A Reduction of Label Ranking to Multiclass Classification

Klaus Brinker[1](✉) and Eyke Hüllermeier[2]

[1] Hamm-Lippstadt University of Applied Sciences, Hamm, Germany
klaus.brinker@hshl.de
[2] Paderborn University, Paderborn, Germany
eyke@upb.de

Abstract. Label ranking considers the problem of learning a mapping from instances to strict total orders over a predefined set of labels. In this paper, we present a framework for label ranking using a decomposition into a set of *multiclass problems*. Conceptually, our approach can be seen as a generalization of pairwise preference learning. In contrast to the latter, it allows for controlling the granularity of the decomposition, varying between binary preferences and complete rankings as extreme cases. It is specifically motivated by limitations of pairwise learning with regard to the minimization of certain loss functions. We discuss theoretical properties of the proposed method in terms of accuracy, error correction, and computational complexity. Experimental results are promising and indicate that improvements upon the special case of pairwise preference decomposition are indeed possible.

Keywords: Label ranking · Multiclass classification · Structured output prediction · Ensemble learning

1 Introduction

In the recent past, various types of ranking problems emerged in the field of preference learning [12]. A well-known example is "learning to rank" or object ranking [19], where the task is to rank any subset of objects (typically described as feature vectors) from a given domain (e.g. documents). In this paper, our focus is on a related but conceptually different problem called *label ranking* (LR). The task in LR is to learn a mapping from an input space $\mathcal{X}$ to strict total orders over a predefined set $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$ of labels (e.g. political parties, music genres, or social emotions). Like in multiclass classification, these labels are only distinguished by their name but not described in terms of any properties.

Previous approaches to label ranking can be grouped into four main categories with respect to model representation [22]: Algorithms learning real-valued scoring functions for each label [8,15,20], instance-based methods [2,6], tree-based methods [5], and binary decomposition methods [18]. In terms of predictive accuracy, especially the latter turned out to be highly competitive.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-46133-1_13) contains supplementary material, which is available to authorized users.

U. Brefeld et al. (Eds.): ECML PKDD 2019, LNAI 11908, pp. 204–219, 2020.
https://doi.org/10.1007/978-3-030-46133-1_13

This paper introduces a generalization of binary decomposition in the sense that LR problems are transformed into multiple *multiclass* classification problems. Thus, the atomic elements of our framework are partial rankings of a fixed length k, including pairwise preferences ($k = 2$) and complete rankings ($k = m$) as special cases. Encoding rankings over fixed subsets of k labels as (meta-)classes, each such subset gives rise to a multiclass classification problem. At prediction time, the corresponding multiclass classifiers (either all of them or a suitable subset) are queried with a new instance, and their predictions are combined into a ranking on the entire label set $\mathcal{L}$.

Intuitively, a decomposition into partial rankings instead of label pairs increases the degree of overlap between the various subproblems, and thereby the ability to correct individual prediction errors. Formally, it can indeed be shown that, to minimize specific loss functions on rankings, knowledge about binary relationships between labels is principally insufficient; an extreme example is the 0/1-loss, which simply checks whether the entire ranking is correct or not.

Our framework for LR is analyzed from a theoretical perspective with respect to both accuracy, error correcting properties, and computational complexity. More precisely, for the aforementioned 0/1-loss and Kendall's tau rank correlation, we present bounds in terms of the average classification error of the underlying multiclass models on the training data, hence, providing a justification for the general consistency of our approach in the sense that accurate multiclass models imply accurate overall LR predictions on the training data. With respect to error correction, we present a theoretical result on the number of multiclass errors which our framework can compensate while still recovering the correct overall ranking.

Empirically, we also analyze the influence of the decomposition granularity k. Our results are promising and suggest that improvements upon the special case of binary decomposition are indeed possible. In the experiments, we observed a consistent relationship between a suitable choice of k and the overall number of labels m.

Section 2 recalls the problem setting of LR, along with notational conventions. Our novel LR framework is presented in Sect. 3. Sections 4 and 5 analyze theoretical properties in terms of accuracy, error correction, and computational complexity, respectively. Section 6 is devoted to an experimental evaluation of our approach. We conclude with a few remarks and open research directions in Sect. 7.

2 Label Ranking

In label ranking, the goal is to learn a predictive model in the form of a mapping $f : \mathcal{X} \rightarrow \mathcal{L}^*$, where $\mathcal{X}$ is an instance space and $\mathcal{L}^*$ the set of all rankings (strict total orders) over a set of labels $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$. Formally, we represent rankings in terms of permutations π of $[m] = \{1, \ldots, m\}$, such that $\pi(i)$ is the (index of the) label at position i. Thus, LR assumes instances $x \in \mathcal{X}$ to be associated with rankings $\pi \in \mathcal{L}^*$. More specifically, we assume this dependency to be probabilistic: Given x as input, there is a certain probability $\mathbf{P}(\pi \,|\, x)$ to observe the ranking $\pi \in \mathcal{L}^*$ as output.

We suppose training data to be given in the form of a set of observations $\mathcal{D} = \{(x_i, \pi_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{L}^*$. Thus, for convenience, we assume the observed rankings to be complete. In practice, this assumption is of course not always fulfilled. Instead, rankings are often incomplete, with some of the labels $\lambda_i \in \mathcal{L}$ being missing. Predictive performance is evaluated in terms of a loss function $\ell : \mathcal{L}^* \times \mathcal{L}^* \rightarrow \mathbb{R}$, which compares observed rankings π with predictions $\hat{\pi}$. Common choices include rank correlation measures such as Kendall's tau and Spearman's rho (which are actually similarity measures).

