse types of statistical information, for example regarding the reliability of an estimation. Empirical experiments show that our approach is competitive to start-of-the-art methods for label ranking and performs quite well even in the case of incomplete ranking information.

1 Introduction

The topic of learning preferences has attracted increasing attention recently and contributes to the more general trend of investigating complex and structured output spaces in machine learning, such as label sequences or natural language parsing trees [1, 2]. Label ranking, a particular preference learning scenario, studies the problem of learning a mapping from instances to rankings over a finite number of predefined labels. It can be considered as a natural generalization of the conventional classification problem, where only a single label is requested instead of a ranking of all labels. Applications of label ranking can be found in various fields such as, e.g., natural language processing and document categorization.

Various approaches for label ranking have been proposed in recent years. Typically, these are extensions of learning algorithms used in binary classification problems. Ranking by pairwise comparison (RPC) is a natural extension of pairwise classification, in which binary preference models are learned for each pair of labels, and the predictions of these models are combined into a ranking of all labels [3]. Two other approaches, constraint classification (CC) and log-linear models for label ranking (LL), seek to learn linear utility functions for each individual label instead of preference predicates for pairs of labels [4, 5].

In this paper, we are interested in yet another alternative, namely the use of an *instance-based* approach. Instance-based or case-based learning algorithms have been applied successfully in various fields, such as machine learning and

pattern recognition, for a long time [6, 7]. These algorithms simply store the training data, or at least a selection thereof, and defer the processing of this data until an estimation for a new instance is requested, a property distinguishing them from typical model-based approaches. Instance-based approaches therefore have a number of potential advantages, especially in the context of the label ranking problem.

As a particular advantage of delayed processing, these learning methods may estimate the target function *locally* instead of inducing a global prediction model for the entire input domain (instance space) $\mathbb{X}$. Predictions are typically obtained using only a small, locally restricted subset of the entire training data, namely those examples that are close to the query $\boldsymbol{x} \in \mathbb{X}$ (hence $\mathbb{X}$ must be endowed with a distance measure). These examples are then *aggregated* in a reasonable way. For example, in conventional classification, the class labels of the query's neighbors are usually aggregated by majority voting. As aggregating a finite set of objects from an output space Ω is often much simpler than representing a complete $\mathbb{X} \longrightarrow \Omega$ mapping in an explicit way, instance-based methods are especially appealing if Ω has a complex structure.

In label ranking, Ω corresponds to the set of all rankings of an underlying label set $\mathcal{L}$. To represent an Ω-valued mapping, the aforementioned model-based approaches encode this mapping in terms of conventional binary models, either by a large set of such models in the original label space $\mathcal{L}$ (RPC), or by a single binary model in an expanded, high-dimensional space (CC, LL). As an aside, we note that this transformation of a label ranking problem, either into several simple or into a single complex binary problem, can also come along with a loss of information, which is caused by decomposing complete rankings into several binary preferences [8]. Since for instance-based methods, there is no need to represent an $\mathbb{X} \longrightarrow \Omega$ mapping explicitly, such methods can operate on the original target space Ω directly.

The paper is organized as follows: In Section 2, we introduce the problem of label ranking in a more formal way. The core idea of our instance-based approach to label ranking, namely maximum likelihood estimation based on a special probability model for rankings, is discussed in Section 4. The model itself is introduced beforehand in Section 3. Section 5 gives an overview of related work, and Section 6 is devoted to experimental results. The paper ends with concluding remarks in Section 7.

2 Label Ranking

Label ranking can be seen as an extension of the conventional setting of classification. Roughly speaking, the former is obtained from the latter through replacing single class labels by complete label rankings. So, instead of associating every instance $\boldsymbol{x}$ from an instance space $\mathbb{X}$ with one among a finite set of class labels $\mathcal{L} = \{\lambda_1 \dots \lambda_m\}$, we now associate $\boldsymbol{x}$ with a total order of the class labels, that is, a complete, transitive, and asymmetric relation $\succ_{\boldsymbol{x}}$ on $\mathcal{L}$ where $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i precedes λ_j in the ranking associated with $\boldsymbol{x}$. It follows that a

ranking can be considered as a special type of preference relation, and therefore we shall also say that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i is *preferred* to λ_j given the instance $\boldsymbol{x}$. 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 fields such as Math, CS, Physics.

Formally, a ranking $\succ_{\boldsymbol{x}}$ can be identified with a permutation $\pi_{\boldsymbol{x}}$ of the set $\{1 \dots m\}$. It is convenient to define $\pi_{\boldsymbol{x}}$ such that $\pi_{\boldsymbol{x}}(i) = \pi_{\boldsymbol{x}}(\lambda_i)$ is the position of λ_i in the ranking. This permutation encodes the (ground truth) ranking:

$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \dots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(m)} \ ,$$

where $\pi_{\boldsymbol{x}}^{-1}(j)$ is the index of the label at position j in the ranking. The class of permutations of $\{1 \dots m\}$ (the symmetric group of order m) is denoted by Ω. By abuse of terminology, though justified in light of the above one-to-one correspondence, we refer to elements $\pi \in \Omega$ as both permutations and rankings.

