A Unified Model for Multilabel Classification and Ranking

Klaus Brinker[1] and **Johannes Fürnkranz**[2] and **Eyke Hüllermeier**[1]

Abstract. Label ranking studies the problem of learning a mapping from instances to rankings over a predefined set of labels. Hitherto existing approaches to label ranking implicitly operate on an underlying (utility) scale which is not calibrated in the sense that it lacks a natural zero point. We propose a suitable extension of label ranking that incorporates the calibrated scenario and substantially extends the expressive power of these approaches. In particular, our extension suggests a conceptually novel technique for extending the common *learning by pairwise comparison* approach to the multilabel scenario, a setting previously not being amenable to the pairwise decomposition technique. We present empirical results in the area of text categorization and gene analysis, underscoring the merits of the calibrated model in comparison to state-of-the-art multilabel learning methods.

1 INTRODUCTION

Label ranking, a particular preference learning scenario [7], studies the problem of learning a mapping from instances (typically represented by feature vectors) to rankings over a finite number of predefined *labels*, in this context also referred to as *alternatives*. Approaches that operate in this framework include *Ranking by pairwise comparison* (RPC) as a natural extension of pairwise classification [6] and *constraint classification* which aims at learning a linear utility function for each label [8].

Although this framework is very general in the sense that simpler learning problems, such as classification or l-multilabel classification can be embedded as special cases, it is restricted to ranking the labels on a non-calibrated scale, that is, a scale which lacks a natural zero-point. Learning problems, such as conventional multilabel learning, which require a bipartite partition into complementary sets (relevant and non-relevant) do not admit a representation as special instances of either RPC or constraint classification. More generally, as a consequence of the underlying non-calibrated scales, both approaches cannot learn to determine the relevant/non-relevant cutoff, even though this information is specified in the training data.

We will present a general model avoiding the aforementioned drawbacks by incorporating a calibrated scale which contains a natural zero-point. Extending conventional label ranking approaches, this novel framework provides a means to represent and learn bipartite partitions of alternatives. In particular, it suggests a conceptually new technique for extending the common pairwise classification learning approach to the multilabel scenario, a setting previously not being amenable to a pairwise decomposition technique.

2 LABEL RANKING

In label ranking, the problem is to learn a mapping from instances $x \in \mathcal{X}$ to rankings $\succ_x$ (total strict orders) over a finite set of labels $\mathcal{L} = \{\lambda_1, \ldots, \lambda_c\}$, where $\lambda_i \succ_x \lambda_j$ means that, for instance x, label λ_i is preferred to λ_j. A ranking over $\mathcal{L}$ can be represented by a permutation as there exists a unique permutation τ such that $\lambda_i \succ_x \lambda_j$ iff $\tau(\lambda_i) < \tau(\lambda_j)$, where $\tau(\lambda_i)$ denotes the position of the label λ_i in the ranking. The target space of all permutations over c labels will be referred to as $\mathcal{S}_c$.

It has been pointed out in several publications [8; 6; 3] that a variety of learning problems may be viewed as special cases of label ranking (perhaps supplemented by a straightforward projection of the output space $\mathcal{S}_c$) hence underscoring the importance of this setting. Among those are the following:

- *Multiclass Classification:* A single class label λ_i is assigned to each example x. This implicitly defines the set of preferences $R_x = \{\lambda_i \succ_x \lambda_j \mid 1 \leq j \neq i \leq c\}$. The output space $\mathcal{S}_c$ is projected to the first component.
- *l-Multilabel Classification:* Each training example x is associated with a subset $P_x \subseteq \mathcal{L}$ of possible labels. This implicitly defines the set of preferences $R_x = \{\lambda_i \succ_x \lambda_j \mid \lambda_i \in P_x, \lambda_j \in \mathcal{L} \setminus P_x\}$. The output space is projected to the first l components.

