Label Ranking with Probabilistic Models

A dissertation submitted to
the Department of Mathematics and Computer Science
of Philipps-Universität Marburg
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Weiwei Cheng
March 2012

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Eyke Hüllermeier) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Johannes Fürnkranz)

Approved for the University Committee on Graduate Studies

To my parents.

Acknowledgment

I would like to thank many people who have helped during my PhD study. Without them, I could not have completed this thesis.

First of all, I would like to express my greatest gratitude to my PhD advisor Prof. Eyke Hüllermeier. Thank him for introducing me to the field of machine learning in general and preference learning in particular. I have learned a lot from his inspiring ideas and our fruitful discussions. It is my honor to be a member of his research group Knowledge Engineering & Bioinformatics at Philipps-Universität Marburg and I have benefited from the active and productive research atmosphere he created. I would also like to thank Prof. Johannes Fürnkranz for being the second reviewer of my thesis. The tight cooperation between our group and his Knowledge Engineering group at Technische Universität Darmstadt has made many challenging research projects successful.

Many of my publications during the PhD study are results of cooperation and teamwork. I would like to thank all my collaborators and, especially, the co-authors of my research papers. Besides Eyke and Johannes, I have also the privilege to co-author papers with Dr. Klaus Brinker, Prof. Bernard De Baets, Dr. Krzysztof Dembczyński, Thomas Fober, Dr. Thore Graepel, Dr. Ralf Herbrich, Dr. Jens Hühn, Dr. Gjergji Kasneci, Sang-Hyeun Park, Dr. Michaël Rademaker, Prof. Bernhard Seeger, Dr. David Stern, Ali Fallah Tehrani, Ilya Vladimirskiy, Dr. Willem Waegeman, and Prof. Volkmar Welker.

Special thanks go to my colleagues and friends at the Knowledge Engineering & Bioinformatics lab. The enjoyable discussions among us helped

me to solve many problems concerning theory, implementation, and everyday lives as well. I would like to thank my colleagues and friends at Deutsche Bank Eschborn and Microsoft Research Cambridge. During my PhD study, I have been lucky enough to work as an intern at these two places. Both internships were valuable and unforgettable experience.

I am deeply indebted to my family in China. My deepest appreciation goes to my parents Zemin Cheng and Ping Wang. I will never forget how supportive they were, when I decided to come to Germany. And I am sure that they care about my thesis as much as I do, if not more. I dedicate this thesis to them.

Contents

List of Abbreviations

AUC	area under the ROC curve
BR	binary relevance
CC	constraint classification
EM	expectation maximization
IB-M	instance-based label ranking with Mallows model
IB-PL	instance-based label ranking with Plackett-Luce model
KNN	k-nearest neighbor
Lin-PL	generalized linear approach with Plackett-Luce model
LL	log-linear models for label ranking
MallowsML	multi-label learning with Mallows model
MAP	maximum a posteriori
MLE	maximum likelihood estimation
MLKNN	multi-label k-nearest neighbor
MM	minorization maximization
NDCG	normalized discounted cumulative gain
NP	non-deterministic polynomial time
PL	Plackett-Luce
PTAS	polynomial-time approximation scheme
RPC	label ranking by pairwise comparisons

Chapter 1

Introduction

This thesis develops a series of probability-based methods for the label ranking problem, an emerging learning task often addressed in the field of machine learning in general and preference learning in particular. In this chapter, we give a general introduction to the thesis, starting with some illustrative examples of label ranking in Section 1.1. We summarize the contributions of the thesis in Section 1.2 and outline its structure in Section 1.3.

1.1 Label Ranking: Illustrative Examples

Label ranking is a key prediction task in preference learning, where the goal is to map instances to a total order of a finite set of predefined labels. Label ranking problems can be found everywhere. As an example, suppose a car dealer sells three brands of cars, BMW, Ford, and Toyota. Each customer may have different preferences on these cars. The car dealer may have records as listed in Table 1.1. An interesting question is, how we can predict the preferences of new customers based on the historical records, i.e., the training data. For example, what would be a reasonable ranking of these three brands for a 32 years old male from Berlin? Such prediction can provide great helps for the sale management. The records in Table 1.1 form a typical label ranking data set. Different from a classification task, where a subset of

customer	preference
male, 49, New York	Ford $\succ$ Toyota $\succ$ BMW
male, 22, Beijing	BMW $\succ$ Ford $\succ$ Toyota
male, 30, Frankfurt	BMW $\succ$ Toyota $\succ$ Ford
female, 27, Tokyo	Toyota $\succ$ BMW $\succ$ Ford
...	

Table 1.1: A label ranking data set based on a fictitious car dealer. The customers are characterized by gender, age, and geographical location. There are three labels in total corresponding to three brands of cars.

labels is selected as the prediction, a complete ordering of labels is required. Predictions in terms of a complete ordering of labels offers some advantages over a subset of labels. In particular, when a top choice for a customer is not available due to some unexpected reason, it is very easy to give the customer a second best offer.

Predicting an ordering of labels is generally much harder than predicting a subset of them, as the search space is of order $n!$ instead of 2^n, where n is the number of labels. Moreover, the evaluation of the predictions in label ranking becomes more complicated than that in the classification setting, simply because comparing two rankings is generally more difficult than comparing two subsets. A degree of similarity must be first defined in order to compare rankings. For example, given the ground-true ranking BMW $\succ$ Ford $\succ$ Toyota, we may say the prediction BMW $\succ$ Toyota $\succ$ Ford is better than the prediction Toyota $\succ$ Ford $\succ$ BMW, if the degree of similarity is defined as the number of paired labels that the prediction agrees with the ground-true ranking.

Another challenge comes from the fact that the training data may be imperfect. The information provided by customers can be incorrect or inconsistent. Particularly in label ranking, the training data may contain incomplete ranking information. For example, a customer may associate with the ranking BMW $\succ$ Ford, but no information is given about the preference on Toyota. In this case, various interpretations exist. One may think this customer's ground-true ranking being one of the following, BMW $\succ$ Ford $\succ$

Toyota, BMW $\succ$ Toyota $\succ$ Ford, or Toyota $\succ$ BMW $\succ$ Ford. When analyzing such piece of incomplete information, it requires us to consider all these possible situations and hence leads to great computational challenges. Note that, the information BMW $\succ$ Ford doesn't necessarily mean BMW is the best choice for this customer nor Ford is the worst one. Such information can be hardly represented with the conventional classification setting.

In some label ranking applications, the reliability of the predictions is of particular concern. To give a striking example, let us consider a learning task for the cancer treatment. We have observed a set of cancer patients, characterized by gender, age, tumor type, tumor size, etc. and for each patient there is an associated ranking of four possible medical actions: surgery, radiotherapy, chemotheraphy, and no treatment. When a new patient arrives, the goal is to predict a ranking of these possible actions that is most suitable for this patient. Needless to say, any prediction must come with extreme caution. We shall only give predictions that we are certain of. For example, if we are sure that the new patient needs a treatment, but uncertain about choices of treatments, the prediction should look like this

$$\text{surgery} \mid \text{radiotherapy} \mid \text{chemotheraphy} \succ \text{no treatment},$$

meaning that any treatment is better than no treatment, but the preference between different treatments is unknown. In order to come up with such predictions, we need to have a label ranking method that is able to assess all the pairwise comparisons it can provide and reject all the comparisons that are unreliable.

1.2 Summary of Contributions

Built upon the existing label ranking research, this thesis attempts to pursue three key directions:

1. To develop new label ranking methods with sound theoretical foundations.

2. To establish relations between label ranking and other learning settings, explore new applications of label ranking.

3. To generalize the label ranking setting.

The contributions of the thesis can be categorized into these three directions correspondingly.

Most of existing approaches to label ranking focus on adapting the established classification methods to label ranking. That is, to reduce the label ranking problem to a set of classification problems and the solutions of these classification problems are then combined to a label ranking. The label ranking methods proposed in this thesis are centered around the probabilistic theory, making use of different statistical models for ranking data, by which such reduction to classifications are avoid. The use of probabilistic models allows theoretically sound analysis of our approaches and comes with a number of other merits as well.

The setting of label ranking is very general and it can be seen as a generalization of a number of other learning settings. For example, as we mentioned in the car dealer example in the previous section, when a subset of some top ranked labels instead of a complete ordering of labels is predicted, it becomes a classification problem. We elaborate the idea using label ranking techniques to solve classification problems.

We propose an extension of the label ranking setting, where the outputs are not necessary a total order, but can be a partial order in general. The idea is to predict only reliable predictions. Unlike most of the existing approaches for label ranking, with the probabilistic approaches we propose, we can derive the degree of confidence of a label ranking prediction in a very natural way.

1.3 Thesis Outline

When predicting label ranking of an instance, one often interprets such ranking on labels as a preference statement. In the car dealer case for example, it can be understood that we are trying to predict the customer's preferences on different brands of cars. In fact, label ranking is often studied by the preference learning community and is considered as one of the key problems in the preference learning field. In Chapter 2, we address label ranking learning in more details under the preference learning framework and, along the way, establish the basic mathematical concepts allowing future discussions in later chapters. Specifically, we give a formal definition of the label ranking learning task and discuss two other related ranking problems. Although the label ranking setting is the focus of this thesis, we believe such a general discussion reveals a better picture and helps for understanding the background of the research problem.

The remainder of the thesis is organized as follows: After an overview of existing label ranking methods in Chapter 3, our probabilistic label ranking approaches are introduced in Chapter 4 and 5. Specifically, Chapter 4 and 5 discuss how to utilize local and global learning methods with probabilistic models, respectively. In Chapter 6 we discuss how to apply the label ranking technique we proposed to solve classification problems. In particular, we will make use of the probabilistic label ranking method to solve the multi-label classification task. Chapter 7 addresses the issues of reliable predictions in label ranking, especially how to design label ranking methods that are able to abstain from any unreliable paired comparison between labels. Chapter 8 concludes the thesis with some final remarks.

Chapter 2

Preference Learning

Preference learning as a new branch of machine learning has attracted considerable attention in recent years. Roughly speaking, preference learning refers to the problem of learning from observations which reveal, either explicitly or implicitly, information about the preferences of an individual or a group of individuals. Generalizing beyond the given training data, the models learned are typically used for preference prediction, i.e., to predict the preferences of a new individual or the same individual in a new situation. Among others, the problem of learning to rank is a representative example and has received the most attention in the machine learning literature; here, the goal is to predict preferences in the form of total or partial orders of alternatives (e.g., a personalized ranking of webpages retrieved by a search engine). Based on the form of the training data and the required predictions, three types of ranking problems are frequently studied in the preference learning literature, namely object ranking, instance ranking, and label ranking [27]. In this chapter, we discuss these three ranking problems with an emphasis on the label ranking task. We try to stick as much as possible to the terminology commonly used in supervised learning, where a labeled instance consists of a set of features (called predictor or independent variables in statistics) and an associated class label (called response or dependent variables in statistics). The former is normally denoted by $\boldsymbol{x}$ with a corresponding instance space

Given:

- a reference set of objects $\mathcal{X}$
- a finite set of pairwise preference $\boldsymbol{x}_i \succ \boldsymbol{x}_j \in \mathcal{X} \times \mathcal{X}$

Find:

- a ranking function $f(\cdot)$ that assumes as input a set of objects and returns a permutation of this set

Performance measures:

- ranking error (e.g., based on the rank correlation) comparing the predicted ranking with the target ranking
- top-k measures comparing the top-positions of the rankings
- retrieval measures such as precision, recall, NDCG

Table 2.1: Definition of object ranking [27]

$\mathcal{X}$,

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_d\,, \tag{2.1}$$

while the label space is denoted by $\mathcal{Y}$.

2.1 Object Ranking

Given objects from an underlying reference set $\mathcal{X}$, the goal in object ranking is to learn a ranking function that produces a ranking of these objects. This is typically done by assigning a score to each instance and then sorting by scores. No output or class label is associated with an object.

An object $\boldsymbol{x} \in \mathcal{X}$ is commonly, though not necessarily, described by a attribute-value representation as Equation (2.1). The training information contains exemplary rankings or pairwise preferences of the form $\boldsymbol{x}_i \succ \boldsymbol{x}_j$

meaning that $\boldsymbol{x}_i$ is ranked higher than $\boldsymbol{x}_j$. This scenario, summarized in Table 2.1, is also referred to as "learning to order things" [13].

The performance can be measured with a distance function or correlation measure on rankings, when the ground-truth is given as rankings. We shall discuss these measures at Section 2.3. Normally, as the number of objects to be ranked is very large, one often prefers measures that emphasize more on the top-ranked objects. Evaluation measures tailored towards such requirements have been frequently used in information retrieval, such as NDCG (normalized discounted cumulative gain) [38].

As an example of an object ranking task, consider the meta-search problem [13], where the goal consists of learning to combine the web search results from different search engines. Here the ranking performance is often provided implicitly by users' click-through data [39].

2.2 Instance Ranking

The setting of instance ranking resembles ordinal classification, where an instance $\boldsymbol{x} \in \mathcal{X}$ belongs to one among a finite set of classes $\mathcal{Y} = \{y_i \,|\, i = 1, \ldots, n\}$ and the classes have an order $y_1 \prec \ldots \prec y_n$. For example, consider the assignment of submitted papers at an academic conference to classes reject, weak reject, weak accept, and accept. In contrast to the classification setting, the goal in instance ranking is not to learn a classifier but a ranking function. Given a subset $X \subset \mathcal{X}$ of instances as input, the function produces a ranking of these instances as output. Hence, instance ranking can be considered as a generic term for bipartite and multipartite ranking [29]. This scenario is summarized in Table 2.2.

As an example, consider the task of the reviewing papers in a conference. Often the labeling of papers is given in terms of different classes, but in the end, a ranking of papers is more desirable than only the classifications of them: If the conference finally decides to accept, say, 100 papers, it is much easier to select according to the ranking, while with the classification setting,

Given:

- a set of training instances $X = \{\boldsymbol{x}_i \,|\, i = 1, \ldots, m\}$
- a set of labels $\mathcal{Y} = \{y_i \,|\, i = 1, \ldots, n\}$ endowed with an order $y_1 \prec \ldots \prec y_n$
- for each training instance $\boldsymbol{x}_i$ an associated label y_i

Find:

- a ranking function $f(\cdot)$ that ranks a new set of instances $\{\boldsymbol{x}_j \,|\, j = 1, \ldots, m'\}$ according to their (underlying) preference degrees

Performance measures:

- the area under the ROC-curve (AUC) in the dichotomous case ($m = 2$)
- generalizations of AUC such as C-index in the polychotomous case ($m > 2$)

Table 2.2: Definition of instance ranking [27]

a further tie-breaking procedure is needed.

Different types of accuracy measures have been proposed for instance ranking. They are normally based on the number of pairs $(\boldsymbol{x}, \boldsymbol{x}') \in X \times X$ such that $\boldsymbol{x}$ is ranked higher than $\boldsymbol{x}'$ while the former belongs to a lower class than the latter. In the two-class case, this amounts to AUC, the area under the ROC-curve [9], which is equivalent to the Wilcoxon-Mann-Whitney statistic [64]. A generalization of this measure to the case of multiple classes is known as the concordance index or C-index in statistics [31].

2.3 Label Ranking

Label ranking can be seen as an extension of the conventional setting of classification. Roughly speaking, the former is obtained from the latter through

Given:

- a set of training instances $\{\boldsymbol{x}_k \,|\, k = 1, \ldots, m\} \subset \mathcal{X}$
- a set of labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$
- for each training instance $\boldsymbol{x}_k$ an associated set of pairwise preferences of the form $y_i \succ_{\boldsymbol{x}_k} y_j$

Find:

- a ranking function $f(\cdot)$ that maps any $\boldsymbol{x} \in \mathcal{X}$ to a ranking $\succ_{\boldsymbol{x}}$ of $\mathcal{Y}$ (permutation $\pi_{\boldsymbol{x}} \in \mathcal{S}_n$)

Performance measures:

- ranking error (e.g., based on rank correlation measures) comparing predicted ranking with target ranking
- position error comparing predicted ranking with a target label

Table 2.3: Definition of label ranking [27]

replacing a selection of class labels by a complete label ranking. So, instead of associating every instance $\boldsymbol{x}$ from the instance space $\mathcal{X}$ with some among a finite set of class labels $\mathcal{Y} = \{y_1, \ldots, y_n\}$, we now associate $\boldsymbol{x}$ with a total order of the class labels, that is, a complete, transitive, and asymmetric relation $\succ_{\boldsymbol{x}}$ on $\mathcal{Y}$, where $y_i \succ_{\boldsymbol{x}} y_j$ indicates that y_i precedes y_j in the ranking associated with $\boldsymbol{x}$. It follows that a ranking can be considered as a special type of preference relation, and therefore we shall also say that $y_i \succ_{\boldsymbol{x}} y_j$ indicates that y_i is preferred to y_j given the instance $\boldsymbol{x}$. To illustrate, suppose that instances are students (characterized by attributes such as gender, age, and major subjects in secondary school) and $\succ$ is a preference relation on a fixed set of study fields such as Math, CS, Physics.

Formally, a ranking $\succ_{\boldsymbol{x}}$ can be identified with a permutation $\pi_{\boldsymbol{x}}$ of the set $\{1, \ldots, n\}$. It is sometimes convenient to define $\pi_{\boldsymbol{x}}(i) = \pi_{\boldsymbol{x}}(y_i)$ as the

position of y_i in the ranking, i.e., the rank of y_i. This permutation encodes the (ground truth) ranking

$$y_{\pi_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} y_{\pi_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \cdots \succ_{\boldsymbol{x}} y_{\pi_{\boldsymbol{x}}^{-1}(n)} \ ,$$

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

To encode a ranking, two representations with integers are often used, namely the rank vector and the order vector. They both match an integer from 1 to n with an object. A rank vector lists the ranks given to objects, where "1" denotes the best and "n" denotes the worst. It presumes the objects are listed in a prespecified order. An order vector, on the other hand, lists the objects themselves with their corresponding indexes, from the best to the worst. For example, considering three subjects

$$\text{1. Math,} \quad \text{2. CS,} \quad \text{3. Physics,}$$

and the ranking

$$\text{Physics} \succ \text{Math} \succ \text{CS},$$

the rank vector representation is $\pi = (2, 3, 1)$, while the order vector representation is $\pi^{-1} = (3, 1, 2)$.

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

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

where $\{i_1, i_2, \ldots, i_k\}$ is a subset of the index set $\{1, \ldots, n\}$ such that $1 \leq i_1 < i_2 < \ldots < i_k \leq n$. 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.

To evaluate the predictive performance of a label ranker, a suitable loss function on Ω is needed. In the statistical literature, several distance measures for rankings have been proposed. One commonly used measure is the Kendall distance based on the number of discordant pairs,

$$T(\pi, \sigma) = \#\{(i,j) \,|\, \pi(i) > \pi(j) \text{ and } \sigma(i) < \sigma(j)\}\,, \tag{2.2}$$

which is closely related to the Kendall's tau coefficient in the case of complete rankings. In fact, the latter is a normalization of (2.2) to the interval $[-1, 1]$ that can be interpreted as a correlation measure (it assumes the value 1 if $\sigma = \pi$ and the value -1 if σ is the reversal of π):

$$\tau = \frac{\#\text{ concordant pairs} - \#\text{ discordant pairs}}{\#\text{ all pairs}}\,, \tag{2.3}$$

where the number of concordant pairs is defined similarly by $\#\{(i,j) \,|\, \pi(i) > \pi(j) \text{ and } \sigma(i) > \sigma(j)\}$.

Kendall distance is a natural, intuitive, and easily interpretable measure [44]. We shall focus on (2.2) throughout our discussions, although other distance measures could of course be used. Other widely used metrics on rankings include the Footrule distance

$$F(\pi, \sigma) = \sum_i |\pi(i) - \sigma(i)| \tag{2.4}$$

and the Spearman distance

$$S(\pi, \sigma) = \sum_i (\pi(i) - \sigma(i))^2\,. \tag{2.5}$$

It can be shown that [21]

$$T(\pi, \sigma) \leq F(\pi, \sigma) \leq 2T(\pi, \sigma) \ , \tag{2.6}$$

$$\frac{1}{\sqrt{n}} T(\pi, \sigma) \leq S(\pi, \sigma) \leq 2T(\pi, \sigma) \ . \tag{2.7}$$

Inequalities (2.6) and (2.7) establish tight relations between these three distances measures, which are of great practical relevance: Two rankings with a small distance in terms of one of these three measures tend to have small distance in terms of the other two measures as well. Based on this theoretical result, efficient approximate algorithms can be invented without much sacrifice of predictive performance, as we shall see in the later chapters.

A desirable property of any distance D on rankings is its invariance toward a renumbering of the elements (renaming of labels). This property is equivalent to the right invariance of D, namely $D(\sigma\nu, \pi\nu) = D(\sigma, \pi)$ for all $\sigma, \pi, \nu \in \Omega$, where $\sigma\nu = \sigma \circ \nu$ denotes the permutation $i \mapsto \sigma(\nu(i))$. The distance (2.2) is right-invariant, and so are most other commonly used metrics on Ω.

Chapter 3

Existing Label Ranking Methods

A number of methods have been proposed for label ranking learning. In this chapter, we give a concise survey of some key references, with a focus on the methods that we are comparing with in the later chapters. Most of the existing methods for label ranking can be categorized as reduction approaches, where a label ranking problem is decomposed into several simpler sub-problems, usually binary classification problems, and then the solutions of these sub-problems are combined into output rankings. In Sections 3.1 and 3.2 we will respectively introduce two widely applied schemes in the reduction approaches, namely label ranking by learning utility functions and label ranking by learning pairwise preferences, with discussions on some representative work. In Section 3.3, we will discuss the work by Brinker and Hüllermeier [10], which applies the instance-based methodology for label ranking and doesn't belong to the paradigm of reduction approaches.

3.1 Label Ranking by Learning Utility Functions

One natural way to represent preferences is to evaluate individual alternatives by means of a real-valued utility function. In the label ranking scenario, a utility function $f_i : \mathcal{X} \to \mathbb{R}$ is needed for each label $y_i, i = 1, \ldots, n$. Here, $f_i(\boldsymbol{x})$ is the utility assigned to label y_i by instance $\boldsymbol{x}$. To obtain a ranking for $\boldsymbol{x}$, the labels are ordered according to these utility scores, such that $y_i \succ_{\boldsymbol{x}} y_j \Leftrightarrow f_i(\boldsymbol{x}) > f_j(\boldsymbol{x})$.

If the training data offer the utility scores directly, preference learning would reduce to a conventional regression problem. But this type of information can rarely be assumed. Instead, usually only constraints derived from comparative preference information of the form "this label should have a higher utility score than that label" are given. Thus, the challenge for the learner is to find a function that is in agreement with all constraints as much as possible. Subsequently, we outline two approaches, constraint classification (CC) and log-linear models for label ranking (LL), which fit in this paradigm.

3.1.1 Constraint Classification

To learn the utility function $f_i(\cdot)$ for each label, the constraint classification framework proposed by Har-Peled et al. [33] proceeds from the following linear models:

$$f_i(\boldsymbol{x}) = \sum_{k=1}^{d} w_{ik} x_k \ , \tag{3.1}$$

with label-specific coefficients $w_{ik}, k = 1, \ldots, d$. A preference $y_i \succ_{\boldsymbol{x}} y_j$ is translated into the constraint $f_i(\boldsymbol{x}) - f_j(\boldsymbol{x}) > 0$ and equivalently $f_j(\boldsymbol{x}) - f_i(\boldsymbol{x}) < 0$. Both constraints, the positive and the negative one, can be expressed in terms of the sign of an inner product $\langle \boldsymbol{z}, \boldsymbol{w} \rangle$, where $\boldsymbol{w} = (w_{11}, \ldots, w_{1d}, w_{21}, \ldots, w_{nd})$ is a concatenation of all label-specific coefficients.

Correspondingly, the vector $\boldsymbol{z}$ is constructed by mapping the original d-dimensional training instance $\boldsymbol{x} = (x_1, \ldots, x_d)$ into an $(n \times d)$-dimensional space: For the positive constraint, $\boldsymbol{x}$ is copied into the components $((i-1) \times d + 1), \ldots, (i \times d)$ and its negation $-\boldsymbol{x}$ into the components $((j-1) \times d + 1), \ldots, (j \times d)$; the remaining entries are filled with 0. For the negative constraint, a vector is constructed with the same elements but reversed signs. Both constraints can be considered as training instances for a conventional binary classifier in an $(n \times d)$-dimensional space: The first vector is a positive and the second one a negative instance. The corresponding learner tries to find a separating hyperplane in this space, that is, a suitable vector $\boldsymbol{w}$ satisfying all constraints. To make a prediction for a new example $\boldsymbol{x}'$, the labels are ordered according to the response resulting from multiplying $\boldsymbol{x}'$ with the i-th d-element section of the hyperplane vector. As this method works solely in an inner product space, it can be kernelized when more complex utility functions are desired [53].

