A Nearest Neighbor Approach to Label Ranking based on Generalized Labelwise Loss Minimization

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

Abstract

In this paper, we introduce a new (meta) learning technique for a preference learning problem called *label ranking*. As opposed to existing meta techniques, which mostly decompose the original problem into *pairwise* comparisons, our approach relies on a *labelwise* decomposition. The basic idea is to train one model per class label, namely a model that maps instances to ranks. We propose a concrete instantiation of this approach by choosing nearest neighbor estimation as a base learner. In an experimental study, we show that this approach is quite competitive to state-of-the-art methods.

1 Introduction

Preference learning is an emerging subfield of machine learning, which deals with the induction of preference models from observed or revealed preference information [7]. Such models are typically used for prediction purposes, for example, to predict context-dependent preferences of individuals on various choice alternatives. Depending on the representation of preferences, individuals, alternatives, and contexts, a large variety of preference models are conceivable, and many such models have already been studied in the literature.

A specific type of preference learning problem is the problem of *label ranking*, namely the problem of learning a model that maps instances to rankings (total orders) over a finite set of predefined alternatives (labels). Several methods for label ranking have already been proposed in the literature [14]. Most of these methods are *reduction techniques* transforming the original learning task into one or several binary classification tasks. Moreover, all existing methods are *relational* in so far as they seek to learn from *relative* or *comparative* preferences, such as pairwise comparisons between alternatives [12]. Since a ranking of alternatives, by its very nature, does indeed inform about *relative* and not about *absolute* preferences, the prevalence of the relational approach is of course completely understandable.

On the other hand, since the number of alternatives in a label ranking problem is fixed, a ranking is uniquely defined by the position (rank) of each of the alternatives, which can be seen as *absolute* preference information. Admittedly, as will be explained in more detail later on, this positional information is not always readily available for training. Yet, it is arguably a bit surprising that, to the best of our knowledge, an approach focused on the learning and prediction of absolute preferences has not even been tried so far.

In this paper, we introduce an approach of that kind, namely a new meta-learning technique for label ranking, which is based on a *labelwise* instead of a *pairwise* decomposition. The basic idea is to train one model per class label, namely a model that maps instances to ranks. In other words, given a new query instance, the idea is to predict the rank of each individual label right away. Unlike existing decomposition techniques, in which the reducts are binary classification problems, this approach leads to a linear number of ordered multi-class problems.

The paper is organized as follows. The next section provides some background of the label ranking problem, and Section 3 reviews existing methods for tackling this problem. Our new approach based on labelwise decomposition (LWD) is introduced in Section 4. Section 5 is devoted to a general discussion of similarities and differences between reduction techniques for label ranking. In Section 6, we provide an experimental study, in which LWD is compared with existing decomposition techniques in a systematic way. The paper end with some concluding remarks in Section 7.

2 Label Ranking

Let $\mathcal{Y} = \{y_1, \ldots, y_K\}$ be a finite set of (choice) alternatives; adhering to the terminology commonly used in supervised machine learning, and accounting for the fact that label ranking can be seen as an extension of multi-class classification, the y_i are also called *class labels*. We consider total order relations $\succ$ on $\mathcal{Y}$, that is, complete, transitive, and antisymmetric relations, where $y_i \succ y_j$ indicates that y_i precedes y_j in the order. Since a ranking can be seen as a special type of preference relation, we shall also say that $y_i \succ y_j$ indicates a preference for y_i over y_j.

Formally, a total order $\succ$ can be identified with a permutation $\bar{\pi}$ of the set $[K] = \{1, \ldots, K\}$, such that $\bar{\pi}(i)$ is the position of y_i in the order. We denote the class of permutations of $[K]$ (the symmetric group of order K) by $\mathbb{S}_K$. By abuse of terminology, though justified in light of the above one-to-one correspondence, we refer to elements $\bar{\pi} \in \mathbb{S}_K$ as both permutations and rankings.