RPC and constraint classification provide a general means to extend arbitrary (linear) binary classification algorithms to the label ranking scenario. Both approaches require (not necessarily complete) sets of pairwise preferences associated with the training instances to learn a ranking model which, as a post-processing step, may be projected from $\mathcal{S}_c$ to the specific output space $\mathcal{Y}$.

The key idea of RPC is to learn, for each pair of labels (λ_i, λ_j), a binary model $\mathcal{M}_{ij}(x)$ that predicts whether $\lambda_i \succ_x \lambda_j$ or $\lambda_j \succ_x \lambda_i$ for an input x. In order to rank the labels for a new instance, predictions for all pairwise label preferences are obtained and a ranking that is maximally consistent with these preferences is derived, typically by means of a voting scheme.[3] This technique describes a natural extension of pairwise classification, i.e., the idea to approach a multiclass classification problem by learning a separate model for each pair of classes.

As we will see in the next section, conventional multilabel classification can not be embedded as a special case of the label ranking setting because, even though RPC or constraint classification could be used to rank the labels, they do not include a mechanism for extracting the set of relevant labels from this ranking.

[1] Otto-von-Guericke-Universität Magdeburg, Germany, email: {brinker,huellerm}@iti.cs.uni-magdeburg.de

[2] TU Darmstadt, Germany, email: juffi@ke.informatik.tu-darmstadt.de

[3] To account for equal vote statistics, we consider random tie breaking in our implementation.

3 MULTILABEL CLASSIFICATION AND RANKING

Multilabel classification refers to the task of learning a mapping from an instance $x \in \mathcal{X}$ to a set $P_x \subset \mathcal{L}$, where $\mathcal{L} = \{\lambda_1, \dots, \lambda_c\}$ is a finite set of predefined labels, typically with a small to moderate number of alternatives. Thus, in contrast to multiclass learning, alternatives are not assumed to be mutually exclusive such that multiple labels may be associated with a single instance. The set of labels P_x are called *relevant* for the given instance, the set $N_x = \mathcal{L} \setminus P_x$ are the *irrelevant* labels.

A common approach to multilabel classification is *binary relevance learning* (BR). BR trains a separate binary relevance model $\mathcal{M}_i$ for each possible label λ_i, using all examples x with $\lambda_i \in P_x$ as positive examples and all those with $\lambda_i \in N_x$ as negative examples. For classifying a new instance, all binary predictions are obtained and then the set of labels corresponding to positive relevance classification is associated with the instance. This scenario is, for example, commonly used for evaluating algorithms on the REUTERS text classification benchmark [10].

In the following, we will study the task of *multilabel ranking*, which is understood as learning a model that associates with a query input x both a ranking of the complete label set $\{\lambda_1, \dots, \lambda_c\}$ and a bipartite partition of this set into relevant and irrelevant labels. Thus, multilabel ranking can be considered as a generalization of both multilabel classification and ranking.

In conventional label ranking, a training example typically consists of an instance $x \in \mathcal{X}$, represented with a fixed set of features, and a set of pairwise preferences over labels $R_x \subset \mathcal{L}^2$, where $(\lambda, \lambda') \in R_x$ is interpreted as $\lambda \succ_x \lambda'$. In multilabel classification, the training information consists of a set P_x of relevant labels and, implicitly, a set $N_x = \mathcal{L} \setminus P_x$ of irrelevant labels. Note that this information can be automatically transformed into a set of preferences $\hat{R}_x = \{(\lambda, \lambda') \mid \lambda \in P_x \wedge \lambda' \in N_x\}$ (cf. Fig. 1 (a)).

While it is straightforward to represent the training information for multilabel classification as a preference learning problem, the algorithms that operate in this framework only produce a ranking of the available options. In order to convert the learned ranking into a multilabel prediction, the learner has to be able to autonomously determine a point at which the learned ranking is split into sets of relevant and irrelevant labels. Previous applications of ranking techniques to multilabel learning, such as [2], have ignored this problem and only focused on producing rankings, but not on determining this correct *zero point* for splitting the ranking.