In the pairwise approach to LR [18], called ranking by pairwise comparison (RPC), one binary model $f_{i,j}$ is trained for each pair of labels (λ_i, λ_j). The task of $f_{i,j}$ is to predict, for a given instance x, whether λ_i precedes or succeeds λ_j in the ranking associated with x. A prediction $\hat{\pi}$ is produced by aggregating the (possibly conflicting) $m(m-1)/2$ pairwise predictions, using techniques such as (weighted) voting.

An important question for any reduction technique is the following: Is it possible to combine the solutions of the individual subproblems into an optimal solution for the original problem? In the case of RPC, this question can be asked more concretely as follows: Is it possible to train and combine the binary predictors $f_{i,j}$ so as to obtain an optimal predictor f for the original LR problem, i.e., a predictor that minimizes the loss ℓ in expectation? Interestingly, this question can be answered affirmatively for several performance measures, including Kendall's tau and Spearman's rho. However, there are also measures for which this is provably impossible. These include the 0/1-loss, the Hamming, the Cayley, and the Ulam distance on rankings, as well as Spearman's footrule [16]. Roughly speaking, for these measures, the loss of information due to projecting the distribution $\mathbf{P}(\cdot \,|\, x)$ on $\mathcal{L}^*$ to its pairwise marginals $\mathbf{P}_{i,j}(\cdot \,|\, x)$ is too high. That is, even knowledge about all pairwise marginals does not allow for reconstructing the risk-minimizing prediction $\hat{\pi}$.

Indeed, pairwise relations may easily lead to ambiguities, such as preferential cycles. Knowledge about the distribution of rankings on larger label subsets, such as triplets or quadruples, may then help to disambiguate. This is a key motivation of the approach put forward in this paper, very much in line with the so-called *listwise* approaches to learning-to-rank problems [3]. In this regard, LR is also quite comparable to multi-label classification (MLC): There are loss functions (such as Hamming) that allow for reducing the original MLC problem to binary classification, but also others (such as the subset 0/1-loss) for which reduction techniques cannot produce optimal results, and which require information about the distribution on larger subsets of labels [9].

3 Reduction to Multiclass Classification

In this section, we present a novel framework for solving LR problems using a decomposition into multiple multiclass classification problems. We refer to this approach as LR2MC (Label Ranking to Multiclass Classification).

3.1 Decomposition

For a given $k \in \{2, \ldots, m\}$, consider all label subsets $L \subset \mathcal{L}$ of cardinality k. Each such L gives rise to a label ranking problem on the reduced set of labels. For the special case $k = 2$, the same decomposition into $\binom{m}{2}$ binary classification problems is produced as for RPC.

To illustrate the decomposition process, consider a setting with $m = 4$ labels $\mathcal{L} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ and subsets of cardinality $k = 3$. Each of the $\binom{4}{3}$ subsets $\{\lambda_1, \lambda_2, \lambda_3\}$, $\{\lambda_1, \lambda_2, \lambda_4\}$, $\{\lambda_1, \lambda_3, \lambda_4\}$, and $\{\lambda_2, \lambda_3, \lambda_4\}$ gives rise to a separate LR problem. In the following, we denote by $\pi^{(L)}$ the restriction of a ranking $\pi \in \mathcal{L}^*$ to the label subset $L \subset \mathcal{L}$, i.e., $\pi^{(L)} = (\lambda_3, \lambda_4, \lambda_2)$ for $\pi = (\lambda_3, \lambda_4, \lambda_1, \lambda_2)$ and $L = \{\lambda_2, \lambda_3, \lambda_4\}$. For each ranking problem on a label subset L of size 3, there are $3! = 6$ elements in the output space L^*, for example $(\lambda_1, \lambda_2, \lambda_3)$, $(\lambda_1, \lambda_3, \lambda_2)$, $(\lambda_2, \lambda_1, \lambda_3)$, $(\lambda_2, \lambda_3, \lambda_1)$, $(\lambda_3, \lambda_1, \lambda_2)$, $(\lambda_3, \lambda_2, \lambda_1)$ in the case of the label subset $L = \{\lambda_1, \lambda_2, \lambda_3\}$.

In general, for a label ranking problem with $|\mathcal{L}| = m$ labels, we construct a set of $\binom{m}{k}$ ranking problems on label subsets of size $|L| = k$. Each of these problems is then converted into a multiclass problem. To this end, each of the $k!$ rankings $\pi^{(L)} \in L^*$ is associated with a class $c \in C = \{c_1, \ldots, c_{k!}\}$. We denote by f_L the multiclass classifier on subset L. For the sake of simplicity, we assume the decoding of class labels to the associated rankings to be done directly, i.e., f_L is viewed as a mapping $\mathcal{X} \to L^*$ (instead of a mapping $\mathcal{X} \to C$), and $f_L(x)$ is the ranking of labels in L predicted for x.

3.2 Aggregation

We will now discuss our approach to combining partial ranking predictions on label subsets into a prediction on the complete label set $\mathcal{L}$. The idea is to find a consensus ranking that disagrees with the minimum number of subset ranking predictions. In Sect. 4, we further elaborate on theoretical properties of this aggregation method.