In analogy with the classification setting, we do not assume that there exists a deterministic $\mathbb{X} \to \Omega$ mapping. Instead, every instance is associated with a *probability distribution* over Ω. This means that, for each $\boldsymbol{x} \in \mathbb{X}$, there exists a probability distribution $\mathbb{P}(\cdot \,|\, \boldsymbol{x})$ such that, for every $\pi \in \Omega$,

$$\mathbb{P}(\pi \,|\, \boldsymbol{x}) \tag{1}$$

is the probability that $\pi_{\boldsymbol{x}} = \pi$. In the above example, the following probability distribution may be given for a particular $\boldsymbol{x}$:

label ranking τ	$\mathbb{P}(\tau \,\|\, \boldsymbol{x})$
Math $\succ$ CS $\succ$ Physics	.4
Math $\succ$ Physics $\succ$ CS	.3
CS $\succ$ Math $\succ$ Physics	.0
CS $\succ$ Physics $\succ$ Math	.2
Physics $\succ$ Math $\succ$ CS	.0
Physics $\succ$ CS $\succ$ Math	.1

The goal in label ranking is to learn a "label ranker" in the form of an $\mathbb{X} \to \Omega$ mapping. As training data, a label ranker uses a set of example instances $\boldsymbol{x}_k$, $k = 1 \dots n$, together with information about the associated rankings $\pi_{\boldsymbol{x}_k}$. Ideally, complete rankings are given as training information. From a practical point of view, however, it is also important to allow for incomplete information in the form of a ranking

$$\lambda_{\pi_{\boldsymbol{x}}^{-1}(i_1)} \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(i_2)} \succ_{\boldsymbol{x}} \dots \succ_{\boldsymbol{x}} \lambda_{\pi_{\boldsymbol{x}}^{-1}(i_k)} \ ,$$

where $\{i_1, i_2 \dots i_k\}$ is a subset of the index set $\{1 \dots m\}$ such that $1 \leq i_1 < i_2 < \dots < i_m \leq m$. For example, for an instance $\boldsymbol{x}_k$, it might be known that $\lambda_2 \succ_{\boldsymbol{x}_k} \lambda_1 \succ_{\boldsymbol{x}_k} \lambda_5$, while no preference information is given about the labels λ_3, and λ_4.

To evaluate the predictive performance of a label ranker, a suitable loss function on Ω is needed. In the statistical literature, several distance measures for

rankings have been proposed. One commonly used measure is the number of discordant pairs,

$$D(\pi, \sigma) = \{ (i,j) \,|\, \pi(i) > \pi(j) \text{ and } \sigma(i) < \sigma(j) \} \ , \tag{2}$$

which is closely related to the Kendall tau-coefficient. In fact, the latter is a normalization of (2) to the interval $[-1, 1]$ that can be interpreted as a correlation measure (it assumes the value 1 if $\sigma = \pi$ and the value -1 if σ is the reversal of π). We shall focus on (2) throughout the paper, even though other distance measures can of course be used. A desirable property of any distance $D(\cdot)$ is its invariance toward a renumbering of the elements (renaming of labels). This property is equivalent to the *right invariance* of $D(\cdot)$, namely $D(\sigma\nu, \pi\nu) = D(\sigma, \pi)$ for all $\sigma, \pi, \nu \in \Omega$, where $\sigma\nu = \sigma \circ \nu$ denotes the permutation $i \mapsto \sigma(\nu(i))$. The distance (2) is right-invariant, and so are most other commonly used metrics on Ω.

3 The Mallows Model

So far, we did not make any assumptions about the probability measure (1) despite its existence. To become more concrete, we resort to a distance-based probability model introduced by Mallows [9]. The standard Mallows model is a two-parameter model that belongs to the exponential family:

$$\mathbb{P}(\sigma \,|\, \theta, \pi) = \frac{\exp(\theta D(\pi, \sigma))}{\phi(\theta, \pi)}, \tag{3}$$

where the two parameters are the location parameter (modal ranking, center ranking) $\pi \in \Omega$ and the spread parameter $\theta \leq 0$. For right-invariant metrics, it can be shown that the normalization constant does not depend on π and, therefore, can be written as a function $\phi(\theta)$ of θ alone. This is due to

$$\begin{aligned} \phi(\theta, \pi) &= \sum_{\sigma \in \Omega} \exp(\theta D(\sigma, \pi)) \\ &= \sum_{\sigma \in \Omega} \exp(\theta D(\sigma\pi^{-1}, e)) \\ &= \sum_{\sigma' \in \Omega} \exp(\theta D(\sigma', e)) \\ &= \phi(\theta) \ , \end{aligned}$$

where $e = (1 \ldots n)$ is the identity ranking. More specifically, it can be shown that the normalization constant is given by [10]

$$\phi(\theta) = \prod_{j=1}^{n} \frac{1 - \exp(j\theta)}{1 - \exp(\theta)}, \tag{4}$$

and that the expected distance from the center is

$$\mathbb{E}\left[D(\sigma, \pi) \,|\, \theta, \pi\right] = \frac{n \exp(\theta)}{1 - \exp(\theta)} - \sum_{j=1}^{n} \frac{j \exp(j\theta)}{1 - \exp(j\theta)} \ . \tag{5}$$

