Pattern mining for label ranking

Pinho Rebelo de Sá, C.F.

Chapter 1

Introduction

Preferences are present in many tasks in our daily lives. Buying the right car, choosing a suitable house or even deciding on the food to eat, are trivial examples of decisions that reveal information, explicitly or implicitly, about our preferences. Hence, extracting and modeling preferences can provide us with invaluable information about the choices of a group of persons or individuals. However, this problem is non-trivial because, quite often, preferences depend on different context and options available [83]. Moreover, in areas like e-commerce, which typically deal with decisions from thousands of users, the acquisition of preferences can be a difficult task [57].

For that reason, artificial intelligent methods have been increasingly important for the discovery and automatic learning of preferences [47]. In particular, a subfield of machine learning which focuses on the study and modeling of preferences is *Preference Learning*.

In this thesis, we focus on one subtask of Preference Learning (introduced in Section 1.1), the prediction and analysis of preferences given a predefined set of objects/labels, commonly referred to as *Label Ranking* (Section 1.2).

1.1 Preference Learning

Preference Learning is an emerging subfield of machine learning that focuses on the study and modeling of preferences[1]. Preference learning methods

[1] A comprehensive overview of the state-of-the-art in the field of preference learning can be found in the *Preference Learning* book [57].

are conceptually different from standard machine learning problems such as classification or regression, as it can involve the prediction of more complex structures [7]. Classification and regression problems focus on the prediction of single values, while preference learning methods are designed to predict the order, or ranking, of a set of objects by relative importance.

In this field, the term *preference* is not strictly referring to preferences of individuals, but can also represent more general order relations. In turn, this flexibility gives an important advantage to the paradigm of preference-based learning, like extracting knowledge which, otherwise, would be harder [14]. However, without loss of generality, the discussion will focus on the more traditional type of preferences for easier interpretation.

Preferences can be extracted in an *explicit* way. As an illustrative example, a person who claims to prefer *apples* to *pears*, represented as:

$$apples \succ pears$$

is giving information about an explicit preference. In [81], 5000 Japanese people were asked to order 10 types of sushi by preference.

However, sometimes, information about preference is only *implicitly* given. Going back to the fruit example, if someone picks *bananas* from a basket containing *apples*, *pears* and *bananas*, one can implicitly infer that:

$$bananas \succ apples \wedge bananas \succ pears$$

One real example can be found in [114], where preferences are implicitly taken from clicking behavior of users.

Regardless of how preferences are extracted, they can be given as *relative* or *absolute*. *Relative* preferences cannot be quantifiable (e.g. sorting fruit by taste: *bananas* $\succ$ *apples* $\succ$ *pears*) [57]. On the other hand, *absolute* preferences are given in a quantitative form (e.g. the cost of the fruit: *bananas* = 2\$, *pears* = 1\$, *apples* = 3\$). Despite its different nature, in preference learning all types of preferences are combined in the same learning perspective [57].

In terms of modeling the preferences, there are two main approaches, learning *utility functions* and learning *preference relations* [57]. Learning utility functions, is learning to assign a relevance score to each object, which can later be ordered by comparison. Learning preference relations, is to learn the relative order relations between the objects being studied. This type of approach can be difficult to learn in cases where there are many objects

to order [42]. For example, consider the ordering of web pages by search engines [78]. In such cases, it is easier to rely on methodologies that learn utility functions.

In short, preference learning, is to learn from empirical data with implicit or explicit preferences. These preferences are explored by preference mining methods [57]. Preference learning is also about predicting preferences in new scenarios, when good generalizations from the given data are possible.

Preference learning can be divided into three main categories [57], *object ranking*, *instance ranking* and *label ranking*.

Object ranking The goal in the *object ranking* task is to output the ranking of a given set of objects, that, in theory, can be infinitely large. It can be considered a regression task whose target variables are orders [82]. A practical example are the lists of ordered web pages generated by search engines [78, 114]. In these case, utility functions are trained to assign a score to each newly given object [57].

Instance ranking In *instance ranking*, the setting is similar to ordinal classification [23], where an instance belongs to a class, among a finite set of classes with a natural order [57]. As an example, consider the assignment of conference papers to categories like: *reject*, *weak reject*, *weak accept* and *accept* [57].

Instance ranking is a generic term for bipartite [89] and multipartite [59] ranking.

