Case-based Multilabel Ranking

Klaus Brinker and **Eyke Hüllermeier**

Data and Knowledge Engineering,
Otto-von-Guericke-Universität Magdeburg, Germany
{brinker,huellerm}@iti.cs.uni-magdeburg.de

Abstract

We present a case-based approach to multilabel ranking, a recent extension of the well-known problem of multilabel classification. Roughly speaking, a multilabel ranking refines a multilabel classification in the sense that, while the latter only splits a predefined label set into relevant and irrelevant labels, the former furthermore puts the labels within both parts of this bipartition in a total order. We introduce a conceptually novel framework, essentially viewing multilabel ranking as a special case of aggregating rankings which are supplemented with an additional virtual label and in which ties are permitted. Even though this framework is amenable to a variety of aggregation procedures, we focus on a particular technique which is computationally efficient and prove that it computes optimal aggregations with respect to the (generalized) Spearman rank correlation as an underlying loss (utility) function. Moreover, we propose an elegant generalization of this loss function and empirically show that it increases accuracy for the subtask of multilabel classification.

1 Introduction

Multilabel ranking (MLR) is a recent combination of two supervised learning tasks, namely multilabel classification (MLC) and label ranking (LR). The former studies the problem of learning a model that associates with an instance x a bipartition of a predefined set of class labels into relevant (positive) and irrelevant (negative) labels, while the latter considers the problem to predict rankings (total orders) of all class labels. A MLR is a consistent combination of these two types of prediction. Thus, it can either be viewed as an extended ranking (containing additional information about a kind of "zero point"), or as an extended MLC (containing additional information about the order of labels in both parts of the bipartition) [Brinker *et al.*, 2006]. For example, in a document classification context, the intended meaning of the MLR $[\texttt{pol} \succ_x \texttt{eco}][\texttt{edu} \succ_x \texttt{spo}]$ is that, for the instance (= document) x, the classes (= topics) politics and economics are relevant, the former even more than the latter, whereas education and sports are irrelevant, the former perhaps somewhat less than the latter.

From an MLC point of view, the additional order information is not only useful by itself but also facilitates the postprocessing of predictions (e.g., considering only the at most top-k relevant labels). Regarding the relation between MLC and MLR, we furthermore like to emphasize two points: Firstly, as will be seen in the technical part below, MLR is not more demanding than MLC with respect to the training information, i.e., a multilabel *ranker* can well be trained on multilabel *classification* data. Secondly, inducing such a ranker can be useful even if one is eventually only interested in an MLC. Roughly speaking, an MLR model consists of two components, a classifier and a ranker. The interdependencies between the labels which are learned by the ranker can be helpful in discovering and perhaps compensating errors of the classifier. Just to illustrate, suppose that the classifier estimates one label to be relevant and a second one not. The additional (conflicting) information that the latter is typically ranked above the former might call this estimation into question and thus repair the misclassification.

Hitherto existing approaches operating in ranking scenarios are typically *model-based* extensions of binary classification techniques which induce a global prediction model for the entire instance space from the training data [Har-Peled *et al.*, 2002; Fürnkranz and Hüllermeier, 2003]. These approaches, briefly reviewed in Section 3, suffer substantially from the increased complexity of the target space in multilabel ranking (in comparison to binary classification), thus having a high level of computational complexity already for a moderate number of class labels.

In Sections 4 and 5, we present an alternative framework for MLR using a *case-based* methodology which is *conceptually simpler* and *computationally less complex*. One of the main contributions of this paper is casting multilabel ranking as a special case of rank aggregation (with ties) within a case-based framework. While our approach is not limited to any particular aggregation technique, we focus on a computationally efficient technique and prove that it computes optimal aggregations with respect to the well-known (generalized) Spearman rank correlation as an accuracy measure. In Section 6, we show that our case-based approach compares favorably with model-based alternatives, not only with respect to complexity, but also in terms of predictive accuracy.