Multilabel ranking can, for example, be realized if the binary classifiers provide real-valued confidence scores or *a posteriori* probability estimates for classification outcomes. Schapire & Singer [12] included an *ad hoc* extension to multilabel ranking in their experimental setup by ordering labels according to decreasing confidence scores.

In the following, we will introduce calibrated ranking, a conceptually new technique for extending the common pairwise learning approach to the multilabel scenario, a setting previously not being amenable to a pairwise decomposition approach. Within our framework, RPC can solve both multilabel classification and ranking problems in a consistent and generally applicable manner.

4 CALIBRATED LABEL RANKING

In this section, we will propose a general model avoiding the aforementioned drawbacks by incorporating a calibrated scale which contains a natural zero-point. This zero-point provides a means to distinguish between the top and the complementary set of labels.

Let us proceed to a formal definition of the hypothesis space underlying the *calibrated label ranking framework*:

Definition 4.1 (Calibrated Label Ranking Model). *Denote by $\mathcal{X}$ a nonempty input space and by $\mathcal{S}_c^0$ the space of permutations over the set $\{\lambda_0, \lambda_1, \dots, \lambda_c\}$, that is, the original set of labels plus an additional* virtual label λ_0. *Then, a model $h : \mathcal{X} \to \mathcal{S}_c^0$ is referred to as a* calibrated label ranking model.

The key idea is to use the virtual label λ_0 as a split point between relevant and irrelevant labels: all relevant labels are preferred to λ_0, which in turn is preferred to all irrelevant labels. Thus, a *calibrated ranking*

$$\lambda_{i_1} \succ \dots \succ \lambda_{i_j} \succ \lambda_0 \succ \lambda_{i_{j+1}} \succ \dots \succ \lambda_{i_c} \tag{1}$$

induces both a ranking among the labels,

$$\lambda_{i_1} \succ \dots \succ \lambda_{i_j} \succ \lambda_{i_{j+1}} \succ \dots \succ \lambda_{i_c}, \tag{2}$$

and a bipartite partition into

$$P = \{\lambda_{i_1}, \dots, \lambda_{i_j}\} \quad \text{and} \quad N = \{\lambda_{i_{j+1}}, \dots, \lambda_{i_c}\} \tag{3}$$

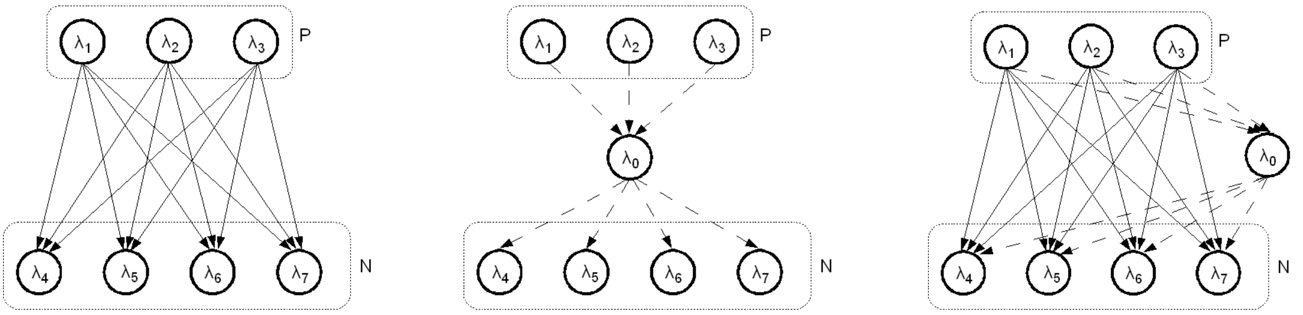
in a straightforward way.