For a new instance $x \in \mathcal{X}$, we first compute the predicted rankings $\{f_L(x) : L \subset \mathcal{L}, |L| = k\}$ on the $\binom{m}{k}$ label subsets. These predictions are combined into a complete ranking $f(x) \in \mathcal{L}^*$ by minimizing the sum of (p‘artial) 0/1-loss values:

$$f(x) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\pi \in \mathcal{L}^*} \sum_{\substack{L \subset \mathcal{L} \\ |L| = k}} \ell_{01}\left(\pi^{(L)}, f_L(x)\right), \tag{1}$$

where

$$\ell_{01}(\rho, \rho') = \begin{cases} 0 & \text{for } \rho = \rho' \\ 1 & \text{for } \rho \neq \rho' \end{cases} \tag{2}$$

for $\rho, \rho' \in L^*$ denotes the 0/1-loss for two label rankings. Ties are broken arbitrarily.

This loss-based aggregation is similar to *Hamming decoding* for computing a multiclass label from binary classifications [1]. Moreover, for $k = 2$, it is identical to the *Slater-optimal* aggregation in pairwise LR [17]. For $k = m$, there is only one label subset $L = \mathcal{L}$. In this case, the risk-minimizing ranking is obviously the multiclass prediction for this set, mapped to the associated label ranking.

3.3 Discussion

What is an optimal choice of k? This question is not easy to answer, and different arguments support different choices:

- Loss of information: The decomposition of a ranking necessarily comes with a loss of information, and the smaller the components, the larger the loss. This argument suggests large values k (close or equal to m).
- Redundancy: The more redundant the predictions, the better mistakes of individual predictors f_L can be corrected. This argument suggests midsize values $k \approx m/2$, for which the number $\binom{m}{k}$ of predictions combined in the aggregation step is largest.
- Simplicity: The difficulty of a multiclass problem increases with the number of classes. This argument suggests small values k, specifically $k = 2$. Practically, it will indeed be difficult to go beyond $k = 4$, since the number of classes $k!$ will otherwise be prohibitive—issues such as class-imbalance and empty classes will then additionally complicate the problem.

As for the last point, also note that for $k > 2$, our reduction to classification comes with a loss of structural information: By mapping partial rankings $\pi^{(L)}$ to class labels c, i.e., replacing the space L^* by the set of classes C, any information about the structure of the former is lost (since C is not equipped with any structure except the discrete metric). This information is only reconsidered in the aggregation step later on. Interestingly, exactly the opposite direction, namely exploiting structure on the label space by turning a multiclass classification into a ranking problem, has recently been considered in [14].

LR2MC is conceptually related to the Ra*k*el method for multilabel classification [21]. Ra*k*el decomposes an MLC problem with m labels into (randomly chosen) MLC problems for subsets of size k, and tackles each of these problems as a multiclass problem with 2^k classes (corresponding to the label subsets). Both approaches share the underlying motivation of taking dependencies between labels and partial rankings, respectively, into account. Moreover, both approaches also resemble error correcting output encodings [10], which aim at improving the overall multiclass classification accuracy by combining multiple binary classifiers and provide a means to distribute the output representation.

4 Theoretical Analysis

In this section, we upper bound the average training 0/1-loss and Kendall's tau rank correlation of LR2MC (as defined in (2)) in terms of the average classification error of the multiclass classifiers $\{f_L\}_{L \subset \mathcal{L}, |L|=k}$ and analyze error correcting properties. Our analysis is similar to [1], where multiclass classification is reduced to learning multiple binary classifiers. In that paper, the authors provide a bound on the empirical multiclass loss in terms of the loss of the binary classifiers and properties and the decomposition scheme.

Theorem 1 (Training loss bound). *Let* $\{f_L\}_{L\subset\mathcal{L},|L|=k}$ *with* $2 \leq k \leq m$ *denote a set of multiclass models and let* ε *denote their average classification error on the training sets decomposed from the LR training data* $\mathcal{D} = \{(x_i, \pi_i)\}_{i=1}^n$. *Then, we can upper bound the average ranking 0/1-loss on the training data as follows:*

$$\frac{1}{n}\sum_{i=1}^{n}\ell_{01}(\pi_i, f(x_i)) \leq \frac{2m(m-1)}{k(k-1)} \cdot \varepsilon$$

Proof. Let $f(x) = \hat{\pi}$ denote the predicted ranking for the training example $(x, \pi) \in \mathcal{D}$. Assume that the predicted and the ground truth rankings disagree, $\pi \neq \hat{\pi}$, hence $\ell_{01}(\pi, \hat{\pi}) = 1$. Moreover, as $\hat{\pi}$ is a minimizer of (1), it holds that

$$\sum_{\substack{L\subset\mathcal{L}\\|L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) \geq \sum_{\substack{L\subset\mathcal{L}\\|L|=k}} \ell_{01}(\hat{\pi}^{(L)}, f_L(x)).$$

Let $S_\Delta \overset{\text{def}}{=} \{L \subset \mathcal{L}^* : |L| = k \wedge \hat{\pi}^{(L)} \neq \pi^{(L)}\}$ denote the label sets of cardinality k, where π and $\hat{\pi}$ disagree. Then,

$$\sum_{L\in S_\Delta} \ell_{01}(\pi^{(L)}, f_L(x)) \geq \sum_{L\in S_\Delta} \ell_{01}(\hat{\pi}^{(L)}, f_L(x)).$$