2 Problem Setting

In so-called *label ranking*, the problem is to learn a mapping from an instance space $\mathcal{X}$ to rankings over a finite set of labels $\mathcal{L} = \{\lambda_1 \ldots \lambda_c\}$, i.e., a function that maps every instance $x \in \mathcal{X}$ to a total strict order $\succ_x$, where $\lambda_i \succ_x \lambda_j$ means that, for this instance, label λ_i is preferred to (ranked higher than) λ_j. A ranking over $\mathcal{L}$ can conveniently be represented by a permutation τ of $\{1 \ldots c\}$, where $\tau(i)$ denotes the position of label λ_i in the ranking. The set of all permutations over c labels, subsequently referred to as $\mathcal{S}_c$, can hence be taken as the target space in label ranking.

Multilabel ranking (MLR) is understood as learning a model that associates with a query input x both a ranking $\succ_x$ and a bipartition (multilabel classification, MLC) of the label set $\mathcal{L}$ into relevant (positive) and irrelevant (negative) labels, i.e., subsets $P_x, N_x \subseteq \mathcal{L}$ such that $P_x \cap N_x = \emptyset$ and $P_x \cup N_x = \mathcal{L}$ [Brinker *et al.*, 2006]. Furthermore, the ranking and the bipartition have to be consistent in the sense that $\lambda_i \in P_x$ and $\lambda_j \in N_x$ implies $\lambda_i \succ_x \lambda_j$.

As an aside, we note that, according to the above consistency requirement, a bipartition (P_x, N_x) implicitly also contains ranking information (relevant labels must be ranked above irrelevant ones). This is why an MLR model can be trained on standard MLC data, even though it considers an extended prediction task.

3 Model-based Multilabel Ranking

A common model-based approach to MLC is *binary relevance learning* (BR). BR trains a separate binary model $\mathcal{M}_i$ for each label λ_i, using all examples x with $\lambda_i \in P_x$ as positive examples and all those with $\lambda_j \in N_x$ as negative ones. To classify a new instance x, the latter is submitted to all models, and P_x is defined by the set of all λ_i for which $\mathcal{M}_i$ predicts relevance.

BR can be extended to the MLR problem in a straightforward way if the binary models provide real-valued confidence scores as outputs. A ranking is then simply obtained by ordering the labels according to these scores [Schapire and Singer, 2000]. On the one hand, this approach is both simple and efficient. On the other hand, it is also ad-hoc and has some disadvantages. For example, good estimations of calibrated scores (e.g., probabilities) are often hard to obtain. Besides, this approach cannot be extended to more general types of preference relations such as, e.g, partial orders. For a detailed survey about MLC and MLR approaches, including case-based methods, we refer the reader to [Tsoumakas *et al.*, 2006].

Brinker *et al.* [2006] presented a unified approach to *calibrated label ranking* which subsumes MLR as a special case. Their framework enables general label ranking techniques, such as the model-based *ranking by pairwise comparison* (RPC) [Fürnkranz and Hüllermeier, 2003] and *constraint classification* (CC) [Har-Peled *et al.*, 2002], to incorporate and exploit partition-related information and to generalize to settings where predicting a separation between relevant and irrelevant labels is required. This approach does not assume the underlying binary classifiers to provide confidence scores. Instead, the key idea in calibrated ranking is to add a virtual label λ_0 as a split point between relevant and irrelevant labels,

i.e., a *calibrated ranking* is simply a ranking of the extended label set $\mathcal{L} \cup \{\lambda_0\}$. Such a ranking induces both a ranking among the (real) labels $\mathcal{L}$ and a bipartite partition (P_x, N_x) in a straightforward way: P_x is given by those labels which are ranked higher than λ_0, N_x by those which are ranked lower. The semantics of the virtual label becomes clear from the construction of training examples for the binary learners: Every label λ_i known to be relevant is preferred to the virtual label $(\lambda_i \succ_x \lambda_0)$; likewise, λ_0 is preferred to all irrelevant labels. Adding these preference constraints to the preferences that can be extracted for the regular labels, a calibrated ranking model can be learned by solving a conventional ranking problem with $c + 1$ labels. We have discussed this approach in more detail as we will advocate a similar idea in extending case-based learning to the multilabel ranking scenario.