In the setting of label ranking, preferences on $\mathcal{Y}$ are "contextualized" by instances $\boldsymbol{x} \in \mathbb{X}$, where $\mathbb{X}$ is an underlying instance space. Thus, each instance $\boldsymbol{x}$ is associated with a ranking $\succ_{\boldsymbol{x}}$ of the label set $\mathcal{Y}$ or, equivalently, a permutation $\bar{\pi}_{\boldsymbol{x}} \in \mathbb{S}_K$. More specifically, since label rankings do not necessarily depend on instances in a deterministic way, each instance $\boldsymbol{x}$ is associated with a probability distribution $\mathbf{P}(\cdot \,|\, \boldsymbol{x})$ on $\mathbb{S}_K$. Thus, for each $\bar{\pi} \in \mathbb{S}_K$, $\mathbf{P}(\bar{\pi} \,|\, \boldsymbol{x})$ denotes the probability to observe the ranking $\bar{\pi}$ in the context specified by $\boldsymbol{x}$.

As an illustration, suppose $\mathbb{X}$ is the set of people characterized by attributes such as sex, age, profession, and marital status, and labels are music genres: $\mathcal{Y} = \{\texttt{Rock}, \texttt{Pop}, \texttt{Classic}, \texttt{Jazz}\}$. Then, for $\boldsymbol{x} = (m, 30, \text{teacher}, \text{married})$ and $\bar{\pi} = (2, 1, 4, 3)$, $\mathbf{P}(\bar{\pi} \,|\, \boldsymbol{x})$ denotes the probability that a 30 years old married man, who is a teacher, prefers Pop music to Rock to Jazz to Classic.

2.1 The Label Ranking Problem

The goal in label ranking is to learn a "label ranker", that is, a model

$$\mathcal{M} : \mathbb{X} \longrightarrow \mathbb{S}_K$$

that predicts a ranking $\hat{\pi}$ for each instance $\boldsymbol{x}$ given as an input. More specifically, seeking a model with optimal prediction performance, the goal is to find a risk (expected loss) minimizer

$$\mathcal{M}^* \in \operatorname*{argmin}_{\mathcal{M} \in \mathbf{M}} \int_{\mathbb{X} \times \mathbb{S}_K} D(\mathcal{M}(\boldsymbol{x}), \bar{\pi}) \, d\mathbf{P} \;\; ,$$

where $\mathbf{M}$ is the underlying model class, $\mathbf{P}$ is the joint measure $\mathbf{P}(\boldsymbol{x}, \bar{\pi}) = \mathbf{P}(\boldsymbol{x})\mathbf{P}(\bar{\pi} \,|\, \boldsymbol{x})$ on $\mathbb{X} \times \mathbb{S}_K$ and D is a loss function on $\mathbb{S}_K$; common choices of D will be introduced below.

As training data $\mathbb{D}$, a label ranker uses a set of instances $\boldsymbol{x}_n$ ($n \in [N]$), together with information about the associated rankings π_n. Ideally, complete rankings are given as training information, i.e., a single observation is a tuple of the form $(\boldsymbol{x}_n, \pi_n) \in \mathbb{X} \times \mathbb{S}_K$; we call an observation of that kind a *complete* example. From a practical point of view, however, it is important to allow for incomplete information in the form of a ranking of some but not all of the labels in $\mathcal{Y}$:

$$y_{\tau(1)} \succ_{\boldsymbol{x}} y_{\tau(2)} \succ_{\boldsymbol{x}} \cdots \succ_{\boldsymbol{x}} y_{\tau(J)} \;\; , \tag{1}$$

where $J < K$ and $\{\tau(1), \ldots, \tau(J)\} \subset [K]$. For example, for an instance $\boldsymbol{x}$, it might be known that $y_2 \succ_{\boldsymbol{x}} y_1 \succ_{\boldsymbol{x}} y_5$, while no preference information is given about the labels y_3 or y_4.

In the following, we will write complete rankings $\bar{\pi}$ with an upper bar (as we already did above). If a ranking π is not complete, then $\pi(j)$ is the position of y_j in the incomplete ranking, provided this label is contained, and $\pi(j) = 0$ otherwise; thus, if $\bar{\pi}$ is a "completion" of π, then $\bar{\pi}(k) \geq \pi(k)$ for all $k \in [K]$. In the above example (1), $\pi = (2, 1, 0, 0, 3)$. We denote by $|\pi| = \{j \,|\, \pi(j) > 0\}$ the size of the ranking; thus, π is complete if $|\pi| = K$.