Therefore,

$$\begin{aligned}
\sum_{\substack{L\subset\mathcal{L}\\|L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) &\geq \sum_{L\in S_\Delta} \ell_{01}(\pi^{(L)}, f_L(x)) \\
&= \frac{1}{2}\left(\sum_{L\in S_\Delta} \left(\ell_{01}(\pi^{(L)}, f_L(x)) + \ell_{01}(\pi^{(L)}, f_L(x))\right)\right) \\
&\geq \frac{1}{2}\left(\sum_{L\in S_\Delta} \underbrace{\left(\ell_{01}(\pi^{(L)}, f_L(x)) + \ell_{01}(\hat{\pi}^{(L)}, f_L(x))\right)}_{\geq 1}\right) \\
&\geq \frac{1}{2}|S_\Delta| \geq \frac{1}{2}\binom{m-2}{k-2}.
\end{aligned}$$

The last step follows from the fact that for the non-equal rankings π and $\hat{\pi}$ at least two labels are in reverse order. Hence, all $\binom{m-2}{k-2}$ restrictions to subsets of k labels which contain the inversed label pair are not equal as well, and therefore elements of S_Δ. In summary, a prediction error, i.e., $\ell_{01}(\pi, f(x)) = 1$ implies that

$$\frac{2}{\binom{m-2}{k-2}} \sum_{\substack{L\subset\mathcal{L}\\|L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) \geq 1\,.$$

Moreover, as

$$\frac{2}{\binom{m-2}{k-2}} \sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) \geq 0$$

for $\ell_{01}(\pi, f(x)) = 0$, it holds for all (x, π) that

$$\ell_{01}(\pi, f(x)) \leq \frac{2}{\binom{m-2}{k-2}} \sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) \, .$$

Therefore, for the average 0/1-loss, it holds that

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} \ell_{01}(\pi_i, f(x_i)) &\leq \frac{1}{n}\sum_{i=1}^{n} \frac{2}{\binom{m-2}{k-2}} \sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi_i^{(L)}, f_L(x_i)) \\ &= \frac{2}{\binom{m-2}{k-2}} \frac{1}{n} \sum_{i=1}^{n} \sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi_i^{(L)}, f_L(x_i)) \\ &= \frac{2\binom{m}{k}}{\binom{m-2}{k-2}} \cdot \underbrace{\frac{1}{n\binom{m}{k}} \sum_{i=1}^{n} \sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi_i^{(L)}, f_L(x_i))}_{=\varepsilon} \\ &= \frac{2m(m-1)}{k(k-1)} \cdot \varepsilon \end{aligned}$$

For the special case of perfect multiclass models on the training data ($\varepsilon = 0$), the 0/1-loss for LR2MC is zero. Moreover, when entirely ignoring the structure of the output space ($m = k$) and approaching label ranking as a multiclass problem with $m!$ classes, the 0/1-loss for label ranking is bounded by twice the average multiclass error.

Corollary 1 (Error correction). *Let* $\{f_L\}_{L \subset \mathcal{L}, |L|=k}$ *with* $2 \leq k \leq m$ *denote a set of multiclass models. For any observation* $(x, \pi) \in \mathcal{X} \times \mathcal{L}^*$ *with*

$$\sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}\left(\pi^{(L)}, f_L(x)\right) < \frac{1}{2}\binom{m-2}{k-2}$$

it holds that $f(x) = \pi$.

Proof. As shown in the proof of Theorem 1, $\sum_{\substack{L \subset \mathcal{L} \\ |L|=k}} \ell_{01}(\pi^{(L)}, f_L(x)) \geq \frac{1}{2}\binom{m-2}{k-2}$ for $f(x) \neq \pi$.

Corollary 1 provides some interesting insights into the error correction properties of our framework:

- For $k = 2$, the right-hand side is $\frac{1}{2}$, and hence the inequality is satisfied only if *all* pairwise predictions are correct. Indeed, it is obvious that a single pairwise error may result in an incorrect overall prediction.
- For $k = m$, there exists only a single multiclass model which is mapped to an LR prediction. In this case, a correct overall prediction is guaranteed if this multiclass prediction is correct.
- For $k = \lfloor \frac{m}{2} + 1 \rfloor$ and $k = \lceil \frac{m}{2} + 1 \rceil$, the right-hand side evaluates to the maximum possible value(s) and Corollary 1 indicates that our framework is capable of correcting a substantial number of multiclass errors.

The last point suggests that values $k \approx \frac{m}{2} + 1$ should lead to optimal performance, because the ability to correct errors of individual classifiers is highest for these values. One should keep in mind, however, that the error probability itself, which corresponds to the performance of a $(k!)$-class classifier[1], will (strongly) increase with k. Depending on how quickly this error increases, and whether or not it can be over-compensated through more redundancy, the optimal value of k is supposed to lie somewhere between 2 and $\frac{m}{2} + 1$. This is confirmed by experimental results for a simplified synthetic setting that we present in the supplementary material (Section A.8).

Corollary 2 (Training bound for Kendall's tau). *Rescaling Kendall's τ rank correlation into a loss function by $\frac{1-\tau}{2} \in [0, 1]$, it holds under the assumptions of Theorem 1 that*

$$\frac{1}{n}\sum_{i=1}^{n} \frac{1 - \tau(\pi_i, f(x_i))}{2} \leq \frac{2m(m-1)}{k(k-1)} \cdot \varepsilon$$

Proof. Direct consequence of Theorem 1 and the fact that for all rankings π and π' it holds that

$$\frac{1 - \tau(\pi, \pi')}{2} \leq \ell_{01}(\pi, \pi').$$

Corollary 2 shows that the average multiclass error on the training data not only bounds the 0/1-loss but also guarantees a certain level of overall ranking accuracy as measured by Kendall's tau.