4 Case-based Multilabel Ranking

Case-based learning algorithms have been applied successfully in various fields such as machine learning and pattern recognition [Dasarathy, 1991]. In previous work, we proposed a case-based approach which is tailored to label ranking, hence, it cannot exploit bipartite data and does not support predicting the zero point for the multilabel ranking scenario [Brinker and Hüllermeier, 2005]. These algorithms defer processing the training data until an estimation for a new instance is requested, a property distinguishing them from model-based approaches. 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 from the data.

A typically small subset of the entire training data, namely those examples most similar to the query, is retrieved and combined in order to make a prediction. The latter examples provide an obvious means for "explaining" a prediction, thus supporting a human-accessible estimation process which is critical to certain applications where black-box predictions are not acceptable. For label ranking problems, this appealing property is difficult to realize in algorithms using complex global models of the target function as the more complex structure of the underlying target space typically entails solving multiple binary classification problems (RPC yields $c(c + 1)/2$ subproblems) or requires embedding the training data in a higher dimensional feature space to encode preference constraints (such as for CC).

In contrast to the model-based methodology which suffers substantially from the increased complexity of the target space in MLR, we will present a case-based approach where the complexity of the target space solely affects the aggregation step which can be carried out in a highly efficient manner.

The k-nearest neighbor algorithm (k-NN) is arguably the most basic case-based learning method [Dasarathy, 1991]. In its simplest version, it assumes all instances to be represented by feature vectors $x = ([x]_1 \ldots [x]_N)^\top$ in the N-dimensional space $\mathcal{X} = \mathbb{R}^N$ endowed with the standard Euclidian metric as a distance measure, though an extension to other instance spaces and more general distance measures $d(\cdot, \cdot)$ is straightforward. When a query feature vector x is submitted to the k-NN algorithm, it retrieves the k training instances closest to

this point in terms of $d(\cdot, \cdot)$. In the case of classification learning, the k-NN algorithm estimates the query's class label by the most frequent label among these k neighbors. It can be adapted to the regression learning scenario by replacing the majority voting step with computing the (weighted) mean of the target values.

In order to extend the basic k-NN algorithm to multilabel learning, the aggregation step needs to be adapted in a suitable manner. To simplify our presentation, we will focus on the standard MLC case where the training data provides only a bipartition into relevant and non-relevant labels for each instance. Later on, we will discuss how to incorporate more complex preference (ranking) data for training.

Let us consider an example (x, P_x, N_x) from a standard MLC training dataset. As stated above, the key idea in calibrated ranking is to introduce a virtual label λ_0 as a split point to separate labels from P_x and N_x, respectively, and to associate a set of binary preferences with x. We will adopt the idea of a virtual label but, instead of associating preferences, use a more direct approach of viewing the sequence of the label sets $(P_x, \{\lambda_0\}, N_x)$ as a *ranking with ties*, also referred to as a *bucket order* [Fagin *et al.*, 2004]. More precisely, a bucket order is a transitive binary relation $\succ$ for which there exist sets $B_1 \ldots B_m$ that form a partition of the domain $\mathcal{D}$ (which is given by $\mathcal{D} = \mathcal{L} \cup \{\lambda_0\}$ in our case) such that $\lambda \succ \lambda'$ if and only if there are i, j with $i < j$ such that $\lambda \in B_i$ and $\lambda' \in B_j$. Using this notation, the MLR scenario corresponds to a generalized ranking setting with three buckets, where $B_1 = P_x$, $B_2 = \{\lambda_0\}$ and $B_3 = N_x$.

If the training data provides not only a bipartition (P_x, N_x) but also a ranking (with ties) of labels within both parts, this additional information can naturally be incorporated: Assume that P_x and N_x form bucket orders $(B_1 \ldots B_{i-1})$ and $(B_{i+1} \ldots B_j)$, respectively. Then, we can combine this additional information into a single ranking with ties in a straightforward way as $(B_1 \ldots B_{i-1}, B_i, B_{i+1} \ldots B_j)$, where $B_i = \{\lambda_0\}$ represents the split point. Note that the following analysis only assumes that the training data can be converted into rankings with ties, with the virtual label specifying the relevance split point. It will hence cover both training data of the standard MLC case as well as the more complex MLR scenario.