In this thesis, we focus on the *label ranking* task (Section 1.2) and its applications.

1.2 Label Ranking

Label ranking is a sub-field of preference learning [57, 26, 123] which studies the problem of learning a mapping from instances to rankings over a finite number of predefined labels. It can be considered a variant of the conventional classification problem [26]. While in classification the goal is to assign examples to a specific class, in label ranking we are interested in assigning

a complete preference order of the labels to every example. If this is not possible, incomplete orders can also be assigned to some examples [28].

There are two approaches to tackle label ranking data [6, 24]. *Reduction techniques* (Section 1.2.3), also known as *decomposition methods*, divide the problem into several simpler problems (e.g. ranking by pairwise comparisons [56]). *Direct methods* (Section 1.2.4) treat the rankings without any transformation (e.g. decision trees adapted for the label ranking task [120, 26] or case-based approaches for label ranking [17, 24]).

Label ranking has been used in different applications, mainly for predictive tasks. For example, in meta-learning [16], to predict a ranking of a set of algorithms according to the best expected accuracy on a given dataset. In microarray analysis [74], to find patterns in genes from Yeast on different micro-array experiments. And also in image categorization [58], to predict the relative importance of categories of elements in landscape pictures (e.g. beach, sunset, field, fall foliage, mountain and urban).

1.2.1 Definition

Given an instance x from the instance space $\mathbb{X}$, the goal is to predict the ranking of the labels $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k\}$ associated with x [74]. The ranking can be represented as permutation or as an ordered vector.[2] The permutation, denoted as π, contains numbers from 1 to k, where 1 indicates the first position and k the last one (e.g. $\pi = (1, 2, 3, 4)$). The ordered vector represents the objects with an operator indicating the order of the preference (e.g. $\lambda_a \succ \lambda_b \succ \lambda_c \succ \lambda_d$).

The goal in label ranking is to learn the mapping $\mathbb{X} \to \Omega$, where Ω is defined as the permutation space. However, as in classification, we do not assume the existence of a deterministic $\mathbb{X} \to \Omega$ mapping. Instead, every instance is associated with a *probability distribution* over Ω [26]. This means that, for each $x \in \mathbb{X}$, there exists a probability distribution $\mathcal{P}(\cdot|x)$ such that, for every ranking $\pi \in \Omega$, $\mathcal{P}(\pi|x)$ is the probability that π is the ranking associated with x. The training data contains a set of instances $D = \{\langle x_i, \pi_i \rangle\}$, $i = 1, \ldots, n$, where x_i is a vector containing the values $x_i^j, j = 1, \ldots, m$ of m independent variables, $\mathcal{A}$, describing instance i and π_i is the corresponding target ranking.

Rankings can be either total or partial orders.

[2] Both notations will be used interchangeably in this dissertation.

Total orders A *strict total order* over $\mathcal{L}$ is defined as:[3]

$$\{\forall\,(\lambda_a, \lambda_b) \in \mathcal{L} | \lambda_a \succ \lambda_b \vee \lambda_b \succ \lambda_a\}$$

which represents a *strict ranking* [123], a *complete ranking* [57], or simply a *ranking*. A strict total order can also be represented as a permutation π of the set $\{1, \ldots, k\}$, such that $\pi(a)$ is the position, or *rank*, of λ_a in π. For example, the *strict total order* $\lambda_1 \succ \lambda_2 \succ \lambda_3 \succ \lambda_4$ can be represented as $\pi = (1, 2, 3, 4)$.

However, in real-world ranking data, we do not always have clear and unambiguous preferences, i.e. strict total orders [15]. Hence, sometimes we have to deal with *indifference* ($\sim$) and *incomparability* ($\perp$) [42]. For illustration purposes, let us consider the scenario of elections. If a voter feels that two candidates have identical proposals, then her preference can be expressed as indifferent, so they are assigned the same rank (i.e. a tie). To represent ties, we need a more relaxed setting, called *non-strict total orders*, or simply *total orders*, over $\mathcal{L}$, by replacing the binary strict order relation, $\succ$, with the binary partial order relation, $\succeq$:

$$\{\forall\,(\lambda_a, \lambda_b) \in \mathcal{L} | \lambda_a \succeq \lambda_b \vee \lambda_b \succeq \lambda_a\}$$

These non-strict total orders can represent *partial rankings* (rankings with ties) [123]. For example, the *non-strict total order* $\lambda_1 \succ \lambda_2 \sim \lambda_3 \succ \lambda_4$ can be represented as $\pi = (1, 2, 2, 3)$.