2.2 Prediction Accuracy

The prediction accuracy of a label ranker is assessed by comparing the true ranking $\bar{\pi}$ with the prediction $\hat{\pi}$, using a distance measure D on rankings. Among the most commonly used measures is the Kendall distance, which is defined by the number of inversions, that is, index pairs $\{i, j\} \subset [K]$ such that the order of y_i and y_j in $\bar{\pi}$ is inverted in $\hat{\pi}$:

$$D(\bar{\pi}, \hat{\pi}) = \sum_{1 \leq i < j \leq K} [\![\operatorname{sign}(\bar{\pi}(i) - \bar{\pi}(j)) \neq \operatorname{sign}(\hat{\pi}(i) - \hat{\pi}(j))]\!] \tag{2}$$

The well-known Kendall rank correlation measure is an affine transformation of (2) to the range $[-1, +1]$. Besides, the sum of L_1 or L_2 losses on the ranks of the individual labels are often used as an alternative distance measures:

$$D_1(\bar{\pi}, \hat{\pi}) = \sum_{i=1}^{M} |\bar{\pi}(i) - \hat{\pi}(i)| \tag{3}$$

$$D_2(\bar{\pi}, \hat{\pi}) = \sum_{i=1}^{M} (\bar{\pi}(i) - \hat{\pi}(i))^2 \tag{4}$$

These measures are closely connected with two other well-known rank correlation measures: Spearman's footrule is an affine transformation of (3) to the interval $[-1, +1]$, and Spearman's rank correlation (Spearman's rho) is such a transformation of (4).

2.3 Label Ranking Methods

Several methods for label ranking have been proposed that try to exploit, in one way or the other, the structure of the output space $\mathbb{S}_K$. These include generalizations of standard machine learning methods such as nearest neighbor estimation [2] and decision tree learning [5], as well as statistical inference based on parametrized models of rank data [4]. Moreover, several *reduction techniques* have been proposed, that is, meta-learning techniques that reduce the original label ranking problem into one or several classification problems that are easier to solve [9; 12].

3 Labelwise Decomposition

In this section, we introduce a new meta-learning technique for label ranking, which is based on the idea of reducing the original problem to standard classification problems in a *labelwise* manner.

3.1 Complete Training Information

If the training data $\mathbb{D}$ consists of complete examples $(\boldsymbol{x}_n, \bar{\pi}_n)$, then each such example informs about the rank $\bar{\pi}(k)$ of the label y_k in the ranking associated with $\boldsymbol{x}_n$. Thus, a quite natural idea is to learn a model

$$\mathcal{M}_k : \mathbb{X} \longrightarrow [K]$$

that predicts the rank of y_k, given an instance $\boldsymbol{x} \in \mathbb{X}$ as an input. Indeed, such a model can be trained easily on the data

$$\mathbb{D}_k = \Big\{ (\boldsymbol{x}_n, r_n) \,|\, (\boldsymbol{x}_n, \bar{\pi}_n) \in \mathbb{D},\, r_n = \bar{\pi}_n(k) \Big\}. \tag{5}$$

The classification problem thus produced are multi-class problems with K classes, where each class corresponds to a possible rank. More specifically, since these ranks have a natural order, we are facing an *ordinal classification* problem. Thus, training of the models $\mathcal{M}_k$ ($k \in [K]$) can in principle be accomplished by any existing method for ordinal classification.

3.2 Incomplete Training Information

As mentioned before, the original training data $\mathbb{D}$ is not necessarily supposed to contain complete rank information; instead, for a training instance $\boldsymbol{x}_n$, only an incomplete ranking π_n of a subset of the labels in $\mathcal{Y}$ might have been observed, while the complete ranking $\bar{\pi}_n$ is not given. In this case, the above method is not directly applicable: If at least one label is missing, i.e., $|\pi_n| < K$, then none of the true ranks $\bar{\pi}_n(k)$ is precisely known; consequently, the training data (5) cannot be constructed.

Nevertheless, even in the case of incomplete rankings, non-trivial information can be derived about the rank $\bar{\pi}(k)$ for at least some of the labels y_k. In fact, if $|\pi| = J$ and $\pi(k) = r > 0$, then

$$\bar{\pi}(k) \in \big\{r, r+1, \ldots, r+K-J\big\} \ .$$

Of course, if $\pi(k) = 0$ (i.e., y_k is not present in the ranking), only the trivial information $\bar{\pi}(k) \in [K]$ can be derived. Yet, more precise information can be obtained under additional assumptions. For example, if π is known to be the top of the ranking $\bar{\pi}$, then

$$\begin{cases} \bar{\pi}(k) = \pi(k) & \text{if } \pi(k) > 0 \\ \bar{\pi}(k) \in \{J+1, \ldots, K\} & \text{if } \pi(k) = 0 \end{cases} \ . \tag{6}$$

This scenario is highly relevant, since top-ranks are observed in many practical applications.