Obviously, the Mallows model assigns the maximum probability to the center ranking π. The larger the distance $D(\sigma, \pi)$, the smaller the probability of σ becomes. The spread parameter θ determines how quickly the probability decreases, i.e., how peaked the distribution is around π. For $\theta = 0$, the uniform distribution is obtained, while for $\theta \to -\infty$, the distribution converges to the one-point distribution that assigns probability 1 to π and 0 to all other rankings.

4 Learning and Inference

Coming back to the label ranking problem and the idea of instance-based learning, consider a query instance $\boldsymbol{x} \in \mathbb{X}$ and let $\boldsymbol{x}_1 \ldots \boldsymbol{x}_k$ denote the nearest neighbors of $\boldsymbol{x}$ (according to an underlying distance measure on $\mathbb{X}$) in the training set, where $k \in \mathbb{N}$ is a fixed integer. Moreover, let $\sigma_1 \ldots \sigma_k \in \Omega$ denote the rankings associated, respectively, with $\boldsymbol{x}_1 \ldots \boldsymbol{x}_k$.

In analogy to the conventional settings of classification and regression, in which the nearest neighbor estimation principle has been applied for a long time, we assume that the probability distribution $\mathbb{P}(\cdot \,|\, \boldsymbol{x})$ on Ω is (at least approximately) *locally constant* around the query $\boldsymbol{x}$. By furthermore assuming independence of the observations, the probability to observe $\boldsymbol{\sigma} = \{\sigma_1 \ldots \sigma_k\}$ given the parameters (θ, π) becomes

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\sigma} \,|\, \theta, \pi) &= \prod_{i=1}^{k} \mathbb{P}(\sigma_i \,|\, \theta, \pi) \\
&= \prod_{i=1}^{k} \frac{\exp\left(\theta D(\sigma_i, \pi)\right)}{\phi(\theta)} \\
&= \frac{\exp\big(\theta(D(\sigma_1, \pi) + \ldots + D(\sigma_k, \pi))\big)}{\phi^k(\theta)} \\
&= \frac{\exp\left(\theta \sum_{i=1}^{k} D(\sigma_i, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(j\theta)}{1-\exp(\theta)}\right)^k}.
\end{aligned} \tag{6}$$

The maximum likelihood estimation (MLE) of (θ, π) is then given by those parameters that maximize this probability. It is easily verified that the MLE of π is given by

$$\hat{\pi} \;=\; \arg\min_{\pi} \sum_{i=1}^{k} D(\sigma_i, \pi), \tag{7}$$

i.e., by the (generalized) median of the rankings $\sigma_1 \ldots \sigma_k$. Moreover, the MLE of θ is derived from the average observed distance from $\hat{\pi}$, which is an estimation of the expected distance $\mathbb{E}\left[D(\sigma, \pi) | \theta, \pi\right]$:

$$\frac{1}{k} \sum_{i=1}^{k} D(\sigma_i, \hat{\pi}) = \frac{n \exp(\theta)}{1 - \exp(\theta)} - \sum_{j=1}^{n} \frac{j \exp(j\theta)}{1 - \exp(j\theta)}. \tag{8}$$

Since the right-hand side of (8) is monotone increasing, a standard line search quickly converges to the MLE [10].

Now, consider the more general case of incomplete preference information, which means that a ranking σ_i does not necessarily contain all labels. The probability of σ_i is then given by

$$\mathbb{P}(E(\sigma_i)) \;=\; \sum_{\sigma \in E(\sigma_i)} \mathbb{P}(\sigma \,|\, \theta, \pi) \;\;,$$

where $E(\sigma_i)$ denotes the set of all *consistent extensions* of σ_i: A permutation $\sigma \in \Omega$ is a consistent extension of σ if it ranks all labels that also occur in σ_i in the same order.

The probability of observing the neighbor rankings $\boldsymbol{\sigma} = (\sigma_1 \ldots \sigma_k)$ then becomes

$$\begin{aligned} \mathbb{P}(\boldsymbol{\sigma} \,|\, \theta, \pi) &= \prod_{i=1}^{k} \mathbb{P}(E(\sigma_i) \,|\, \theta, \pi) \\ &= \prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \mathbb{P}(\sigma \,|\, \theta, \pi) \\ &= \frac{\prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \exp\left(\theta D(\sigma, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(j\theta)}{1-\exp(\theta)}\right)^k} \;\;. \end{aligned} \tag{9}$$

Computing the MLE of (θ, π) by maximizing this probability now becomes more difficult. Our current implementation uses a simple brute force approach, namely an exhaustive search over Ω combined with a numerical procedure to optimize the spread θ (given a center ranking π). This approach works for label sets of small to moderate size but becomes infeasible for larger number of labels. Yet, as we are first of all interested in validating the approach from a conceptual point of view, we leave the problem of a more efficient implementation for future work. In this regard, we especially plan to use sophisticated sampling methods [11, 12].