Additionally, real-world data may lack preferences data regarding two or more labels, which is known as *incomparability*. Continuing with the elections example, if the voter is familiar with the proposals of λ_a but not those of λ_b, she is unable to compare them, $\lambda_a \perp \lambda_b$. In other words, the voter cannot decide whether the candidates are equivalent or select one as her favorite. In this case, we can use *partial orders*.

Partial orders Similar to *total orders*, there are *strict* and *non-strict partial orders*. Let us consider the *non-strict partial orders* (which can also be referred to as *partial orders*) over $\mathcal{L}$:

$$\{\forall\,(\lambda_a, \lambda_b) \in \mathcal{L} | \lambda_a \succeq \lambda_b \vee \lambda_b \succeq \lambda_a \ \vee \lambda_a \perp \lambda_b\}$$

We can represent partial orders with *subrankings* [70]. For example, the *partial order* $\lambda_1 \succ \lambda_2 \succ \lambda_4$ can be represented as $\pi = (1, 2, 0, 4)$, where 0 represents that λ_3 is incomparable to the others, i.e. $\lambda_1, \lambda_2, \lambda_4 \perp \lambda_3$.

[3]For convenience, we say *total order* but in fact we mean a *totally ordered set*. Strictly speaking, a *total order* is a binary relation.

1.2.2 Evaluation

Given an instance x_i with label ranking π_i and a ranking $\hat{\pi}_i$ predicted by a label ranking model, several loss functions on Ω can be used to evaluate the accuracy of the prediction. One such function is the number of discordant label pairs:

$$\mathcal{D}(\pi, \hat{\pi}) = \#\{(a, b) \,|\pi(a) > \pi(b) \wedge \hat{\pi}(a) < \hat{\pi}(b)\}$$

If there are no discordant label pairs, the distance $\mathcal{D} = 0$. On the other hand, the function to define the number of concordant pairs is:

$$\mathcal{C}(\pi, \hat{\pi}) = \#\{(a, b) \,|\pi(a) > \pi(b) \wedge \hat{\pi}(a) > \hat{\pi}(b)\}$$

These concepts are used in the definition of several metrics that can be used for evaluation in label ranking:

Kendall Tau Kendall's τ coefficient [85] is the normalized difference between the number of concordant, $\mathcal{C}$, and discordant pairs, $\mathcal{D}$:

$$\tau(\pi, \hat{\pi}) = \frac{\mathcal{C} - \mathcal{D}}{\frac{1}{2}k(k-1)}$$

where $\frac{1}{2}k(k-1)$ is the number of possible pairwise combinations, $\binom{k}{2}$. The values of this coefficient range from $[-1, 1]$, where $\tau(\pi, \pi) = 1$ (i.e. when the rankings are equal) and $\tau(\pi, \pi^{-1}) = -1$ if π^{-1} denotes the inverse order of π (e.g. $\pi = (1, 2, 3, 4)$ and $\pi^{-1} = (4, 3, 2, 1)$). Kendall's τ can also be computed in the presence of ties, using τ_B [5].

Gamma coefficient If we want to measure the correlation between two partial orders (subrankings), or between total and partial orders, we can use the Gamma coefficient [93]:

$$\gamma(\pi, \hat{\pi}) = \frac{\mathcal{C} - \mathcal{D}}{\mathcal{C} + \mathcal{D}}$$

Note that the Gamma coefficient is identical to Kendall's τ coefficient in the presence of strict total orders, because, in this case, $\mathcal{C} + \mathcal{D} = \frac{1}{2}k(k-1)$.

Spearman distance One other commonly used measure is the Spearman's rank correlation coefficient [118]. It is defined as:

$$\rho\left(\pi,\hat{\pi}\right)=1-\frac{6d_S\left(\pi,\hat{\pi}\right)}{k\left(k^2-1\right)}$$

where d_S is the squared sum of rank differences, also referred as *Spearman distance* [82]:

$$d_S\left(\pi,\hat{\pi}\right)=\sum_{a=1}^{k}\left(\pi\left(a\right)-\hat{\pi}\left(a\right)\right)^2$$

In other words, the Spearman's rank correlation coefficient is the normalized version of the *Spearman distance* into the interval $[-1,1]$.