Alternatively, [33] proposes an online version of constraint classification, namely an iterative algorithm that maintains weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{R}^d$ for each label individually. In every iteration, the algorithm checks each constraint $y_i \succ_{\boldsymbol{x}} y_j$ and, in case the associated inequality $\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle = f_i(\boldsymbol{x}) > f_j(\boldsymbol{x}) = \langle \boldsymbol{w}_j, \boldsymbol{x} \rangle$ is violated, adapts the weight vectors $\boldsymbol{w}_i, \boldsymbol{w}_j$ appropriately. In particular, this algorithm can be implemented in terms of a multi-output perceptron in a way quite similar to the approach of Grammer and Singer [15]. We list the pseudo code proposed by [33] in Algorithm 1 with slight modifications tailored to the label ranking learning. When the training data are noise-free, that is, all the pairwise preferences $y_j \succ_{\boldsymbol{x}_i} y_{j'}$ are correctly given, the convergence of Algorithm 1 can be guaranteed. It is of course not often the case in a real-world application. In practice a noise-tolerant version of this algorithm can be applied, namely setting an upper bound α to the number of updates that can be made on one particular instance (or preference). This is often called the α-bound trick in the literature [42].

Algorithm 1 Online constraint classification for label ranking

Require: training data of size m as defined in Table 2.3
Ensure: weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{R}^d$ for ranking the labels

initialize $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n \in \mathbb{R}^d$
repeat until converge
for $i = 1, \ldots, m$ **do**
  for all pairwise preference $y_j \succ_{\boldsymbol{x}_i} y_{j'}$ **do**
    if $\langle \boldsymbol{w}_j, \boldsymbol{x}_i \rangle \leq \langle \boldsymbol{w}_{j'}, \boldsymbol{x}_i \rangle$ **then**
      promote $\boldsymbol{w}_j$
      demote $\boldsymbol{w}_{j'}$
    end if
  end for
end for

3.1.2 Log-Linear Model

The log-linear models for label ranking have been proposed by Dekel et al. [17]. Here, utility functions are expressed in terms of linear combinations of a set of base ranking functions:

$$f_i(\boldsymbol{x}) = \sum_j v_j h_j(\boldsymbol{x}, y_i), \tag{3.2}$$

where a base function $h_j(\cdot)$ maps instance-label pairs to real numbers. In particular, for the case in which instances are represented as feature vectors $\boldsymbol{x} = (x_1, \ldots, x_d)$ and the base functions are of the form

$$h_{ki}(\boldsymbol{x}, y) = \begin{cases} x_k & y = y_i \\ 0 & y \neq y_i \end{cases} \qquad (1 \leq k \leq d, 1 \leq i \leq n), \tag{3.3}$$

the model is essentially equivalent to constraint classification, as it amounts to learning label-specific utility functions (3.1). Algorithmically, however, the underlying optimization problem is approached in a different way, by means of a boosting-based algorithm that seeks to minimize a generalized

ranking error

$$l(\boldsymbol{f}, \mathbf{G}) = \sum_{i=1}^{m} \frac{1}{|G_i|} \sum_{G_i} \log\left(1 + \exp(f_k(\boldsymbol{x}_i) - f_{k'}(\boldsymbol{x}_i))\right) \tag{3.4}$$

in an iterative way, where $G_i = \{(k, k') \mid y_k \succ_{\boldsymbol{x}_i} y_{k'}\}$ is the set of pairwise preferences associated with instance $\boldsymbol{x}_i$. The corresponding pseudo code, a modified version of the one stated in [17], can be found at Algorithm 2.

Algorithm 2 A boosting-based algorithm for label ranking

Require: training data of size m as defined in Table 2.3 and a set of base ranking functions $\{h_0, \ldots, h_{d \cdot n}\}$ in the form of Equation (3.3)

Ensure: a corresponding weight vector $v_1, \ldots, v_{d \cdot n} \in \mathbb{R}$ for base ranking functions

Initialize:

$\boldsymbol{v}_1 = \{0, \ldots, 0\}$

$\pi_{i,p,j} = h_j(\boldsymbol{x}_i, \text{term}(p)) - h_j(\boldsymbol{x}_i, \text{init}(p))$, with $1 \leq i \leq m$, $1 \leq j \leq d \cdot n$, $p \in \{y_k \succ_{\boldsymbol{x}_i} y_{k'}\}$, and for $p = a \succ b, \text{init}(p) = a, \text{term}(p) = b$

$z = \max_{i,p} \sum_j |\pi_{i,p,j}|$

Iterate:

for $t = 1, 2, \ldots$ **do**

  $q_{t,i,p} = \frac{\exp\langle \boldsymbol{v}_t, \boldsymbol{\pi}_{i,p} \rangle}{1 + \exp\langle \boldsymbol{v}_t, \boldsymbol{\pi}_{i,p} \rangle}$, with $1 \leq i \leq m, p \in \{y_k \succ_{\boldsymbol{x}_i} y_{k'}\}$

  $w^+_{t,j} = \sum_{i,p:\pi_{i,p,j}>0} \frac{q_{t,i,p}\pi_{i,p,j}}{d \cdot n}$ and $w^-_{t,j} = \sum_{i,p:\pi_{i,p,j}<0} \frac{-q_{t,i,p}\pi_{i,p,j}}{d \cdot n}$, with $1 \leq j \leq d \cdot n$

  $\lambda_{t,j} = \frac{1}{2} \ln\left(\frac{w^+_{t,j}}{w^-_{t,j}}\right)$, with $1 \leq j \leq d \cdot n$

  $\boldsymbol{v}_{t+1} = \boldsymbol{v}_t - \frac{\boldsymbol{\lambda}_t}{z}$

end for

3.1.3 Related Methods

The maximum-margin approach [24] proposed for multi-label classification has a straightforward generalization to the label ranking problem. This approach tries to minimize the rank loss defined as

$$l(f, G_i) = \frac{1}{|G_i|} \left| f_p(\boldsymbol{x}_i) \leq f_q(\boldsymbol{x}_i) \right| , \tag{3.5}$$

where $f_p(\boldsymbol{x}_i) = \langle \boldsymbol{w}_p, \boldsymbol{x}_i \rangle$ and $G_i = \{(p,q) \mid y_q \succ_{\boldsymbol{x}_i} y_p\}$ is the set of pairwise preferences associated with instance $\boldsymbol{x}_i$. The corresponding optimization problem can be formalized as follows:

$$\begin{aligned} \min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_n} & \sum_{j=1}^{n} \|\boldsymbol{w}_j\|^2 + C \sum_{i=1}^{m} \frac{1}{|G_i|} \sum_{(p,q)\in G_i} \xi_{ipq} \\ \text{subject to: } & \langle \boldsymbol{w}_p - \boldsymbol{w}_q, \boldsymbol{x}_i \rangle \geq 1 - \xi_{ipq}, \\ & \xi_{ipq} \geq 0, \\ & \text{for all } (p,q) \in G_i, \forall i = 1, \ldots, m\,, \end{aligned} \tag{3.6}$$

where $C > 0$ is the hyper-parameter that balances the loss term and the regularization term. This formulation is closely related to Algorithm 1, the online constraint classification for label ranking: (3.6) can be considered as a regularized, maximum margin, batch version of Algorithm 1. Despite a higher computation cost, (3.6) has a better generalization guarantee. The empirical performance of both algorithms are, however, generally quite comparable [36].

The method proposed in [24] is further generalized in [55], where one assumes the existence of a feedback vector $\boldsymbol{v} \in \mathbb{R}^n$ that can be induced by a decomposition framework on the preference graphs of labels. Moreover, $y_i \succ_{\boldsymbol{x}} y_j$ if and only if $v_i > v_j$, and the difference $v_i - v_j$, representing the importance of the pairwise preference $y_i \succ_{\boldsymbol{x}} y_j$, is used in the optimization problem. The loss function considered in this work is a generalized hinge-loss for label ranking defined as follows:

$$l_{i,j}(\boldsymbol{f}, \boldsymbol{v}) = \left[(v_i - v_j) - (f_i(\boldsymbol{x}) - f_j(\boldsymbol{x}))\right]_+\,, \tag{3.7}$$

where $f_i(\boldsymbol{x}) = \langle \boldsymbol{w}_i, \boldsymbol{x} \rangle$ and $[a]_+ = \max(a, 0)$. The form of the feedback vector $\boldsymbol{v}$ can be very flexible and hence makes this method a very general one: The quadratic programming formulation in [24] can be recovered as a special case of this method.

3.2 Label Ranking by Learning Pairwise Preferences

Label ranking by learning pairwise preferences is motivated by the idea of the one-vs-one framework, a decomposition technique extensively used in multi-class classification [26]. One-vs-one is a pairwise learning framework known by a variety of names, such as all pairs, round robin, etc. The key idea is to transform a n-class problem with class labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$ into $n(n-1)/2$ binary problems, one for each pair of class labels. For each pair of labels $(y_i, y_j) \in \mathcal{Y} \times \mathcal{Y}, 1 \leq i < j \leq n$, a separate model $\mathcal{M}_{ij}$ is trained using the instances from these two labels as the training set. A model $\mathcal{M}_{ij}$ is intended to separate the objects with label y_i from those having label y_j. At classification time, a query instance $\boldsymbol{x} \in \mathcal{X}$ is submitted to all models $\mathcal{M}_{ij}$, and the predictions $\mathcal{M}_{ij}(\boldsymbol{x})$ are combined into an overall prediction. Often, the prediction $\mathcal{M}_{ij}(\boldsymbol{x})$ is interpreted as a vote for either y_i or y_j, and the label with the highest overall votes is proposed as the final prediction. Comparing to alternative decomposition techniques, such as the one-vs-all approach which learns one model for each label, the one-vs-one approach often leads to simpler problems. In particular, since all instances having neither of the two labels are ignored, pairwise problems contain fewer training instances and are hence computationally less complex. Moreover, these problems typically lead to simpler decision boundaries. See an illustration in Figure 3.1.

To demonstrate how the one-vs-one decomposition principle can be applied in a label ranking problem, we illustrate an example in Figure 3.2. Even though we assume the existence of an underlying ranking, we do not expect the training data to provide full information about this ranking. Inconsistencies may also appear, such as pairwise preferences that conflict with each other (e.g., lead to cycles). At prediction time, similar to classification, a query instance $\boldsymbol{x} \in \mathcal{X}$ is submitted to all learned models $\mathcal{M}_{ij}$, and the prediction $\mathcal{M}_{ij}(\boldsymbol{x})$ is often interpreted as a vote for either y_i or y_j. Instead

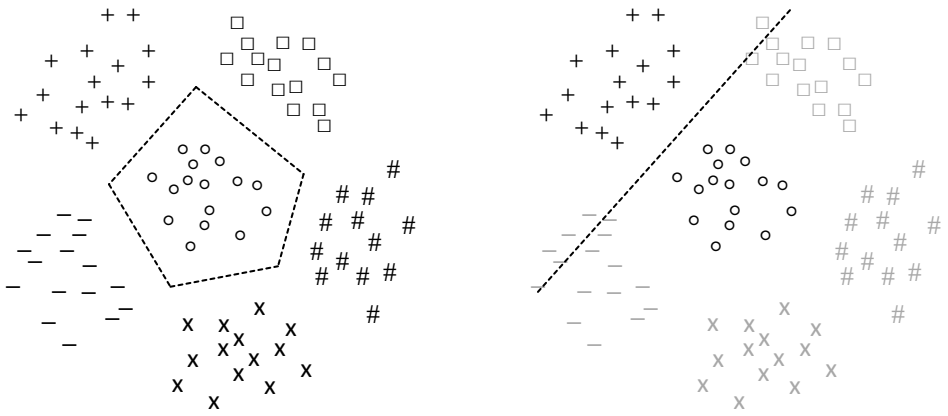

Figure 3.1: One-vs-all classification (figure on left) transforms a 6-class problem into 6 binary problems, one for each class, where each of these problems uses the instances of its class label as the positive ones (here ∘), and all other instances as negative ones. One-vs-one classification (figure on right) solves $6 \cdot (6-1)/2$ binary problems, one for each pair of labels (here ∘ and +) ignoring the instances with other labels.

of outputting the label with the highest value of votes, a ranking of labels is generated according to their scores (i.e., replacing the arg max operation with arg sort).

3.2.1 Complexity Analysis

In this section, we discuss the runtime complexity of the previously mentioned label ranking methods. Let $|G_i|$ be the number of pairwise preferences that are associated with instance $\boldsymbol{x}_i$, we denote by $z = 1/m \cdot \sum_i |G_i|$ the average number of pairwise preferences over all instances throughout this section. The two following theorems, due to [26] and [36], serve as a basic guideline for choosing between RPC and CC in practice, as long as the runtime requirement is a major concern:

Theorem 1. *For a base learner with complexity $\mathcal{O}(m^c)$, the complexity of Label Ranking by Learning Pairwise Preferences (RPC for short) is $\mathcal{O}(zm^c)$.*

Proof. Let m_{ij} be the number of training instances for model $\mathcal{M}_{ij}$. Each

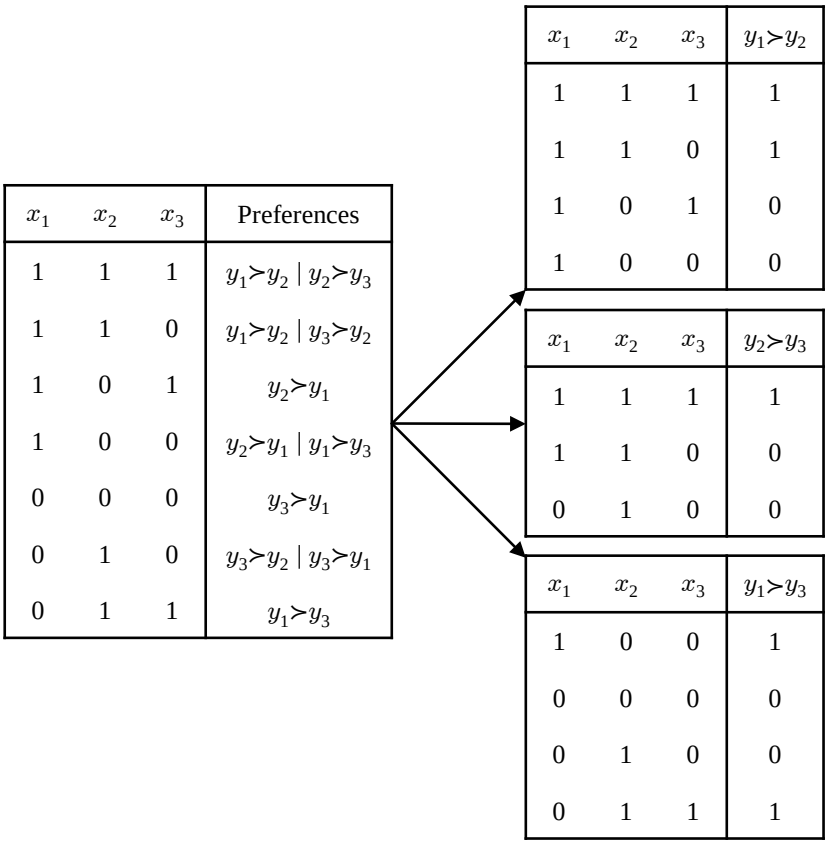

Figure 3.2: The decomposition scheme of label ranking by learning pairwise preferences. In the original data, each instance is associated with a subset of pairwise preferences. According to these pairwise preferences, a set of corresponding binary classification data is established.

instance corresponds to a single preference, i.e.,

$$\sum_{1 \leq i < j \leq n} m_{ij} = \sum_{k=1}^{m} |G_i| = zm \tag{3.8}$$

and the total learning complexity is $\sum \mathcal{O}(m_{ij}^c)$. We now obtain

$$\begin{aligned}\frac{\sum \mathcal{O}(m_{ij}^c)}{\mathcal{O}(zm^c)} &= \frac{1}{z}\sum \frac{\mathcal{O}(m_{ij}^c)}{\mathcal{O}(m^c)} = \frac{1}{z}\sum \mathcal{O}\left(\left(\frac{m_{ij}}{m}\right)^c\right) \\ &\leq \frac{1}{z}\sum \mathcal{O}\left(\frac{m_{ij}}{m}\right) = \frac{\sum \mathcal{O}(m_{ij})}{z\,\mathcal{O}(m)} = \frac{\mathcal{O}(\sum m_{ij})}{\mathcal{O}(zm)} = \frac{\mathcal{O}(zm)}{\mathcal{O}(zm)} \\ &= \mathcal{O}(1)\ .\end{aligned} \tag{3.9}$$

This inequality holds since each instance has at most one preference involving the label pair (y_i, y_j), and hence $m_{ij} \leq m$. □

Theorem 2. *For a base learner with complexity $\mathcal{O}(m^c)$, the complexity of constraint classification (CC for short) is $\mathcal{O}(z^c m^c)$.*

Proof. CC transforms the original training data into a set of $2\sum_{i=1}^{m}|G_i| = 2zm$ instances, which means that CC constructs twice as many training examples as RPC. If this problem is solved with a base learner with complexity $\mathcal{O}(m^c)$, the total complexity is $\mathcal{O}\left((2zm)^c\right) = \mathcal{O}(z^c m^c)$. □

Generally, for a base learner with a polynomial time complexity, RPC is at least as efficient as CC; but in cases where the base learner has a sub-linear time complexity (i.e., $c < 1$), CC is faster. In practice, of course, many other factors have to be taken into consideration. For example, given a base learner with a linear runtime (and hence the same total runtime complexity for both RPC and CC), CC might be preferable due to the quadratic numbers of models RPC needs to store for binary predictions.

A direct comparison is less obvious for the online version and other large-margin variants of CC, since the complexity strongly depends on the number of iterations needed to achieve convergence for the former and the selected optimization routine for the latter. For the online version of CC, as depicted in Algorithm 1, the algorithm checks all constraints for every instance in a single iteration and, in case a constraint is violated, adapts the weight vector correspondingly. The complexity is hence $\mathcal{O}(zdmt)$, where d is the number of features of an instance and t is the number of iterations.

The complexity for the boosting-based algorithm proposed for log-linear models also depends on the number of iterations. In each iteration, the algorithm essentially updates the weights that are associated with each instance and preference constraint. The complexity of this step is $\mathcal{O}(zm)$. Moreover, the algorithm maintains the weights for each base ranking function. If specified as in (3.3), the number of these functions is dn. Therefore, the total complexity is $\mathcal{O}((zm + dn) \cdot t)$, with t iterations.

3.3 Case-Based Label Ranking

Instance-based or case-based learning algorithms have been applied successfully in various fields, such as machine learning and pattern recognition, for a long time [1, 48]. The key characteristic of instance-based learning algorithms, which distinguishes them from global function approximation approaches, i.e., model-based approaches, is they don't form the target functions directly based on the entire instance space. Instead, the target functions are formed locally, dependent on the query instances. Often, the training instances (or a selection thereof) are stored but not processed until an estimation for a new instance is requested. A different local approximation may be obtained for each query instance. As a result, instance-based learning comes along with a number of advantages. Since the training instances are explicitly stored, the information present in the data is always preserved; and as the target function is estimated locally, highly complex hypotheses can be formulated. We shall come back to these advantages in Chapter 4.

Among other instance-based approaches (e.g., locally weighted regression, radial basis function, etc.), k-nearest neighbor (KNN) approach is the most prominent one, which has been thoroughly analyzed in machine learning. The popularity of KNN is partially due to its simplicity: For a query instance, KNN first retrieves the k most "similar" training instances, and the estimation for this query instance is then given by an aggregation of these instances' outputs. In classification, the mostly applied aggregation operator

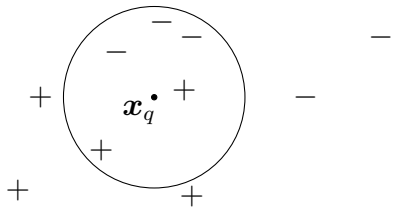

Figure 3.3: An illustration of KNN (with the Euclidean distance) for binary classification. The query instance $\boldsymbol{x}_q$ will be classified as positive with 1NN, and negative with 5NN.

is majority voting (i.e., mode of the output classes), while in regression the mean and the median are often used. See Figure 3.3 for an illustration of the KNN approach for classification.

When applying instance-based approaches to label ranking, the aggregation step becomes much more challenging. It essentially boils down to the ranking aggregation problem. Ranking aggregation is a special case of the weighted feedback arc set problem [2]. Informally, the goal is to combine many different rankings on the same set of objects, in order to obtain a "better" ranking that is close to these given rankings. Ranking aggregation has been studied in many disciplines, most extensively in the context of social choice theory. It has gained much attention in the field of computer science in recent years. A number of applications make use of its results, such as the meta-search problem mentioned in Section 2.1.

An intuitive approach to ranking aggregation is majority voting with the pairwise preferences on objects, but the optimality is not guaranteed. Already in the 17th century, Condorcet has shown that the majority preferences can be irrational: The majority may prefer pairwise preferences that lead to a cycle. This observation is often referred to as "The Condorcet Paradox". Indeed, even considering preferences of 3 individuals with 3 objects, e.g., $y_1 \succ y_2 \succ y_3, y_2 \succ y_3 \succ y_1$, and $y_3 \succ y_1 \succ y_2$, we already have 2/3 of the group prefer y_1 to y_2, 2/3 prefer y_2 to y_3, and 2/3 prefer y_3 to y_1. McGarvey further showed that the majority may exhibit any pattern of pairwise preferences (Figure 3.4) [47]. Moreover, it was shown by Arrow [4] that, for

$y_1 \succ y_2 \succ y_3 \succ y_4 \succ y_5$
$y_3 \succ y_5 \succ y_2 \succ y_4 \succ y_1$
$y_4 \succ y_5 \succ y_1 \succ y_2 \succ y_3$

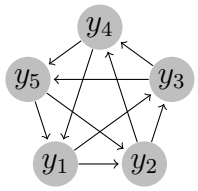

Figure 3.4: Preferences from individuals and the corresponding preference graph based on the majority voting on pairs. The direction of the arrows indicates the preference of the majority. Formally, every tournament on n vertices is a $2k-1$ majority tournament for a large enough k, where a tournament is an oriented complete graph and it is a $2k-1$ majority tournament if there are $2k-1$ linear orders on the vertices, and $y_i \to y_j$ if and only if y_i precedes y_j in at least k of them.

3 or more objects, there is no voting scheme that satisfies (a) unanimity, (b) independence of irrelevant alternatives, and (c) non-dictatorship.[1] This result is known as Arrow's impossibility theorem.

A widely accepted objective for aggregating preferences, if each individual provides a complete ranking of the objects, is the Kemeny optimum, which is defined with the Kendall distance (2.2). Kemeny-optimal ranking aggregation seeks a ranking π that minimizes the number of pairwise disagreements with the input rankings $\sigma_1, \ldots, \sigma_k$, i.e., $\arg\min_{\pi\in\Omega} \sum_{i=1,\ldots,k} T(\pi, \sigma_i)$. Kemeny-optimal ranking satisfies the generalized Condorcet criterion [23]:

Theorem 3. *Let π be a Kemeny-optimal ranking. If Y and Y' partition the set of objects, and for every $y \in Y$ and $y' \in Y'$ the majority ranks y ahead of y', then $\pi(y) < \pi(y')$ for every $y \in Y$ and $y' \in Y'$.*

Loosely speaking, the generalized Condorcet criterion has a partition step in addition to the majority voting and hence the Kemeny-optimal ranking can be seen as an approximation of the majority voting result.

While it is not hard to compute the Kendall distance T for n objects in $O(n \log n)$, finding a Kemeny-optimal ranking aggregation is known to be

[1]Unanimity: If all individuals rank y_i above y_j, then so does the resulting order. Independence of irrelevant alternatives: The group's relative ranking of any pair of objects is determined by the individuals' relative ranking of this pair. Non-dictatorship: The group's ranking is not determined by that of one individual.

NP-hard, even in a special case of four individuals [23]. In recent years, many efforts have been made in theoretical computer science in order to produce good approximations. Several algorithms are known with performance guarantees within a factor two or less of the optimal one [51]. The very first polynomial-time approximation scheme (PTAS) for finding the Kemeny optimum was proposed by Kenyon-Mathieu and Schudy [41].[2]

The idea of using the instance-based framework for label ranking has been pioneered by Brinker and Hüllermeier [10]. For aggregating the rankings of the neighbors, they make use of the Borda count method, which can be traced back to the 17th century. Given a ranking σ_i of n labels, the top-ranked label receives n votes, the second-ranked $n-1$ votes, and so on. Given k rankings $\sigma_1, \ldots, \sigma_k$, the sum of the k votes are computed for each label, and the labels are then ranked according to their total votes. Despite its simplicity, Borda count is provably optimal for minimizing the sum of the Spearman distances, and correspondingly, maximizing the sum of the Spearman's rank correlation coefficients [35]. As we discussed in Section 2.3, due to the tight relations between the widely used distance measures for rankings, Borda count often leads to satisfactory results for other measures as well.