In general, the type of training data that can be derived for a label y_k in the case of incomplete rank information are examples of the form

$$\big(\boldsymbol{x}_n, R_n\big) \in \mathbb{X} \times 2^{[K]} \ , \tag{7}$$

that is, an instance $\boldsymbol{x}_n$ together with a set of possible ranks R_n. The problem of learning from data with *imprecise* class information has recently been studied in the literature, where it is called learning from *ambiguously labeled examples* [11] or learning from *partial labels* [8; 6].

3.3 Generalized Nearest Neighbor Estimation

As explained in [10], a reasonable approach to learning from imprecise data is to combine model identification and *data disambiguation*, that is, trying to fit an optimal model while simultaneously finding the "true data". Concretely, this can be accomplished by means of generalized loss functions, which, roughly speaking, compare a (point) prediction with a set of possible "true" values in an optimistic way. In our case, a loss function of that kind is of the form

$$L(R, \hat{r}) = \min_{r \in R} \ell(\hat{r}, r) \ , \tag{8}$$

where $R \subseteq [K]$ is a set of ranks, $\hat{r}$ is the predicted rank, and $\ell : [K]^2 \to \mathbb{R}$ is the original loss (comparing predicted and true ranks). An example for $\ell(\hat{r}, r) = |\hat{r} - r|$, i.e.,

$$L(R, \hat{r}) = \min_{r \in R} |\hat{r} - r| \ , \tag{9}$$

is shown in Figure 1. As can be seen, the generalized loss is 0 as long as $\hat{r} \in R$, that is, as long as $\hat{r}$ possibly corresponds to the true rank.

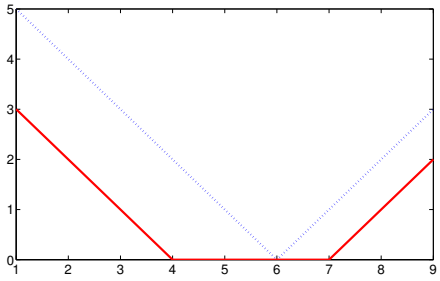

Figure 1: Loss function $\ell(\hat{r}, r) = |\hat{r} - r|$ for $r = 6$ (dashed line) and generalized version (9) for $R = \{4, 5, 6, 7\}$.

Now, consider a nearest neighbor approach to label ranking: Given a new query instance $\boldsymbol{x}_0$, a prediction $\hat{\pi}$ is obtained by combining the (incomplete) rankings $\pi_1, \ldots, \pi_{nn}$ coming from the nn nearest neighbors of $\boldsymbol{x}_0$ in the training data $\mathbb{D}$. Denote by $R_{k,n}$ $(k \in [K], n \in [nn])$ the (possibly imprecise) rank information for label y_k provided by π_n. Moreover, consider a loss function D on $\mathbb{S}_K$ that is labelwise decomposable, i.e., which can be written in the form

$$D(\bar{\pi}, \hat{\pi}) = \sum_{k=1}^{K} \ell(\bar{\pi}(k), \hat{\pi}(k)).$$

Obviously, the L_1 and L_2 loss (3) and (4) are both of this type. Then, the empirical risk of $\hat{\pi}$, i.e., the loss of this prediction in the neighborhood of $\boldsymbol{x}_0$, is given by

$$\sum_{n=1}^{nn} D(\bar{\pi}_n, \hat{\pi}) = \sum_{n=1}^{nn} \sum_{k=1}^{K} \ell(\bar{\pi}_n(k), \hat{\pi}(k)) \tag{10}$$

$$= \sum_{k=1}^{K} \sum_{n=1}^{nn} \ell(\bar{\pi}_n(k), \hat{\pi}(k)) \tag{11}$$

$$= \sum_{k=1}^{K} L_k(\bar{\pi}_n(k), \hat{\pi}(k)), \tag{12}$$

where $L_k(r)$ is the cost of putting label y_k on position r. Taking into account that in general only incomplete rankings π_n are observed, the loss $\ell(\cdot)$ should be replaced by the generalized loss (8) and, therefore, L_k should be defined as

$$L_k(r) = \sum_{n=1}^{nn} L(R_{k,n}, r) \ .$$

Thus, an optimal solution would consists of assigning y_k the position $\hat{\pi}(k) = r$ for which $L_k(r)$ is minimal. However, noting that each position $r \in [K]$ must be assigned at most once, this approach is obviously not guaranteed to produce a feasible solution. Instead, the minimization of (10) requires the solution of an *optimal assignment problem* [3]:

- labels $y_k \in \mathcal{Y}$ must be uniquely assigned to ranks $r = \hat{\pi}(k) \in [K]$;
- assigning y_k to rank r causes a cost of $L_k(r)$;
- the goal is to minimize the sum of all assignment costs.