5 Computational Complexity

The set of multiclass models is at the core of LR2MC. The training data for each multiclass model consists of the original instances x_i and the appropriate class labels which represent rankings on label subsets. Hence, we need to train and evaluate $\binom{m}{k}$ multiclass models using n training examples with up to $k!$ different class labels.

[1] The effective number of classes a classifier is trained on corresponds to the distinct number of permutations of the k labels in the training data, which is normally $< k!$.

Depending on the influence of the number of classes on the complexity of the classification learning algorithm and the choice of k, the computational complexity can increase substantially compared to the pairwise ranking approach, where $\binom{m}{2} = \frac{m(m-1)}{2}$ binary models are required. More precisely, the maximum number of multiclass models for a given m is required for $k = \lfloor \frac{m}{2} \rfloor$ as a consequence of basic properties of binomial coefficients. For this case, it holds that

$$\binom{m}{\lfloor \frac{m}{2} \rfloor} \geq \left(\frac{m}{\lfloor \frac{m}{2} \rfloor} \right)^{\lfloor \frac{m}{2} \rfloor} \geq 2^{\lfloor \frac{m}{2} \rfloor} \geq 2^{\frac{m-1}{2}} = (\sqrt{2})^{m-1}.$$

Hence, the maximum number of multiclass problems is lower bounded by a term which grows exponentially in the number of labels m.

For testing, we need to evaluate all $\binom{m}{k}$ multiclass models and solve the optimization problem (1). Even when ignoring the multiclass model evaluation complexity, the computational complexity for this combinatorial optimization problem increases substantially in m as there are $m!$ possible rankings to be considered. Moreover, as stated above, for $k = 2$, this aggregation method is identical to computing the *Slater-optimal* label ranking which is known to be NP-complete [17].

Overall, even considering that the number of labels m in LR is normally small (mostly < 10, comparable to multiclass classification), computational complexity is clearly an issue for LR2MC. Of course, there are various directions one could think of to improve efficiency. For example, instead of decomposing into all label subsets of size k, one may (randomly) chose only some of them, like in Ra$k$el. Besides, aggregation techniques less expensive than Hamming decoding could be used, such as (generalized) Borda count. While these are interesting directions for future work, the main goal of this paper is to elaborate on the usefulness of the approach in principle, i.e., to answer the question whether or not a generalization from pairwise to multiclass decomposition is worthwhile at all.

6 Experimental Evaluation

6.1 Setup

This section presents an experimental evaluation of the accuracy of LR2MC, which essentially consists of two studies. For the first study, we replicated a setting previously proposed in [4], where a suite of benchmark datasets for label ranking was used[2]; see Table 1 for an overview of the dataset properties. The second study is based on an artificial setting from [13], where the underlying problem is to learn the ranking function induced by an expected utility

[2] These are classification and regression datasets from the UCI and Statlog repository, which were turned into label ranking problems. The datasets are publicly available at https://cs.uni-paderborn.de/is/research/research-projects/software/. Due to the computational demands of LR2MC, we restricted our evaluation to datasets with $m \leq 7$ labels.

Table 1. Dataset characteristics

Dataset	Instances	Attributes	Labels
authorship	841	70	4
bodyfat	252	7	7
calhousing	20640	4	4
cpu-small	8192	6	5
fried	40769	9	5
glass	214	9	6
housing	506	6	6
iris	150	4	3
segment	2310	18	7
stock	950	5	5
vehicle	846	18	4
wine	178	13	3

maximizing agent. This setting allows for varying dataset properties, i.e., the number of instances, features, and labels, to study specific aspects of our novel approach in a more controlled setting.

We consider the following evaluation measures: Kendall's tau and Spearman's rho rank correlation coefficients, 0/1 loss, and Hamming distance (number of items for which the predicted rank deviates from the true rank). For coherence, we turn the last two into $[0, 1]$-valued similarity scores: Match is defined as 1 minus 0/1 loss, and Hamming similarity is 1 minus normalized Hamming distance.

For the first study, the empirical results are computed as in [4] using five repetitions of ten-fold cross-validation. In the second study, we averaged the results over 100 repetitions of the synthetic data generation process for each dataset property configuration with separate test sets consisting of 1000 examples.

As base learners, we use the decision tree (J48) and random forest (Random Forest) implementations from the Weka machine learning suite with default parameters [11]. Both methods combine fast training and testing with a natural means of handling multiclass problems without using any further decomposition techniques. This is in contrast to learning methods that make use of reduction techniques themselves (multiclass SVM, for example, uses a decomposition into one-vs-rest or all pairs). By excluding such methods, we try to avoid blending multiclass to binary and label ranking to multiclass decomposition together, which may yield empirical results that are difficult to interpret.

Table 2. Experimental results of the label ranking techniques with decision trees (J48) as base learner in terms of Kendall's tau (in brackets the ranks; best average rank in boldface).