The methods proposed in this thesis extend the one proposed in [10]. Since the conventional Borda count operates only on complete rankings, the application of [10] is limited to complete rankings, too. One natural question is how the instance-based framework can be generalized to the incomplete ranking case, while preserving the optimality with respect to some measures on rankings. We will come back to this issue in Chapter 4.

[2] A PTAS is an algorithm that for any fixed $\epsilon > 0$ produces, in polynomial time, a solution that is within a factor $1 + \epsilon$ of being optimal.

3.4 Chapter Conclusions

This chapter has covered a wide spectrum of methods for label ranking learning. Most methods we discussed will be empirically tested against our methods in the forthcoming chapters. Needless to say, there are more methods that can be used for label ranking than the ones mentioned here.

The outputs in label ranking have a complex structure, so in a sense, it can be considered as a particular type of structured prediction [5]. Roughly speaking, structured prediction algorithms infer a joint scoring function on input-output pairs and, for a given input, predict the output that maximizes this scoring function. The scoring function is parameterized by a weight vector $\mathbf{w}$ and is defined as $f(\mathbf{x}, y; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, y) \rangle$. Here, $\Phi(\mathbf{x}, y)$ defines the (possibly infinite dimensional) feature map of an input-output pair. The prediction rule can then be written as $\hat{y} = \arg\max_{y \in \mathcal{Y}} f(x, y) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}, \Phi(\mathbf{x}, y) \rangle$. Hence, the setting is reduced to a label ranking framework, if $\mathcal{Y}$ corresponds to the space for all possible label rankings.

Other types of classification algorithms can be modified for label ranking learning as well. A notable example is [11], where the authors make use of tree-based models for label ranking. In a decision tree, each leaf node represents a (typically rectangular) part of the instance space and is labeled with a local model for prediction. In regression, the model is given in the form of a constant or linear function, while in classification, it is simply a class assignment. In [11], the leaf nodes of decision trees are associated with (possibly incomplete) label rankings.

Despite many possible variations of methods, the distinctions of label ranking by (a) learning utility functions, (b) learning pairwise preference, and (c) case-based approaches are very general and should cover most of the existing label ranking methods.

Chapter 4

Instance-Based Label Ranking with Probabilistic Models

We have discussed various approaches to label ranking in Chapter 3. Existing methods for label ranking are typically extensions of binary classification algorithms. For example, ranking by pairwise comparison (RPC) is an extension of pairwise classification [36], while constraint classification (CC) and log-linear models for label ranking (LL) seek to learn linear utility functions for each individual label instead of preference predicates for pairs of labels [33, 17].

Even though these approaches have shown good performance in the empirical studies [36], the reduction of the complex label ranking problem to the simple binary classification problem is not self-evident and does not come for free. Such reduction becomes possible only through the use of an ensemble of binary models; in CC and LL, the size of this ensemble is linear in the number of labels, while in RPC it is quadratic. Some problems come along with such an ensemble. First, the representation of a "ranking-valued" mapping in terms of an aggregation (e.g., argsort) of an ensemble of simple mappings (e.g., real-valued utility functions) typically comes along with a strong bias. This is especially true for methods such as constraint classification, for which

the transformation from ranking to classification strongly exploits the linearity of the underlying utility functions. Likewise, it is often not clear (and mostly even wrong) that minimizing the classification error, or a related loss function, on the binary problems leads to maximizing the (expected) performance of the label ranking model in terms of the desired evaluation function on rankings [22]. A proper aggregation of the ensemble results is challenging for many performance measures on rankings. Second, a representation in terms of an ensemble of models is not always desired, mainly since single models are considered more comprehensible and interpretable. This point is particularly relevant for the pairwise approach, as the size of the model ensemble is quadratic in the number of class labels. Comprehensibility and interpretability of a model are critical for certain learning tasks, such as the decision making processes in, e.g., medical applications.

To overcome these problems, we advocate extensions of instance-based learning to the label ranking setting. They are based on local estimation principles, which are known to have a rather weak bias. Instance-based or case-based learning algorithms simply store the training data, or at least a selection thereof, and defer the processing of these data until an estimation for a new instance is requested, a property distinguishing them from typical model-based approaches. Instance-based approaches therefore have a number of potential advantages, especially in the context of the label ranking problem.

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

appealing if Ω has a complex structure. In analogy with the classification setting, we do not assume such mapping is deterministic. Instead, every instance is associated with a probability distribution over Ω. This means, for each $\boldsymbol{x} \in \mathcal{X}$, there exists a probability distribution $\Pr(\cdot \,|\, \boldsymbol{x})$ such that, for every $\sigma \in \Omega$, $\Pr(\sigma \,|\, \boldsymbol{x})$ is the probability that $\boldsymbol{x}$ having ranking σ, i.e., $\sigma_{\boldsymbol{x}} = \sigma$.

In label ranking, Ω corresponds to the set of all rankings of an underlying label set $\mathcal{L}$. To represent an Ω-valued mapping, the aforementioned reduction approaches encode this mapping in terms of conventional binary models, either by a large set of such models in the original label space $\mathcal{L}$ (RPC), or by a single binary model in an expanded, high-dimensional space (CC, LL). Since for instance-based methods, there is no need to represent an $\mathcal{X} \to \Omega$ mapping explicitly, such methods can operate on the original target space Ω directly.

This chapter is organized as follows: We first introduce two probability models for rankings in Section 4.1. The core idea of our instance-based local approach to label ranking, namely maximum likelihood estimation based on probability models for rankings, is discussed in Section 4.2. Section 4.3 is devoted to experimental results. The chapter ends with concluding remarks in Section 4.4.

4.1 Probability Models for Rankings

So far, we did not make any assumptions about the probability measure $\Pr(\cdot \,|\, \boldsymbol{x})$ despite its existence. In statistics, different types of probability distributions on rankings have been proposed. A detailed review can be found in [45]. Roughly speaking, two ways of modeling rankings have been developed in the literature: (a) modeling the population of the rankers; and (b) modeling the ranking process. While the first approach is more data-analytic, trying to describe parametrically the distribution of rankings attached to a population of rankers, the second approach tries to describe the underlying

processes that a ranker undergoes to produce the rankings. In this section, we introduce two widely-used models, the Mallows model and the Plackett-Luce (PL) model, which fall into these two categories respectively. More specifically, the Mallows model is a distance-based model, where one often assumes there is a center ranking $\pi \in \Omega$ and the observed rankings are more or less close to π. An appropriate model gives higher probability to rankings closer to π. On the other hand, the PL model is a multi-stage model, where one assumes a ranking is produced in a stagewise way: First, one considers which object should be ranked first, and then which object should be ranked second, so on and and so forth. We begin our discussion with the Mallows model.

4.1.1 The Mallows Model

The Mallows model is a distance-based probability model first introduced by Mallows in the 1950s [44]. The standard Mallows model is a two-parameter model that belongs to the exponential family:

$$\Pr(\sigma \,|\, \theta, \pi) = \frac{\exp(-\theta T(\sigma, \pi))}{\phi(\theta, \pi)} \,, \tag{4.1}$$

where the two parameters are the center ranking (modal ranking, location parameter) $\pi \in \Omega$ and the spread parameter $\theta \geq 0$. Here, $\phi(\theta, \pi)$ is the normalization constatnt. The Mallows model assigns the maximum probability to the center ranking π. The larger the Kendall distance $T(\sigma, \pi)$, the smaller the probability of σ becomes. The spread parameter θ determines how quickly the probability decreases, i.e., how peaked the distribution is around π. For $\theta = 0$, the uniform distribution is obtained, while for $\theta \to \infty$, the distribution converges to the one-point distribution that assigns probability 1 to π and 0 to all other rankings.

For a right-invariant metric D, it can be shown that the normalization constant does not depend on π and, therefore, can be written as a function

$\phi(\theta)$ of θ alone. This is due to

$$\begin{aligned}\phi(\theta,\pi) &= \sum_{\sigma\in\Omega}\exp(-\theta D(\sigma,\pi)) \\ &= \sum_{\sigma\in\Omega}\exp(-\theta D(\sigma\pi^{-1},e)) \\ &= \sum_{\sigma'\in\Omega}\exp(-\theta D(\sigma',e)) = \phi(\theta)\ ,\end{aligned} \tag{4.2}$$

where $e=(1,\ldots,n)$ stands for the identity ranking. Moreover for $D=T$, it can be shown that (see, e.g., [25]) the normalization constant is given by

$$\phi(\theta) = \prod_{j=1}^{n}\frac{1-\exp(-j\theta)}{1-\exp(-\theta)}\ , \tag{4.3}$$

and the expected distance from the center is

$$\mathbb{E}\left[T(\sigma,\pi)\,|\,\theta,\pi\right] = \frac{n\exp(-\theta)}{1-\exp(-\theta)} - \sum_{j=1}^{n}\frac{-j\exp(j\theta)}{1-\exp(-j\theta)}\ . \tag{4.4}$$

The model we discussed here is referred as the Mallows ϕ model in statistics, where the Kendall distance T is used. Applying other distance measures leads to different distance-based models. Especially, replacing T with the Spearman distance S yields the Mallows θ model. But then (4.3) and (4.4) generally do not hold anymore, which often leads to higher computation cost. Notice that in the case when the normalization is no longer a function of the spread θ alone, enumerating Ω can be very costly.

4.1.2 The Plackett-Luce Model

First studied by Luce [43] and subsequently by Plackett [49], the PL model is specified by a parameter vector $\boldsymbol{v}=(v_1,\ldots,v_n)\in\mathbb{R}_+^n$:

$$\Pr(\sigma\,|\,\boldsymbol{v}) \;=\; \prod_{i=1}^{n}\frac{v_{\sigma^{-1}(i)}}{v_{\sigma^{-1}(i)}+v_{\sigma^{-1}(i+1)}+\ldots+v_{\sigma^{-1}(n)}}\ . \tag{4.5}$$

This model is a generalization of the well-known Bradley-Terry model, a model for the pairwise comparison of alternatives, which specifies the probability that "a wins against b" in terms of

$$\Pr(a \succ b) = \frac{v_a}{v_a + v_b} . \tag{4.6}$$

Obviously, the larger v_a in comparison to v_b, the higher the probability that a is chosen. Likewise, the larger the parameter v_i in (4.5) in comparison to the parameters v_j, $j \neq i$, the higher the probability that the label y_i appears on a top rank. Hence, the parameter vector $\boldsymbol{v}$ is often referred to as a "skill" vector indicating each object's skill, score, popularity, etc. An intuitively appealing explanation of the PL model can be given by a vase model: If v_i corresponds to the relative frequency of the i-th label in a vase filled with labeled balls, then $\Pr(\sigma \,|\, \boldsymbol{v})$ is the probability to produce the ranking σ by randomly drawing balls from the vase in a sequential way and putting the label drawn in the k-th trial on position k (unless the label was already chosen before, in which case the trial is annulled).

4.1.3 Other Models

In addition to the distance-based model and the multi-stage model, two other types of ranking models are often found in the statistical literature: (a) the order statistics model and (b) the paired comparison model.[1] We briefly introduce these two models and discuss their relation to the models we previously introduced.

An order statistic model is often called a Thurstonian model as it is pioneered by Thurstone during the 1920s [57]. In a general order statistic model, a joint model is assumed for the vector $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$, where z_i is a continuous but unobserved random variable associated with label y_i.

[1]The terms "paired" and "pairwise" are used exchangeably in this thesis. Depending on the context, the choices between these two terms are made in order to be consistent with the literature.

The ordering of labels is given by the vector $\boldsymbol{z}$, that is

$$y_{i_1} \succ y_{i_2} \succ \ldots \succ y_{i_n} \Leftrightarrow z_{i_1} > z_{i_2} > \ldots > z_{i_n}. \tag{4.7}$$

It essentially corresponds to the utility-based label ranking setting that we discussed in Section 3.1. In Thurstone's original paper, he proposed that $\boldsymbol{z}$ follows a Gaussian distribution, and hence the model parameters include n means, n variances, and $n(n-1)/2$ correlations. Straightforward simplifications of this setting were also proposed in that paper, such as equating the correlations, equating the variances, or assuming z_i's are independent, i.e, setting the correlations to zero. It is further showed by Yellott that, if $\boldsymbol{z}$ follows the Gumbel distribution function $G(z) = \exp(-\exp(-z))$ for $z \in \mathbb{R}$, this model turns out to be the same as the PL model [66].

A paired comparison model is often referred to as a Babington Smith model in statistics. Given a ranking of n items, $n(n-1)/2$ pairwise preferences can be easily identified; but it is not always straightforward to recover a ranking from a set of pairwise preferences (see Section 3.3). A general paired comparison model constructs a ranking by starting with pairwise preferences, but only the consistent set of preferences are considered. Given a ranking σ, it has the density

$$\Pr(\sigma) = \frac{n!}{c(\boldsymbol{p})} \prod_{(i,j):\, \sigma(i)<\sigma(j)} p_{ij}\,, \tag{4.8}$$

where the model parameter $\boldsymbol{p}$ is a vector of size $n(n-1)/2$ indexed by i and $j, i < j$. The p_{ij} equals $\Pr(y_i \succ y_j)$, which is the probability that label y_i is preferred to label y_j. A direct use of the general paired comparison model is of less practical interest, especially when the number of items to be ranked is large: It has a quadratic number of parameters with respect to n and the normalization constant $c(\boldsymbol{p})$ sums up $n!$ products of $n(n-1)/2$ terms. Usually, simplifications are made by restricting p_{ij} to a certain form, such as defining it with the Bradley-Terry model (4.6).

4.2 Instance-Based Label Ranking

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

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

$$\Pr(\boldsymbol{\sigma} \,|\, \boldsymbol{\omega}) = \prod_{i=1}^{k} \Pr(\sigma_i \,|\, \boldsymbol{\omega}) \; . \tag{4.9}$$

The model parameters $\boldsymbol{\omega}$ are then trained through a learning process. A common way of doing this is to fit the data with the maximum likelihood principle, leading to the maximum likelihood estimation (MLE). In the following sections we respectively study the parameter estimation for the Mallows and the PL model under this framework.

4.2.1 Ranking with the Mallows Model

In the case of the Mallows model, the model parameters $\boldsymbol{\omega}$ correspond to the center ranking π and the spread θ, and (4.9) becomes

$$\begin{aligned}\Pr(\boldsymbol{\sigma} \,|\, \boldsymbol{\omega}) &= \Pr(\boldsymbol{\sigma} \,|\, \theta, \pi) \\ &= \prod_{i=1}^{k} \Pr(\sigma_i \,|\, \theta, \pi) \\ &= \prod_{i=1}^{k} \frac{\exp\left(-\theta T(\sigma_i, \pi)\right)}{\phi(\theta)} \\ &= \frac{\exp\left(-\theta \sum_{i=1}^{k} T(\sigma_i, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(-j\theta)}{1-\exp(-\theta)}\right)^k} \,.\end{aligned} \tag{4.10}$$

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

$$\hat{\pi} \;=\; \arg\min_{\pi} \sum_{i=1}^{k} T(\sigma_i, \pi) \;, \tag{4.11}$$

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

$$\frac{1}{k} \sum_{i=1}^{k} T(\sigma_i, \hat{\pi}) = \frac{n \exp(-\theta)}{1 - \exp(-\theta)} - \sum_{j=1}^{n} \frac{j \exp(-j\theta)}{1 - \exp(-j\theta)}. \tag{4.12}$$

Since the right-hand side of (4.12) is monotone increasing, a standard line search quickly converges to the MLE of θ [25].

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

probability of σ_i is then given by

$$\Pr(E(\sigma_i)) \;=\; \sum_{\sigma \in E(\sigma_i)} \Pr(\sigma \,|\, \theta, \pi)\;, \tag{4.13}$$

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

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

$$\begin{aligned} \Pr(\boldsymbol{\sigma} \,|\, \theta, \pi) &= \prod_{i=1}^{k} \Pr(E(\sigma_i) \,|\, \theta, \pi) \\ &= \prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \Pr(\sigma \,|\, \theta, \pi) \\ &= \frac{\prod_{i=1}^{k} \sum_{\sigma \in E(\sigma_i)} \exp\left(-\theta T(\sigma, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(-j\theta)}{1-\exp(-\theta)}\right)^k}\;. \end{aligned} \tag{4.14}$$

Computing the MLE of (θ, π) by maximizing this probability now becomes more difficult. For label sets of small to moderate size, say, up to seven, one can afford a straightforward brute force approach, namely an exhaustive search over Ω to find the center ranking π, combined with a numerical procedure to optimize the spread θ. For larger label sets, this procedure becomes too inefficient. Here, we propose an approximation algorithm that can be seen as an instance of the EM (expectation-maximization) family [19].

Our algorithm works as follows (see Algorithm 3). Starting from an initial center ranking $\pi \in \Omega$, each incomplete neighboring ranking σ_i is replaced by the most probable linear extension, i.e., by the ranking $\sigma_i^* \in E(\sigma_i)$ whose probability is maximal given $\hat{\pi}$ as a center (first M-step). Having replaced all neighboring rankings by their most probable extensions, an MLE $(\hat{\theta}, \hat{\pi})$ can be derived as described for the case of complete rankings above (second M-step). The center ranking π is then replaced by $\hat{\pi}$, and the whole procedure

Algorithm 3 IB-M

Require: query $\boldsymbol{x} \in \mathcal{X}$, training data $\mathcal{T}$, integer k
Ensure: label ranking estimation for $\boldsymbol{x}$

1: find the k nearest neighbors of $\boldsymbol{x}$ in $\mathcal{T}$
2: get neighboring rankings $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_k\}$
3: use generalized Borda count to get $\hat{\pi}$ from $\boldsymbol{\sigma}$
4: **for** every ranking $\sigma_i \in \boldsymbol{\sigma}$ **do**
5: **if** σ_i is incomplete **then**
6: $\sigma_i^* \leftarrow$ most probable extension of σ_i given $\hat{\pi}$
7: **end if**
8: **end for**
9: use Borda count to get π from $\boldsymbol{\sigma}^* = \{\sigma_1^*, \ldots, \sigma_k^*\}$
10: **if** $\pi \neq \hat{\pi}$ **then**
11: $\hat{\pi} \leftarrow \pi$
12: go to Step 4
13: **else**
14: estimate $\hat{\theta}$ given $\hat{\pi}$ and $\boldsymbol{\sigma}^*$
15: return $(\hat{\pi}, \hat{\theta})$
16: **end if**

is iterated until the center does not change any more; $\hat{\pi}$ is then output as a prediction. In the following, we discuss three sub-problems of the algorithm in more detail, namely (a) the problem to find most probable extensions in the first M-step, (b) the solution of the median problem (4.11) in the second M-step, and (c) the choice of an initial center ranking.

(a) Regardless of the spread θ, a most probable extension $\sigma_i^* \in E(\sigma_i)$ of an incomplete ranking σ_i, given π, is obviously a minimizer of $T(\pi, \cdot)$. Such a ranking can be found efficiently, as shown in the following theorem:

Theorem 4. *Let π be a ranking of $Y = \{y_1, y_2, \ldots, y_n\}$, and let σ be a ranking of a subset $C \subseteq Y$ with $|C| = m \leq n$. The linear extension σ^* of σ that minimizes $T(\pi, \cdot)$ can be found as follows. First, each $y_i \in Y \setminus C$ is optimally inserted in σ, i.e., it is inserted between the labels on position j and $j+1$ in σ, where $j \in \{0, \ldots, m\}$ ($j = 0$ means before the first and $j = m$*

after the last label), if j is a minimizer of

$$\begin{aligned} &\#\{y_k \in C \,|\, \sigma(y_k) \leq j \wedge \pi(y_k) > \pi(y_i)\} \\ +\,&\#\{y_k \in C \,|\, \sigma(y_k) > j \wedge \pi(y_k) < \pi(y_i)\}\,. \end{aligned} \tag{4.15}$$

In the case of a tie, the position with the smallest index is chosen. Then, those $y_i \in Y \setminus C$ that are inserted at the same position are put in the same order as in π.

This theorem is a direct result of Lemma 1 and Corollary 1, which are introduced next.

Lemma 1. *If $y_i, y_j \in Y \setminus C$ and $\pi(y_i) < \pi(y_j)$, then $\sigma^*(y_i) < \sigma^*(y_j)$ in the optimal ranking.*

Proof. Suppose that y_j precedes y_i in σ^*, i.e., we have

$$\boxed{L} \quad y_j \quad \boxed{M} \quad y_i \quad \boxed{R}\,.$$

By swapping them we produce

$$\boxed{L} \quad y_i \quad \boxed{M} \quad y_j \quad \boxed{R}\,.$$

The number of introduced conflicts is then

$$\begin{aligned} z = -1 &+ \#\{y_k \in M \,|\, \pi(y_k) > \pi(y_j)\} \\ &- \#\{y_k \in M \,|\, \pi(y_k) < \pi(y_j)\} \\ &+ \#\{y_k \in M \,|\, \pi(y_k) < \pi(y_i)\} \\ &- \#\{y_k \in M \,|\, \pi(y_k) > \pi(y_i)\} \\ = -1 &+ |A_1| - |A_2| + |A_3| - |A_4|\,. \end{aligned} \tag{4.16}$$

Since $A_1 \subseteq A_4$ and $A_3 \subseteq A_2$, we have $z < 0$. Thus, swapping y_i and y_j improves the result, which contradicts the optimality of σ^*. □

Given $\sigma = y_{c_1} \succ y_{c_2} \succ \ldots \succ y_{c_{m-1}} \succ y_{c_m}$ with $C = \{y_{c_1}, \ldots, y_{c_m}\}$, we can put the label $y_i \in Y \setminus C$ into $m + 1$ buckets:

$$\boxed{B_1} \quad y_{c_1} \quad \boxed{B_2} \quad \dots \quad \boxed{B_m} \quad y_{c_m} \quad \boxed{B_{m+1}} \; .$$

The ordering of the labels $y_i \in Y \setminus C$ within the same bucket is straightforward. Let $K(y_i, B_b)$ be the number of conflicts with C produced by putting y_i in bucket B_b, i.e.,

$$\begin{aligned} K(y_i, B_b) \; = & \#\big\{k \in \{1, \dots, m\} \,|\, k < b \wedge \pi(y_i) < \pi(y_{c_k})\big\} \\ & + \#\big\{k \in \{1, \dots, m\} \,|\, k \geq b \wedge \pi(y_i) > \pi(y_{c_k})\big\} \, . \end{aligned} \tag{4.17}$$

Corollary 1. *Let $y_i, y_j \in Y \setminus C$, $\pi(y_i) < \pi(y_j)$. Let*

$$\begin{aligned} b_i &= \arg\min_{b=1,\dots,m+1} K(y_i, B_b) \, , \\ b_j &= \arg\min_{b=1,\dots,m+1} K(y_j, B_b) \, . \end{aligned} \tag{4.18}$$

Then $b_i \leq b_j$.

Proof. Suppose $b_i > b_j$, then $\sigma^*(y_i) > \sigma^*(y_j)$, which leads to a contradiction of Lemma 1. □

From Lemma 1 and Corollary 1 it follows that the optimal ranking is obtained by finding, for each $y_i \in Y \setminus C$, the optimal bucket (position) that minimizes the conflict with C, and ordering the free labels optimally within each bucket. This is exactly the statement of Theorem 4.

Example 1. *Suppose $Y = \{y_1, y_2, y_3, y_4, y_5\}$ and $\sigma = y_2 \succ y_4 \succ y_1$, so we have $C = \{y_1, y_2, y_4\}$, $Y \setminus C = \{y_3, y_5\}$, and $j \in \{0, 1, 2, 3\}$. Further, assume $\pi = y_1 \succ y_2 \succ y_3 \succ y_4 \succ y_5$ is the center ranking.*

We can write the quantity (4.15) as a function of $y_i \in Y \setminus C$ and j. That is,

$$\begin{aligned} f(y_i, j) \; = & \#\{y_k \in C \,|\, \sigma(y_k) \leq j \wedge \pi(y_k) > \pi(y_i)\} \\ & + \#\{y_k \in C \,|\, \sigma(y_k) > j \wedge \pi(y_k) < \pi(y_i)\} \, . \end{aligned} \tag{4.19}$$

Then we have

$$\begin{aligned} &f(y_3,0)=2,\ f(y_3,1)=1,\ f(y_3,2)=2,\ f(y_3,3)=1, \textit{ and} \\ &f(y_5,0)=3,\ f(y_5,1)=2,\ f(y_5,2)=1,\ f(y_5,3)=0. \end{aligned} \tag{4.20}$$

So y_3 is put at position $j=1$ and y_5 is put at position $j=3$. We eventually have $\sigma^* = y_2 \succ y_3 \succ y_4 \succ y_1 \succ y_5$.