A bucket order induces binary preferences among labels but moreover forms a natural representation for generalizing various metrics on strict rankings to rankings with ties. To this end, we define a generalized rank $\sigma(i)$ for each label $\lambda_i \in \mathcal{D}$ as the average overall position $\sigma(i) = \sum_{l < j} |B_l| + \frac{1}{2}(|B_j| + 1)$ within the bucket B_j which contains λ_i. Fagin *et al.* [2004] proposed several generalizations of well-known metrics such as Kendall's tau and the Spearman footrule distance, where the latter can be written as the l_1 distance of the generalized ranks σ, σ' associated with the bucket orders, $l_1(\sigma, \sigma') = \sum_{\lambda_i \in \mathcal{D}} |\sigma(i) - \sigma'(i)|$.

Given a metric l, a natural way to measure the quality of a single ranking σ as an aggregation of the set of rankings $\sigma_1 \ldots \sigma_k$ is to compute the sum of pairwise distances: $L(\sigma) = \sum_{j=1}^{k} l(\sigma, \sigma_j)$. Then, aggregation of rankings leads to the optimization problem of computing a consensus ranking σ (not necessarily unique) such that $L(\sigma) = \min_\tau L(\tau)$.

The remaining step to actually solve multilabel ranking using the case-based methodology is to incorporate methods which compute (approximately) optimal solutions for the latter optimization problem. As we do not exploit any particular property of the metric l, this approach provides a general framework which allows us to plug in any optimization technique suitable for a metric on rankings with ties in order to aggregate the k nearest neighbors for a query instance x.

The complexity of computing an optimal aggregation depends on the underlying metric and may form a bottleneck as this optimization problem is NP-hard for Kendall's tau [Bartholdi *et al.*, 1989] and Spearman's footrule metric on bucket orders [Dwork *et al.*, 2001].[1] Hence, computing an optimal aggregation is feasible only for relatively small label sets $\{\lambda_1 \ldots \lambda_c\}$. There exist, however, approximate algorithms with quadratic complexity in c which achieve a constant factor approximation to the minimal sum of distances L for Kendall's tau and the footrule metric [Fagin *et al.*, 2004].

While approximate techniques in fact provide a viable option, we will present a computationally efficient and exact method for a generalization of the sum of squared rank differences metric in the following section to implement a version of our case-based multilabel ranking framework.

5 Aggregation Analysis

The Spearman rank correlation coefficient, a linear transformation of the sum of squared rank differences metric, is a natural and well-known *similarity measure* on strict rankings [Spearman, 1904]. It can be generalized to the case of rankings with ties in the same way as the Spearman footrule metric, where (integer) rank values for strict rankings are substituted with average bucket locations. Hence, for any two bucket orders σ, σ', the generalized squared rank difference metric is defined as

$$l_2(\sigma, \sigma') = \sum_{\lambda_i \in \mathcal{D}} (\sigma(i) - \sigma'(i))^2. \qquad (1)$$

The following theorem shows that an optimal aggregation with respect to the l_2 metric can be computed by ordering the labels according to their (generalized) mean ranks.

Theorem 1. *Let $\sigma_1 \ldots \sigma_k$ be rankings with ties on $\mathcal{D} = \{\lambda_1 \ldots \lambda_c\}$. Suppose σ is a permutation such that the labels λ_i are ordered according to $\frac{1}{k} \sum_{j=1}^{k} \sigma_j(i)$ (ties are broken arbitrarily). Then,*

$$\sum_{j=1}^{k} l_2(\sigma, \sigma_j) \leq \min_{\tau \in \mathcal{S}_c} \sum_{j=1}^{k} l_2(\tau, \sigma_j) \qquad (2)$$

Before we proceed to the formal proof, note that the key point in Theorem 1 is that the minimum is taken over $\mathcal{S}_c$ while it is well-known that the minimizer in $\mathbb{R}^c$ would be the mean rank vector. For strict rankings with unique mean rank values, the optimal-aggregation property was proved in [Dwork *et*

---

[1] In the case of strict complete rankings, solving the aggregation problem requires polynomial time for Spearman's footrule metric [Dwork *et al.*, 2001].

al., 2001]. A proof for the more general case of non-unique rank values can be derived from [Hüllermeier and Fürnkranz, 2004].