Dataset	RPC	LR2MC-2	LR2MC-3	LR2MC-4	LR2MC-5	LR2MC-6	LR2MC-7
authorship	0.787 (2)	0.789 (1)	0.782 (3)	0.771 (4)			
bodyfat	0.153 (1)	0.143 (2)	0.106 (3)	0.097 (4)	0.073 (5)	0.046 (7)	0.051 (6)
calhousing	0.299 (1)	0.297 (2)	0.206 (3)	0.191 (4)			
cpu-small	0.311 (2)	0.311 (1)	0.265 (3)	0.242 (4)	0.224 (5)		
fried	0.808 (4)	0.814 (3)	0.862 (1)	0.836 (2)	0.749 (5)		
glass	0.801 (5)	0.818 (3)	0.830 (1)	0.826 (2)	0.812 (4)	0.790 (6)	
housing	0.310 (6)	0.333 (4)	0.402 (1)	0.375 (2)	0.341 (3)	0.313 (5)	
iris	0.780 (1)	0.765 (3)	0.777 (2)				
segment	0.830 (7)	0.839 (5)	0.887 (2)	0.888 (1)	0.869 (3)	0.854 (4)	0.832 (6)
stock	0.729 (4)	0.727 (5)	0.781 (1)	0.768 (2)	0.755 (3)		
vehicle	0.815 (2)	0.819 (1)	0.815 (3)	0.795 (4)			
wine	0.862 (2)	0.863 (1)	0.838 (3)				
Average rank	3.08	2.58	**2.17**	2.90	4.00	5.50	6.00

6.2 First Study (Real Data)

Results for the first study are shown in Table 2 for Kendall's tau as performance metric and decision trees as base learner—corresponding results for other metrics and learners can be found in the supplementary material. Note that the rank statistic shown in that table (ranking of methods per dataset, and average ranks per method) is arguably biased, because not all decompositions are applicable to all datasets (for example, LR2MC-5 cannot be used for the 4-label authorship dataset).

Therefore, Table 3 shows average *normalized* ranks, where normalization means that, for each dataset, the ranks are linearly rescaled to have unit sum. As can be seen, none of the approaches consistently outperforms the others. However, a more fine-grained, bipartite analysis of the results demonstrates the dependence of the optimal choice of k on the overall number of labels m (see Corollary 1 and the subsequent theoretical discussion): The pairwise approach outperforms all other methods for datasets with *up to 4 labels* (middle entry in each triplet), while the LR2MC-3 approach outperforms all other methods for datasets with *more than 4 labels* (right entry).

The results for RPC [18] and LR2MC-2 are similar, as expected, since both methods use the same decomposition and only differ in the aggregation step. RPC uses a voting approach that is specifically tailored for Spearman's rho, which in turn bounds Kendall's tau [7]. For these metrics, RPC is indeed very strong.

In agreement with our theoretical arguments, the situation looks different when measuring performance in terms of Match and Hamming. Here, the pairwise approach performs worse, and best results are obtained for LR2MC with

$k = 4$ or $k = 5$. As already explained, pairwise information is not enough to optimize these measures. Theoretically, one may even expect larger values for the optimal k, but practically, classifiers are of course difficult to train on problems with too many classes (and limited data).

6.3 Second Study (Synthetic Data)

In a second study, we carried out a set of experiments using the synthetic setting of [13], which allows for controlling dataset properties. More precisely, in a first scenario, we considered the following setup: $250, 500, 1000, \ldots, 16000$ training instances with 10 features and 7 labels, Match and Hamming as evaluation measures.

Table 3. Experimental results in terms of average *normalized* ranks (best result for each metric/learner combination in boldface) for different combinations of measure (Kendall, Spearman, Match, Hamming) and base learner (J48 and Random Forest): all datasets/datasets with $m \leq 4$ / datasets with $m > 4$.

		RPC	LR2MC-2	LR2MC-3	LR2MC-4	LR2MC-5	LR2MC-6	LR2MC-7
K	J48	.21/**.20**/.21	**.19**/.21/.17	.20/.35/**.09**	.21/.40/.13	.21/—/.21	.23/—/.23	.21/—/.21
K	RF	.20/**.20**/.19	.22/.31/.15	**.17**/.27/**.09**	.20/.37/.13	.23/—/.23	.23/—/.23	.25/—/.25
S	J48	**.15/.15**/.16	.22/.27/.18	.19/.35/**.08**	.22/.40/.15	.22/—/.22	.25/—/.25	.23/—/.23
S	RF	**.15/.18**/.13	.22/.29/.18	.17/.29/**.07**	.22/.40/.15	.26/—/.26	.24/—/.24	.25/—/.25
M	J48	.31/.33/.29	.28/.35/.23	.20/.24/.18	.11/**.13**/.10	**.10**/—/**.10**	.13/—/.13	.11/—/.11
M	RF	.33/.40/.29	.29/.35/.24	.16/.17/.16	.11/**.13**/.10	**.09**/—/**.09**	.13/—/.13	.17/—/.17
H	J48	.29/.32/.27	.29/.37/.23	.19/.23/.16	**.11/.13/.09**	.10/—/.10	.16/—/.16	.16/—/.16
H	RF	.29/.32/.27	.32/.40/.27	.19/.22/.17	**.09/.10/.09**	.10/—/.10	.13/—/.13	.14/—/.14

The experimental results with decision tree base learners are given in Table 4 (detailed results for random forests are available as supplementary material). Here, the ranks are computed *without* normalization in compliance with [4]. Since only the overall number of training examples varies, all considered methods are applicable, for which reason a normalization is not needed.

As expected, the absolute accuracy increases with the number of training examples for all methods. Moreover, the relative performance (ranks) of the methods is identical, regardless of the training set size, except for RPC and LR2MC-2 which achieve very similar results. With decision trees as base learners, LR2MC-4 consistently outperforms all other approaches, including LR2MC-7. This observation again supports the hypothesis that pure classification methods that do not leverage the structure of the output space are more difficult to train. For random forests as base learner, the experimental results are in line with these observations, with the slight difference that LR2MC-3 is the overall winner and a bit better than LR2MC-2 and LR2MC-4, which yield very similar absolute results.

Table 4. Experimental results using decision trees (J48) as base learners in terms of Match (top) and Hamming (bottom), in parentheses the ranks.