As sketched in the previous section, the training information for a multilabel ranking problem consists of a set of preferences R_x, and subsets of labels $P_x, N_x \subset \mathcal{L}$ with $P_x \cap N_x = \emptyset$, which distinguish, respectively, *positive* labels that should be ranked above the zero-point element λ_0 and *negative* labels to be ranked below.[4] The bipartite partitions associated with the training instances is used to, with the help of the virtual label λ_0, induce additional constraints: the calibrated classifier h should predict $\lambda \succ_x \lambda_0$ for all $\lambda \in P_x$ and vice-versa $\lambda_0 \succ_x \lambda'$ for all $\lambda' \in N_x$ (cf. Fig. 1 (b)). Moreover, as a consequence of transitivity, it should predict $\lambda \succ_x \lambda'$ for all $\lambda \in P_x$ and $\lambda' \in N_x$ (Fig. 1 (c)). Combining the new partition-induced preference constraints with the original set of pairwise preferences for the training data, i.e.,

$$\begin{aligned} R'_x \overset{\text{def}}{=} R_x &\cup \{(\lambda, \lambda_0) \mid \lambda \in P_x\} \\ &\cup \{(\lambda_0, \lambda') \mid \lambda' \in N_x\} \\ &\cup \{(\lambda, \lambda') \mid \lambda \in P_x \wedge \lambda' \in N_x\}, \end{aligned} \tag{4}$$

the calibrated ranking model becomes amenable to previous approaches to the original label ranking setting: The calibrated ranking model can be learned by solving a conventional ranking problem in the augmented calibrated hypothesis space, which may be viewed as a ranking problem with $c + 1$ alternatives, with respect to the modified sets of constraints R'_x on the original labels $\lambda_1, \dots, \lambda_c$ *and* the virtual label λ_0. Therefore, this unified approach to the calibrated setting enables many existing techniques, such as RPC and constraint classification [1], to incorporate and exploit partition-related preference information and to generalize to settings where predicting the zero-point is required. We will discuss an exemplary application of this framework to pairwise ranking in Section 5.

[4] In general, we do not need to assume complete training data, neither for the sets of preferences (R_x might even be empty) nor for the partitions (which do not necessarily have to cover all the labels, i.e., $P_x \cup N_x \neq \mathcal{L}$). Besides, in a *noisy learning scenario*, it may happen that $(\lambda', \lambda) \in R_x$ even though $\lambda \in P_x$ and $\lambda' \in N_x$. In this paper, we will not further consider these cases, and assume a strict multilabel scenario.



(a) the set of preferences representing a multilabel classification problem

(b) introducing a virtual label λ_0 that separates P and N

(c) the set of preferences representing a calibrated label ranking problem

Figure 1. Calibrated Label Ranking

4.1 Relation to Binary Relevance Learning

Conventional pairwise label ranking learns a binary preference model $\mathcal{M}_{ij}$ for all combinations of labels λ_i and λ_j with $1 \leq i < j \leq c$,[5] where instances x with $(\lambda_i, \lambda_j) \in R_x$ are associated with positive and those with $(\lambda_j, \lambda_i) \in R_x$ with negative class labels (cf. Figure 1 (b)). In the calibrated scenario, the partition-related constraints with respect to λ_0 as defined in (4) are required to learn an additional set of binary preference models $\mathcal{M}_{0j}$ with $1 \leq j \leq c$. It is important to notice, that these additional models are identical to the common binary-relevance models $\mathcal{M}_j$.

Theorem 4.2. *The models $\mathcal{M}_{0j}$ that are learned by a pairwise approach to calibrated ranking, and the models $\mathcal{M}_j$ that are learned by conventional binary ranking are equivalent.*

Proof. Each training example x, for which label λ_j is relevant, is, by definition, a positive example in the training set for model $\mathcal{M}_j$. The calibrated ranking approach adds the preference $\lambda_j \succ_x \lambda_0$, which means that x will be a negative example for $\mathcal{M}_{0j}$. Similarly, if λ_j is irrelevant, x is negative for $\mathcal{M}_j$ and positive for $\mathcal{M}_{0j}$. Assuming a symmetric learner, the learned models will be the equivalent in the sense that $\mathcal{M}_j = -\mathcal{M}_{0j}$. □

Thus, calibrated RPC may be viewed as a method for combining RPC with conventional binary ranking, in the sense that the binary models that are learned for RPC, and the binary models that are learned for BR, are pooled into a single ensemble. However, CRPC provides a new interpretation to the BR models, that not only allows for ranking the labels, but also to determine a suitable split into relevant and irrelevant categories.