(b) Solving the (generalized) median problem (4.11) is known to be NP-hard, i.e., if the distance D is given by the number of rank inversions [3]. To solve this problem approximately, we make use of the fact that the Kendall distance is well approximated by the Spearman distance (see (2.7)), and that the median can be computed for this measure (i.e., for $D(\cdot)$ given by the sum of squared rank differences) by the efficient Borda count procedure [36]: Given a (complete) ranking σ_i of n labels, the top-label receives n votes, the second-ranked $n-1$ votes, and so on. Given k rankings $\sigma_1, \ldots, \sigma_k$, the sum of the k votes are computed for each label, and the labels are then ranked according to their total votes.

(c) The choice of the initial center ranking in the above algorithm is of course critical. To find a good initialization, we again resort to the idea of solving the problem (4.11) approximately using the Borda count principle. At the beginning, however, the neighboring rankings σ_k are still incomplete (and, since there is no π either, cannot be completed by an M-step). To handle this situation, we make the assumption that the completions are uniformly distributed in $E(\sigma_i)$. In other words, we start with the initial guess $\theta = 0$ (uniform distribution). Based on this assumption, we can show the following result that suggests an good initial center π^*.

Theorem 5. *Let a set of incomplete rankings $\sigma_1, \ldots, \sigma_k$ be given, and suppose the associated complete rankings $\sigma_1^*, \ldots, \sigma_k^*$ to be distributed uniformly in $E(\sigma_1), \ldots, E(\sigma_k)$, respectively. The expected sum of distances $D(\pi, \sigma_1^*) + \ldots + D(\pi, \sigma_k^*)$, with D the sum of squared rank distances, becomes minimal for the ranking π^* which is obtained by a generalized Borda count, namely*

a Borda count with a generalized distribution of votes from incomplete rankings: If σ_i is an incomplete ranking of $m \leq n$ labels, then the label on rank $i \in \{1, \ldots, m\}$ receives $(m - i + 1)(n + 1)/(m + 1)$ votes, while each missing label receives a vote of $(n + 1)/2$.

Proof. The optimality follows from the results of [23]. We show in the following how to derive the expected votes of a label with respect to the uniform distribution in $E(\sigma_i)$.

Since $\sigma_1^*, \ldots, \sigma_k^*$ are uniformly distributed in $E(\sigma_1), \ldots, E(\sigma_k)$, respectively, and the sum of all votes for one complete ranking is $n(n + 1)/2$, it is easy to show that each missing label receives a vote of $(n + 1)/2$.

We denote the vote for label at the j-th rank in σ as v_j, where $j = 1, \ldots, m$. The sum of votes of these m labels equals the votes of all labels minus the votes for the $n - m$ missing labels:

$$\begin{aligned} v_1 + \cdots + v_m &= \frac{n(n+1)}{2} - (n-m)\frac{n+1}{2} \\ &= \frac{m(n+1)}{2}\,. \end{aligned} \tag{4.21}$$

With $\delta = v_p - v_{p+1}$, where $p = 1, \ldots, m - 1$, we have

$$\begin{aligned} v_1 + \cdots + v_m &= v_1 + (v_1 - \delta) + \ldots + (v_1 - (m-1)\,\delta) \\ &= m\,v_1 - \frac{m(m-1)}{2}\,\delta\,. \end{aligned} \tag{4.22}$$

From (4.21) and (4.22) we have

$$\delta = \frac{2v_1 - n - 1}{m - 1}\,. \tag{4.23}$$

We have $n - m$ labels that have to be uniformly put into $m + 1$ buckets. So every bucket has on average $(n-m)/(m+1)$ labels. It means, between every adjacent labels among the m labels, there are $\gamma = (n - m)/(m + 1)$ labels.

Since $\delta = \gamma + 1$, we have

$$\frac{2v_1 - n - 1}{m - 1} = \frac{n - m}{m + 1} + 1\,, \tag{4.24}$$

which lead to

$$v_1 = \frac{m(n+1)}{m+1} \text{ and } \delta = \frac{n+1}{m+1}\,. \tag{4.25}$$

Finally with v_1 and δ, we can calculate the votes for all labels in σ. □

As a nice feature of applying the Mallows model, not shared by most existing methods (including reduction techniques) for label ranking, we note that it comes with a natural measure of the reliability of a prediction $\hat{\pi}$, namely the estimation of the spread θ. In fact, the larger the parameter θ, the more peaked the distribution around the center ranking and, therefore, the more reliable this ranking becomes as a prediction.

4.2.2 Ranking with the PL Model

By integrating the Mallows model into the instance-based learning framework, we have avoided the problems encountered by the reduction approaches to some extent. Generally speaking, Section 4.2.1 outlines the basic idea to develop label ranking methods on the basis of statistical models for ranking data, that is, parameterized (conditional) probability distributions on the class of all rankings. Given assumptions of that kind, the learning problem can be posed as a problem of maximum likelihood estimation (or, alternatively, as a problem of Bayesian inference) and thus can be solved in a theoretically sound way. In this section, we advocate the PL model as an alternative, especially since this model is more apt to learning from possibly incomplete label rankings.

Similarly as in Section 4.2.1, we assume that the probability distribution $\Pr(\cdot \mid \boldsymbol{x})$ on Ω is (at least approximately) locally constant around the query $\boldsymbol{x}$. By further assuming that the rankings σ_i have been produced independently of each other by the PL model (4.5), the probability to observe the rankings

$\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_k\}$ in the neighborhood, given the parameters $\boldsymbol{v} = (v_1, \ldots, v_n)$, becomes

$$\Pr(\boldsymbol{\sigma} \mid \boldsymbol{v}) = \prod_{i=1}^{k} \prod_{q=1}^{n_i} \frac{v_{\sigma_i^{-1}(q)}}{\sum_{j=q}^{n_i} v_{\sigma_i^{-1}(j)}} \,, \tag{4.26}$$

where we denote by $n_i \in \{2, \ldots, n\}$ the number of labels ranked by σ_i. Moreover, recall that $\sigma_i^{-1}(j)$ denotes the index of the label ranked on position j. The MLE of $\boldsymbol{v}$ is then given by those parameters that maximize this probability or, equivalently, the log-likelihood function

$$L(\boldsymbol{v}) = \sum_{i=1}^{k} \sum_{q=1}^{n_i} \left[\log \left(v_{\sigma_i^{-1}(q)} \right) - \log \sum_{j=q}^{n_i} v_{\sigma_i^{-1}(j)} \right]. \tag{4.27}$$

Finding the MLE parameters of the PL model is a problem that has already been considered in the statistical literature. We resort to a minorization and maximization (MM) algorithm that is advocated by Hunter [37]. It is an iterative algorithm whose idea is to maximize, in each iteration, a function that minorizes the original log-likelihood, namely

$$Q_t(\boldsymbol{v}) = \sum_{i=1}^{k} \sum_{q=1}^{n_i} \left[\log \left(v_{\sigma_i^{-1}(q)} \right) - \frac{\sum_{j=q}^{n_i} v_{\sigma_i^{-1}(j)}}{\sum_{j=q}^{n_i} v^{(t)}_{\sigma_i^{-1}(j)}} \right]. \tag{4.28}$$

Here, $\boldsymbol{v}^{(t)} = (v_1^{(t)}, \ldots, v_n^{(t)})$ is the estimation of the PL parameters in the t-th iteration. Considering these values as fixed, the problem to maximize $Q_t(\cdot)$ as a function of $\boldsymbol{v}$ can be solved analytically. The corresponding solution, i.e., the parameter vector $\boldsymbol{v}^*$ for which $Q_t(\cdot)$ is maximal, is then used as a new solution: $\boldsymbol{v}^{(t+1)} = \boldsymbol{v}^*$. This procedure provably converges to an MLE estimation of the PL parameters.

Given the MLE $\boldsymbol{v}^*$, a prediction of the ranking associated with $\boldsymbol{x}$ can be derived from the distribution $\Pr(\cdot \mid \boldsymbol{v}^*)$ on Ω. In particular, a MAP (maximum a posteriori) estimate, i.e., a ranking with the highest posterior probability, is given by

$$\sigma^* \in \arg\max_{\sigma \in \Omega} \Pr(\sigma \mid \boldsymbol{v}^*) \,. \tag{4.29}$$

A ranking of this kind can easily be produced by sorting the labels y_i in decreasing order according to the respective parameters v_i^*, i.e., such that

$$v_{(\sigma^*)^{-1}(i)} \geq v_{(\sigma^*)^{-1}(j)} \tag{4.30}$$

for all $1 \leq i < j \leq n$. More generally, given a loss function $\ell(\cdot)$ to be minimized, the best prediction is

$$\sigma^* \;=\; \arg\min_{\sigma \in \Omega} \sum_{\tau \in \Omega} \ell(\sigma, \tau) \cdot \Pr(\tau \mid \boldsymbol{v}^*) \;. \tag{4.31}$$

In general, an interesting question concerns the complexity of the minimization problem (4.31). An explicit computation of the expected loss for each ranking σ is feasible only for a small label set $\mathcal{Y}$, since the cardinality of Ω, which is given by $|\Omega| = |\mathcal{Y}|! = n!$, grows very fast. However, depending on the loss function $\ell(\cdot)$ and the probability distribution $\Pr(\cdot \mid \boldsymbol{v}^*)$, an explicit enumeration of this type can often be avoided.

The PL model appears to be especially appealing from this point of view. In fact, due to the special structure of the probability distribution (4.5), a ranking of the form (4.30) is not only the most intuitive prediction, but also provably optimal for virtually all common loss functions on rankings. In particular, it is a risk minimizer for the 0/1 loss function (defined by $\ell(\sigma^*, \sigma) = 0$ if $\sigma^* = \sigma$ and $= 1$ if $\sigma^* \neq \sigma$) and, likewise, a maximizer of the expected rank correlation in terms of (2.3).

In contrast to other methods that simply produce a prediction in terms of a ranking, a probabilistic approach to label ranking allows one to complement predictions by diverse types of statistical information, for example regarding the reliability of a prediction. In this regard, the PL model shares the same merits as the Mallows model. The distribution $\Pr(\cdot \mid \boldsymbol{v}^*)$ also supports various types of generalized predictions, such as credible sets of rankings covering the true one with a high probability.

4.3 Experiments

In this section, we present an empirical evaluation of the instance-based label ranking framework with the Mallows model (IB-M) and the PL model (IB-PL) as introduced in the previous sections. We use three state-of-the-art methods, constraint classification (CC), log-linear model (LL), and ranking by pairwise comparisons (RPC) as baselines to compare with. Those methods have been already discussed in Chapter 3. Concretely, CC is implemented which its online-variant as proposed in [33], using a noise-tolerant perceptron algorithm as a base learner [42][2]; we use (3.3) as base ranking functions in LL; logistic regression is used as the base learner of RPC, which has been empirically justified in [36]. For IB-M and IB-PL, the neighborhood size $k \in \{5, 10, 15, 20\}$ is selected through cross validation on the training set. As a distance measure in the instance space, we simply used the Euclidean distance (after normalizing the attributes).

4.3.1 Data

We have produced a number of label ranking data sets from real-world applications. Specifically, we resorted to multi-class and regression data sets from the UCI repository and the Statlog collection, turned them into label ranking data in two different ways. (A) For classification data, we followed the procedure proposed in [36]: A naive Bayes classifier is first trained on the complete data set. Then, for each example, all the labels present in the data set are ordered with respect to the predicted class probabilities (in the case of ties, labels with lower index are ranked first). (B) For regression data, a certain number of (numerical) attributes is removed from the set of predictors, and each one is considered as a label. To obtain a ranking, the attributes are standardized and then ordered by size. Given that the original attributes are correlated, the remaining predictive features will contain information about

[2]This algorithm is based on the "α-bound trick" introduced in Section 3.1.1. The corresponding parameter α is set to 500.

the produced ranking. Yet, as will be confirmed by the experimental results, this second type of data generation leads to more difficult learning problems. A summary of the data sets and their properties is given in Table 4.1.[3]

data set	type	# instances	# features	# labels
authorship	A	841	70	4
bodyfat	B	252	7	7
calhousing	B	20640	4	4
cpu-small	B	8192	6	5
elevators	B	16599	9	9
fried	B	40769	9	5
glass	A	214	9	6
housing	B	506	6	6
iris	A	150	4	3
pendigits	A	10992	16	10
segment	A	2310	18	7
stock	B	950	5	5
vehicle	A	846	18	4
vowel	A	528	10	11
wine	A	178	13	3
wisconsin	B	194	16	16

Table 4.1: Label ranking data sets and their properties (the type refers to the way in which the data has been generated).

4.3.2 Results

Results were derived in terms of Kendall's tau correlation coefficient from five repetitions of a ten-fold cross-validation. To model incomplete observations, we modified the training data as follows: A biased coin was 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 $p \in [0, 1]$. Hence, $p \times 100\%$ of the labels will be missing on average.

The summary of the results is shown in Table 4.2. To analyze these results, we performed a two-step statistical test, consisting of a Friedman test

[3]The data sets, along with a detailed description, are available at `http://www.uni-marburg.de/fb12/kebi/research`.

	complete rankings				
	CC	IB-M	IB-PL	LL	RPC
authorship	.920(3)	.936(1)	.936(2)	.657(5)	.910(4)
bodyfat	.281(2)	.229(5)	.230(4)	.266(3)	.285(1)
calhousing	.250(3)	.344(1)	.326(2)	.223(5)	.243(4)
cpu-small	.475(3)	.496(1)	.495(2)	.419(5)	.450(4)
elevators	.768(1)	.727(3)	.721(4)	.701(5)	.749(2)
fried	.999(2)	.900(4)	.894(5)	.989(3)	.999(1)
glass	.846(2)	.842(3)	.841(4)	.818(5)	.882(1)
housing	.660(4)	.736(1)	.711(2)	.626(5)	.671(3)
iris	.836(4)	.925(2)	.960(1)	.818(5)	.885(3)
pendigits	.903(4)	.941(1)	.939(2)	.814(5)	.932(3)
segment	.914(3)	.802(5)	.950(1)	.810(4)	.934(2)
stock	.737(4)	.925(1)	.922(2)	.696(5)	.777(3)
vehicle	.855(3)	.855(2)	.859(1)	.770(5)	.854(4)
vowel	.623(4)	.882(1)	.851(2)	.601(5)	.647(3)
wine	.933(4)	.944(2)	.947(1)	.942(3)	.921(5)
wisconsin	.629(2)	.501(4)	.479(5)	.542(3)	.633(1)
avg. rank	3.00	2.31	2.50	4.44	2.75
	30% missing labels				
	CC	IB-M	IB-PL	LL	RPC
authorship	.891(3)	.913(2)	.927(1)	.656(5)	.884(4)
bodyfat	.260(2)	.198(5)	.204(4)	.251(3)	.272(1)
calhousing	.249(3)	.310(1)	.303(2)	.223(5)	.243(4)
cpu-small	.474(2)	.473(3)	.477(1)	.419(5)	.449(4)
elevators	.767(1)	.683(5)	.702(3)	.699(4)	.748(2)
fried	.998(2)	.850(5)	.861(4)	.989(3)	.999(1)
glass	.835(2)	.776(5)	.809(4)	.817(3)	.851(1)
housing	.655(3)	.669(1)	.654(4)	.625(5)	.667(2)
iris	.807(4)	.867(3)	.926(1)	.804(5)	.871(2)
pendigits	.902(4)	.902(3)	.918(2)	.802(5)	.932(1)
segment	.911(2)	.735(5)	.874(3)	.806(4)	.933(1)
stock	.735(4)	.855(2)	.877(1)	.691(5)	.776(3)
vehicle	.839(1)	.822(4)	.838(2)	.769(5)	.834(3)
vowel	.615(4)	.810(1)	.785(2)	.598(5)	.644(3)
wine	.911(4)	.930(2)	.926(3)	.941(1)	.902(5)
wisconsin	.617(1)	.464(4)	.453(5)	.533(3)	.607(2)
avg. rank	2.63	3.19	2.63	4.13	2.44
	60% missing labels				
	CC	IB-M	IB-PL	LL	RPC
authorship	.835(4)	.849(2)	.886(1)	.650(5)	.872(3)
bodyfat	.224(3)	.160(5)	.151(4)	.241(1)	.235(2)
calhousing	.247(3)	.263(1)	.259(2)	.221(5)	.242(4)
cpu-small	.470(1)	.428(4)	.437(3)	.418(5)	.448(2)
elevators	.765(1)	.596(5)	.633(4)	.696(3)	.748(2)
fried	.997(2)	.777(5)	.797(4)	.987(3)	.997(1)
glass	.789(3)	.611(5)	.675(4)	.808(1)	.799(2)
housing	.638(2)	.543(4)	.492(5)	.614(3)	.641(1)
iris	.743(5)	.799(2)	.868(1)	.768(4)	.779(3)
pendigits	.900(2)	.781(5)	.794(3)	.787(4)	.929(1)
segment	.902(2)	.612(5)	.674(4)	.801(3)	.920(1)
stock	.724(3)	.724(4)	.740(2)	.689(5)	.771(1)
vehicle	.810(1)	.736(5)	.765(3)	.764(4)	.786(2)
vowel	.598(3)	.638(1)	.588(5)	.591(4)	.612(2)
wine	.853(5)	.893(3)	.907(1)	.894(2)	.864(4)
wisconsin	.566(1)	.399(4)	.381(5)	.518(3)	.536(2)
avg. rank	2.56	3.75	3.25	3.44	2.00

Table 4.2: Performance of the label ranking methods in terms of Kendall's tau (in brackets the rank). The average ranks are listed at the last rows.

of the null hypothesis that all learners have equal performance and, in case this hypothesis is rejected, a two-tailed sign test to compare learners in a pairwise way. Such a two-step procedure is recommended, because straightforward paired tests on multiple methods often make little sense: When so many tests are made, a certain proportion of the null hypotheses might get rejected due to random chance [20]. The Friedman test is based on the average ranks (for each problem, the methods are ranked in decreasing order of performance, and the ranks thus obtained are averaged over the problems) as shown in the bottom lines in Table 4.2. At a significance level of 0.05, the Friedman test rejects the null hypothesis in all three cases, suggesting significant differences among these five methods. Pairwise comparisons between them are then summarized in terms of win statistics in Table 4.3. The critical value for the two-tailed sign test at the significance level 0.05 is 12 in our case. It means, a method is significantly better than another if it performs better on at least 12 out of all 16 data sets.

	CC	IB-M	IB-PL	LL	RPC
CC	–	6	5	15	6
IB-M	10	–	11	12	10
IB-PL	11	5	–	13	11
LL	1	4	3	–	1
RPC	10	6	5	15	–
CC	–	8	8	15	7
IB-M	8	–	5	9	7
IB-PL	8	11	–	11	8
LL	1	7	5	–	1
RPC	9	9	8	15	–
CC	–	11	11	12	5
IB-M	5	–	5	6	4
IB-PL	5	11	–	8	4
LL	4	10	8	–	3
RPC	11	12	12	13	–

Table 4.3: Win statistics (number of data sets on which the first method is better than the second one) for complete rankings, 30% missing labels, and 60% missing labels, from top to bottom.

Comparing two instance-based approaches, IB-M has a better performance in the case of complete rankings, while when the ratio of missing labels becomes higher, IB-PL surpasses IB-M. With incomplete rankings, as already discussed in Section 4.1, computational challenges arise by the Mallows model and non-trivial approximations are required, which somehow compromise its performance. On the other hand, the PL model is more capable of dealing with incomplete rankings, due to its stagewise nature.

Among the three reduction approaches, CC and RPC have achieved quite comparable performance across different data sets on all settings, while the results are not in favor of LL. These results are in agreement with some previous studies [36].

The comparison between local and reduction approaches is of particular interest. As mentioned earlier, we hypothesize that, since our instance-based methods for label ranking fit local models to the data, they are especially useful for problems requiring complex decision boundaries. Some evidence supporting this hypothesis is indeed provided by the learning curves depicting the performance as a function of the fraction of missing label information. While the learning curves of CC, LL, and RPC are often rather flat, showing a kind of saturation effect, they are much steeper for IB-M and IB-PL. This suggests that additional label information is still beneficial for the instance-based methods even when the reduction approaches, due to a lack of flexibility, are no longer able to exploit and adapt to extra data. As an illustration, the learning curves of IB-M and RPC on the housing data are shown in Figure 4.1, which nicely demonstrate this typical difference between the instance-based approach and the reduction approach.

We conclude this section with short remarks on two issues. First, as to the computational complexity of the label ranking methods, a direct comparison is complicated by the fact that IB-M and IB-PL are lazy learners, with almost no cost at training time but higher cost at prediction time. Anyway, in the current implementation, they are very efficient and quite comparable, in terms of runtime, to the corresponding counterpart for classification. This

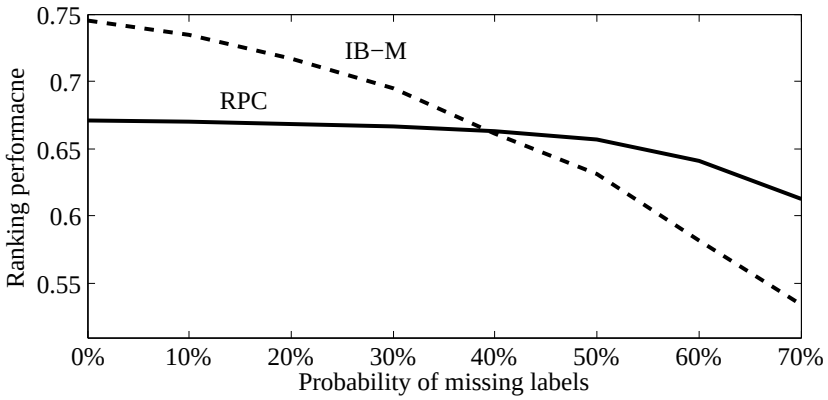

Figure 4.1: Ranking performance (in terms of Kendall's tau) as a function of the missing label rate on the housing data.

is true despite the more complex local estimation procedures: the approximate EM procedure in IB-M and the MM procedure in IB-PL converge very quickly.

Second, as mentioned earlier, an advantage of our probabilistic methods is that it delivers, as a byproduct, natural measures of the reliability of a prediction. In particular, the estimated spread $\hat{\theta}$ in the Mallows model is such a measure, which showed a very high correlation with the quality of prediction (in terms of Kendall's tau), suggesting that it is indeed a reasonable indicator of the uncertainty of a prediction. In general, the most straightforward measure of this kind is perhaps the probability of the prediction itself, namely $\hat{p} = \Pr(\sigma \mid \boldsymbol{\omega})$. To illustrate, we use it to compute a kind of accuracy-rejection curve: Using IB-PL in a leave-one-out cross validation, we compute the accuracy of the prediction (in terms of Kendall's tau) and its reliability (in terms of $\hat{p}$) for each instance $\boldsymbol{x}$. Subsequent to sorting the instances in decreasing order of reliability, we plot the function $t \mapsto f(t)$, where $f(t)$ is the mean accuracy of the top t percent of the instances. Given that $\hat{p}$ is indeed a good indicator of reliability, this curve should be decreasing, because the higher t, the more instances with a low reliability are taken into consideration. This expectation is confirmed with our data sets. Figure 4.2 shows the exemplary curve for the housing data.

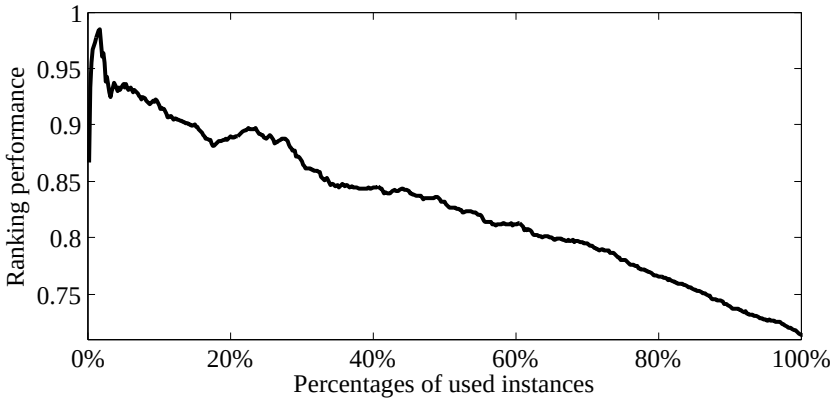

Figure 4.2: The accuracy-rejection curve computed on the basis of $\Pr(\sigma \mid \boldsymbol{\omega})$ using the housing data.

4.4 Chapter Conclusions

In this chapter, we have introduced an instance-based framework for label ranking. While the basic inference principle is a consistent extension of the nearest neighbor estimation principle, as used previously for well-known learning problems such as classification and regression, this framework is based on sound probabilistic models for rankings: Assuming that the conditional probability distribution of the output given the query is locally constant, we derive the maximum likelihood estimations based on the Mallows model and the PL model. The empirical results are quite promising and suggest that our approach is particularly strong in terms of predictive accuracy. Specifically, as instance-based methods are able to produce quite flexible models, our method appears to be especially advantageous for problems requiring complex decision boundaries. Besides, it has some further advantages, as it does not only produce a single ranking as an estimation but instead delivers a probability distribution over all rankings. This distribution can be used, e.g., to quantify the reliability of the predicted ranking. Since a single model are arguably more transparent than an ensemble of models, the proposed method might be preferred to existing approaches for reasons of interpretability.