Assignment problems of that kind have been studied extensively in the literature, and efficient algorithms for their solution are available. The well-known Hungarian algorithm [13], for example, solves the above problem in time $O(K^3)$. Such algorithms can be used to produce a prediction $\hat{\pi}$ that minimizes

$$\sum_{k=1}^{K} L_k(\hat{\pi}(k)) \ ,$$

and therefore to realize our nearest neighbor approach to label ranking. In the next section, we experimentally analyze this approach with L given by (9).

4 Experiments

In this section, we experimentally compare our new method, referred to as LWD, with another nearest neighbor approach to label ranking. This approach is based on the (local) estimation of the parameters of a probabilistic model called the Plackett-Luce (PL) model [4]. It is known to achieve state-of-the-art performance, not only among the nearest neighbor approaches but among label ranking methods in general.

4.1 Data

We used several benchmark data sets for label ranking that have also been used in previous studies [12]; these are semi-synthetic data sets, namely label ranking versions of (real) UCI multi-class data. Moreover, we used two real label ranking data sets: The Sushi data[1] consists of 5000 instances (customers) described by 11 features, each one associated with a ranking of 10 types of sushis. The Students data [1] consists of 404 students (each characterized by 126 attributes) with associated rankings of five goals (want to get along with my parents, want to feel good about myself, want to have nice things, want to be different from others, want to be better than others). See Table 1 for a summary of the data.

Two missing label scenarios were simulated, namely a "missing-at-random" setting and the top-rank setting (6). In the first case, a biased coin is flipped for every label in a ranking to decide whether to keep or delete that label; the probability for a deletion is specified by a parameter $p \in [0, 1]$. Thus, $p \times 100\%$ of the labels will be missing on average. Similarly, in the second case, only the J top-labels in a ranking are kept, where J has a binomial distribution with parameters K and $1 - p$.

4.2 Results

The results in Tables 2 and 3 are presented as averages of 5×10-fold cross validation in terms of the Kendall correlation measure; other measures such as (3) and (4) led to similar results. The number of nearest neighbors (see column nn in Table 1) was determined through internal cross-validation using the PL method in the full ranking setting; the same number was then adopted for LWD.[2] As a distance measure on $\mathbb{X}$, the standard Euclidean distance was used.

[1] http://kamishima.new/sushi/

[2] Thus, any bias will be more in favor of PL than LWD.

Table 1: Properties of the data sets.

data set	# inst. (N)	# attr. (d)	# labels (K)	nn
authorship	841	70	4	10
glass	214	9	6	5
iris	150	4	3	5
pendigits	10992	16	10	10
segment	2310	18	7	5
vehicle	846	18	4	10
vowel	528	10	11	5
wine	178	13	3	10
sushi	5000	11	10	100
students	404	126	5	100

These results clearly support the conclusion that, while LWD and PL are quite en par in the complete ranking case, the latter is much more sensitive toward missing label information than the former. In fact, the performance of LWD is comparably stable, and its drop in performance due to missing label information is less pronounced than in the case of PL; this observation is especially clear in the missing-at-random setting, whereas the differences in performance are less visible in the top-rank setting (see Figures 2 and 3).

5 Summary and Conclusion

In this paper, we introduced labelwise decomposition (LWD) as a new meta-learning technique for label ranking, and realized this technique for the specific case of nearest neighbor estimation as an underlying base learner. In contrast to existing techniques, which are mostly based on decomposing training information into *comparative* preferences, this approach is based on *absolute* preference information in the form of ranks. The idea is quite simple: For each individual label, a model is learned that, given a query instance as an input, predicts the rank of the label in the associated ranking. Since these predictions need to guarantee that each rank is assigned exactly once, the individual predictions must be aggregated appropriately—as we have shown, the problem of finding an overall (empirical) risk minimizing prediction can be formalized as an optimal assignment problem.