By training a larger number of pairwise models, the calibrated extension of RPC achieves two potential advantages over simple relevance learning. Firstly, it provides additional information about the ranking of labels. Secondly, it may also improve the discrimination between relevant and irrelevant labels. In fact, it is legitimate to assume (and indeed supported by empirical evidence in the next section) that the additional binary models can somehow "stabilize" the related classification. For example, while an error of model $\mathcal{M}_j = -\mathcal{M}_{0j}$ definitely causes a misclassification of label λ_j in simple relevance learning, this error might be compensated by the models $\mathcal{M}_{ij}$, $1 \leq i \neq j \leq c$, in the case of RPC. The prize to pay is, of course, a higher computational complexity, which, as we will show in the next section, depends on the maximum number of labels for an example.

4.2 Computational Complexity

In this section, we will derive bounds for the number of training examples that are constructed for training the pairwise classifiers. The total computational complexity depends on the complexity of the base classifier used for processing these examples. In brief, superlinear classifiers like SVMs will profit from distributing the workload on many small problems instead of fewer larger problems as for the BR approach.

Theorem 4.3. *CRPC is trained on $O(lcn)$ examples, where $l = max_x|P_x|$ is the maximum number of relevant labels that may be associated with a single training example, c is the number of possible labels, and n is the number of original training examples.*

Proof. In previous work, it has been shown that training a pairwise classifier requires $O(cn)$ training examples [5]. To see this, note that each of the n original training examples will only appear in the training sets of $c - 1$ models, therefore the total number of training examples that have to be processed is $(c-1)n$.

For multilabel classification, RPC will compare a training example's $|P_x|$ relevant labels to all $|N_x| = c - |P_x|$ labels that are not relevant for this example. In addition, CPRC will include every example in the c training sets of the models $\mathcal{M}_{0j}, j = 1 \dots c$.

Thus, each example occurs in $|P_x| \times |N_x| + c = |P_x|(c - |P_x|) + c < |P_x|c + c < (l+1)c$ training sets, and the total number of training examples is therefore $O(lcn)$. □

This result shows that the complexity of training CRPC depends crucially on the maximum number of labels in the training examples. For the case of $l = 1$, i.e., for conventional pairwise classification, we get the linear bound that was shown in [5].[6] For multilabel classification with a maximum number of l labels per example, we still have a bound that is linear in the number of examples and the number of classes, i.e., the complexity is within a factor of l of the $O(cn)$ examples needed for training a BR. We would like to point out that in many practical applications, the bound l is determined by the procedure that is used for labeling the training examples, and is independent of c. A typical example is the number of keywords that are assigned to a text, which is rarely ever more than ten.

Thus, for practical purposes, the complexity of CRPC is within a constant factor of l of the complexity of BR. Of course, the worst-case complexity, which would occur when all possible label subsets are *a priori* equally likely, is $O(c^2n)$.

[5] The case $i > j$ is typically not required as a consequence of the symmetry of binary classifiers with respect to positive and negative instances.

[6] Note that the O-notation hides, in this case, a constant factor of two, which results from the introduction of the virtual label λ_0.

Table 1. Experimental Results on the Yeast Dataset.