Bibliographical Notes

Parts of the results presented in this chapter have been published in:

- Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 161-168. Omnipress, 2009.

- Weiwei Cheng and Eyke Hüllermeier. A new instance-based label ranking approach using the Mallows model. In Wen Yu, Haibo He, and Nian Zhang, editors, *Proceedings of the 6th International Symposium on Neural Networks*, pages 707-716. Springer, 2009.

- Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Label ranking methods based on the Plackett-Luce model. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 215-222. Omnipress, 2010.

Chapter 5

Probabilistic Label Ranking Models: A Global Extension

The learning method proposed in the previous chapter is local and lazy in the sense that an individual probabilistic model, i.e., an individual parameter vector $\boldsymbol{\omega}$, is estimated for each query instance $\boldsymbol{x} \in \mathcal{X}$ based on a part of the entire training data. (Recall that, in the Mallows model, the parameter vector corresponds to the center ranking and the spread, while in the PL model it corresponds to the skill vector.) In general, three choices one could make when designing a probabilistic method for label ranking.

- The (parameterized) probabilistic model that generates the rankings, e.g., the PL model.
- The dependency between the parameters of the probabilistic model and the input attributes, e.g., a linear dependency.
- How the parameters are learned. The parameters can be fit either locally or globally.

In Chapter 4 for example, we have studied the local learning methods with the Mallows and the PL model, where the instances and the model parameter have an implicit non-linear dependency. In this chapter we briefly discuss the learning of a global label ranking function as an alternative, in which

the model parameters are learned in an eager way. To this end, the model parameters are defined as a function of attributes describing an instance, namely $\boldsymbol{\omega} = f(\boldsymbol{\alpha}, \boldsymbol{x})$. While this extension is seemingly complicated in the case of the Mallows model, it is fairly intuitive for the PL model, as we shall see in the subsequent sections.

This chapter is organized as follows. We first propose a generalized linear method based on the PL model and discuss the parameter estimation in Section 5.1. An experimental evaluation is then presented in Section 5.2. Section 5.3 concludes this chapter.

5.1 Generalized Linear Models

We define the PL parameters v_i, quantifying the propensity for the i-th label y_i, as a linear function of the input attributes. Despite its simplicity, such a linear model is often more interpretable than its non-linear counterpart. To guarantee the non-negativity of the parameters, we model their logarithm as a linear function:

$$v_i \;=\; \exp\left(\sum_{j=1}^{d} \alpha_j^{(i)} \cdot x_j\right) \;, \tag{5.1}$$

where we assume an instance to be represented in terms of a feature vector $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathcal{X} = \mathbb{R}^d$.

The model parameters to be estimated are now the $\alpha_j^{(i)}$ $(1 \leq i \leq n,\, 1 \leq j \leq d)$. Given a training data set

$$\mathcal{T} \;=\; \left\{\left(\boldsymbol{x}^{(q)}, \pi^{(q)}\right)\right\}_{q=1}^{m} \tag{5.2}$$

with $\boldsymbol{x}^{(q)} = \left(x_1^{(q)}, \ldots, x_d^{(q)}\right)$, the log-likelihood function is given by

$$L = \sum_{q=1}^{m} \sum_{i=1}^{n_q} \left[\log\left(v\left((\pi^{(q)})^{-1}(i), q\right)\right) - \log \sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right)\right], \tag{5.3}$$

where n_q is the number of labels in the ranking $\pi^{(q)}$, and

$$v(i,q) \;=\; \exp\left(\sum_{j=1}^{d} \alpha_j^{(i)} \cdot x_j^{(q)}\right) \;. \tag{5.4}$$

The first derivatives of L are given by

$$\frac{\partial L}{\partial \alpha_p^{(a)}} = \sum_{q=1}^{m} \delta(p,q,1) \cdot x_p^{(q)} - \sum_{q=1}^{m} \sum_{i=1}^{n_q} \delta(a,q,i) \cdot \frac{v(a,q) \cdot x_p^{(q)}}{\sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right)} \;, \tag{5.5}$$

where

$$\delta(a,q,i) = \begin{cases} 1 & \pi^{(q)}(a) \geq i \\ 0 & \text{otherwise} \end{cases} \;. \tag{5.6}$$

Moreover, the second derivatives (for $a \neq b$, $p \neq \ell$) are as follows:

$$\begin{aligned}
\frac{\partial^2 L}{\partial \left(\alpha_p^{(a)}\right)^2} =& - \sum_{q=1}^{m} \sum_{i=1}^{n_q} \delta(a,q,i) \cdot v(a,q) \cdot \left(x_p^{(q)}\right)^2 \\
& \cdot \frac{\left[\sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right) - v(a,q)\right]}{\left(\sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right)\right)^2} \;, \\
\frac{\partial^2 L}{\partial \alpha_p^{(a)} \partial \alpha_\ell^{(a)}} =& - \sum_{q=1}^{m} \sum_{i=1}^{n_q} \delta(a,q,i) \cdot v(a,q) \cdot x_p^{(q)} \cdot x_\ell^{(q)} \\
& \cdot \frac{\left[\sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right) - v(a,q)\right]}{\left(\sum_{j=i}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right)\right)^2} \;, \\
\frac{\partial^2 L}{\partial \alpha_p^{(a)} \partial \alpha_\ell^{(b)}} =& \sum_{q=1}^{m} \sum_{i=1}^{n_q} \delta(a,q,i) \cdot \delta(b,q,i) \\
& \cdot \frac{v(a,q) \cdot x_p^{(q)} \cdot v(b,q) \cdot x_\ell^{(q)}}{\left(\sum_{j=1}^{n_q} v\left((\pi^{(q)})^{-1}(j), q\right)\right)^2} \;.
\end{aligned} \tag{5.7}$$

Note that $\partial^2 L / \partial (\alpha_q^{(a)})^2 \leq 0$ for all $1 \leq a \leq n$ and $1 \leq p \leq d$. Based on these derivatives, the maximization of the log-likelihood can be accomplished by means of gradient-based optimization methods. In our implementation, we

use a standard stochastic gradient descent algorithm [7] that is, in terms of efficiency, compared quite favorably with other gradient-based methods.

5.2 Experiments

In this section, we present an empirical evaluation of our generalized linear approach (Lin-PL) to label ranking using the PL model. For comparison, Lin-PL is tested against the instance-based method with the PL model (IB-PL) discussed in Chapter 4. Results are given in Table 5.1. We have tested these two approaches on the same data sets with an identical experimental setting as in Section 4.3. So the results in Table 5.1 can be directly compared with the ones we reported there. The numbers of wins for each method are summarized at the bottom of the table.

	complete ranking		30% missing labels		60% missing labels	
	IB-PL	Lin-PL	IB-PL	Lin-PL	IB-PL	Lin-PL
authorship	.936	.930	.927	.899	.886	.846
bodyfat	.230	.272	.204	.266	.151	.222
calhousing	.326	.220	.303	.229	.259	.229
cpu-small	.495	.426	.477	.418	.437	.412
elevators	.721	.712	.702	.706	.633	.704
fried	.894	.996	.861	.993	.797	.990
glass	.841	.825	.809	.825	.675	.807
housing	.711	.659	.654	.658	.492	.636
iris	.960	.832	.926	.823	.868	.778
pendigits	.939	.909	.918	.909	.794	.907
segment	.950	.902	.874	.895	.674	.888
stock	.922	.710	.877	.701	.740	.687
vehicle	.859	.838	.838	.817	.765	.804
vowel	.851	.586	.785	.581	.588	.575
wine	.947	.954	.926	.931	.907	.915
wisconsin	.479	.635	.453	.615	.381	.585
# wins	12	4	8	8	6	10

Table 5.1: Performance of label ranking methods in terms of Kendall's tau.

Since we now compare two methods, the results can be directly analyzed with a two-tailed sign test. At the significance level 0.05, the critical value for

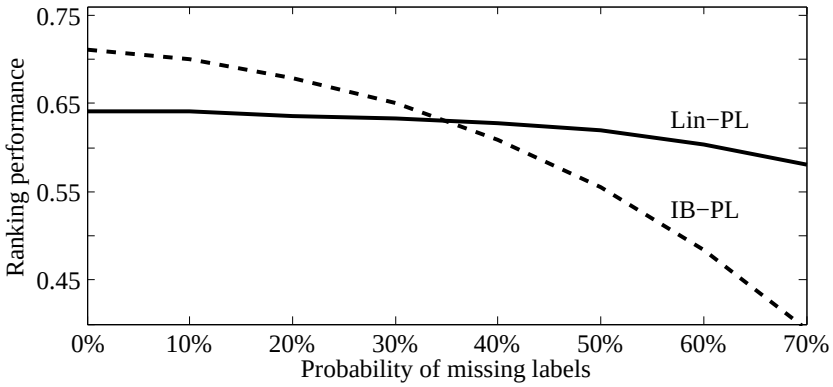

Figure 5.1: Ranking performance (in terms of Kendall's tau) as a function of the missing label rate on the housing data.

this test is 12, meaning that a method is significantly better than the other if it outperforms on at least 12 out of all 16 data sets. Despite being statistically non-significant in some cases, the results are still quite informative and show an important trend (which are likely to become significant when increasing the number of data sets): The instance-based approach IB-PL performs better in the complete ranking scenario, but its performance drops more quickly when missing label information.

This observation is plausible and coherent with the complementary nature of global and local methods. Like in the case of conventional classification, instance-based methods are advantageous for problems requiring complex decision boundaries, for which the strong bias of linear methods prevents them from achieving a good separation. On the other hand, if the linearity assumption is (at least approximately) valid, better models can be learned with fewer data. Correspondingly, instance-based learners are more sensitive toward the amount of training data. Some evidence supporting this hypothesis is provided by the learning curves depicting the performance as a function of the fraction of missing label information. The learning curves for the housing data in Figure 5.1 have demonstrated this typical observation.

5.3 Chapter Conclusions

The basic idea of this chapter is to parameterize the coefficients of a probabilistic model and expressing them as functions of the input attributes. By doing so, we end up with global probabilistic models, which often have lower variance compared with the methods we proposed in Chapter 4. Although the general principle of this idea holds for every probabilistic model for rankings, it should be noted that for some models, to parameterize the coefficients by the input attributes is not straightforward. For example in the case of the Mallows model, it is cumbersome to explicitly express the center ranking and the spread as a function of the input attributes. While for some other models, such as the order statistic model and the PL model, the parameterization is much more intuitive. In this chapter, we have demonstrated this idea with the PL model and it leads to fitting a model in the form of log-linear (utility) functions. Empirically, we have compared this global method with the local method that is also based on the PL model. The results have confirmed the bias-variance trade-off we expected. Finally, let us remark that, since the global method we discussed here is based on a probabilistic model for rankings as well, it comes with a very natural way to measure the reliability of the predictions. From this perspective, it shares the same merits with the methods in Chapter 4.

Bibliographical notes

Parts of the results presented in this chapter have been published in:

- Weiwei Cheng, Krzysztof Dembczyński, and Eyke Hüllermeier. Label ranking methods based on the Plackett-Luce model. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 215-222. Omnipress, 2010.

Chapter 6

A Label Ranking Approach to Multi-Label Classification

This chapter addresses the application of instance-based label ranking in multi-label classification. In conventional classification, each instance is assumed to belong to exactly one among a finite set of candidate classes. As opposed to this, the setting of multi-label classification allows an instance to belong to several classes simultaneously or, say, to attach more than one label to an instance.

Even though quite a number of sophisticated methods for multi-label classification has been proposed in the literature, the application of instance-based learning has not been studied very deeply in this context. This is a bit surprising, given that the instance-based learning algorithms based on the nearest neighbor estimation principle have been applied quite successfully in classification and pattern recognition [1]. A notable exception is the multi-label k-nearest neighbor (MLKNN) method that was proposed in [68], where it was shown to be competitive to state-of-the-art multi-label learning methods.

In this chapter, we introduce an instance-based approach to multi-label classification, which is based on calibrated label ranking, a recently proposed

framework that unifies multi-label classification and label ranking (see Section 6.1). Within this framework, instance-based prediction is realized in the form of a maximum a posteriori (MAP) estimation, assuming a ranking distribution follows the Mallows model (see Section 6.2). After the discussion of the related work (see Section 6.3), we provide an empirical study of this approach focusing on its predictive accuracy (see Section 6.4).

6.1 Multi-Label Classification as Calibrated Label Ranking

Let $\mathcal{X}$ denote an instance space and let $\mathcal{Y} = \{y_1, \ldots, y_n\}$ be a finite set of class labels. Moreover, suppose that each instance $\boldsymbol{x} \in \mathcal{X}$ can be associated with a subset of labels $Y \in 2^{\mathcal{Y}}$; this subset is often called the set of relevant labels, while the complement $\mathcal{Y} \backslash Y$ is considered as irrelevant for $\boldsymbol{x}$. Given the training data, i.e., a finite set of observations in the form of tuples $(\boldsymbol{x}, Y_{\boldsymbol{x}}) \in \mathcal{X} \times 2^{\mathcal{Y}}$, typically assumed to be drawn independently from an (unknown) probability distribution on $\mathcal{X} \times 2^{\mathcal{Y}}$, the goal in multi-label classification is to learn a classifier $h : \mathcal{X} \to 2^{\mathcal{Y}}$ that generalizes well beyond these observations in the sense of minimizing the expected prediction loss with respect to a specific loss function.

Note that multi-label classification can be reduced to a conventional classification problem in a straightforward way, namely by considering each label subset $Y \in 2^{\mathcal{Y}}$ as a distinct (meta-)class. This approach is referred to as label powerset in the literature [8]. An obvious drawback of this approach is the potentially large number of classes that one has to deal with in the newly generated problem. Another way of reducing multi-label to conventional classification is offered by the binary relevance (BR) approach. Here, a single binary classifier h_i is trained for each label $y_i \in \mathcal{Y}$. For a query instance $\boldsymbol{x}$, this classifier is supposed to predict whether y_i is relevant for $\boldsymbol{x}$ ($h_i(\boldsymbol{x}) = 1$) or not ($h_i(\boldsymbol{x}) = 0$). A multi-label prediction for $\boldsymbol{x}$ is then given by $h(\boldsymbol{x}) = \{y_i \in \mathcal{Y} \,|\, h_i(\boldsymbol{x}) = 1\}$. Since binary relevance learning treats every

label independently of all other labels, an obvious disadvantage of this approach is that it ignores potential correlations and interdependencies between labels.

Some of the more sophisticated approaches learn a multi-label classifier h in an indirect way via a scoring function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that assigns a real number to each instance-label combination. Such a function does not only allow one to make multi-label predictions (via thresholding the scores), but also offers the possibility to produce a ranking of the class labels, simply by ordering them according to their scores. Sometimes, this ranking is even more desirable as a prediction, and indeed, there are several widely-used evaluation metrics that compare a true label subset with a predicted ranking instead of a predicted label subset.

Despite the tight relation between a label ranking and a multi-label classification (label subset), label ranking methods cannot be directly applied to multi-label learning, since a label ranking provides information about the relative preference for labels, but not about the absolute preference or, say, relevance of a label. To combine the information offered by these two, the concept of a calibrated label ranking has been proposed in [28]. A calibrated label ranking is a ranking of the label set Ω extended by a neutral label y_0. The idea is that y_0 splits a ranking into two parts, the positive (relevant) part consisting of those labels y_i preceding y_0, i.e., $y_i \succ_{\boldsymbol{x}} y_0$, and the negative (irrelevant) part given by those labels y_j ranked lower than y_0, i.e., $y_0 \succ_{\boldsymbol{x}} y_j$. In this way, a multi-label prediction can be derived from a predicted calibrated label ranking.

The other way around, a multi-label set $Y_{\boldsymbol{x}}$ translates into the set of pairwise preferences $\{y \succ_{\boldsymbol{x}} y' \,|\, y \in Y_{\boldsymbol{x}}, y' \in \mathcal{Y} \setminus Y_{\boldsymbol{x}}\}$, and can hence be considered as incomplete information about an underlying calibrated label ranking. More specifically, $Y_{\boldsymbol{x}}$ is consistent with the set of label rankings $E(Y_{\boldsymbol{x}})$ given by those permutations $\pi \in \Omega$ that rank all labels in $Y_{\boldsymbol{x}}$ higher and all labels in $\mathcal{Y} \setminus Y_{\boldsymbol{x}}$ lower than the neutral label y_0. In the rest of this chapter, when we speak about a ranking, we always mean a calibrated

ranking (i.e., Ω contains the neutral label y_0).

6.2 Instance-Based Multi-Label Classification

Coming back to the label ranking problem and the idea of instance-based learning, i.e., local prediction based on the nearest neighbor estimation principle, consider a query instance $\boldsymbol{x} \in \mathcal{X}$ and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ denote the nearest neighbors of $\boldsymbol{x}$ (according to an underlying distance measure on $\mathcal{X}$) in the training set, where $k \in \mathbb{N}$ is a fixed integer. Each neighbor $\boldsymbol{x}_i$ is associated with a subset $Y_{\boldsymbol{x}_i} \subseteq \mathcal{Y}$ of labels. In analogy to the conventional settings of classification and regression, we assume that the probability distribution $\Pr(\cdot \,|\, \boldsymbol{x})$ on Ω is (at least approximately) locally constant around the query $\boldsymbol{x}$, so that the neighbors can be considered as a sample on the basis of which $\Pr(\cdot \,|\, \boldsymbol{x})$ can be estimated.

Thus, assuming an underlying (calibrated) label ranking, the probability to observe $Y_{\boldsymbol{x}_i}$ is given by

$$\Pr(E(Y_{\boldsymbol{x}_i})) \;=\; \sum_{\sigma \in E(Y_{\boldsymbol{x}_i})} \Pr(\sigma \,|\, \theta, \pi) \;, \tag{6.1}$$

where $E(Y_{\boldsymbol{x}_i})$ denotes the set of all label rankings consistent with $Y_{\boldsymbol{x}_i}$. Making a simplifying assumption of independence under the Mallows model (4.1), the probability of the complete set of observations $\mathbf{Y} = \{Y_{\boldsymbol{x}_1}, \ldots, Y_{\boldsymbol{x}_k}\}$ then becomes

$$\begin{aligned} \Pr(\mathbf{Y} \,|\, \theta, \pi) \;&=\; \prod_{i=1}^{k} \Pr(E(Y_{\boldsymbol{x}_i}) \,|\, \theta, \pi) \\ &=\; \prod_{i=1}^{k} \sum_{\sigma \in E(Y_{\boldsymbol{x}_i})} \Pr(\sigma \,|\, \theta, \pi) \\ &=\; \frac{\prod_{i=1}^{k} \sum_{\sigma \in E(Y_{\boldsymbol{x}_i})} \exp\left(-\theta T(\sigma, \pi)\right)}{\left(\prod_{j=1}^{n} \frac{1-\exp(-j\theta)}{1-\exp(-\theta)}\right)^k} \;. \end{aligned} \tag{6.2}$$

Instance-based prediction of the label ranking $Y_{\boldsymbol{x}}$ can now be posed as a maximum likelihood problem, namely as finding the MLE of π (and θ) in (6.2). This problem is extremely difficult in general. Fortunately, in the context of multi-label classification, we are able to exploit the special structure of the observations. More specifically, based on the results of Section 4.2.1 we can show the following theorem.

Theorem 6. *For each label $y_i \in \mathcal{Y}$, let $f(y_i)$ denote the frequency of occurrence of this label in the neighborhood of $\boldsymbol{x}$, i.e., $f(y_i) = \#\{j \mid y_i \in Y_{\boldsymbol{x}_j}\}/k$. Moreover, let $f(y_0) = 1/2$ by definition. Then, a ranking $\pi \in \Omega$ is an MLE in (6.2) iff it guarantees that $f(y_i) > f(y_j)$ implies $\pi(i) < \pi(j)$.*

According to this result, an optimal ranking and, hence, an optimal multi-label prediction can be simply found by sorting the labels according to their frequency of occurrence in the neighborhood. A disadvantage of this estimation is its ambiguity in the presence of ties: If two labels have the same frequency, they can be ordered in either way. We can remove this ambiguity by replacing the MLE by the MAP estimation.

Corollary 2. *Let $g(y_i)$ denote the frequency of occurrence of the label y_i in the complete training set. There exists a prior distribution $\Pr$ on Ω such that, for large enough k, a ranking $\pi \in \Omega$ is an MAP estimation iff it guarantees the following: If $f(y_i) > f(y_j)$ or $f(y_i) = f(y_j)$ and $g(y_i) > g(y_j)$, then $\pi(i) < \pi(j)$.*

This result suggests a very simple prediction procedure: Labels are sorted according to their frequency in the neighborhood of the query, and ties are broken by resorting to global information outside the neighborhood, namely the label frequency in the complete training data (which serve as estimates of the unconditional probability of a label).

6.3 Related Work in Multi-Label Classification

Multi-label classification has received a great deal of attention in machine learning in recent years, and a number of methods has been developed, often motivated by specific types of applications such as text categorization [52, 60, 40, 67], computer vision [8], and bioinformatics [12, 24, 67]. Besides, several well-established methods for conventional classification have been extended to the multi-label case, including support vector machines [30, 24, 8], neural networks [67], and decision trees [62]. Detailed discussions of multi-label learning methods can be referred to [18].

Our interest in instance-based multi-label classification is largely motivated by the multi-label k-nearest neighbor (MLKNN) method that has recently been proposed in [68]. In that paper, the authors show that MLKNN performs quite well in practice. In the concrete experiments presented, MLKNN outperformed some state-of-the-art model-based approaches to multi-label classification, including RankSVM and AdaBoost.MH [24, 14].

MLKNN is a binary relevance learner, i.e., it learns a single classifier h_i for each label $y_i \in \mathcal{Y}$. However, instead of using the standard KNN classifier as a base learner, it implements the h_i by means of a combination of KNN and Bayesian inference: Given a query instance $\boldsymbol{x}$ with unknown multi-label classification $Y \subseteq \mathcal{Y}$, it finds the k nearest neighbors of $\boldsymbol{x}$ in the training data and counts the number of occurrences of y_i among these neighbors. Considering this number, c, as information in the form of a realization of a random variable C, the posterior probability of $y_i \in Y$ is given by

$$\Pr(y_i \in Y \,|\, C = c) = \frac{\Pr(C = c \,|\, y_i \in Y) \cdot \Pr(y_i \in Y)}{\Pr(C = c)} \,, \tag{6.3}$$

which leads to the decision rule

$$h_i(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \Pr(C = c \,|\, y_i \in Y) \Pr(y_i \in Y) \geq \Pr(C = c \,|\, y_i \notin Y) \Pr(y_i \notin Y) \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

The prior probabilities $\Pr(y_i \in Y)$ and $\Pr(y_i \notin Y)$ as well as the conditional probabilities $\Pr(C = c \,|\, y_i \in Y)$ and $\Pr(C = c \,|\, y_i \notin Y)$ are estimated from the training data in terms of corresponding relative frequencies. While the estimation of the former probabilities is uncritical from a computational point of view, the estimation of the conditional probabilities can become quite expensive. Essentially, it requires the consideration of all k-neighborhoods of all training instances, and the counting of the number of occurrences of each label within these neighborhoods. Implementing nearest neighbor search in a naive way, namely by linear search, this would mean a complexity of $O(kn)$, where n is the size of the training data. Of course, this complexity can be reduced by using more efficient algorithms and data structures for nearest neighbor search; for example, the all nearest neighbors problem, i.e., the problem to find the (first) nearest neighbor for each element of a data set, can be solved in time $O(k \log n)$ [61]. Nevertheless, the computational overhead produced by this kind of preprocessing on the training data will remain a dominating factor for the overall runtime of the method.

6.4 Experiments

This section is devoted to experimental studies that we conducted to get a concrete idea of the performance of our method. Before presenting results, we give some information about the learning algorithms and data sets included in the study, as well as the criteria used for evaluation.

data set	domain	# instances	# attributes	# labels	cardinality
emotions	music	593	72	6	1.87
image	vision	2000	135	5	1.24
genbase	biology	662	1186*	27	1.25
mediamill	multimedia	5000	120	101	4.27
reuters	text	7119	243	7	1.24
scene	vision	2407	294	6	1.07
yeast	biology	2417	103	14	4.24

Table 6.1: Statistics for the multi-label data sets used in the experiments. The symbol * indicates that the data set contains nominal features; the cardinality is the average number of labels per instance.