5 Related Work

As mentioned previously, several approaches to label ranking have been proposed in recent years. This section gives a brief review of these approaches that we shall include in the empirical study in Section 6.

5.1 Ranking by Pairwise Comparison

The key idea of pairwise learning is well-known in the context of classification [13], where it allows one to transform a polychotomous 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) $\mathcal{M}_{i,j}$ is trained for each

pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L} \times \mathcal{L}$, $1 \leq i < j \leq m$; thus, a total number of $m(m-1)/2$ models is needed. $\mathcal{M}_{i,j}$ 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 $\mathcal{M}_{i,j}$, and their predictions are combined into an overall prediction. In the simplest case, each prediction of a model $\mathcal{M}_{i,j}$ is interpreted as a vote for either λ_i or λ_j, and the label with the highest number of votes is proposed as the final prediction.

The above procedure can be extended to the case of preference learning in a natural way [14, 15]. Again, a preference (order) information of the form $\lambda_a \succ_{\boldsymbol{x}} \lambda_b$ is turned into a training example $(\boldsymbol{x}, y)$ for the learner $\mathcal{M}_{i,j}$, where $i = \min(a, b)$ and $j = \max(a, b)$. Moreover, $y = 1$ if $a < b$ and $y = 0$ otherwise. Thus, $\mathcal{M}_{i,j}$ is intended to learn the mapping that outputs 1 if $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ and 0 if $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$:

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

The model is trained with all examples $\boldsymbol{x}_k$ for which either $\lambda_i \succ_{\boldsymbol{x}_k} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}_k} \lambda_i$ 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. By using base classifiers that map into the unit interval $[0, 1]$, one obtains a *valued preference relation* $\mathcal{R}(\boldsymbol{x})$ for every (query) instance $\boldsymbol{x} \in \mathbb{X}$:

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

for all $\lambda_i \neq \lambda_j \in \mathcal{L}$. This preference relation provides the point of departure for deriving an associated ranking $\pi_{\boldsymbol{x}}$. A simple though effective strategy is a generalization of the aforementioned voting strategy: Each alternative (label) λ_i is evaluated by the sum of (weighted) votes

$$S(\lambda_i) \; = \sum_{\lambda_j \neq \lambda_i} \mathcal{R}(\boldsymbol{x})(\lambda_i, \lambda_j),$$

and all labels are then ordered according to these evaluations, i.e., such that

$$(\lambda_i \succ_{\boldsymbol{x}} \lambda_j) \Rightarrow (S(\lambda_i) \geq S(\lambda_j)).$$

5.2 Constraint Classification

Instead of comparing pairs of alternatives (labels), another natural way to represent preferences is to evaluate individual alternatives by means of a (real-valued) utility function. Suppose to be given a utility function $f_i : \mathbb{X} \to \mathbb{R}$ for each of the labels λ_i, $i = 1 \ldots m$. Here, $f_i(\boldsymbol{x})$ is the utility assigned to alternative λ_i by instance $\boldsymbol{x}$. To obtain a ranking for $\boldsymbol{x}$, the labels can then be ordered according to these utility scores, i.e., such that $(\lambda_i \succeq_{\boldsymbol{x}} \lambda_j) \Rightarrow (f_i(\boldsymbol{x}) \geq f_j(\boldsymbol{x}))$.

A corresponding method for learning the functions $f_i(\cdot)$, $i = 1 \dots m$, from training data has been proposed in the framework of *constraint classification* [16, 4]. Proceeding from linear utility functions

$$f_i(\boldsymbol{x}) = \sum_{k=1}^{n} \alpha_{ik} x_k \tag{12}$$

with label-specific coefficients α_{ik}, $k = 1 \dots n$, a preference $\lambda_i \succ_x \lambda_j$ translates into the constraint $f_i(\boldsymbol{x}) - f_j(\boldsymbol{x}) > 0$ or, equivalently, $f_j(\boldsymbol{x}) - f_i(\boldsymbol{x}) < 0$. Both constraints, the positive and the negative one, can be expressed in terms of the sign of an inner product $\langle \boldsymbol{z}, \alpha \rangle$, where $\alpha = (\alpha_{11} \dots \alpha_{1n}, \alpha_{21} \dots \alpha_{mn})$ is a concatenation of all label-specific coefficients. Correspondingly, the vector $\boldsymbol{z}$ is constructed by mapping the original ℓ-dimensional training example $\boldsymbol{x} = (x_1 \dots x_\ell)$ into an $(m \times \ell)$-dimensional space: For the positive constraint, $\boldsymbol{x}$ is copied into the components $((i-1) \times \ell + 1) \dots (i \times \ell)$ and its negation $-\boldsymbol{x}$ into the components $((j-1) \times \ell + 1) \dots (j \times \ell)$; the remaining entries are filled with 0. For the negative constraint, a vector is constructed with the same elements but reversed signs. Both constraints can be considered as training examples for a conventional binary classifier in an $(m \times \ell)$-dimensional space: The first vector is a positive and the second one a negative example. The corresponding learner tries to find a separating hyperplane in this space, that is, a suitable vector α satisfying all constraints. For classifying a new example $\boldsymbol{e}$, the labels are ordered according to the response resulting from multiplying $\boldsymbol{e}$ with the i-th ℓ-element section of the hyperplane vector.