The following proof is an adaptation of [Hüllermeier and Fürnkranz, 2004] where the *ranking by pairwise comparison* voting procedure for complete strict rankings was analyzed in a probabilistic risk minimization scenario. An essential building block of our proof is the subsequent observation on permutations:

Lemma 2 ([Hüllermeier and Fürnkranz, 2004])**.** *Let m_i, $i = 1 \dots c$, be real numbers ordered such that $0 \le m_1 \le m_2 \le \cdots \le m_c$. Then, for all permutations $\tau \in \mathcal{S}_c$,*

$$\sum_{i=1}^{c} (i - m_i)^2 \le \sum_{i=1}^{c} (i - m_{\tau(i)})^2. \qquad (3)$$

Proof [Theorem 1]. Let us define $m_i \stackrel{\text{def}}{=} \frac{1}{k} \sum_{j=1}^{k} \sigma_j(i)$, $i = 1 \dots c$. Then,

$$\sum_{j=1}^{k} l_2(\tau, \sigma_j) = \sum_{j=1}^{k} \sum_{i=1}^{c} (\tau(i) - \sigma_j(i)^2$$

$$= \sum_{i=1}^{c} \sum_{j=1}^{k} (\tau(i) - m_i + m_i - \sigma_j(i))^2$$

$$= \sum_{i=1}^{c} \sum_{j=1}^{k} (\tau(i) - m_i)^2 - 2(\tau(i) - m_i) \cdot$$
$$\qquad (m_i - \sigma_j(i)) + (m_i - \sigma_j(i))^2$$

$$= \sum_{i=1}^{c} \Big(\sum_{j=1}^{k} (\tau(i) - m_i)^2$$
$$\qquad - 2(\tau(i) - m_i) \sum_{j=1}^{k} (m_i - \sigma_j(i))$$
$$\qquad + \sum_{j=1}^{k} (m_i - \sigma_j(i))^2 \Big).$$

In the last equation, the mid-term equals 0 as

$$\sum_{j=1}^{k} (m_i - \sigma_j(i)) = \sum_{j=1}^{k} \frac{1}{k} \sum_{l=1}^{k} \sigma_l(i) - \sum_{j=1}^{k} \sigma_j(i).$$

Furthermore, the last term is a constant $t \stackrel{\text{def}}{=} \sum_{j=1}^{k} (m_i - \sigma_j(i))^2$ which does not depend on τ. Hence, we obtain

$$\sum_{j=1}^{k} l_2(\tau, \sigma_j) = c\, t + k \sum_{i=1}^{c} (\tau(i) - m_i)^2.$$

The proof follows directly from Lemma 2. $\qquad \square$

We have proved that an l_2-optimal aggregation with respect to the set of permutations can be computed by ordering the labels according to their mean ranks. Regarding the complexity, this method requires computational time in the order of $\mathcal{O}(kc + c \log c)$ for computing and sorting the mean ranks,

hence, providing a very efficient aggregation technique. Note that this method aggregates *rankings with ties* into a single *strict ranking*. The related problem of aggregating into a ranking where ties are allowed forms an interesting area of research in itself and for the case of the l_2-metric the required complexity is an open question. Moreover, multilabel ranking requires predicting strict rankings such that an intermediate aggregation into a ranking with ties would entail an additional postprocessing step and hence forms a less intuitive approach to this problem.

As stated above, the virtual label λ_0 is associated with the second bucket $B_2 = \{\lambda_0\}$ in order to provide a relevance split point. In an initial empirical investigation, we observed that l_2-optimal rankings in k-NN multilabel ranking yield good performance with respect to standard evaluation measures on the *ranking* performance, while the accuracy in terms of multilabel *classification* measures reached a reasonable, yet not entirely satisfactory level. This observation may be attributed to the fact that the l_2-metric penalizes misplaced labels equally for all labels including λ_0. However, particularly in the context of multilabel classification, λ_0 carries a special degree of importance and therefore misclassifications in the aggregation step should be penalized more strongly. In other words, reversing the preference between two labels is especially bad if one of these labels is λ_0, as it means misclassifying the second label in an MLC sense.