Examples	RPC	LR2MC-2	LR2MC-3	LR2MC-4	LR2MC-5	LR2MC-6	LR2MC-7
250	0.128 (4)	0.127 (5)	0.157 (2)	0.165 (1)	0.151 (3)	0.111 (6)	0.081 (7)
500	0.151 (4)	0.149 (5)	0.188 (2)	0.199 (1)	0.178 (3)	0.134 (6)	0.096 (7)
1000	0.176 (4)	0.174 (5)	0.216 (2)	0.231 (1)	0.209 (3)	0.155 (6)	0.111 (7)
2000	0.197 (5)	0.197 (4)	0.246 (2)	0.263 (1)	0.239 (3)	0.178 (6)	0.128 (7)
4000	0.219 (4)	0.216 (5)	0.277 (2)	0.294 (1)	0.269 (3)	0.203 (6)	0.147 (7)
8000	0.243 (5)	0.244 (4)	0.306 (2)	0.323 (1)	0.298 (3)	0.230 (6)	0.166 (7)
16000	0.264 (5)	0.264 (4)	0.337 (2)	0.353 (1)	0.328 (3)	0.256 (6)	0.186 (7)
Average rank	4.43	4.57	2.00	**1.00**	3.00	6.00	7.00
250	0.567 (4)	0.566 (5)	0.595 (2)	0.599 (1)	0.573 (3)	0.514 (6)	0.468 (7)
500	0.591 (5)	0.592 (4)	0.627 (2)	0.629 (1)	0.604 (3)	0.541 (6)	0.492 (7)
1000	0.614 (4)	0.613 (5)	0.652 (2)	0.655 (1)	0.630 (3)	0.568 (6)	0.518 (7)
2000	0.634 (5)	0.635 (4)	0.674 (2)	0.679 (1)	0.653 (3)	0.592 (6)	0.538 (7)
4000	0.655 (5)	0.655 (4)	0.696 (2)	0.700 (1)	0.675 (3)	0.615 (6)	0.560 (7)
8000	0.672 (4)	0.672 (5)	0.715 (2)	0.721 (1)	0.697 (3)	0.635 (6)	0.580 (7)
16000	0.690 (4)	0.689 (5)	0.733 (2)	0.740 (1)	0.717 (3)	0.658 (6)	0.601 (7)
Average rank	4.43	4.57	2.00	**1.00**	3.00	6.00	7.00

In a second scenario, we used the following setting: 2000 training instances with 10 features and $3, 4, \ldots, 7$ labels, Kendall and Spearman as evaluation measures. The experimental results are given in Table 5. This set of experiments completely supports the finding from the first study: For datasets with $m \leq 4$ labels, the pairwise approaches RPC and LR2MC-2 outperform all other decompositions. However, for datasets with $m > 4$ labels, LR2MC-3 consistently achieves the highest accuracy in terms of both Kendall's tau and Spearman's rho.

6.4 Discussion

Our experimental results suggest that the number of labels is an essential property for choosing a suitable decomposition approach in label ranking. More precisely, while standard pairwise decomposition appears to be favorable for datasets with $m \leq 4$ labels, our novel multiclass decomposition with $k = 3$ (LR2MC-3) seems to provide a promising alternative for $m > 4$ labels and rank correlation (Kendall's tau or Spearman's rho) as performance measure. If other measures are used, such as 0/1 loss (Match) or Hamming, higher order decompositions are even more advantageous. Interestingly, in the context of multilabel learning, the experimental conclusions for the conceptually related Ra*k*el method [21] are similar, since label set sizes around $k = 3$ often yield the best accuracy.

As our experimental evaluation suggests that multiclass decomposition with $k = 3$ is a particularly promising alternative to the well-known pairwise label ranking approach, we will add some remarks regarding its computational complexity in terms of the number of classification problems: For a dataset with m

Table 5. Experimental results for the second controlled scenario in terms of *normalized* ranks (best result for each metric/learner combination in boldface) for different combinations of measure (Kendall, Spearman, Match, Hamming) and base learner (J48 and Random Forest): all datasets/datasets with $m \leq 4$/datasets with $m > 4$.

		RPC	LR2MC-2	LR2MC-3	LR2MC-4	LR2MC-5	LR2MC-6	LR2MC-7
K	J48	**.18**/.18	.22/.15	.40/**.05**	.40/.14	—/.23	—/.25	—/.25
K	RF	**.13**/.12	.27/.14	.40/**.05**	.40/.19	—/.25	—/.25	—/.25
S	J48	**.13**/.13	.27/.18	.40/**.05**	.40/.14	—/.25	—/.25	—/.25
S	RF	.22/.08	**.18**/.15	.40/**.07**	.40/.20	—/.25	—/.25	—/.25

labels, pairwise label ranking considers an overall number of $\binom{m}{2}$ binary classification problems, while LR2MC-3 makes use of a decomposition into $\binom{m}{3}$ multiclass problems. Hence, ignoring special cases, there is an increase in the number of classification problems by a factor of $\frac{m-2}{3}$. For example, for $m = 7$ this amounts to roughly 67% more classification problems with the training set size being identical for both approaches. Depending on the application field, the increase in accuracy may justify these computational requirements.