Alternatively, [16, 4] propose an online version of constraint classification, namely an iterative algorithm that maintains weight vectors $\alpha_1 \dots \alpha_m \in \mathbb{R}^\ell$ for each label individually. In every iteration, the algorithm checks each constraint $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ and, in case the associated inequality $\alpha_i \times x = f_i(\boldsymbol{x}) > f_j(\boldsymbol{x}) = \alpha_j \times \boldsymbol{x}$ is violated, adapts the weight vectors α_i, α_j appropriately. In particular, using perceptron training, the algorithm can be implemented in terms of a multi-output perceptron in a way quite similar to the approach of [17].

5.3 Log-Linear Models for Label Ranking

So-called log-linear models for label ranking have been proposed in [18]. Here, utility functions are expressed in terms of linear combinations of a set of *base ranking functions*:

$$f_i(\boldsymbol{x}) = \sum_j \alpha_j h_j(\boldsymbol{x}, \lambda_i),$$

where a base function $h_j(\cdot)$ maps instance/label pairs to real numbers. Interestingly, for the special case in which instances are represented as feature vectors $\boldsymbol{x} = (x_1 \dots x_\ell)$ and the base functions are of the form

$$h_{kj}(\boldsymbol{x}, \lambda) = \begin{cases} x_k & \lambda = \lambda_j \\ 0 & \lambda \neq \lambda_j \end{cases} \qquad (1 \leq k \leq \ell,\ 1 \leq j \leq m), \tag{13}$$

the approach is essentially equivalent to CC, as it amounts to learning class-specific utility functions (12). Algorithmically, however, the underlying optimization problem is approached in a different way, namely by means of a boosting-based algorithm that seeks to minimize a (generalized) ranking error in an iterative way.

5.4 Instance-Based Label Ranking

The idea of using an instance-based (case-based) approach to label ranking has already been presented earlier in [19]. There, however, the estimation step is realized by means of an ad-hoc aggregation procedure and not, as in this paper, based on a sound probabilistic inference principle. Besides, the approach is restricted to the case of complete preference information, which is why we did not include it in the experimental study (and also refrain from a more detailed discussion here).

6 Experimental Results

6.1 Methods

In this section, we compare our instance-based (nearest neighbor, NN) approach to label ranking with ranking by pairwise comparison (RPC), constraint classification (CC), and log-linear models for label ranking (LL) as outlined, respectively, in the previous section. CC is implemented in its online-variant using a noise-tolerant perceptron algorithm as a base learner [20].[1] To guarantee a fair comparison, we use LL with (13) as base ranking functions, which means that it is based on the same underlying model class as CC. Moreover, we implement RPC with simple logistic regression as a base learner, which comes down to fitting a linear model and using the logistic link function ($\text{logit}(\pi) = \log(\pi/(1-\pi))$) to derive $[0, 1]$-valued scores, the type of model output requested in RPC. For our NN method, the parameter k (neighborhood size) was selected through cross-validation on the training set. As a distance measure on the instance space we used the Euclidean distance (after normalizing the attributes).

6.2 Data

We used two real-world data sets, dtt and spo, from the bioinformatics field. These data sets contain two types of genetic data, namely phylogenetic profiles and DNA microarray expression data for the Yeast genome.[2] The genome consists of 2465 genes, and each gene is represented by an associated phylogenetic profile of length 24. Using these profiles as input features, we investigated the task of predicting a "qualitative" representation of an expression profile:

[1] This algorithm is based on the "alpha-trick". We set the corresponding parameter α to 500.

[2] This data is publicly available at http://www1.cs.columbia.edu/compbio/exp-phylo

Table 1. Statistics for the semi-synthetic and real datasets

dataset	#examples	#classes	#features
iris	150	3	4
wine	178	3	13
glass	214	6	9
vehicle	846	4	18
ddt	2465	4	24
cold	2465	4	24

Actually, the expression profile of a gene is an ordered sequence of real-valued measurements, such as $(2.1, 3.5, 0.7, -2.5)$, where each value represents the expression level of that gene measured at a particular point of time. A qualitative representation can be obtained by converting the expression levels into ranks, i.e., ordering the time points (= labels) according to the associated expression values. In the above example, the qualitative profile would be given by $(2, 1, 3, 4)$, which means that the highest expression was observed at time point 2, the second-highest at time point 1, and so on. The use of qualitative profiles of that kind, and a rank correlation measure as a similarity measure between them, was motivated in [21], both biologically and from a data analysis point of view.