Comparing LWD with a state-of-the-art nearest neighbor approach to label ranking, we found clear improvements in terms of prediction performance, notably in the case of missing label information.

Table 2: Performance in terms of Kendall's tau on synthetic data: missing-at-random (above) and top-rank setting (below).

	complete ranking		30% missing labels		60% missing labels	
	LWD	PL	LWD	PL	LWD	PL
authorship	.933±.016	.936±.015	.925±.018	.833±.030	.891±.021	.601±.054
glass	.840±.075	.841±.067	.819±.078	.669±.064	.721±.072	.395±.068
iris	.960±.036	.960±.036	.932±.051	.896±.069	.876±.068	.787±.111
pendigits	.940±.002	.939±.002	.924±.002	.770±.004	.709±.005	.434±.007
segment	.953±.006	.950±.005	.914±.009	.710±.013	.624±.020	.381±.020
vehicle	.853±.031	.859±.028	.836±.032	.753±.032	.767±.037	.520±.050
vowel	.876±.021	.851±.020	.821±.022	.612±.027	.536±.034	.327±.033
wine	.938±.050	.947±.047	.933±.054	.919±.059	.921±.062	.863±.094
authorship	.933±.016	.936±.015	.932±.017	.927±.017	.923±.015	.886±.022
glass	.840±.075	.841±.067	.838±.074	.809±.066	.815±.075	.675±.069
iris	.960±.036	.960±.036	.956±.036	.926±.051	.932±.048	.868±.070
pendigits	.940±.002	.939±.002	.933±.002	.918±.002	.837±.004	.794±.004
segment	.953±.006	.950±.005	.943±.005	.874±.008	.844±.010	.674±.015
vehicle	.853±.031	.859±.028	.851±.033	.838±.030	.818±.032	.765±.035
vowel	.876±.021	.851±.020	.867±.021	.785±.020	.800±.021	.588±.024
wine	.938±.050	.947±.047	.936±.049	.926±.061	.930±.059	.907±.066

Table 3: Performance in terms of Kendall's tau on real-world data: missing-at-random (above) and top-rank setting (below).

sushi	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.323±.012	.322±.011	.320±.011	.319±.010	.315±.011	.308±.011	.296±.011	.277±.010
PL	.321±.010	.320±.010	.318±.010	.311±.010	.298±.011	.278±.010	.246±.010	.203±.012
LWD	.325±.012	.324±.011	.324±.011	.323±.011	.323±.011	.323±.011	.321±.011	.316±.011
PL	.321±.010	.320±.010	.320±.011	.320±.011	.319±.010	.316±.010	.310±.010	.303±.011

students	0%	10%	20%	30%	40%	50%	60%	70%
LWD	.641±.051	.641±.051	.640±.050	.640±.051	.638±.052	.637±.051	.633±.054	.626±.055
PL	.386±.028	.384±.027	.382±.026	.377±.029	.365±.025	.350±.027	.327±.027	.274±.033
LWD	.641±.051	.641±.051	.641±.051	.641±.051	.640±.051	.640±.052	.638±.050	.628±.052
PL	.386±.028	.385±.028	.386±.028	.385±.027	.383±.029	.379±.026	.377±.026	.371±.028

References

[1] M. Boekaerts, K. Smit, and F.M.T.A. Busing. Salient goals direct and energise students' actions in the classroom. *Applied Psychology: An International Review*, 4(S1):520–539, 2012.

[2] K. Brinker and E. Hüllermeier. Case-based label ranking. In *Proceedings ECML–06, 17th European Conference on Machine Learning*, pages 566–573, Berlin, September 2006. Springer-Verlag.

[3] R.E. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. SIAM, 2009.

[4] W. Cheng, K. Dembczynski, and E. Hüllermeier. Label ranking based on the Plackett-Luce model. In J. Fürnkranz and T. Joachims, editors, *Proceedings ICML–2010, International Conference on Machine Learning*, Haifa, Israel, 2010.

[5] W. Cheng, J. Hühn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings ICML–2009, 26th International Conference on Machine Learning*, Montreal, Canada, 2009. [27% acceptance rate].

[6] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

[7] J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2011.

[8] Y. Grandvalet. Logistic regression for partial labels. In *IPMU–02, Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1935–1941, Annecy, France, 2002.

[9] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS-02)*, pages 785–792, 2003.