6.4.1 Learning Algorithms

For the reasons mentioned previously, our main interest is focused on MLKNN, which is the state-of-the-art in instance-based multi-label ranking; we used its implementation in the MULAN package [59].[1] MLKNN is parameterized by the size of the neighborhood, for which we adopted the value $k = 10$. This value is recommended in [68], where it was found to yield the best performance. For the sake of fairness, we use the same neighborhood size for our method (MallowsML). In both cases, the Euclidean metric (on the complete normalized attribute space) was used as a distance function. As an additional baseline, we used binary relevance learning (BR) with C4.5 (the WEKA [65] implementation J48 in its default setting) as a base learner.

6.4.2 Data Sets

Benchmark data for multi-label classification is not as abundant as for conventional classification, and indeed, experiments in this field are often restricted to a very few or even only a single data set. For our experimental study, we have collected a comparatively large number of seven data sets from different domains; an overview is given in Table 6.1.[2]

[1] `http://mlkd.csd.auth.gr/multi-label.html`

[2] Data sets are public available at `http://mlkd.csd.auth.gr/multi-label.html` and `http://lamda.nju.edu.cn/data.htm`.

The emotions data was created from a selection of songs from 233 musical albums [58]. From each song, a sequence of 30 seconds after the initial 30 seconds was extracted. The resulting sound clips were stored and converted into wave files of 22050 Hz sampling rate, 16-bit per sample and mono. From each wave file, 72 features have been extracted, falling into two categories: rhythmic and timbre. Then, in the emotion labeling process, 6 main emotional clusters are retained corresponding to the Tellegen-Watson-Clark model of mood: amazed-surprised, happy-pleased, relaxing-clam, quiet-still, sad-lonely, and angry-aggressive.

Image and scene are semantic scene classification data sets proposed, respectively, by [69] and [8], in which a picture can be categorized into one or more classes. In the scene data, for example, pictures can have the following classes: beach, sunset, foliage, field, mountain, and urban. Features of this data set correspond to spatial color moments in the LUV space. Color as well as spatial information have been shown to be fairly effective in distinguishing between certain types of outdoor scenes: bright and warm colors at the top of a picture may correspond to a sunset, while those at the bottom may correspond to a desert rock. Features of the image data set are generated by the SBN method [46] and essentially correspond to attributes in an RGB color space.

From the biological field, we have chosen the two data sets yeast and genbase. The yeast data set is about predicting the functional classes of genes in the Yeast Saccharomyces cerevisiae. Each gene is described by the concatenation of micro-array expression data and a phylogenetic profile, and is associated with a set of 14 functional classes. The data set contains 2417 genes in total, and each gene is represented by a 103-dimensional feature vector. In the genbase data, 27 important protein families are considered, including, for example, PDOC00064 (a class of oxydoreductases) and PDOC00154 (a class of isomerases). After the preprocessing, a training set is exported, consisting of 662 proteins that belong to one or more of these 27 classes.

From the text processing field, we have chosen a subset of the widely

studied Reuters-21578 collection [54]. The seven most frequent categories are considered. After removing documents whose label sets or main texts are empty, 8866 documents are retained where only 3.37% of them are associated with more than one class label. After randomly removing documents with only one label, a text categorization data set containing 2000 documents is obtained. Functional words are removed from the vocabulary and the remaining words are stemmed. Instances adopt the bag-of-words representation based on term frequencies. Without loss of effectiveness, dimensionality reduction is performed by retaining the top 2% words with highest document frequency. Thereafter, each instance is represented as a 243-dimensional feature vector.

The mediamill data set is from the field of multimedia indexing and originates from the well-known TREC Video Retrieval Evaluation data (TRECVID 2005/2006) initiated by American National Institute of Standards and Technology (NIST), which contains 85 hours of international broadcast news data. The task in this data set is the automated detection of a lexicon of 101 semantic concepts in videos. Every instance of this data set has 120 numeric features including visual, textual, as well as fusion information. The trained classifier should be able to categorize an unseen instance to some of these 101 labels, e.g., face, car, male, soccer, and so on. More details about this data set can be found at [56].

6.4.3 Evaluation Measures

To evaluate the performance of multi-label classification methods, a number of criteria and metrics have been proposed in the literature. For a classifier h, let $h(\boldsymbol{x}) \subseteq \mathcal{Y}$ denote its multi-label prediction for an instance $\boldsymbol{x}$, and let $Y_{\boldsymbol{x}}$ denote the true set of relevant labels. The Hamming loss computes the percentage of labels whose relevance is predicted incorrectly:

$$\textsc{HamLoss}(h) = \frac{1}{|\mathcal{Y}|}\big|h(\boldsymbol{x}) \,\Delta\, Y_{\boldsymbol{x}}\big| \,, \tag{6.5}$$

data set	MLKNN	MallowsML	BR	MLKNN	MallowsML	BR
emotions	0.261	0.197	0.253	0.262	0.163	0.352
genbase	0.005	0.003	0.001	0.006	0.006	0.006
image	0.193	0.192	0.243	0.214	0.208	0.398
mediamill	0.027	0.027	0.032	0.037	0.036	0.189
reuters	0.073	0.085	0.057	0.068	0.087	0.089
scene	0.087	0.094	0.131	0.077	0.088	0.300
yeast	0.194	0.197	0.249	0.168	0.165	0.360

Table 6.2: Experimental results in terms of Hamming loss (left) and rank loss (right).

where Δ is the symmetric difference between two sets.

To measure the ranking performance, we used the rank loss, which computes the average fraction of label pairs that are not correctly ordered:

$$\textsc{RankLoss}(\pi) = \frac{\#\{(y, y') \,|\, \pi_{\boldsymbol{x}}(y) \leq \pi_{\boldsymbol{x}}(y'), (y, y') \in Y_{\boldsymbol{x}} \times \overline{Y}_{\boldsymbol{x}}\}}{|Y_{\boldsymbol{x}}||\overline{Y}_{\boldsymbol{x}}|} \,, \qquad (6.6)$$

where $\pi_{\boldsymbol{x}}(y)$ denotes the position assigned to label y for instance $\boldsymbol{x}$, and $\overline{Y}_{\boldsymbol{x}} = \mathcal{Y} \setminus Y_{\boldsymbol{x}}$ is the set of irrelevant labels.

A detailed analysis of these two losses can be found in [18]. It turns out, both our approach MallowsML and MLKNN are theoretically optimal in terms of minimizing these two losses.

6.4.4 Results

The results of a cross validation study (10-fold, 5 repeats) are summarized in Table 6.2. As can be seen, both instance-based approaches perform quite strongly in comparison to the baseline, which is apparently not competitive. The instance-based approaches themselves are more or less en par, with a slight though statistically non-significant advantage for our method.

As discussed in the previous section, MLKNN is expected to be less efficient from a computational point of view, and this expectation was confirmed by our experiments. Indeed, our approach scales much better than MLKNN.

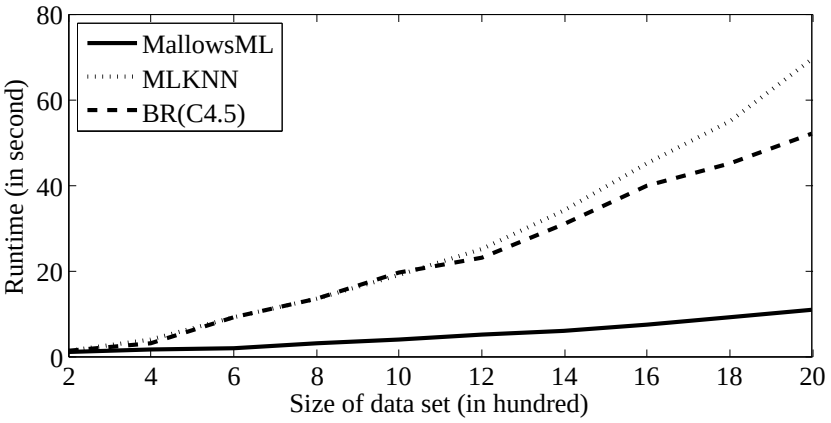

Figure 6.1: Runtime of the methods on the image data.

A typical example is shown in Figure 6.1, where the runtime (total time needed to conduct a 10-fold cross validation) is plotted as a function of the size of the data. To obtain data sets of different size, we sampled from the image data.

6.5 Chapter Conclusion

In this chapter, we have presented an alternative instance-based multi-label classifier, which is (at least) competitive in terms of predictive accuracy, while being computationally more efficient than MLKNN, which is considered as a state-of-the-art approach in instance-based multi-label classification. In fact, our approach comes down to a very simple prediction procedure, in which labels are sorted according to their local frequency in the neighborhood of the query, and ties are broken by global frequencies. Despite its simplicity, this approach is well justified with the underlying theoretical framework based on the Mallows model and the calibrated label ranking method.

Bibliographical notes

Parts of the results presented in this chapter have been published in:

- Weiwei Cheng and Eyke Hüllermeier. A simple instance-based approach to multi-label classification using the Mallows model. In Grigorios Tsoumakas, Minling Zhang, and Zhihua Zhou, editors, *Proceedings of Learning from Multi-Label Data Workshop*, pages 28-38, 2009.

Chapter 7

Ranking with Abstention

A ranking is commonly understood as a strict total order, i.e., an irreflexive, asymmetric, and transitive relation. In this chapter, we propose a generalization of the standard setting, allowing a model to make predictions in the form of partial instead of total orders. We interpret such kind of prediction as a ranking with partial abstention: If the ranker is not sufficiently certain regarding the relative order of two alternatives and, therefore, cannot reliably decide whether the former should precede the latter or the other way around, it may abstain from this decision and instead declare these alternatives as being incomparable.

The notion of abstention is already well-known for conventional classification, and the corresponding extension is usually referred to as classification with a reject option [34, 6]: The classifier is allowed to abstain from a prediction for a query instance in case it is not sure enough. An abstention of this kind is an obvious means to avoid unreliable predictions. Needless to say, the same idea makes sense also in the context of ranking. In fact, one may even argue that a reject option becomes even more interesting here: While a conventional classifier has only two choices, namely to predict a class or to abstain, a ranker can abstain to a certain degree: The order relation predicted by the ranker can be more or less complete or, stated differently, more or less partial, ranging from a total order (conventional ranking) to

the empty relation in which all alternatives are incomparable. Later on, we will express the degree of abstention of a ranker more precisely in terms of a degree of completeness of the partial order it predicts.

This chapter proposes two approaches to label ranking with partial abstention. The first approach, a distribution-free approach, consists of two main steps. First, a preference relation is derived that specifies, for each pair of labels y_i and y_j, a degree of preference for y_i over y_j and, vice versa, a degree of preference for y_j over y_i. The idea is that, the more similar these two degrees are, the more uncertain the learner is. Then, in a second step, a partial order maximally compatible with this preference relation, in a sense to be specified later on, is derived as a prediction. The second approach assumes the degree of preference for the pairs are induced from a certain probabilistic models for rankings, i.e., it is assumed the underlying rankings of the labels are generated from a particular parameterized probability distribution. By making such a stronger model assumption, this approach is able to avoid inconsistencies that may occur in the first approach and hence simplifies the construction of consistent partial order relations.

The remainder of the chapter is organized as follows. Our approaches to label ranking with partial abstention are detailed in the next two section. In Section 7.3, we address the question of how to evaluate predictions in the form of partial orders and propose suitable performance metrics for measuring the correctness and completeness of such predictions. Section 7.4 is devoted to experimental studies. We show that our approaches are indeed able to achieve a reasonable trade-off between these two criteria. The chapter ends with a couple of concluding remarks in Section 7.5.

7.1 Ranking with Partial Abstention

The set of labels $\mathcal{Y}$ to be ordered by a ranker depends on the type of ranking problem. A ranking on $\mathcal{Y}$ is an irreflexive, total, and transitive relation $\succ$, specifying for all pairs $y_i, y_j \in \mathcal{Y}$ whether y_i precedes y_j, denoted $y_i \succ y_j$, or

y_j precedes y_i. The key property of transitivity can be seen as a principle of consistency: If y_i is preferred to y_j and y_j is preferred to y_k, then y_i must be preferred to y_k.

A partial order $\sqsupset$ on $\mathcal{Y}$ is a generalization that sticks to this consistency principle but is not necessarily total. If, for two alternatives y_i and y_j, neither $y_i \sqsupset y_j$ nor $y_j \sqsupset y_i$, then these alternatives are considered as incomparable, written as $y_i \perp y_j$. Note that, in the following, we still assume irreflexivity of $\sqsupset$, even if this is not always mentioned explicitly.

7.1.1 Partial Orders in Learning to Rank

As mentioned before, our idea is to make use of the concept of a partial order in a machine learning context, namely to generalize the problem of learning to rank. More specifically, the idea is that, for each pair of labels y_i and y_j, the ranker can decide whether to make a prediction about the order relation between these labels, namely to hypothesize that y_i precedes y_j or that y_j precedes y_i, or to abstain from this prediction. We call a ranker having this possibility of abstention a ranker with partial reject option. Note that, for different pairs of alternatives, the reject decisions cannot be made independently of each other. Instead, the pairwise predictions should be of course consistent in the sense of being transitive and acyclic. In other words, a ranker with a (partial) reject option is expected to make a prediction in the form of a (strict) partial order $\sqsupset$ on the set of alternatives. This partial order is considered as an incomplete estimation of an underlying (ground-truth) order relation $\succ$: For labels $y_i, y_j \in \mathcal{Y}$, $y_i \sqsupset y_j$ corresponds to the prediction that $y_i \succ y_j$ (and not $y_j \succ y_i$) holds, whereas $y_i \perp y_j$ indicates an abstention on this pair of alternatives.

In the following sections, we propose a method that enables a ranker to make predictions of such kind. Roughly speaking, this approach consists of two main steps:

- The first step is the prediction of a preference relation P that specifies, for each pair of labels y_i and y_j, a degree of uncertainty regarding their

relative comparison.

- In the second step, a (strict) partial order maximally compatible with this preference relation is derived.

7.1.2 Prediction of a Binary Preference Relation

Let P be an $\mathcal{Y} \times \mathcal{Y} \to [0, 1]$ mapping, so that $P(y_i, y_j)$ is a measure of support for the order (preference) relation $y_i \succ y_j$. We assume P to be reciprocal, i.e.,

$$P(y_j, y_i) \;=\; 1 - P(y_i, y_j) \tag{7.1}$$

for all $y_i, y_j \in \mathcal{Y}$. A relation of that kind can be produced in different ways. For example, some ranking methods explicitly train models that compare alternatives in a pairwise way, e.g., by training a single classifier for each pair of labels [36]. If these models are able to make probabilistic predictions, these can be used directly as preference degrees $P(y_i, y_j)$.

However, since probability estimation is known to be a difficult problem, we like to emphasize that the method we introduce here only assumes an ordinal structure of the relation P. In fact, as will be seen below, the partial order induced by P is invariant toward monotone transformations of P. In other words, only the order relation of preference degrees is important, not the degrees themselves: If $P(y_i, y_j) > P(y_s, y_t)$, then $y_i \succ y_j$ is considered as more certain than $y_s \succ y_t$.

Here, we propose a generic approach that allows one to turn every ranker into a partial ranker. To this end, we resort to the idea of ensembling, although other possibilities are also conceivable. Let L be a learning algorithm that, given a set of training data, induces a model $\mathcal{M}$ that in turn makes predictions in the form of rankings (total orders) $\succ$ of a set of labels $\mathcal{Y}$. Now, instead of training a single model, our idea is to train k such models $\mathcal{M}_1, \ldots, \mathcal{M}_k$ by resampling from the original data set, i.e., by creating k bootstrap samples and giving them as input to L. Consequently, by querying all these models, k rankings $\succ_1, \ldots, \succ_k$ will be produced instead of a single

prediction.[1]

For each pair of alternatives y_i and y_j, we then define the degree of preference $P(y_i, y_j)$ in terms of the fraction of rankings in which y_i precedes y_j:

$$P(y_i, y_j) \;=\; \frac{1}{k} \left| \{t \,|\, y_i \succ_t y_j\} \right| . \tag{7.2}$$

Thus, $P(y_i, y_j) = 1$ suggests a consensus among the ensemble members, since all of them agree that y_i should precede y_j. On the other hand, $P(y_i, y_j) \approx 1/2$ indicates a highly uncertain situation.

7.1.3 Prediction of a Strict Partial Order Relation

On the basis of the preference relation P, we seek to induce a (partial) order relation $\sqsupset$ on $\mathcal{Y}$, that we shall subsequently also denote by $\mathcal{R}$. Thus, $\mathcal{R}$ is an $\mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ mapping or, equivalently, a subset of $\mathcal{Y} \times \mathcal{Y}$, where $\mathcal{R}(y_i, y_j) = 1$, also written as $(y_i, y_j) \in \mathcal{R}$ or $y_i \,\mathcal{R}\, y_j$, indicates that $y_i \sqsupset y_j$.

The simplest idea is to let $y_i \,\mathcal{R}\, y_j$ iff $P(y_i, y_j) = 1$. The relation $\mathcal{R}$ thus defined indeed leads to a (strict) partial order, but since a perfect consensus ($P(y_i, y_j) \in \{0, 1\}$) is a strong requirement, most alternatives will be declared incomparable. Seeking a prediction that is as informative as possible, it is therefore natural to reduce the required degree of consensus. We therefore proceed from an "α-cut" of the relation P, defined as

$$\mathcal{R}_\alpha = \{(y_i, y_j) \mid P(y_i, y_j) > \alpha\} \tag{7.3}$$

for $1/2 \leq \alpha \leq 1$. A cut of that kind provides a reasonable point of departure, as it comprises the most certain preference statements while ignoring those comparisons (y_i, y_j) with $P(y_i, y_j) \leq \alpha$. However, it is not necessarily transitive and may even contain cycles. For example, suppose $y_i \succ_1 y_j \succ_1 y_k$, $y_j \succ_2 y_k \succ_2 y_i$ and $y_k \succ_3 y_i \succ_3 y_j$. Clearly, $P(y_i, y_j) = P(y_j, y_k) = P(y_k, y_i) = 2/3$, rendering $\mathcal{R}_{2/3}$ a cyclical relation. While transitivity is easily enforced by

[1] In the case of an instance-based label ranker, instead of using the ensemble, such k rankings can also be provided by the k nearest neighbors of the query instance.

computing the transitive closure of $\mathcal{R}_\alpha$, absence of cycles is not as easily obtained. Intuitively, it seems natural that for larger α, cycles become less probable. However, as the example shows, for $\alpha > 1/2$, cycles can still occur. Furthermore, the larger α, the less informative the corresponding $\mathcal{R}_\alpha$.

Consequently, we propose to look for a minimal α (denote it as α^*) such that the transitive closure of $\mathcal{R}_\alpha$ (denote it as $\overline{\mathcal{R}}_\alpha$) is a strict partial order relation [50]. This $\overline{\mathcal{R}}_{\alpha^*}$ will be the predicted strict partial order relation $\mathcal{R}$, and we call α^* the consensus threshold. By minimizing this threshold, we maximize $\mathcal{R}_\alpha$ as well as its transitive closure $\overline{\mathcal{R}}_\alpha$, and thereby also the information extracted from the ensemble on the basis of which P was computed. In what follows, we deal with the problem of computing α^* in an efficient way.

7.1.4 Determination of an Optimal Threshold

Suppose that P can assume only a finite number of values. In our case, according to (7.2), this set is given by $D = \{0, 1/k, 2/k, \ldots, 1\}$, and its cardinality by $k + 1$, where k is the ensemble size. Obviously, the domain of α can then be restricted to D. The simplest approach, therefore, is to test each value in D, i.e., to check for each value whether $\mathcal{R}_\alpha$ is acyclic, and hence $\overline{\mathcal{R}}_\alpha$ a partial order. Of course, instead of trying all values successively, it makes sense to exploit a monotonicity property: If $\mathcal{R}_\alpha$ is not acyclic, then $\mathcal{R}_\beta$ cannot be acyclic either, unless $\beta > \alpha$. Consequently, α^* can be found in at most $\log_2(k + 1)$ steps using bisection. More specifically, by noting that α^* is lower-bounded by

$$\alpha_l = \frac{1}{k} + \max_{y_i, y_j} \min\left(P(y_i, y_j), P(y_j, y_i)\right) \tag{7.4}$$

(which is larger than 1/2) and trivially upper-bounded by $\alpha_u = 1$, one can repeatedly update the bounds as follows, until $\alpha_u - \alpha_l < 1/k$:

1. set α to the middle point between α_l and α_u

2. compute $\mathcal{R}_\alpha$

3. compute $\overline{\mathcal{R}}_\alpha$ (e.g., using the Floyd-Warshall's algorithm [63])

4. if $\overline{\mathcal{R}}_\alpha$ is a partial order, set α_u to α

5. else set α_l to α

This procedure stops with $\alpha^* = \alpha_l$. The complexity of this procedure is not worse than the transitive closure operation, i.e., it is at most $\mathcal{O}(|\mathcal{Y}|^3)$.

As shown in [50], the same result can be computed with another algorithm that is conceptually simpler (though equally costly in terms of complexity, at least theoretically). This algorithm operates on an $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix $\mathbf{R}$ initialized with the entries $P(y_i, y_j)$ (recall that $\mathcal{Y}$ is the set of alternatives). It repeatedly performs a transitive closure operation on all the levels of D simultaneously:

$$\mathbf{R}(y_i, y_j) \leftarrow \max\left(\mathbf{R}(y_i, y_j), \max_{y_k \in \mathcal{Y}}(\min(\mathbf{R}(y_i, y_k), \mathbf{R}(y_k, y_j)))\right) \tag{7.5}$$

for all $y_i, y_j \in \mathcal{Y}$, until no further changes occur. These transitive closure operations can be seen as a correction of inconsistences in P (y_i is to some degree preferred to y_j, which in turn is to some degree preferred to y_k, but y_i is not sufficiently preferred to y_k). Since these inconsistencies do not occur very often, the number of update operations needed to stabilize $\mathbf{R}$ is normally quite small. In practice, we found that we rarely need more than one or two iterations.

By construction, thresholding the final relation $\mathbf{R}$ at a level α will yield the transitive closure of relation $\mathcal{R}_\alpha$ in (7.3). Therefore, α^* can be taken as

$$\alpha^* = \frac{1}{k} + \max\left(\mathbf{R}(y_i, y_j) \,|\, \mathbf{R}(y_i, y_j) \leq \mathbf{R}(y_j, y_i)\right) , \tag{7.6}$$

which is obviously the smallest α that avoids cycles. The whole procedure is summarized in Algorithm 4.

Algorithm 4

Require: training data $\mathcal{T}$ as defined in Table 2.3, a test instance $\boldsymbol{x}$, ensemble size k, base learner L

Ensure: a matrix $\mathbf{R}$ encoding partial order information of labels for $\boldsymbol{x}$ ($\mathbf{R}(i,j) = 1$ means $y_i \succ_{\boldsymbol{x}} y_j$, where $y_i, y_j \in \mathcal{Y}$)

initialize $\mathbf{R}$ as zero matrix, $\alpha := 1/2$
generate k bootstrap samples from $\mathcal{T}$
constitute the ensemble with k rankers trained using L
get k rankings of labels for $\boldsymbol{x}$
for each of k rankings **do**
  for every pair y_i, y_j in the ranking **do**
    if $y_i \succ y_j$ **then**
      set $\mathbf{R}(i,j) := \mathbf{R}(i,j) + 1/k$
    end if
  end for
end for
repeat
  for every entry in $\mathbf{R}$ **do**
    $\mathbf{R}(i,j) := \max\big(\mathbf{R}(i,j), \max_{k\in\{1,\ldots,|\mathcal{Y}|\}}(\min(\mathbf{R}(i,k), \mathbf{R}(k,j)))\big)$
  end for
until No entry in $\mathbf{R}$ is changed.
for every entry in $\mathbf{R}$ **do**
  if $\alpha < \min(\mathbf{R}(i,j), \mathbf{R}(j,i))$ **then**
    $\alpha := \min(\mathbf{R}(i,j), \mathbf{R}(j,i))$
  end if
end for
for every entry in $\mathbf{R}$ **do**
  if $\mathbf{R}(i,j) > \alpha$ **then**
    $\mathbf{R}(i,j) := 1$
  end if
end for

Finally, we note that, as postulated above, α^* in (7.6) yields a maximal partial order as a prediction. In principle, of course, any larger value can be used as well, producing a less complete relation and, therefore, a more "cautious" prediction. We shall come back to this issue in Section 7.3.

7.1.5 An Illustrative Example

To illustrate the idea of using ensemble approach, we demonstrate it by means of a small two-dimensional toy example for bipartite ranking, where the goal is to rank positive classes ahead of the negative classes (see Table 2.2).[2] Suppose that the conditional class distributions of the positive and the negative class are two overlapping Gaussians. A training data set may then look like the one depicted in Figure 7.1 (left), with positive examples as black and negative examples as white dots. Given a new set of query instances X to be ranked, one may expect that a learner will be uncertain for those instances lying close to the overlap region, and may hence prefer to abstain from comparing them.