To remedy this problem, our approach can be extended in a consistent and elegant manner: Instead of a single virtual label λ_0, we consider a set of virtual labels $\{\lambda_{0,1} \dots \lambda_{0,p}\}$ which is associated with the split bucket B_i. In doing so, the theoretical analysis on the aggregation remains valid and the parameter p provides a means to control the penalty for misclassifications in aggregating rankings. Note that the computational complexity does not increase as the expansion into a set of virtual split labels can be conducted implicitly. Moreover, on computing a prediction, the set of virtual labels can be merged into a single label again in a consistent way as all labels have the same mean rank value.

To illustrate this "gap broadening" control mechanism, let us take a look at a simple aggregation example with three MLC-induced rankings using a single virtual label:

$$\{\lambda_1\} \succ \{\lambda_0\} \succ \{\lambda_2, \lambda_3, \lambda_4, \lambda_5\}$$
$$\{\lambda_1\} \succ \{\lambda_0\} \succ \{\lambda_2, \lambda_3, \lambda_4, \lambda_5\}$$
$$\{\lambda_2\} \succ \{\lambda_0\} \succ \{\lambda_1, \lambda_3, \lambda_4, \lambda_5\}$$

These bucket orders would be aggregated into a total order such that $P = \emptyset$ and $N = \{\lambda_1 \dots \lambda_5\}$ as $m_0 = 2$ (mean rank of λ_0) and every other mean rank is greater, including $m_1 = 2.17$. Using a set of two virtual labels, we obtain $m_0 = m_1 = 2.5$, hence, the order of these labels is determined randomly. Finally, for three virtual labels, $m_0 = 3$ and $m_1 = 2.83$ such that the aggregated calibrated ranking corresponds to a multilabel classification $P = \{\lambda_1\}$ and $N = \{\lambda_2, \lambda_3, \lambda_4\, \lambda_5\}$.

6 Empirical Evaluation

The purpose of this section is to provide an empirical comparison between state-of-the-art model-based approaches and
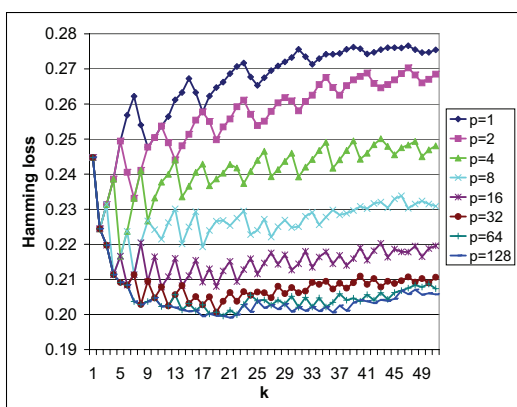
Figure 1: Gap amplification on the (functional) Yeast dataset: The estimated Hamming loss clearly decreases in the parameter p, which controls the number of virtual labels used for splitting relevant and irrelevant labels.

our novel case-based framework (using the l_2-minimizing aggregation technique). The datasets that were included in the experimental setup originate from the bioinformatics fields where multilabeled data can frequently be found. More precisely, our experiments considered two types of genetic data, namely phylogenetic profiles and DNA microarray expression data for the Yeast genome, consisting of 2465 genes.[2] Every gene was represented by an associated phylogenetic profile of length 24. Using these profiles as input features, we investigated the task of predicting a "qualitative" (MLR) representation of an expression profile: Actually, the profile of a gene is a sequence of real-valued measurements, each of which represents the expression level of that gene at a particular time point. Converting the expression levels into ranks, i.e., ordering the time points (= labels) according to the associated expression values) and using the Spearman correlation as a similarity measure between profiles was motivated in [Balasubramaniyan *et al.*, 2005].[3] Here, we further extend this representation by replacing rankings with multilabel rankings. To this end, we use the zero expression level as a natural split point. Thus, the sets P_x and N_x correspond, respectively, to the time points where gene x is over- and under-expressed and, hence, have an important biological meaning.