	Degree	1	2	3	4	5	6	7	8	9	Optimum
CRPC	PREC	**0.762***	**0.768***	**0.760***	**0.767***	**0.772***	**0.772***	**0.773**	0.770	0.767	**0.773**
	RANKLOSS	**0.168***	**0.164***	**0.170***	**0.164***	**0.159***	**0.158***	**0.158***	0.160	0.162	**0.158**
	ONEERROR	**0.230**	**0.224***	**0.230***	**0.229***	**0.229***	**0.233**	0.230	0.234	0.234	**0.224**
	HAMLOSS	0.199	**0.197**	**0.208***	**0.204***	**0.199***	**0.194**	0.193	0.192	0.193	0.192
BR	PREC	0.746	0.755	0.733	0.742	0.755	0.762	0.768	**0.772**	**0.771**	0.772
	RANKLOSS	0.199	0.183	0.197	0.188	0.178	0.171	0.165	**0.161**	**0.160**	0.160
	ONEERROR	0.241	0.248	0.277	0.268	0.244	0.236	**0.229**	**0.227**	**0.229**	0.227
	HAMLOSS	0.199	0.198	0.219	0.209	0.202	0.196	**0.192**	0.192	**0.192***	0.192

Bold face indicates superior performance comparing models with the same kernel parameters (except for the optimum-related column).
Stars indicate statistical significance at the $\alpha = 0.05$ level using a paired t-test.

Table 2. Experimental Results on the Reuters2000 Dataset.

	C	1	10	100	Optimum
CRPC	PREC	**0.943**	**0.944**	**0.943**	**0.944**
	RANKLOSS	**0.031**	**0.031**	**0.031**	**0.031**
	ONEERROR	**0.052**	**0.052**	**0.052**	**0.052**
	HAMLOSS	0.067	0.069	**0.070**	0.067
BR	PREC	0.940	0.935	0.933	0.940
	RANKLOSS	0.035	0.038	0.039	0.035
	ONEERROR	0.053	0.061	0.063	0.053
	HAMLOSS	0.067	0.069	0.071	0.067

Note: Bold face indicates superior performance comparing models with the same kernel parameters (except for the optimum-related column).

5 EMPIRICAL EVALUATION

The purpose of the following section is to provide an empirical comparison of calibrated RPC with the common binary-relevance approach in the domain of multilabel classification and ranking. The datasets that were included in the experimental setup cover two application areas in which multilabeled data are frequently observed: *text categorization* and *bioinformatics*.

- *Yeast:* The Yeast gene functional multiclass classification problem consists of 1500 genes in the training and 917 in the test set, represented by 103 features, where each gene is associated with a subset of the 14 functional classes considered [4].
- *Reuters2000:* The Reuters Corpus Volume I is one of the currently most widely used test collection for text categorization research. As the full corpus contains more than 800000 newswire documents, we restricted our experiments to a subset distribution of five times 3000 training and 3000 test documents which is publically available as a preprocessed version.[7] The documents are represented using stemmed word frequency vectors (normalized to unit length) with a TFIDF weighting scheme and elimination of stopwords resulting in 47152 features. Each document is associated with a subset of the 101 categories present in the dataset. The number of categories was reduced to 10 in our experiments by sequentially grouping the original categories into buckets of roughly equal size where each bucket corresponds to a single new label which is associated with positive relevance if at least one of the labels in the bucket is relevant.

We replicated the experimental setup in [4] to conduct the empirical evaluation on the Yeast dataset where a predefined split into training and test data was given. The four different measures considered cover both multilabel classification and ranking performance:

Preliminaries: For a multilabel ranking model h and a given instance x, let $\tau(\lambda_i)$ denote the position of λ_i in the predicted ranking (with λ_0 being removed from the ranking), $\tau^{-1}(i)$ the label λ having assigned position i, and P the set of relevant labels as predicted by CRPC.