[10] E. Hüllermeier. Learning from imprecise and fuzzy data: Data disambiguation through generalized loss minimization. In revision.

[11] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–440, 2006.

[12] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

[13] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

[14] S. Vembu and T. Gärtner. Label ranking: a survey. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*. Springer-Verlag, 2010.

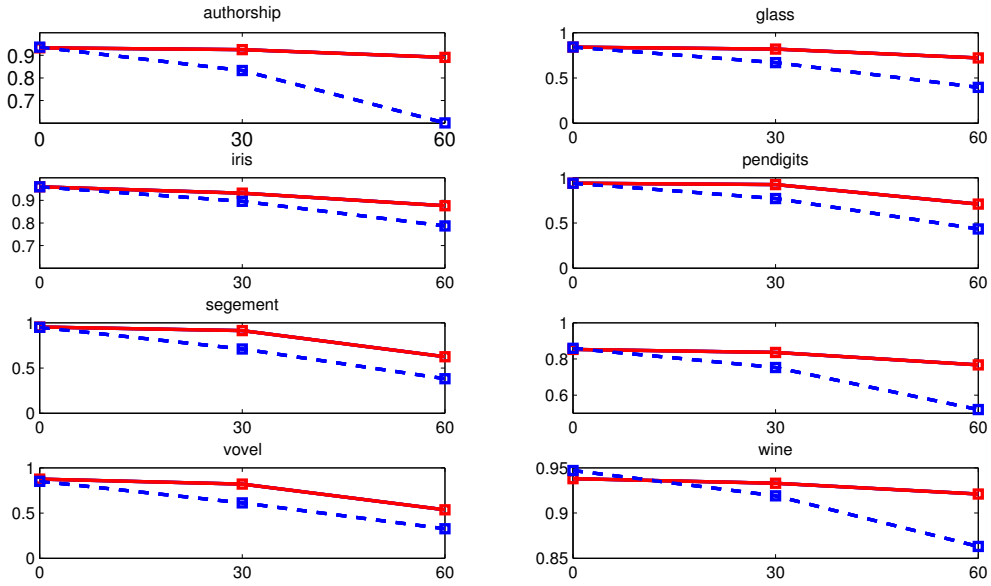

Figure 2: Performance of LWD (solid lines) and PL (dashed line) in the missing-at-random setting.

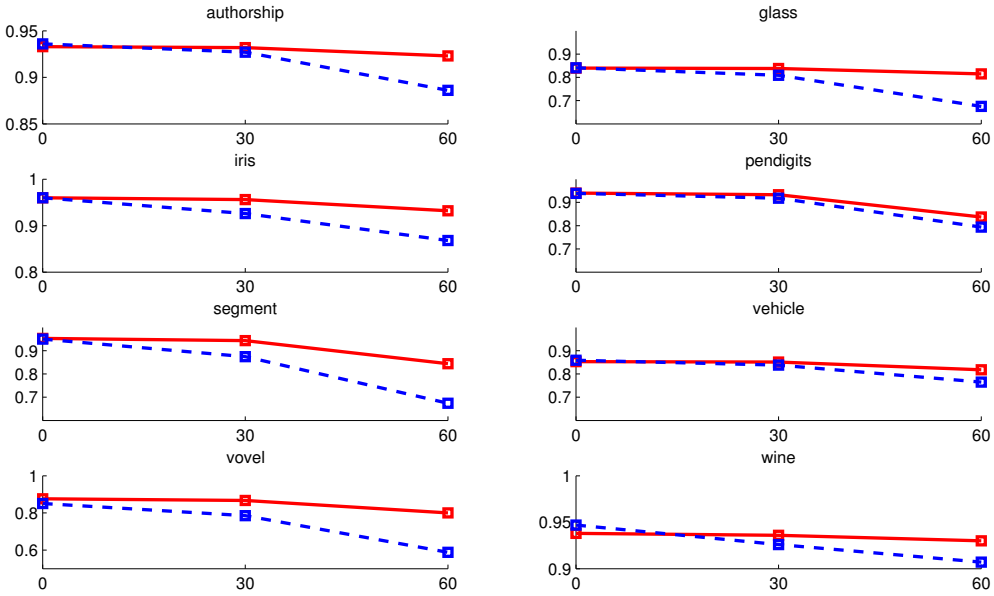

Figure 3: Performance of LWD (solid lines) and PL (dashed line) in the top-rank setting.