We used data from eight microarray experiments, giving rise to eight prediction problems all using the same input features but different target rankings. It is worth mentioning that these experiments involve different numbers of measurements, ranging from 4 to 18. Since in our context, each measurement corresponds to a label, we obtain ranking problems of quite different complexity. Besides, even though the original measurements are real-valued, there are many expression profiles containing ties. Each of the datasets was randomly split into a training and a test set comprising 70% and 30%, respectively, of the instances. In compliance with [Balasub-

---

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

[3]This transformation can be motivated from a biological as well as data analysis point of view.

ramaniyan *et al.*, 2005], we measured accuracy in terms of the (generalized) Spearman rank correlation coefficient, normalized such that it evaluates to -1 for reversed and to $+1$ for identical rankings (see Section 5).

Support vector machines have demonstrated state-of-the-art performance in a variety of classification tasks, and therefore have been used as the underlying binary classifiers for the binary relevance (BR) and calibrated ranking by pairwise comparison (CRPC) approaches to multilabel learning in previous studies [Elisseeff and Weston, 2001; Brinker *et al.*, 2006]. Regarding the associated kernel, we considered both linear kernels (LIN) with the margin-error penalty $C \in \{2^{-4} \ldots 2^4\}$ and polynomial kernels (POLY) where the degree varied from 1 to 5 and $C \in \{2^{-2} \ldots 2^2\}$. For each parameter (combination) the validation accuracy was estimated by training on a randomly selected subsample comprising 70% of the training set and testing on the remaining 30%. Then, the final model was trained on the whole training set using the parameter combination which achieved the best validation accuracy. Similarly, the number of nearest neighbors $k \in \{1, 3, .., 11, 21, .., 151\}$ was determined.

In addition to the original k-NN MLR approach, we included a version (denoted by the suffix "-r") which only exploits the MLC training data, (P_x, N_x), and a common extension in k-NN learning leading to a slightly modified aggregation step where average ranks are weighted by the distances of the respective feature vectors to the query vector (referred to as k-NN*).

The experimental results in Table 2 clearly demonstrate that our k-NN approach is competitive with state-of-the-art model-based methods. More precisely, k-NN* and CRPC-POLY achieve the highest level of accuracy, followed by k-NN with only a small margin. BR is outperformed by the other methods, an observation which is not surprising as BR only uses the relevance partition of labels for training and cannot exploit the additional rankings of labels. Similarly, the MLC versions of k-NN perform worse than their MLR counterparts. Moreover, CRPC with polynomial kernels performs slightly better than with linear kernels, whereas for BR a substantial difference cannot be observed. The influence of gap amplification is demonstrated in Figure 1 on an MLC task replicated from [Elisseeff and Weston, 2001], where genes from the same Yeast dataset discussed above have to be associated with functional categories. Moreover, as already anticipated on behalf of our theoretical analysis in Section 5, Table 1 impressively underpins the computational efficiency of our approach from an empirical perspective.

7 Concluding Remarks

We presented a general framework for multilabel ranking using a case-based methodology which is *conceptually simpler* and *computationally less complex* than previous model-based approaches to multilabel ranking. From an empirical perspective, this approach is highly competitive with state-of-the-art methods in terms of accuracy, while being substantially faster.

Conceptually, the modular aggregation step provides a means to extend this approach in several directions. For example, Ha and Haddawy [2003] proposed an appealing

Dataset	Labels	k-NN	k-NN-r	k-NN*	k-NN*-r	CRPC-POLY	CRPC-LIN	BR-POLY	BR-LIN
alpha	18	0.2126	0.2096	0.2164	0.2153	0.2241	0.2167	0.2040	0.2070
elu	14	0.2332	0.2255	0.2367	0.2274	0.2285	0.2164	0.1983	0.2021
cdc	15	0.2021	0.1834	0.2047	0.1871	0.2092	0.1825	0.1867	0.1737
spo	11	0.1858	0.1691	0.1882	0.1715	0.1750	0.1710	0.1496	0.1368
heat	6	0.1576	0.1186	0.1507	0.1206	0.1509	0.1517	0.1040	0.1417
dtt	4	0.3089	0.3155	0.3105	0.3117	0.3117	0.3034	0.2894	0.2303
cold	4	0.2739	0.2719	0.2840	0.2678	0.2975	0.2741	0.2415	0.2421
diau	7	0.4074	0.4069	0.4135	0.4114	0.4122	0.3996	0.3897	0.3609