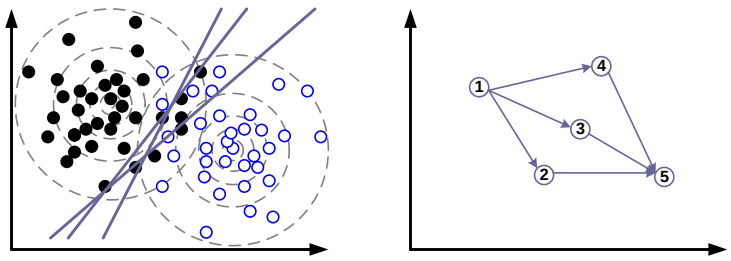

Figure 7.1: Left: training data and ensemble models; right: partial order predicted for a set of five query instances.

Specifically, suppose that a linear model is used to train a ranker. Roughly speaking, this means fitting a separating line and sorting instances according to their distance from the decision boundary. Figure 7.1 (left) shows several such models that may result from different bootstrap samples. Now, consider the five query instances shown in the right picture of Figure 7.1. While all

[2]We use the bipartite ranking problem as it is much easy to visualize. Algorithm 4 can be applied to a bipartite ranking problem with minor changes.

these models will rank instance 1 ahead of 2, 3 and 4, and these in turn ahead of 5, instances 2, 3 and 4 will be put in various orders. Applying our approach as outlined before, with a proper choice of the threshold α, may then yield the strict partial order indicated by the arrows in the right picture of Figure 7.1. A prediction of that kind agrees with our expectation: Instance 1 is ranked first and instance 5 last; instances 2, 3 and 4 are put in the middle, but the learner abstains from comparing them in a mutual way.

7.2 Abstention by Thresholding Probability Distributions in Label Ranking

The method proposed in Section 7.1 consists of two main steps and can be considered as a pairwise approach in the sense that, as a point of departure, a valued preference relation $P : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is produced, where $P(y_i, y_j)$ is interpreted as a measure of support of the pairwise preference $y_i \succ y_j$. Support is commonly interpreted in terms of probability, hence P is assumed to follow (7.1), that is, to be reciprocal for all $y_i, y_j \in \mathcal{Y}$. Then, in a second step, a partial order $\mathcal{R}$ is derived from P via thresholding: $\mathcal{R}(y_i, y_j) = 1$ if $P(y_i, y_j) > \alpha$ and $\mathcal{R}(y_i, y_j) = 0$ otherwise, where $1/2 \leq \alpha < 1$ is a threshold. Thus, the idea is to predict only those pairwise preferences that are sufficiently likely, while abstaining on pairs (y_i, y_j) for which the probability $P(y_i, y_j)$ is too close to $1/2$.

The first step of deriving the relation P is realized by means of an ensemble learning technique, and the preference degrees $P(y_i, y_j)$ are essentially independent of each other. Or, stated differently, they do not guarantee any specific properties of the relation P except being reciprocal. For the relation $\mathcal{R}$ derived from P via thresholding, this has two important consequences:

- If the threshold α is not large enough, then $\mathcal{R}$ may have cycles. Thus, not all thresholds in $[1/2, 1)$ are actually feasible. In particular, if $\alpha = 1/2$ cannot be chosen, this also implies that the method may not

be able to predict a total order as a special case.

- Even if $\mathcal{R}$ does not have cycles, it is not guaranteed to be transitive.

In order to tackle the above problems in the label ranking setting, the idea we discuss here is to restrict the relation P so as to exclude the possibility of cycles and violations of transitivity from the very beginning. To this end, we take advantage of methods for label ranking that produce (parameterized) probability distributions over Ω as predictions. Our main theoretical result is to show that thresholding pairwise preferences induced by such distributions yields preference relations with the desired properties, that is, partial order relations $\mathcal{R}$.

Given a probability distribution on the set of rankings Ω, the probability of a pairwise preference $y_i \succ y_j$ (and hence the corresponding entry in the preference relation P) can be derived through marginalization:

$$P(i,j) = \Pr(y_i \succ y_j) = \sum_{\pi \in E(i,j)} \Pr(\pi) \,, \tag{7.7}$$

where $E(i,j)$ denotes the set of linear extensions of the incomplete ranking $y_i \succ y_j$, i.e., the set of all rankings $\pi \in \Omega$ in which y_i precedes y_j. Our main theoretical result states that thresholding (7.7) yields a proper partial order relation $\mathcal{R}$, both for the Mallows and the PL model.

Theorem 7. *Let* Pr *in (7.7) be the Mallows model (4.1), with a distance D having the so-called transposition property, or the PL model (4.5). Moreover, let $\mathcal{R}$ be defined by the thresholded relation $\mathcal{R}(y_i, y_j) = 1$ if $P(y_i, y_j) > \alpha$ and $\mathcal{R}(y_i, y_j) = 0$ otherwise. Then $\mathcal{R}$ defines a proper partial order relation for all $\alpha \in [1/2, 1)$.*

Definition 1. *A distance D on rankings is said to have the transposition property, if the following holds: Let π and π' be rankings and let (i,j) be a conflict pair, i.e., $(i,j) \in \{1,\ldots,n\}^2$ such that $i < j$ and $(\pi(i) - \pi(j))(\pi'(i) - \pi'(j)) < 0$. Let the ranking π'' be constructed from π by swapping y_i and*

y_j, *that is,* $\pi''(i) = \pi'(j)$, $\pi''(j) = \pi'(i)$ *and* $\pi''(m) = \pi'(m)$ *for all* $m \in \{1, \ldots, n\} \setminus \{i, j\}$. *Then,* $D(\pi, \pi'') \leq D(\pi, \pi')$.

Lemma 2. *The Kendall distance* T*, the Spearman distance* S*, and the Spearman footrule* F *satisfy the transposition property.*

Proof. See [16]. □

Lemma 3. *Suppose that, in the Mallows model,* D *satisfies the transposition property, and that* y_i *precedes* y_j *in the center ranking* π_0*. Then,* $\Pr(y_i \succ y_j) \geq 1/2$.

Proof. For every ranking $\pi \in \Omega$, let $b(\pi) = \pi$ if y_i precedes y_j in π; otherwise, $b(\pi)$ is defined by swapping y_i and y_j in π. Obviously, $b(\cdot)$ defines a bijection between $E(i, j)$ and $E(j, i)$. Moreover, since D has the transposition property, $D(b(\pi), \pi_0) \leq D(\pi, \pi_0)$ for all $\pi \in \Omega$. Therefore, according to the Mallows model, $\Pr(b(\pi)) \geq \Pr(\pi)$, and hence

$$\begin{aligned} \Pr(y_i \succ y_j) &= \sum_{\pi \in E(i,j)} \Pr(\pi) \\ &\geq \sum_{\pi \in E(i,j)} \Pr(b^{-1}(\pi)) \\ &= \sum_{\pi \in E(j,i)} \Pr(\pi) \\ &= \Pr(y_j \succ y_i) \ . \end{aligned} \tag{7.8}$$

Since, moreover, $\Pr(y_i \succ y_j) = 1 - \Pr(y_j \succ y_i)$, it follows that $\Pr(y_i \succ y_j) \geq 1/2$. □

Lemma 4. *Suppose that, in the Mallows model,* D *satisfies the transposition property, and that* $\Pr(y_i \succ y_j) > \alpha \geq 1/2$*. Then,* y_i *precedes* y_j *in the center ranking* π_0.

Proof. It follows from Lemma 3 by contradiction: If y_j would precede y_j, then $\Pr(y_j \succ y_i) \geq 1/2$, and therefore $\Pr(y_i \succ y_j) = 1 - \Pr(y_j \succ y_i) \leq 1/2$. □

Lemma 5. *Suppose that, in the Mallows model, D satisfies the transposition property. Moreover, suppose that y_i precedes y_j and y_j precedes y_k in the center ranking π_0. Then,*

$$\Pr(y_i \succ y_k) \geq \max\left(\Pr(y_i \succ y_j), \Pr(y_j \succ y_k)\right) \ . \tag{7.9}$$

Proof. We show that $\Pr(y_i \succ y_k) \geq \Pr(y_i \succ y_j)$. The second inequality $\Pr(y_i \succ y_k) \geq \Pr(y_j \succ y_k)$ can be shown analogously.

Let $E(i,j,k)$ denote the set of linear extensions of $y_i \succ y_j \succ y_k$, i.e., the set of rankings $\pi \in \Omega$ in which y_i precedes y_j and y_j precedes y_k.

Now, for every $\pi \in E(k,j,i)$, define $b(\pi)$ by first swapping y_k and y_j and then y_k and y_i in π. Obviously, $b(\cdot)$ defines a bijection between $E(k,j,i)$ and $E(j,i,k)$. Moreover, due to the transposition property, $D(b(\pi),\pi_0) \leq D(\pi,\pi_0)$, and therefore $\Pr(b(\pi)) \geq \Pr(\pi)$ under the Mallows model. Consequently, since

$$E(i,j) = E(i,j,k) \cup E(i,k,j) \cup E(k,i,j) \ , \tag{7.10}$$

$$E(i,k) = E(i,k,j) \cup E(i,j,k) \cup E(j,i,k) \ , \tag{7.11}$$

it follows that

$$\begin{aligned} \Pr(y_i \succ y_k) - \Pr(y_i \succ y_j) &= \sum_{\pi \in E(i,k) \setminus E(i,j)} \Pr(\pi) \\ &= \sum_{\pi \in E(j,i,k)} \Pr(\pi) - \sum_{\pi \in E(k,j,i)} \Pr(\pi) \\ &= \sum_{\pi \in E(k,j,i)} \underbrace{\Pr(b(\pi)) - \Pr(\pi)}_{\geq 0} \geq 0 \ . \end{aligned} \tag{7.12}$$

□

Lemma 6. *Suppose that, in the Mallows model, D satisfies the transposition property. Then, the relation P defined by $P(i,j) = \Pr(y_i \succ y_j)$ satisfies the*

following property (closely related to strong stochastic transitivity):

$$\big((P(i,j) > \alpha) \wedge (P(j,k) > \alpha)\big) \Rightarrow\ P(i,k) \geq \max(P(i,j), P(j,k)) \quad (7.13)$$

for all $\alpha \geq 1/2$ *and all* $i, j, k \in \{1, \ldots, n\}$.

Proof. This follows from Lemmas 3, 4 and 5. □

The proof of Theorem 7 for the Mallows model follows immediately: Since $\Pr(y_i \succ y_j) = 1 - \Pr(y_j \succ y_i)$, it follows that $\mathcal{R}(y_i, y_j) = 1$ implies $\mathcal{R}(y_j, y_i) = 0$. Moreover, Lemma 6 implies that $\mathcal{R}$ is transitive. Consequently, $\mathcal{R}$ defines a proper partial order relation.

The proof of Theorem 7 for the PL model is more straightforward: For $\alpha \geq 1/2$, $\mathcal{R}$ being irreflexive is directly obtained due to (4.6) and its transitivity can be shown by the following lemma.

Lemma 7. *For any* $i, j, k \in \{1, \ldots, n\}$, $1/2 \leq \alpha < 1$, *if* $\frac{v_i}{v_i + v_j} > \alpha$ *and* $\frac{v_j}{v_j + v_k} > \alpha$, *we have* $\frac{v_i}{v_i + v_k} > \alpha$.

Proof. Since $\frac{v_i}{v_i + v_j} > \alpha$, we have

$$\frac{1-\alpha}{\alpha} v_i > v_j\ . \quad (7.14)$$

Analogously,

$$\frac{1-\alpha}{\alpha} v_j > v_k\ . \quad (7.15)$$

From (7.14) and (7.15), we have

$$\frac{v_i}{v_i + v_k} = \frac{1}{1 + \frac{v_k}{v_i}} > \frac{1}{1 + (\frac{1-\alpha}{\alpha})^2} \quad (7.16)$$

For $1/2 \leq \alpha < 1$, since

$$\frac{1}{1 + (\frac{1-\alpha}{\alpha})^2} - \alpha \geq 0 \iff (1-\alpha)(1-2\alpha) \leq 0 \quad (7.17)$$

always holds, it leads to $\frac{v_i}{v_i + v_k} > \alpha$. □

Since the results of ranking are invariant towards relabeling of the labels and, correspondingly, the indexes of $\mathbf{v}$, Lemma 7 effectively means that, under the PL model, any threshold larger than or equal to $1/2$ guarantees an transitive relation obtained by P.

7.3 Evaluation Measures

If a model is allowed to abstain from making predictions, it is expected to reduce its error rate. In fact, it can trivially do so, namely by rejecting all predictions, in which case it avoids any mistake. Clearly, this is not a desirable solution. Indeed, in the setting of prediction with reject option, there is always a trade-off between two criteria: correctness on the one side and completeness on the other side. An ideal learner is correct in the sense of making few mistakes, but also complete in the sense of abstaining but rarely. The two criteria are normally conflicting: increasing completeness typically comes along with reducing correctness and vice versa.

7.3.1 Correctness

As a measure of correctness, we propose a quantity that is also known as the gamma rank correlation [32] in statistics, although it is not applied to partial orders. Instead, it is used as a measure of correlation between rankings (with ties). As will be seen, however, it can also be used in a more general way.

Let $\succ_*$ be the ground-true relation on the set of labels $\mathcal{Y}$. This relation is a total order, so $y_i \succ_* y_j$ if y_i precedes y_j and $y_j \succ_* y_i$ if y_j precedes y_i; exactly one of these two cases is true, i.e., we never have $y_i \perp_* y_j$. Now, let $\sqsupset$ be a predicted (strict) partial order, i.e., a prediction of $\succ_*$. We call a pair of labels y_i and y_j concordant if they are compared in the correct way, that is,

$$(y_i \succ_* y_j \wedge y_i \sqsupset y_j) \vee (y_j \succ_* y_i \wedge y_j \sqsupset y_i) \,. \tag{7.18}$$

Likewise, we call y_i and y_j discordant if the comparison is incorrect, that is,

$$(y_i \succ_* y_j \wedge y_j \sqsupset y_i) \vee (y_j \succ_* y_i \wedge y_i \sqsupset y_j) \,. \tag{7.19}$$

Given these notions of concordance and discordance, we can define

$$\mathrm{CR}(\sqsupset, \succ_*) = \frac{|C| - |D|}{|C| + |D|} \,, \tag{7.20}$$

where C and D denote, respectively, the set of concordant and discordant pairs of labels. Obviously, $\mathrm{CR}(\sqsupset, \succ_*) = 1$ for $\succ_* = \sqsupset$ and $\mathrm{CR}(\sqsupset, \succ_*) = -1$ if $\sqsupset$ is the inversion of $\succ_*$.

Note that (7.20) reduces to commonly used Kendall's tau (2.3) in the complete (non-partial) case, that is, when $\sqsupset$ is a total order.

7.3.2 Completeness

To measure the degree of completeness of a prediction, a straightforward idea is to punish the abstention from comparisons that should actually be made. This leads to the following measure of completeness:

$$\mathrm{CP}(\sqsupset) = \frac{|C| + |D|}{|\succ_*|} = \frac{2}{n(n-1)} \left(|C| + |D|\right) \,, \tag{7.21}$$

where $n = |\mathcal{Y}|$ is the number of labels to be ranked.

7.4 Experiments

In this section, we empirically evaluate these two proposed approaches, where we have tested on a selection of label ranking data sets presented in Chapter 4. We performed five repetitions of the 10-fold cross-validation and used an ensemble size of 10. As a label ranking method, we used the RPC approach with logistic regression as a base learner.

The averaged results of the approach proposed in Section 7.1 are summarized in Table 7.1. It can be seen that our approach of partial abstention leads to improved performance. In fact, it is never worse and sometimes yields better results with significant margins. Moreover, this gain in performance comes with an acceptable loss in terms of completeness. The degrees of completeness are quite high throughout, significantly above 90%.

data set	#attr.	#cls.	#inst.	correctness with abstention	correctness w/o abstention	completeness
iris	4	3	150	0.910±0.062	0.885±0.068	0.991±0.063
wine	13	3	178	0.940±0.051	0.921±0.053	0.988±0.067
glass	9	6	214	0.892±0.039	0.882±0.042	0.990±0.030
vowel	10	11	528	0.657±0.019	0.647±0.019	0.988±0.016
vehicle	18	4	846	0.858±0.026	0.854±0.025	0.992±0.039
authorship	70	4	841	0.941±0.016	0.910±0.015	0.989±0.043
pendigits	16	10	10992	0.933±0.002	0.932±0.002	0.999±0.005
segment	18	7	2310	0.938±0.006	0.934±0.006	0.998±0.011

Table 7.1: Results for label ranking: mean values and standard deviations for correctness and completeness.

We conducted a second experiment with the aim to investigate the trade-off between correctness and completeness. As was mentioned earlier, and to some extent already confirmed by our first experiment, we expect a compromise between both criteria insofar as it should be possible to increase correctness at the cost of completeness. To verify this conjecture, we varied the threshold α in (7.3) in the range $[\alpha^*, 1]$. Compared to α^*, larger thresholds will make the predictions increasingly incomplete; at the same time, however, they should also become more correct. Indeed, the results we obtained are well in agreement with these expectations. Figure 7.2 shows typical examples of the trade-off between correctness and completeness for two data sets.

For the alternative approach we proposed in Section 7.2 for the label ranking problem, it should be noticed that this approach, despite being appealing from the theoretical point of view, does not automatically imply a practical advantage, especially since it makes strong model assumptions (in

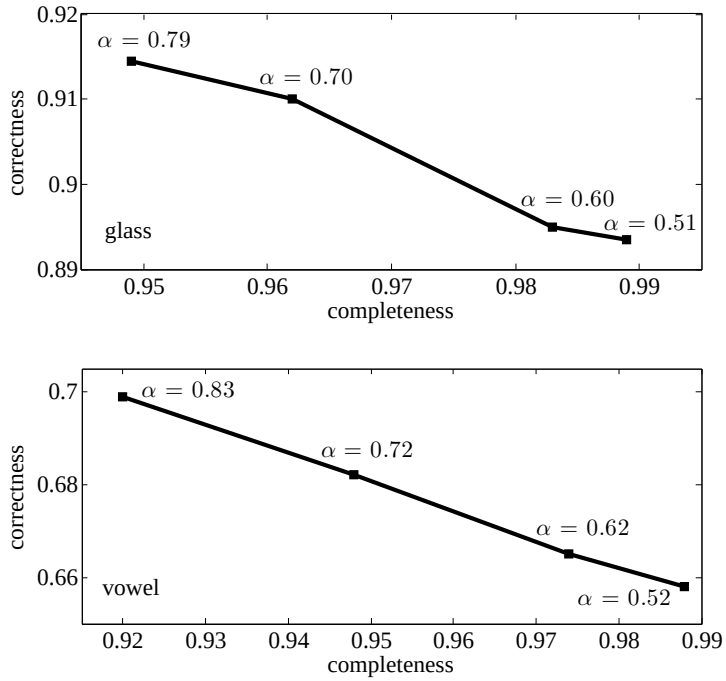

Figure 7.2: Label ranking with partial abstention: Trade-off between correctness and completeness for selected data sets.

terms of the Mallows or PL model) that are not necessarily satisfied. Therefore, we complement our theoretical results by an empirical study shown in Figure 7.3.

The main conclusion can be drawn from our results is that, as expected, the probabilistic approach does indeed achieve a better trade-off between completeness and correctness, especially in the sense that it spans a wider range of values for the former. Besides, we often observe that the level of correctness is increased, too.

7.5 Chapter Conclusion

In this chapter, we have addressed the problem of reliable prediction in the context of learning to rank. Based on the idea of allowing a learner to abstain from an uncertain comparison of alternatives, together with the requirement

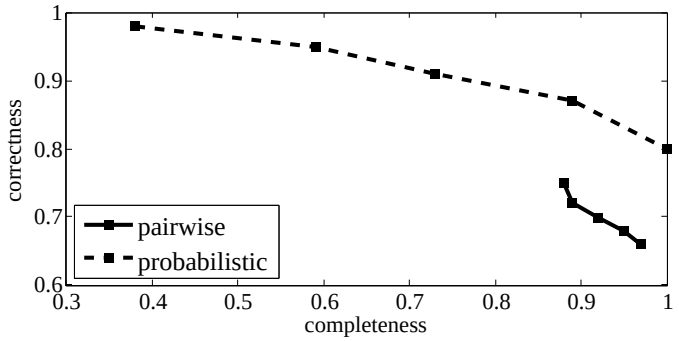

Figure 7.3: Trade-off between completeness and correctness for a label ranking variant of the UCI benchmark data set vowel: The pairwise method proposed in Section 7.1 (solid line) versus the approach based on probabilistic models proposed in Secion 7.2 (dashed line).

that predictions are consistent, we have proposed a relaxation of the conventional setting in which predictions are given in terms of partial instead of total orders. We have proposed a generic approach to predicting partial orders or, according to our interpretation, ranking with partial abstention. We have also proposed a method based on the idea of thresholding the probabilities of pairwise preferences between labels. It can be shown that, when such pairwise preferences are induced by some particular probability distribution on rankings, thresholding can be safely done in a sense that it guarantees that a proper partial order relation is predicted. To evaluate the performance of a ranker with (partial) reject option, measures of correctness and completeness are introduced. Empirically, we have shown that our methods are indeed able to trade off accuracy against completeness: The correctness of a prediction can be increased at the cost of reducing the number of alternatives that are compared.

The extension from predicting total to predicting partial orders as proposed in this chapter opens the door for a multitude of further studies. In this chapter, we have essentially assumed that the target is a complete order, and a prediction in terms of a partial order $\sqsupset$ is an incomplete estimation thereof. Therefore, we do not penalize the case where $y_i \sqsupset y_j$ even though

$y_i \perp_* y_j$. Now, if $\sqsupset_*$ is a true partial order, it clearly makes sense to request, not only the correct prediction of order relations $y_i \sqsupset_* y_j$ between alternatives, but also of incomparability relations $y_i \perp_* y_j$. Although the difference may look subtle at first sight, the changes will go beyond the evaluation of predictions and instead call for different learning algorithms. In particular, in this latter scenario, $y_i \perp y_j$ will be interpreted as a prediction that y_i and y_j are incomparable ($y_i \perp_* y_j$), and not as a rejection of the decision whether $y_i \sqsupset_* y_j$ or $y_j \sqsupset_* y_i$. Nevertheless, the two settings are of course related, and their connection is worth a deep study in the future work.

Bibliographical notes

Parts of the results presented in this chapter have been published in:

- Weiwei Cheng, Michaël Rademaker, Bernard De Baets, and Eyke Hüllermeier. Predicting partial orders: ranking with abstention. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 215-230. Springer, 2010.

Chapter 8

Conclusion

The topic of label ranking has attracted increasing attention in the recent machine learning literature [27, 36, 33, 17, 10]. It is a particular preference learning scenario, studies the problem of learning a mapping from instances to rankings of over a finite number of predefined labels. This setting is versatile and generalizes a number of different other learning settings. When only the top ranked label is requests, label ranking reduces to a conventional classification problem; when a calibrated label is introduced, the output of label ranking can be considered as a multi-label prediction. Because of the versatility, label ranking has its application in a lot of learning tasks, such as natural language processing, customer modeling, bioinformatics, etc. [27]

Not surprisingly, quite a number of label ranking algorithms have been proposed in the literature, where two main general frameworks exist, namely label ranking by learning utility functions and label ranking by learning preference relations. In Chapter 3 we have discussed some of them. As we showed, most of the existing approaches use reduction techniques to approach label ranking problem indirectly by solving a set of classification problems. Reduction techniques have shown promising performance in experimental studies. Moreover, the reduction of the label ranking problem to the simpler problem of classification is appealing for several reasons. Notably, it makes

the label ranking problem amenable to the large repertoire of (binary) classification methods and existing algorithms in this field. On the other hand, reduction techniques also come with some disadvantages. In particular, theoretical assumptions on the sought "ranking-valued" mapping, which may serve as a proper learning bias, may not be easily translated into corresponding assumptions for the classification problems. Likewise, it is often not clear (and mostly even wrong) that minimizing the classification error, or a related loss function, on the binary problems is equivalent to maximizing the (expected) performance of the label ranking model in terms of the desired loss function on rankings. In this thesis, to avoid these problems to some extent, we propose the label ranking methods on the basis of statistical models for ranking data, that is, parameterized (conditional) probability distributions on the class of all rankings. Given assumptions of that kind, the learning problem can be posed as a problem of maximum likelihood estimation and thus be solved in a theoretically sound way. In particular, in Chapter 4 we have made use of the Mallows and Plackett-Luce model and developed an instance-based (nearest neighbor) learning algorithm to estimate the models in a local way. Moreover, apart from the estimation of locally constant models suitable for instance-based learning, we also develop a method for estimating generalized linear models based on the Plackett-Luce model in Chapter 5. An advantage of using probabilistic methods is that it delivers, as a byproduct, natural measures of the reliability of a prediction, which are often not directly provided by existing approaches. Moreover, due to the versatility of the label ranking setting, the use of probabilistic methods also provides means to analyze other learning problems. In Chapter 6, a simple yet powerful algorithm for multi-label learning is proposed, based on the theoretical analysis with the Mallows model.