In addition to the real-world data sets, the following multiclass datasets from the UCI Repository of machine learning databases [22] and the Statlog collection [23] were included in the experimental evaluation: iris, wine, glass, vehicle. For each of these datasets, a corresponding ranking dataset was generated in the following manner: We trained a naive Bayes classifier on the respective dataset. Then, for each example, *all* the labels present in the dataset were ordered with respect to decreasing predicted class probabilities (in the case of ties, labels with lower index are ranked first). Thus, by substituting the single labels contained in the original multiclass datasets with the complete rankings, we obtain the label ranking datasets required for our experiments. The fundamental underlying learning problem may also be viewed as learning a qualitative replication of the probability estimates of a naive Bayes classifier. A summary of the data sets and their properties is given in Table 1.

6.3 Experiments and Results

Results were derived in terms of the Kendall tau correlation coefficient from five repetitions of a ten-fold cross-validation. To model incomplete preferences, we modified the training data as follows: A biased coin was flipped for every label in a ranking in order to decide whether to keep or delete that label; the probability for a deletion is specified by a parameter p.

The results are summarized in Table 2 and furthermore presented graphically in Fig. 1. As can be seen, NN is quite competitive to the model-based approaches and sometimes even outperforms these methods. In any case, it is always close to the best result. It is also remarkable that NN seems to be quite robust toward

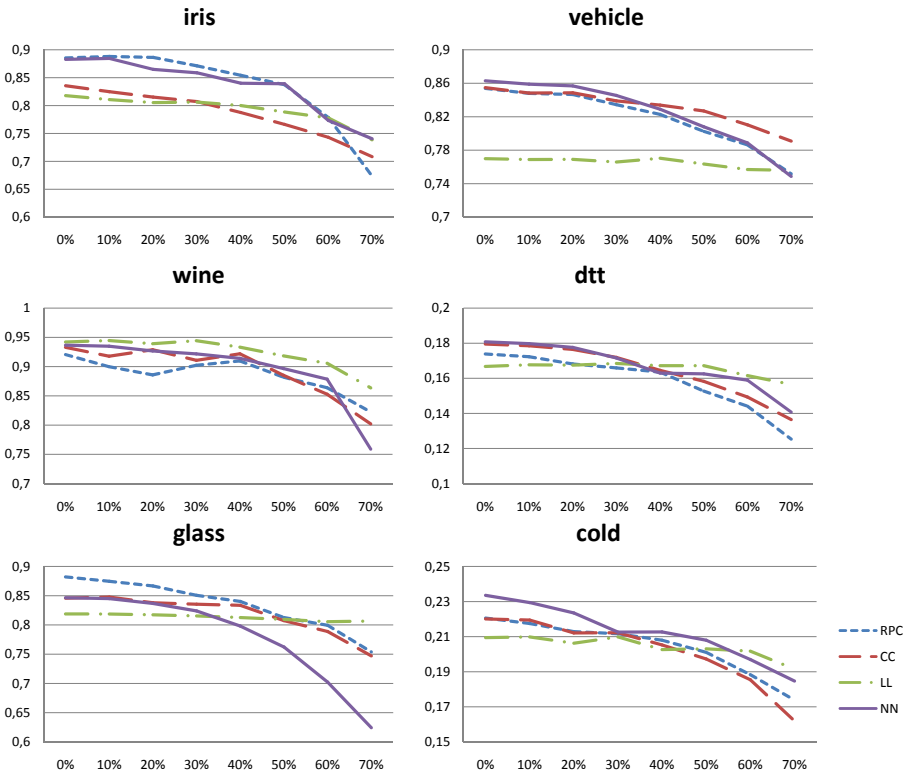

Fig. 1. Graphical illustration of the experimental results in terms of mean values.

Table 2. Experimental results in terms of Kendall's tau (mean and standard deviation) for different missing label rates (parameter p).