Weighted rank correlation measures Sometimes it is more important to predict the items in the top ranks than the ones ranked lower. For instance, when predicting the ranking of financial analysts to choose which ones to follow [6], it is more important to predict the best ones correctly than the worst ones. That is because it would not be very wise to follow the recommendations of the worst analysts. Thus, labels could be associated with cost and benefit values, which determine the real value of the ranking. For instance, to follow a given analyst, I have to buy the stocks he recommends. On the other hand, following different analysts will likely yield different gains or losses in the market. The empirical evaluation of ranking methods will only be useful in practice if these issues are taken into account.

In these cases, a weighted rank correlation coefficient can be used. They are typically adaptations of existing similarity measures, such as a weighted version of the Spearman's rank coefficient [110].

In terms of evaluation techniques, the usual resampling strategies, such as holdout or cross-validation, can be used to estimate the accuracy of a label ranking algorithm [26]. The accuracy of a label ranker can be estimated by averaging the values of any of the measures explained here, over the rankings predicted for a set of test examples.

To assess the significance of differences between models, using paired tests directly is not advised, since straightforward paired tests on multiple methods might reject the null hypothesis due to random chance [43]. For this reason, two-step statistical tests are usually performed [17, 26]. The first step, consists of a Friedman test, where the null hypothesis is that all learners have

equal performance. If this hypothesis is rejected, a two-tailed sign test to compare learners such as the Dunn's Multiple Comparison Procedure [104] is performed.

1.2.3 Reduction techniques

Because label ranking is a relatively new field in machine learning, some methods were basically approaching a reduction to a classification or regression problem [24], i.e. *Reduction techniques*. One great advantage of the reduction is that it makes a label ranking problem viable to be transformed into classification [74] or regression [41] problems. Also, reduction techniques can be quite efficiently implemented and easily applied for distributed systems [124]. On the other hand, there are also some disadvantages.

One option is to reduce the problem to the prediction of the *best label* (multi-label classification). This, however, will come with loss of information [23]. Assume we have the ranking of 3 algorithms in two scenarios: $Alg_1 \succ Alg_2 \succ Alg_3$ and $Alg_2 \succ Alg_1 \succ Alg_3$. A classifier, by focusing on the best one, will struggle to predict the most accurate, while a ranker will conclude that algorithms 1 and 2 perform better than 3.

One most commonly accepted reduction technique is to decompose rankings into binary preference relations, referred to as *pairwise comparisons* [74]. In simple words, it consists into reducing the problem of ranking into several classification problems. Examples of that are: Ranking by Pairwise Comparison (RPC) [74], Likelihood Pairwise Comparisons (LPC) [44] and Rule-based Label Ranking [64]. However, it has been noted that minimizing the classification error on several binary problems is not always equivalent to minimizing a loss function on rankings [23].

Ranking by Pairwise Comparisons

The method Ranking by Pairwise Comparisons (RPC) [74] is a well known reduction technique in the label ranking field. In simple terms, RPC can be divided in two phases, prediction of pairwise preferences and derivation of the rankings [74].

Before the first step, one needs to decomposed rankings into pairwise comparisons for each pair of labels of the form:

$$(\lambda_a, \lambda_b) \in \mathcal{L}, 1 \leq a < b \leq k$$

Considering that $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k\}$, there will be $\frac{k(k-1)}{2}$ different pairwise comparisons.

The first step is to learn a classification model from the training data for each pair of labels. This is, considering each pairwise comparison as a class, a separate model, $\mathcal{M}_{ab}$, is called to learn a mapping of the form:

$$x_i \rightarrow \left\{ \begin{array}{ll} 1 & \text{if } \lambda_a \succ \lambda_b \\ 0 & \text{if } \lambda_b \succ \lambda_a \end{array} \right\}, x_i \in D$$

This mapping can be done by any classifier at hand [74].

This approach has the advantage that it can be used with partial rankings. For any instance x_i, where nothing is known about the preference relation of a pair of labels $(\lambda_a, \lambda_b) \in \mathcal{L}$, the model $\mathcal{M}_{ab}$ ignores x_i in the training.