Table 2: Experimental results on the Yeast dataset using the Spearman rank correlation as the evaluation measure.

	k-NN	CRPC		BR	
Dataset	Test	Train	Test	Train	Test
alpha	1.77	416.90	145.61	44.60	15.84
elu	1.71	240.23	84.41	33.38	12.02
cdc	1.73	280.39	100.47	36.66	13.13
spo	1.71	154.09	54.47	24.32	9.11
heat	1.68	52.98	17.86	15.20	5.17
dtt	1.64	20.90	7.18	6.68	2.59
cold	1.66	22.81	8.07	8.91	3.15
diau	1.69	52.63	18.85	13.02	4.88

Table 1: Computational complexity (in seconds) for training and testing on a Pentium 4 with 2.8GHz (where $k = 100$ for the k-NN approach). The Yeast training and test set consist of 1725 and 740 instances, respectively.

probabilistic loss on preferences which originates from the Kendall tau loss and extends to both *partial* and *uncertain* preferences. Efficient methods for (approximate) rank aggregation with respect to this measure have not been developed yet but could potentially be plugged into our case-based framework in order to generalize to the uncertainty case. Moreover, Chin et al. [2004] studied a weighted variant of the Kendall tau loss function and proposed an approximate aggregation algorithm which requires polynomial time.

Acknowledgments

References

[Balasubramaniyan *et al.*, 2005] R. Balasubramaniyan, E. Hüllermeier, N. Weskamp, and Jörg Kämper. Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics*, 21(7):1069–1077, 2005.

[Bartholdi *et al.*, 1989] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989.

[Brinker *et al.*, 2006] Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A unified model for multilabel classification and ranking. In *Proceedings of the 17th European Conference on Artificial Intelligence* , 2006.

[Brinker and Hüllermeier, 2005] Klaus Brinker and Eyke Hüllermeier. Case-based label ranking. In *Proceedings of ECML 2006*, pages 566–573, 2006.

[Chin *et al.*, 2004] Francis Y. L. Chin, Xiaotie Deng, Qizhi Fang, and Shanfeng Zhu. Approximate and dynamic rank aggregation. *Theor. Comput. Sci.*, 325(3):409–424, 2004.

[Dasarathy, 1991] B.V. Dasarathy. Nearest neighbor (NN) norms: NN pattern classification techniques, 1991.

[Dwork *et al.*, 2001] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation revisited. In *World Wide Web*, pages 613–622, 2001.

[Elisseeff and Weston, 2001] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in NIPS 14*, pages 681–687, 2001.

[Fagin *et al.*, 2004] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing and aggregating rankings with ties. In *Proc. 23rd ACM Symposium on PODS*, pages 47–58, 2004.

[Fürnkranz and Hüllermeier, 2003] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *Proceedings of ECML 2003*, pages 145–156, 2003.

[Ha and Haddawy, 2003] Vu Ha and Peter Haddawy. Similarity of personal preferences: theoretical foundations and empirical analysis. *Artif. Intell.*, 146(2):149–173, 2003.

[Har-Peled *et al.*, 2002] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification: A new approach to multiclass classification and ranking. In *Advances in NIPS 15*, 2002.

[Hüllermeier and Fürnkranz, 2004] Eyke Hüllermeier and Johannes Fürnkranz. Comparison of ranking procedures in pairwise preference learning. In *IPMU–04*, pages 535–542, 2004.

[Schapire and Singer, 2000] Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[Spearman, 1904] Charles Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.

[Tsoumakas *et al.*, 2006] G. Tsoumakas, I. Katakis, and I. Vlahavas. A review of multi-label classification methods. In *2nd ADBIS Workshop on Data Mining and Knowledge Discovery*, pages 99–109, 2006.