iris	*0%*	*10%*	*20%*	*30%*	*40%*	*50%*	*60%*	*70%*
RPC	**.885±.068**	**.888±.064**	**.886±.060**	**.871±.074**	**.854±.082**	.837±.089	**.779±.110**	.674±.139
CC	.836±.089	.825±.095	.815±.088	.807±.099	.788±.105	.766±.115	.743±.131	.708±.105
LL	.818±.088	.811±.089	.805±.087	.806±.087	.800±.091	.788±.087	.778±.096	.739±.186
NN	.883±.060	.884±.066	.865±.072	.859±.069	.840±.082	**.839±.089**	.773±.102	**.740±.117**
wine								
RPC	.921±.053	.900±.067	.886±.073	.902±.063	.910±.065	.882±.082	.864±.097	.822±.118
CC	.933±.043	.918±.057	.929±.058	.911±.059	.922±.057	.885±.074	.853±.078	.802±.123
LL	**.942±.043**	**.944±.046**	**.939±.051**	**.944±.042**	**.933±.062**	**.918±.065**	**.906±.072**	**.864±.094**
NN	.937±.050	.935±.053	.927±.052	.922±.059	.914±.054	.897±.059	.878±.082	.759±.165
glass								
RPC	**.882±.042**	**.875±.046**	**.867±.044**	**.851±.052**	**.840±.053**	**.813±.062**	.799±.054	.754±.076
CC	.846±.045	.848±.053	.838±.059	.835±.054	.833±.051	.807±.066	.789±.052	.747±.061
LL	.817±.060	.815±.061	.813±.063	.819±.062	.819±.060	.809±.066	**.806±.065**	**.807±.063**
NN	.846±.072	.845±.071	.837±.070	.824±.062	.798±.068	.762±.084	.702±.072	.624±.069
vehicle								
RPC	.854±.025	.848±.025	.847±.024	.834±.026	.823±.032	.803±.033	.786±.036	.752±.041
CC	.855±.022	.848±.026	.849±.026	.839±.025	**.834±.026**	**.827±.026**	**.810±.026**	**.791±.030**
LL	.770±.037	.769±.035	.769±.033	.766±.040	.770±.038	.764±.031	.757±.038	.756±.036
NN	**.863±.029**	**.859±.031**	**.857±.028**	**.845±.026**	.829±.033	.808±.029	.789±.032	.749±.040
dtt								
RPC	.174±.034	.172±.034	.168±.036	.166±.036	.164±.034	.153±.035	.144±.028	.125±.030
CC	.180±.037	.178±.034	.176±.033	**.172±.032**	.165±.033	.158±.033	.149±.031	.136±.033
LL	.167±.034	.168±.033	.168±.034	.168±.034	**.167±.033**	**.167±.036**	**.162±.032**	**.156±.034**
NN	**.181±.033**	**.180±.031**	**.178±.034**	**.172±.034**	.163±.034	.163±.038	.159±.037	.141±.033
cold								
RPC	.221±.028	.217±.028	.213±.030	.212±.030	.208±.030	.201±.030	.188±.030	.174±.031
CC	.220±.029	.219±.030	.212±.030	.212±.028	.205±.024	.197±.030	.185±.031	.162±.035
LL	.209±.028	.210±.031	.206±.030	.210±.030	.203±.031	.203±.031	**.202±.032**	**.192±.031**
NN	**.234±.025**	**.229±.028**	**.223±.027**	**.213±.027**	**.213±.027**	**.208±.029**	.197±.024	.185±.027

missing preferences and compares comparably well in this regard. This was not necessarily expected, since NN uses only local information, in contrast to the other approaches that induce global models.

As a nice feature of our approach, let us mention that it comes with a natural measure of the reliability of a prediction. In fact, the smaller the parameter θ, the more peaked the distribution around the center ranking and, therefore, the more reliable this ranking becomes as a prediction. To test whether (the estimation of) θ is indeed a good measure of uncertainty of a prediction, we used it to compute a kind of *accuracy-rejection* curve: By averaging over five 10-fold cross validations, we computed an accuracy degree $\tau_{\boldsymbol{x}}$ (the average Kendall-tau) and a reliability degree $\theta_{\boldsymbol{x}}$ for each instance $\boldsymbol{x}$. The instances are then sorted in decreasing order of reliability. Our curve plots a value p against the mean τ-value of the first p percent of the instances. Given that θ is indeed a good indicator of reliability, this curve should be decreasing, because the higher p, the more instances with a less strong θ-value are taken into consideration. As can be seen in Fig. 2, the curves obtained for our data sets are indeed decreasing and thus provide evidence for our claim that θ may serve as a reasonable indicator of the reliability of a prediction.

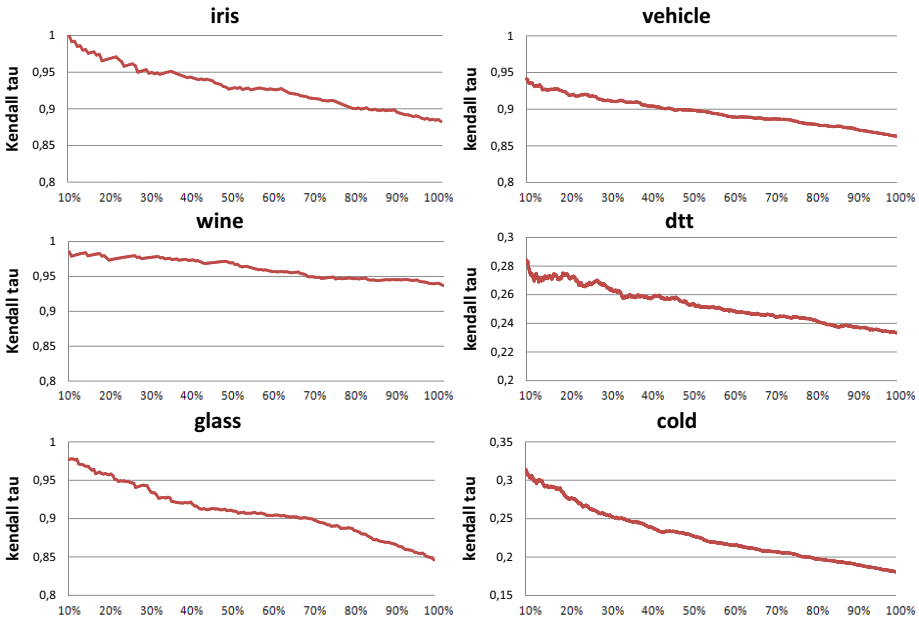

Fig. 2. Accuracy-rejection curves computed on the basis of the parameter θ.