One may hypothesize that the increase in accuracy may be attributed to the increased number of underlying classifiers, and hence may be a straightforward ensemble size consequence. We have conducted some preliminary experiments to further investigate this hypothesis. In these experiments, a maximum number of $\binom{m}{2}$ classifiers was subsampled from the overall set of multiclass classifiers to remove any potential ensemble advantage over the pairwise setting. Interestingly, the experimental results when using the limited number of underlying classifiers are comparable to those of the overall set of classifiers (see supplementary material for some more detailed results), suggesting that ensemble size cannot explain the observed difference in accuracy.

7 Conclusion and Future Research

Accepting pairwise learning as a state-of-the-art approach to label ranking, the major objective of this work was to address the question whether, in principle, going beyond pairwise comparisons and decomposing preference information into partial rankings of length $k > 2$ could be useful. Technically, this question comes down to comparing a reduction to binary classification with a reduction to multiclass classification.

The answer we can give is clearly affirmative: Our method, called LR2MC, tends to be superior to pairwise learning as soon as the number of labels exceeds four. In agreement with our theoretical arguments, this superiority is especially pronounced for performance metrics that are difficult to optimize based on pairwise preference information. Practically, $k = 3$ or $k = 4$ seem to be reasonable choices, which optimally balance various factors responsible for the success of the learning process. Such factors include the inherent loss of information caused by decomposition, the redundancy of the reduction, and the practical difficulty of the individual classification problems created.

While this is an important insight into the nature of label ranking, which paves the way for new methods beyond pairwise preference learning, the increased computational complexity of LR2MC is clearly an issue. In future work, we will therefore elaborate on various ways to reduce complexity and increase efficiency, both in the decomposition and aggregation step. One may think, for example, of incomplete decompositions that do not comprise projections to all label subsets, or mixed decompositions including rankings of different length (instead of using a fixed k). Moreover, the efficiency of the aggregation step could be increased by computationally less complex aggregation techniques, such as the well-known Borda-count. Preliminary experiments suggest that it is indeed possible to substantially reduce the computational complexity for aggregation while still benefit from superior ranking accuracy for datasets with $m > 4$ labels.

Currently, our approach is limited to complete label rankings as training data. Therefore, another direction of future research is to develop a generalization that allows for incorporating partial ranking data. For example, we may generalize LR2MC by inserting a preprocessing step for training examples which are associated with a partial ranking only, and consider all compatible ranking extensions as (weighted) virtual training examples in the decomposition process.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. **1**, 113–141 (2000)
2. Brinker, K., Hüllermeier, E.: Case-based label ranking. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 566–573. Springer, Heidelberg (2006). https://doi.org/10.1007/11871842_53
3. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine learning, ICML, pp. 129–136 (2007)
4. Cheng, W., Dembczyński, K., Hüllermeier, E.: Label ranking methods based on the Plackett-Luce model. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, pp. 215–222. Omnipress, June 2010
5. Cheng, W., Hühn, J., Hüllermeier, E.: Decision tree and instance-based learning for label ranking. In: Bottou, L., Littman, M. (eds.) Proceedings of the 26th International Conference on Machine Learning (ICML-09), Montreal, Canada, pp. 161–168. Omnipress, June 2009
6. Cheng, W., Hüllermeier, E.: A new instance-based label ranking approach using the mallows model. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009. LNCS, vol. 5551, pp. 707–716. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01507-6_80
7. Coppersmith, D., Fleischer, L.K., Rurda, A.: Ordering by weighted number of wins gives a good ranking for weighted tournaments. ACM Trans. Algorithms **6**(3), 55:1–55:13 (2010)

8. Dekel, O., Manning, C.D., Singer, Y.: Log-linear models for label ranking. In: Thrun, S., Saul, L.K., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16 (NIPS 2003), pp. 497–504. MIT Press, Cambridge (2004)
9. Dembczynski, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence and loss minimization in multi-label classification. Mach. Learn. **88**(1–2), 5–45 (2012)
10. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**, 263–286 (1995)
11. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Fourth edn. (2016)
12. Fürnkranz, J., Hüllermeier, E.: Preference learning: an introduction. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 1–18. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-14125-6_1
13. Fürnkranz, J., Hüllermeier, E.: Pairwise preference learning and ranking. In: Lavrač, N., Gamberger, D., Blockeel, H., Todorovski, L. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 145–156. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39857-8_15
14. Hamm, J., Belkin, M.: Probabilistic zero-shot classification with semantic rankings. CoRR abs/1502.08039 (2015). http://arxiv.org/abs/1502.08039
15. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification: a new approach to multiclass classification and ranking. In: Advances in Neural Information Processing Systems 15 (NIPS 2002) (2002)
16. Hüllermeier, E., Fürnkranz, J.: On predictive accuracy and risk minimization in pairwise label ranking. J. Comput. Syst. Sci. **76**(1), 49–62 (2010)
17. Hüllermeier, E., Fürnkranz, J.: Comparison of ranking procedures in pairwise preference learning. In: 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-04), pp. 535–542 (2004)
18. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artif. Intell. **172**(16–17), 1897–1916 (2008)
19. Liu, T.Y.: Learning to Rank for Information Retrieval. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-14267-3
20. Shalev-Shwartz, S., Singer, Y.: Efficient learning of label ranking by soft projections onto polyhedra. J. Mach. Learn. Res. **7**, 1567–1599 (2006)
21. Tsoumakas, G., Katakis, I., Ioannis, V.: Random *k*-labelsets for multilabel classification. IEEE Trans. Knowl. Data Eng. **23**(7), 1079–1089 (2011)
22. Zhou, Y., Lui, Y., Yang, J., He, X., Liu, L.: A taxonomy of label ranking algorithms. J. Comput. **9**(3), 557–565 (2014)