Unlike classification, a lot of aspects in label ranking have not yet been addressed in the literature and worth further investigation. For example, Chapter 7 dedicates to learning with reject option, which is a well-studied topic in classification but not in label ranking so far. As we argued earlier,

this setting is even more interesting and challenging in label ranking, since the learner can reject to a certain degree. Moreover, in order to guarantee the outputs are proper rankings, i.e., partial orders, the abstention of comparisons between labels cannot be made independently. In that chapter, two approaches have been proposed to solve this problem.

As we already discussed in the beginning of the thesis, the ranking of different alternatives can often be interpreted as preference information, and indeed the label ranking problem is intensively studied as a sub-field of preference learning [27]. Roughly speaking, preference learning is about inducing predictive preference models from empirical data. In Chapter 2, we have outlined three intensively studied preference learning problems, object ranking, instance ranking, and label ranking. Although this thesis studies label ranking exclusively, many developed techniques for label ranking apply to the other two settings as well. In fact, the theoretical analysis of relationships between these settings and unifying different learning to rank problems in a sound way are valuable future research topics in preference learning.

Bibliography

[1] Davia Aha, Dennis Kibler, and Marc Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In Harold Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 684–693. ACM Press, 2005.

[3] Noga Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20(1):134–142, 2006.

[4] Kenneth Arrow. *Social Choice and Individual Values*. Yale University Press, 2nd edition, 1963.

[5] Gökhan Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander Smola, Ben Taskar, and S. V. N. Vishwanathan, editors. *Predicting Structured Data*. MIT Press, 2007.

[6] Peter Bartlett and Marten Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

[7] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics*, pages 177–187. Springer, 2010.

[8] Matthew Boutell, Jiebo Luo, Xipeng Shen, and Christopher Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[9] Andrew Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[10] Klaus Brinker and Eyke Hüllermeier. Case-based label ranking. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 566–573. Springer, 2006.

[11] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 161–168. Omnipress, 2009.

[12] Amanda Clare and Ross King. Knowledge discovery in multi-label phenotype data. In Luc De Raedt and Arno Siebes, editors, *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2001.

[13] William Cohen, Robert Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[14] Francesco De Comite, Remi Gilleron, and Marc Tommasi. Learning multi-label alternating decision tree from texts and data. In Petra Perner and Azriel Rosenfeld, editors, *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 35–49. Springer, 2003.

[15] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

[16] Douglas Critchlow, Michael Fligner, and Joseph Verducci. Probability models on rankings. *Journal of Mathematical Psychology*, 35:294–318, 1991.

[17] Ofer Dekel, Christopher Manning, and Yoram Singer. Log-linear models for label ranking. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 497–504. MIT Press, 2004.

[18] Krzysztof Dembczyński, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multi-label classification via probabilistic classifier chains. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 279–286. Omnipress, 2010.

[19] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

[20] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[21] Persi Diaconis and Ronald Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society Series B*, 39(2):262–268, 1977.

[22] John Duchi, Lester Mackey, and Michael Jordan. On the consistency of ranking algorithms. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 327–334. Omnipress, 2010.

[23] Cynthia Dwork, Ravi Kumary, Moni Naorz, and D. Sivakumar. Rank aggregation methods for the web. In Vincent Shen, Nobuo Saito, Michael Lyu, and Mary Zurko, editors, *Proceedings of the 10th International Conference on World Wide Web*, pages 613–622. ACM P, 2001.

[24] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In Thomas Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 681–687. MIT Press, 2002.

[25] Michael Fligner and Joseph Verducci. Distance based ranking models. *Journal of the Royal Statistical Society B*, 48(3):359–369, 1986.

[26] Johannes Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

[27] Johannes Fürnkranz and Eyke Hüllermeier, editors. *Preference Learning*. Springer, 2010.

[28] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[29] Johannes Fürnkranz, Eyke Hüllermeier, and Stijn Vanderlooy. Binary decomposition methods for multipartite ranking. In Wray Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 359–374. Springer, 2009.

[30] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, editors, *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 20–33. Springer, 2004.

[31] Mithat Gönen and Glenn Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

[32] Leo Goodman and William Kruskal. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49(268):732–764, 1954.

[33] 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*, volume 15, pages 785–792. MIT Press, 2003.

[34] Radu Herbei and Marten Wegkamp. Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721, 2006.

[35] Eyke Hüllermeier and Johannes Fürnkranz. Comparison of ranking procedures in pairwise preference learning. In *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knoweldge-Based Systems*, pages 535–542. Università La Sapienza, 2004.

[36] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

[37] David Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):386–408, 2004.

[38] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[39] Thorsten Joachims. Optimizing search engines using clickthrough data. In Osamar Zaiane, Randy Goebel, David Hand, Daniel Keim, and Raymond Ng, editors, *Proceedings of the 8th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, pages 133–142. ACM Press, 2002.

[40] Hideto Kazawa, Tomonori Izumitani, Hirotoshi Taira, and Eisaku Maeda. Maximal margin labeling for multi-topic text categorization. In Lawrence Saul, Yair Weiss, and Leon Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 649–656. MIT Press, 2005.

[41] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors: a PTAS for weighted feedback arc set on tournaments. In David Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 95–103. ACM Press, 2007.

[42] Roni Khardon and Gabriel Wachman. Noise tolerant variants of the perceptron algorithm. *The Journal of Machine Learning Research*, 8:227–248, 2007.

[43] Robert Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

[44] Colin Mallows. Non-null ranking models. *Biometrika*, 44(1):114–130, 1957.

[45] John Marden. *Analyzing and Modeling Rank Data*. CRC Press, 1995.

[46] Oded Maron and Aparna Ratan. Multiple-instance learning for natural scene classification. In Jude Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning*, pages 341–349. Morgan Kaufmann, 1998.

[47] David McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[48] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[49] Robin Plackett. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1975.

[50] Michaël Rademaker and Bernard de Baets. A threshold for majority in the context of aggregating partial order relations. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1–4. IEEE, 2010.

[51] Frans Schalekamp and Anke van Zuylen. Rank aggregation: together we're strong. In Irene Finocchi and John Hershberger, editors, *Proceedings of the 11th Workshop on Algorithm Engineering and Experiments*, pages 38–51. SIAM, 2009.

[52] Robert Schapire and Yoram Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[53] Bernhard Schölkopf and Alexander Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization*. MIT Press, 2001.

[54] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[55] Shai Shalev-Shwartz and Yoram Singer. Efficiently learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7:1567–1599, 2006.

[56] Cees Snoek, Marcel Worring, Jan van Gemert, Jan-Mark Geusebroek, and Arnold Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In Klara Nahrstedt and Matthew Turk, editors, *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 421–430. ACM Press, 2006.

[57] Louis Thurstone. A law of comparative judgement. *Psychological Review*, 34:273–286, 1927.

[58] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis Vlahavas. Multilabel classification of music into emotions. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 325–330. Drexel University, 2008.

[59] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: an overview. *International Journal of Data Warehousing and Mining*, 3(3):1–17, 2007.

[60] Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-label text. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 721–728. MIT Press, 2003.

[61] Paravin Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete and Computational Geometry*, 4(1):101–115, 1989.

[62] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

[63] Stephen Warshall. A theorem on Boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.

[64] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[65] Ian Witten, Eibe Frank, and Mark Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[66] John Yellott. A relationship between Luce's choice axiom, Thurstone's theory of comparative judgement, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977.

[67] Minling Zhang and Zhihua Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[68] Minling Zhang and Zhihua Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[69] Zhihua zhou and Minling Zhang. Multi-instance multi-label learning with application to scene classification. In Bernhard Schölkopf, John Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, pages 1609–1616. MIT Press, 2007.

Weiwei Cheng

Curriculum Vitae

roywwcheng@gmail.com
www.chengweiwei.com

WORK EXPERIENCE

2007 – now **Researcher**
Machine learning research funded by German Research Foundation and Hessian Ministry of Science and the Arts
Knowledge Engineering and Bioinformatics Lab
Faculty of Mathematics and Computer Science
Philipps University Marburg, Germany

2011 – 2012 **Student consultant**
Consulting service for Asian students
Faculty of Mathematics and Computer Science
Philipps University Marburg, Germany

2010 **Research intern**
Improving machine learning with large-scale ontological knowledge
Machine Learning and Perception Group
Microsoft Research Cambridge, United Kingdom

2009 **Data mining consulting intern**
Analyzing and mining IT-related audit issues
Group Technology and Operations
Deutsche Bank
Eschborn, Germany

2006 – 2007 **Research assistant**
Efficient preference operators in very large database systems
Faculty of Computer Science
Otto-von-Guericke University Magdeburg, Germany

2003 **Software development intern**
Geographic information system for electricity distribution management
Xinli Software Co. Ltd.
Hefei, China

EDUCATION

2007 – now **PhD candidate in Computer Science**
Research interests: machine learning, data mining, preference learning, ranking, multi-label classification
Faculty of Mathematics and Computer Science
Philipps University Marburg, Germany

2005 – 2007 **Master's degree in Computer Science**
Subject: Data and Knowledge Engineering
GPA: 4.0/4.0 *with highest distinction* + *best graduate award*
Master's thesis: Interactive ranking of skylines using machine learning techniques
Otto-von-Guericke University Magdeburg, Germany

2000 – 2004 **Bachelor's degrees in Computer Science and Business Administration**
Bachelor's thesis of CS: Database systems for personnel management
Bachelor's thesis of BA: Retail analysis and development in Zhengzhou
Zhengzhou University, China

PUBLICATIONS

Dec. 2011 Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, Eyke Hüllermeier
An exact algorithm for F-measure maximization
Advances in Neural Information Processing Systems 24 (NIPS-11): 223-230, Curran Associates
Granada, Spain *NIPS travel grant*

Dec. 2011 Weiwei Cheng, Eyke Hüllermeier
Label ranking with abstention: predicting partial orders by thresholding probability distributions
Workshop Proceedings of Choice Models and Preference Learning (CMPL-11): arXiv:1112.0508v1 [cs.AI]
The 25th Annual Conference on Neural Information Processing Systems (NIPS-11)
Sierra Nevada, Spain *NIPS travel grant*

Dec. 2011 Thomas Fober, Weiwei Cheng, Eyke Hüllermeier
Focusing search in multiobjective evolutionary optimization through preference learning from user feedback
Proceedings of the 21st Workshop Computational Intelligence (WCI-11): 107-117, KIT Scientific Publishing
Dortmund, Germany

Oct. 2011 Weiwei Cheng, Gjergji Kasneci, Thore Graepel, David Stern, Ralf Herbrich
Automated feature generation from structured knowledge
Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM-11): 1395-1404, ACM
Glasgow, UK

Sep. 2011 Ali Fallah Tehrani, Weiwei Cheng, Krzysztof Dembczyński, Eyke Hüllermeier
Learning monotone nonlinear models using the Choquet integral
LNAI 6913 Machine Learning and Knowledge Discovery in Databases: 414-429, Springer
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-11)
Athens, Greece *invited to Machine Learning Journal Special Issue*

Sep. 2011 Weiwei Cheng, Johannes Fürnkranz, Eyke Hüllermeier, Sang-Hyeun Park
Preference-based policy iteration: leveraging preference learning for reinforcement learning
LNAI 6911 Machine Learning and Knowledge Discovery in Databases: 312-327, Springer
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-11)
Athens, Greece *invited to Machine Learning Journal Special Issue*

Jul. 2011 Ali Fallah Tehrani, Weiwei Cheng, Eyke Hüllermeier
Choquistic regression: generalizing logistic regression using the Choquet integral
Proceedings of the 7th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011): 868-875, Atlantis Press
Aix-les-Bains, France *nominated for best student paper award*

Dec. 2010 Ali Fallah Tehrani, Weiwei Cheng, Eyke Hüllermeier
Preference learning using the Choquet integral: the case of multipartite ranking
Proceedings of the 20th Workshop Computational Intelligence (WCI-10): 119-130, KIT Scientific Publishing
Dortmund, Germany

Sep. 2010 Weiwei Cheng, Michaël Rademaker, Bernard De Baets, Eyke Hüllermeier
Predicting partial orders: ranking with abstention
LNAI 6321 Machine Learning and Knowledge Discovery in Databases: 215-230, Springer
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-10)
Barcelona, Spain *UNESCO ECMLPKDD conference grant*

Sep. 2010 Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, Eyke Hüllermeier
Regret analysis for performance metrics in multi-label classification: the case of Hamming and subset zero-one loss
LNAI 6321 Machine Learning and Knowledge Discovery in Databases: 280-295, Springer
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-10)
Barcelona, Spain *UNESCO ECMLPKDD conference grant*

Jun. 2010 Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, Eyke Hüllermeier
On label dependence in multi-label classification
Workshop Proceedings of Learning from Multi-Label Data 2010 (MLD-10): 5-12
The 27th International Conference on Machine Learning (ICML-10)
Haifa, Israel *invited paper*

Jun. 2010 Weiwei Cheng, Krzysztof Dembczyński, Eyke Hüllermeier
Label ranking methods based on the Plackett-Luce model
Proceedings of the 27th International Conference on Machine Learning (ICML-10): 215-222, Omnipress
Haifa, Israel *ICML scholarship*

Jun. 2010 Weiwei Cheng, Krzysztof Dembczyński, Eyke Hüllermeier
Graded multi-label classification: the ordinal case
Proceedings of the 27th International Conference on Machine Learning (ICML-10): 223-230, Omnipress
Haifa, Israel *ICML scholarship*

Jun. 2010 Krzysztof Dembczyński, Weiwei Cheng, Eyke Hüllermeier
Bayes optimal multi-label classification via probabilistic classifier chains
Proceedings of the 27th International Conference on Machine Learning (ICML-10): 279-286, Omnipress
Haifa, Israel *ICML scholarship*

Sep. 2009 Weiwei Cheng, Eyke Hüllermeier
Label ranking with partial abstention using ensemble learning
Workshop Proceedings of Preference Learning 2009 (PL-09): 17-23
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-09)
Bled, Slovenia

Sep. 2009 Weiwei Cheng, Eyke Hüllermeier
A simple instance-based approach to multi-label classification using the Mallows model
Workshop Proceedings of Learning from Multi-Label Data 2009 (MLD-09): 28-38
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-09)

Bled, Slovenia

Sep. 2009 — Weiwei Cheng, Eyke Hüllermeier
Combining instance-based learning and logistic regression for multi-label classification
Machine Learning 76: 211-225, Springer
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-09)
Bled, Slovenia *Machine Learning Journal Best Student Paper Award*

Jun. 2009 — Weiwei Cheng, Jens Hühn, Eyke Hüllermeier
Decision tree and instance-based learning for label ranking
Proceedings of the 26th International Conference on Machine Learning (ICML-09): 161-168, Omnipress
Montreal, Canada *ICML scholarship*

May 2009 — Weiwei Cheng, Eyke Hüllermeier
A new instance-based label ranking approach using the Mallows model
LNCS 5551 Advances in Neural Networks: 707-716, Springer
The 6th International Symposium on Neural Networks (ISNN-09)
Wuhan, China

Oct. 2008 — Weiwei Cheng, Eyke Hüllermeier
Ranking skylines using active learning techniques
Proceedings of Chinese Intelligent Systems Engineering 2008 (CNISE-08)
Chengdu, China

Sep. 2008 — Weiwei Cheng, Eyke Hüllermeier
Instance-based label ranking using the Mallows model
Workshop Proceedings of Preference Learning 2008 (PL-08)
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-08)
Antwerp, Belgium

Sep. 2008 — Weiwei Cheng, Eyke Hüllermeier
Instance-based label ranking using the Mallows model
Workshop Proceedings of ECCBR-08: 143-157
Uncertainty and Knowledge Discovery in CBR
The 9th European Conference on Case-Based Reasoning (ECCBR-08)
Trier, Germany

Sep. 2008 — Weiwei Cheng, Eyke Hüllermeier
Learning similarity functions from qualitative feedback
LNAI 5239 Advances in Case-Based Reasoning: 120-134, Springer
The 9th European Conference on Case-Based Reasoning (ECCBR-08)
Trier, Germany *nominated for best paper award*

Aug. 2008 — Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, Klaus Brinker
Label ranking by learning pairwise preferences
Artificial Intelligence 172: 1897-1916, Elsevier *listed in Most Cited Artificial Intelligence Articles 2007-2012*

Oct. 2007 — Weiwei Cheng
Interactive ranking of skylines using machine learning techniques
Master's thesis *summa cum laude*
Faculty of Computer Science, Otto-von-Guericke University Magdeburg
Magdeburg, Germany

Sep. 2007 — Weiwei Cheng, Eyke Hüllermeier, Bernhard Seeger, Ilya Vladimirskiy
Interactive ranking of skylines using machine learning techniques

Proceedings of Lernen-Wissen-Adaptivität 2007 (LWA-07): 141-148, Martin Luther University
Halle, Germany

TALKS & TUTORIALS

Jul. 2012 **Preference-based reinforcement learning**
The 25th European Conference on Operational Research (EURO-2012)
Vilnius, Lithuania *DAAD conference scholarship*

Sep. 2011 **Learning monotone nonlinear models using the Choquet integral**
Lernen-Wissen-Adaptivität (LWA-11)
Magdeburg, Germany

Dec. 2010 **The RipOff! game: a tutorial of cooperative game theory**
with Yoram Bachrach
Think Computer Science
Cambridge, UK

Sep. 2010 **Graded multi-label classification: the ordinal case**
Lernen-Wissen-Adaptivität (LWA-10)
Kassel, Germany

Jun. 2010 **Acceptance speech of 2009 Chinese Government Award for Outstanding Self-Financed Students Abroad**
Embassy of China, Berlin, Germany

Nov. 2009 **Text classification: concepts and methods**
Association of Chinese Computer Scientists in Germany (GCI) Annual Conference
Fürth, Germany

Nov. 2009 **Text classification: a back-to-school tutorial**
GTO IES Audit & Risk Management Department, Deutsche Bank
Frankfurt, Germany

Oct. 2009 **Human vs. computer: case studies**
Association of Chinese Scholars and Students in Magdeburg (VCWSM e.V.)
Magdeburg, Germany

Oct. 2009 **Multi-label classification: a new machine learning problem**
Lightning Talk, the 4th Annual Google Test Automation Conference (GTAC-09)
Zurich, Switzerland

Sep. 2009 **Combining instance-based learning and logistic regression for multi-label classification**
Lernen-Wissen-Adaptivität (LWA-09)
Darmstadt, Germany

Sep. 2009 **Human-computation in Internet**
Association of Chinese Computer Scientists in Germany (GCI) IT Strategy Workshop
Schwetzingen, Germany

Nov. 2008 **Ranking the skyline: an application of preference learning**
Association of Chinese Computer Scientists in Germany (GCI) Annual Conference
Lahnstein, Germany

Sep. 2008 **Instance-based label ranking**
Student Talk, the 10th Machine Learning Summer School (MLSS-08)
Ile De Re, France *MLSS scholarship*

Mar. 2007 **A brief introduction to preference handling**
Colloquium for Graduate Students in Mathematics and Computer Science
Philipps University Marburg, Germany

PROFESSIONAL ACTIVITIES

Membership

Association for Computing Machinery (ACM)
Association of Chinese Computer Scientists in Germany (GCI)
Institute of Electrical and Electronics Engineers (IEEE)
International Machine Learning Society (IMLS)

Journal reviewer

Neural Processing Letters (Springer)
Artificial Intelligence (Elsevier)
Engineering Applications of Artificial Intelligence (Elsevier)
IEEE Transactions on Fuzzy Systems (IEEE)
Information Retrieval (Springer)
Journal of Algorithms (Elsevier)
Journal of Artificial Intelligence Research (AAAI Press)
Journal of Machine Learning Research (Microtome Publishing)
Machine Learning (Springer)
Pattern Recognition (Elsevier)
Soft Computing (Springer)
Uncertainty, Fuzziness and Knowledge-Based Systems (World Scientific)

Program committee membership

Feb. 2012 Knowledge Discovery and Data Mining Meets Linked Open Data workshop at the 9th Extended Semantic Web Conference (ESWC-12)

Jan. 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-12)

Jan. 2012 The 20th European Conference on Artificial Intelligence (ECAI-2012)

Mar. 2011 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-11)

Sep. 2009 Preference Learning workshop at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-09)

Jul. 2009 International Fuzzy Systems Association World Congress and European Society for Fuzzy Logic and Technology Conference (IFSA/EUSFLAT-09)

Sep. 2008 Preference Learning workshop at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-08)

Conference reviewer

Mar. 2012 The 29th International Conference on Machine Learning (ICML-12)

Feb. 2012 The 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-12)

Nov. 2011 2012 SIAM International Conference on Data Mining (SDM-12)

Nov. 2011	The 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-12)
Jun. 2011	The 20th ACM Conference on Information and Knowledge Management (CIKM-11)
May 2011	The 5th International Conference on Scalable Uncertainty Management (SUM-11)
Mar. 2011	The 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-11)
Jul. 2010	2010 IEEE International Conference on Data Mining (IEEEICDM-10)
Jul. 2010	The 19th ACM International Conference on Information and Knowledge Management (CIKM-10)
May 2010	The 11th International Conference on Parallel Problem Solving from Nature (PPSN-10)
Mar. 2010	The 10th Industrial Conference on Data Mining (ICDM-10)
Mar. 2010	The 36th Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB-10)
Feb. 2010	The 27th International Conference on Machine Learning (ICML-10)
Aug. 2009	The 26th IEEE International Conference on Data Engineering (ICDE-10)
Jul. 2009	2009 IEEE International Conference on Data Mining (IEEEICDM-09)
Jun. 2009	The 32nd Annual Conference on Artificial Intelligence (KI-09)
May 2009	European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD-09)
Apr. 2009	The 8th International Symposium on Intelligent Data Analysis (IDA-09)
Nov. 2008	2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM-09)
Jul. 2008	Advances in Data Analysis, Data Handling and Business Intelligence – 32nd Annual Conference of the German Classification Society, Joint Conference with the British Classification Society and the Dutch/Flemish Classification Society (GfKl-08)
May 2008	The 34th International Conference on Very Large Data Bases (VLDB-08)

TEACHING EXPERIENCE

2008 – 2010	Machine Learning

HONORS & AWARDS

Awards

Apr. 2012 **Data mining competition award**
The 2nd place at JRS 2012 Data Mining Competition
Chengdu, China

Jul. 2011 **Nomination for best student paper award**
The 7th Conference of the European Society for Fuzzy Logic and Technology,
Aix-les-Bains, France

Jun. 2010	**Outstanding Chinese student award** Chinese Government Award for Outstanding Self-Financed Students Abroad Ministry of Education, China
Sep. 2009	**Best paper award** Machine Learning Journal Best Student Paper Award The 19th European Conference on Machine Learning, Bled, Slovenia
Oct. 2008	**Best graduate award** Best Graduate 2007/2008 in Master of Data and Knowledge Engineering Otto-von-Guericke University Magdeburg, Germany
Sep. 2008	**Nomination for best paper award** The 9th European Conference on Case-Based Reasoning, Trier, Germany
Jun. 2002	**Best debater award** The 1st Varsity Debate Tournament of Henan, China
Aug. 1999	**Speech contest award** The 3rd prize at Speech Contest "The 50 Years", Anqing, China

Scholarships & Grants

Jul. 2012	**DAAD conference scholarship** German Academic Exchange Service (DAAD) Vilnius, Lithuania, July, 2012
Dec. 2011	**NIPS travel grant** Sponsored by Winton Capital, etc. The 25th Annual Conference on Neural Information Processing Systems, Granada, Spain
Apr. 2011	**DAAD STIBET scholarship** Sponsored by German Academic Exchange Service (DAAD) Marburg University Research Academy (MARA)
Sep. 2010	**UNESCO ECMLPKDD conference grant** Sponsored by UNESCO Chair in Data Privacy European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Barcelona, Spain
Jun. 2010	**ICML scholarship** Sponsored by IBM, National Science Foundation (NSF), etc. The 27th International Conference on Machine Learning, Haifa, Israel
Jun. 2009	**ICML scholarship** Sponsored by National Science Foundation (NSF), etc. The 26th International Conference on Machine Learning, Montreal, Canada
Sep. 2008	**MLSS scholarship** Sponsored by Predict & Control and Lille's Computer Science Laboratory (LIFL) The 10th Machine Learning Summer School, Ile De Re, France
Oct. 2006	**Outstanding international student scholarship** Ministry of Education and Cultural Affairs of Saxony-Anhalt, Germany

INTERESTS & SKILLS

Programming	Java, MATLAB, Octave, Python, C#, C, HTML, SQL, SAP's ABAP, etc.

Languages	Chinese, English, German
Hobbies	Mandarin debate, investing, electronic sports, table tennis, etc.

SOME LINKS

Research	work.chengweiwei.com
Blog	blog.chengweiwei.com
Photos	photo.chengweiwei.com
Twitter	twitter.com/chengweiwei
Video lectures	videos.chengweiwei.com