Precision (PREC) assesses the multilabel ranking performance and is a frequently used measure in Information Retrieval:

$$\text{PREC}(h, x, P_x) \overset{\text{def}}{=} \frac{1}{|P_x|} \sum_{\lambda \in P_x} \frac{|\{\lambda' \in P_x | \tau(\lambda') \leq \tau(\lambda)\}|}{\tau(\lambda)} \tag{5}$$

The ranking loss (RANKLOSS) also measures ranking performance as it evaluates the average fraction of pairs of labels which are not correctly ordered:

$$\text{RANKLOSS}(h, x, P_x) \overset{\text{def}}{=} \frac{\left|\{(\lambda, \lambda') \in P_x \times N_x \mid \tau(\lambda) > \tau(\lambda')\}\right|}{|P_x||N_x|}$$

The one-error loss (ONEERROR) evaluates the multilabel ranking performance from a restricted perspective as it only determines if the top-ranked label is actually relevant:

$$\text{ONEERROR}(h, x, P_x) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } \tau^{-1}(1) \notin P_x, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The Hamming loss (HAMLOSS) assesses the multilabel classification performance in terms of the average binary (non-)relevant error:

$$\text{HAMLOSS}(h, x, P_x) \overset{\text{def}}{=} \frac{1}{|\mathcal{L}|} \left| P \triangle P_x \right| \tag{7}$$

In compliance with [4], support vector machines (SVMs) [13] provided the underlying classifiers for the binary-relevance multilabel model, as well as for the calibrated ranking model which was built on top of SVMs with polynomial kernels. The degree of the kernel varied from 1 to 9 and the margin-error penalty was fixed to $C = 1$. Linear kernels with $C \in \{1, 10, 100\}$ were used on the Reuters2000 dataset as they have demonstrated excellent performance in many publications studying text mining problems [9].

The empirical results on the Yeast dataset, depicted in Table 1, demonstrate that the calibrated ranking model is a promising alternative to the common binary-relevance approach to multilabel classification and ranking. Over a wide range of parameter values, the performance of the calibrated ranking is superior to the binary-relevance approach and comparatively stable, thus indicating a high level of robustness. For the optimal choice of parameters with respect to the testset (which could be approximately determined by cross-validation or leave-one-out estimation in practice), the gap is decreasing while calibrated ranking still outperforms its binary-relevance counterpart on three of the four evaluation measures. Interestingly, the only measure for which both approaches achieve comparable accuracy, namely, the Hamming loss, evaluates the multilabel accuracy

[7] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

in a manner that seems to be more tailored to the binary-relevance approach from a theoretical point of view as it computes the average (independently measured) binary-relevance error. Note that although the pattern observed in Table 1 might suggest that further increasing the degree of the polynomial kernel could improve the experimental results of the binary-relevance approach, this assumption could not be validated in an additional experiment. Despite that we could not set up a perfect replication of [4] as certain parameters (e.g., the margin-penalty values C) were not published, we nevertheless note that the performance of calibrated RPC compares favorably to those reported in [4] for the ranking and Hamming loss, while the results in terms of precision are comparable to those of the herein proposed SVM ranking algorithm. With respect to the one-error loss the performance is slightly worse. Moreover, there is a substantial margin between the performance of Boostexter [12], a boosting-based multilabel learning system, on the Yeast benchmark dataset (as published in [4]) and the calibrated ranking technique with respect to all four evaluation measures considered in the experimental setup.

The empirical results on the Reuters2000 dataset (see Table 2) clearly support the conclusions drawn from the Yeast dataset. In terms of the Hamming loss, both approaches achieve comparable accuracy, while calibrated ranking outperforms binary-relevance multilabel learning for the remaining evaluation measures. Again, the margin is typically smaller for the optimal choice of parameter values, whereas the difference in accuracy is larger when comparing the results with the parameter C being fixed. Moreover, calibrated ranking tends to be more stable than binary-relevance learning with respect to the choice of the underlying binary classifiers. Furthermore, additional experiments using different methods for reducing the number of labels, such as selecting only the most frequent relevant labels (cf., [12]), showed qualitatively comparable outcomes.