7 Conclusions and Future Work

In this paper, we have introduced an instance-based (nearest neighbor) approach to the label ranking problem that has recently attracted attention in the field of machine learning. Our basic inference principle is a consistent extension of the nearest neighbor estimation principle, as used previously for well-known learning problems such as classification and regression: Assuming that the conditional (probability) distribution of the output given the query is locally constant, we derive a maximum likelihood estimation based on the Mallows model, a special type of probability model for rankings. Our first empirical results are quite promising and suggest that this approach is competitive to (model-based) state-of-the-art methods for label ranking.

Currently, we are working on a more efficient implementation of the estimation step in the case of incomplete preference information. In this case, there is no analytical solution for the MLE problem and, as mentioned previously, a naive implementation (exhaustive search) becomes too expensive. In particular, we plan to use efficient sampling methods to overcome this problem. Besides, we are looking at extensions and variants of the label ranking problem, such a calibrated label ranking and multi-label classification [24, 25].

References

1. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML'04: Proceedings of the 21st International Conference on Machine Learning, New York, USA, ACM Press (2004) 823–830
2. Altun, Y., McAllester, D., Belkin, M.: Margin semi-supervised learning for structured variables. In: In Y. Weiss, B. Schölkopf, J. Platt, eds.: Advances in Neural Information Processing Systems. Volume 18. (2006)
3. Brinker, K., Fürnkranz, J., Hüllermeier, E.: Label ranking by learning pairwise preferences. Technical Report TUD-KE-2007-01, TU Darmstadt, Knowledge Engineering Group (2007)
4. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification for multiclass classification and ranking. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems 15 (NIPS-02). (2003) 785–792
5. Dekel, O., Manning, C., Singer, Y.: Log-linear models for label ranking. In: Advances in Neural Information Processing Systems. (2003)
6. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning **6**(1) (1991) 37–66
7. Hüllermeier, E.: Case-Based Approximate Reasoning. Volume 44 of Theory and Decision Library, Series B: Mathematical and Statistical Methods. Springer-Verlag, Heidelberg, Berlin (2007)
8. Hüllermeier, E., Fürnkranz, J.: Ranking by pairwise comparison: A note on risk minimization. In: IEEE International Conference on Fuzzy Systems. (2004)
9. Mallows, C.: Non-null ranking models. In: Biometrika. Volume 44., Biometrika Trust (1957) 114–130
10. Fligner, M., Verducci, J.: Distance based ranking models. In: Royal Statistical Society. Volume 48. (1986) 359–369
11. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. In: The American Statistician. Volume 49., American Statistical Association (1995) 327–335
12. Diaconis, P., Saloff-Coste, L.: What do we know about the metropolis algorithm? In: Journal of Computer and System Sciences. Volume 57. (1998) 20–36
13. Fürnkranz, J.: Round robin classification. Journal of Machine Learning Research **2** (2002) 721–747
14. Fürnkranz, J., Hüllermeier, E.: Pairwise preference learning and ranking. In: Proc. ECML–03, 13th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia (September 2003)
15. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artificial Intelligence To appear.
16. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification: a new approach to multiclass classification. In: Proceedings 13th Int. Conf. on Algorithmic Learning Theory, Lübeck, Germany, Springer (2002) 365–379
17. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. In: Journal of Machine Learning Research. Volume 3. (2003) 951–991
18. Dekel, O., Manning, C., Singer, Y.: Log-linear models for label ranking. In: Advances in Neural Information Processing Systems. (2003)
19. Brinker, K., Hüllermeier, E.: Case-based label ranking. In: Proceedings ECML–06, 17th European Conference on Machine Learning, Berlin, Springer-Verlag (September 2006) 566–573

20. Khardon, R., Wachman, G.: Noise tolerant variants of the perceptron algorithm. In: The journal of machine learning research. 8 (2007) 227–248
21. Balasubramaniyan, R., Hüllermeier, E., Weskamp, N., Kämper, J.: Clustering of gene expression data using a local shape-based similarity measure. In: Bioinformatics. Volume 21. (2005) 1069–1077
22. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
23. Michie, D., Spiegelhalter, D., Taylor, C.: Machine learning, neural and statistical classification. Ellis Horwood (1994)
24. Brinker, K., Fürnkranz, J., Hüllermeier, E.: A unified model for multilabel classification and ranking. In: Proceedings ECAI–2006, 17th European Conference on Artificial Intelligence, Riva del Garda, Italy (2006) 489–493
25. Brinker, K., Hüllermeier, E.: Case-based multilabel ranking. In: Proc. IJCAI–07, 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (January 2007) 701–707