6 RELATED WORK

Schapire & Singer [11] derived a family of multilabel extensions in the framework of AdaBoost (referred to as AdaBoost.MH and AdaBoost.MR) which provided the algorithmic basis for the Boostexter text and speech categorization system [12]. For the online learning setting, a computationally efficient class of perceptron-style algorithms for multilabel classification and ranking was proposed in [2]. In a manner similar to the above mentioned approaches, the multilabel generalization of support vector machines advocated by Elisseeff & Weston [4] exploits comparative preferences among pairs of labels as defined for the multilabel case in Section 2, while lacking a natural approach to determine the relevance zero-point in the multilabel ranking.

7 CONCLUDING REMARKS

We have proposed a unified extension to overcome the severe restriction on the expressive power of previous approaches to label ranking induced by the lack of a calibrated scale. This unified approach to the calibrated ranking setting enables general ranking techniques, such as ranking by pairwise comparison and constraint classification, to incorporate and exploit partition-related preference information and to generalize to settings where predicting the zero-point is required. In particular, the calibrated extension suggests a conceptually novel technique for extending the common learning by pairwise comparison technique to the multilabel scenario, a setting previously not being amenable to pairwise decomposition. A particular limitation of the binary-relevance extension to multilabel ranking, which is not shared by the calibrated framework, consists in the fact that it only applies to soft-classifiers providing confidence scores in the prediction. Exploring the benefits of calibrated ranking with binary-valued classifiers is a promising aspect of future work. Experimental results in the areas of text categorization and gene analysis underscore the merits of our calibrated framework from an empirical point of view.

ACKNOWLEDGEMENTS

This research was supported by the German Research Foundation (DFG) and Siemens Corporate Research (Princeton, USA).

REFERENCES

[1] Klaus Brinker and Eyke Hüllermeier, 'Calibrated Label-Ranking', *Proceedings of the NIPS-2005 Workshop on Learning to Rank*, S. Agarwal and C. Cortes and R. Herbrich (eds.), pp. 1–6, Whistler, BC, Canada, (2005).

[2] Koby Crammer and Yoram Singer, 'A family of additive online algorithms for category ranking', *Journal of Machine Learning Research*, **3**, 1025–1058, (2003).

[3] Ofer Dekel, Christopher Manning, and Yoram Singer, 'Log-linear models for label ranking', in *Advances in Neural Information Processing Systems 16*, eds., Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, MIT Press, Cambridge, MA, (2004).

[4] André Elisseeff and Jason Weston, 'A kernel method for multi-labelled classification', in *Advances in Neural Information Processing Systems 14*, eds., T. G. Dietterich, S. Becker, and Z. Ghahramani, pp. 681–687, Cambridge, MA, (2002). MIT Press.

[5] Johannes Fürnkranz, 'Round robin classification', *Journal of Machine Learning Research*, **2**, 721–747, (2002).

[6] Johannes Fürnkranz and Eyke Hüllermeier, 'Pairwise preference learning and ranking', in *Proceedings of the 14th European Conference on Machine Learning (ECML 2003)*, pp. 145–156, Cavtat, Croatia, (2003). Springer-Verlag.

[7] Johannes Fürnkranz and Eyke Hüllermeier, 'Preference learning', *Künstliche Intelligenz*, **19**(1), 60–61, (2005).

[8] Sariel Har-Peled, Dan Roth, and Dav Zimak, 'Constraint classification: A new approach to multiclass classification and ranking', in *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, (2002).

[9] Thorsten Joachims, 'Text categorization with support vector machines: Learning with many relevant features', in *Proceedings of the European Conference on Machine Learning (ECML 1998)*, eds., Claire Nédellec and Céline Rouveirol, pp. 137–142, Berlin, (1998). Springer.

[10] David D. Lewis. Reuters-21578 text categorization test collection. README file (V 1.2), available from `http://www.research.att.com/~lewis/reuters21578/README.txt`, September 1997.

[11] Robert E. Schapire and Yoram Singer, 'Improved boosting using confidence-rated predictions', *Machine Learning*, **37**(3), 297–336, (1999).

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

[13] Bernhard Schölkopf and Alexander J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2002.