As a matter of choice, this can be easily adapted to deal with the interval $[0, 1]$. This will result in a *valued preference relation*, vpr_x, for every instance $x \in \mathbb{X}$:

$$vpr_x\left(\lambda_a, \lambda_b\right) \left\{ \begin{array}{ll} \mathcal{M}_{ab} & \text{if } a < b \\ 1 - \mathcal{M}_{ab} & \text{if } a > b \end{array} \right.$$

Finally, there is the aggregation step, where the predictions are combined to derive the rankings. Given the predicted pairwise comparisons for each x, the simplest approach is to order the labels, considering the predictions of the model $\mathcal{M}_{ab}$ as weights. Each label λ_a is ranked depending on the sum of the weights:

$$\sum_{\lambda_a \neq \lambda_b} vpr_x\left(\lambda_a, \lambda_b\right)$$

This task may not be trivial as there are possibilities of ties. In this regard, there are some well studied and documented approaches [55, 74]. However, one simple approach is to favor the most common classes according to the class distribution [74].

1.2.4 Direct approaches

Direct methods treat the rankings without any transformation. Hence, avoiding some of the problems of the reduction approaches [23], mentioned in Section 1.2.3. In this section, we outline some direct approaches for label ranking problems which have been proposed in recent years.

The most prominent approaches in the label ranking field are based on probabilistic distribution of rankings, like Mallow's Model [26] or Plackett-Luce [24]. These probabilistic methods estimate the conditional probability $\mathcal{P}(\pi|x)$ from the training data. This gives methods the advantage that, besides predicting a ranking, also provide a reliability score [24].

Case-based methods are also highly competitive direct approaches in label ranking (e.g. k-Nearest Neighbor [17, 26]). In [17] a nearest neighbor approach was proposed to deal with the problem of meta-learning. From a different perspective, in [24], the authors combined case-based with probabilistic models using the Instance-Based Label Ranking method.

A different group of label ranking methods tackle the ranking similarities with distance-based approaches (e.g., [120, 36, 116]). A relatively recent example is a neural networks adaptation proposed with Multilayer Perceptron for Label Ranking [116]. Also, in the naive Bayes for Label Ranking method [6], the prior probabilities of the rankings are similarity-based. In this cases, ranking correlation measures, like Kendall's τ coefficient [85] or the *Spearman distance* [82], are used to calculate the distance between rankings. These so-called distance-based models, make the prediction problem more similar to a regression task, where the difference between two rankings is similar to the error in a regression setting.

Tree-based models are popular in label ranking [120, 115, 26]. Decision trees are known to be competitive methods which are relatively easy to interpret [26]. In [120], Predictive Clustering Trees, successfully combine hierarchical clustering with decision trees for predicting rankings. Probabilistic models are combined in the tree generation to derive the nodes in Label Ranking Trees [26].

1.3 Contributions of this thesis

In this section, we give an overview of the contributions of this thesis, and its motivations. As mentioned in Section 1.2, there are two main approaches to the problem of label ranking [6, 24]. *Decomposition approaches* which divide the problem into several simpler problems and *Direct methods* that treat the rankings as target objects without any transformation. We focus more on direct methods but we also propose decomposition approaches.

The first part of this PhD project extends the work started with the MSc thesis [33] of the candidate. In the latter, Label Ranking Association Rules

(LRAR) were proposed [36]. LRARs are based on traditional Association Rules, redefining the support and confidence measures, in order to take into account the nature of label rankings. However, in the MSc project the empirical study was limited and little information about the behavior of LRARs was obtained. In the PhD project, this work was consolidated, namely to better understand how the rules perform in extreme conditions and in which cases are correctly applied (Section 1.3.1).

In this project we also addressed the lack of pre-processing methods that are specific to label ranking problems. LRARs, like Association Rules, cannot handle numeric data directly, which needs to be discretized beforehand. We proposed two discretization approaches that are specific for label ranking problems (Section 1.3.2). Both approaches are based on a new measure of *ranking entropy* which was developed as part of this work.

The new measure of *ranking entropy* was also the basis for a third contribution. We proposed Entropy Ranking Trees (Section 1.3.3), which is an adaptation to the problem of label ranking of a Top-Down Induction of Decision Trees algorithm. Based on this new algorithm, we made a fourth contribution, which is an ensemble method for label ranking. The algorithm is Label Ranking Forests (Section 1.3.3), which, as the name indicates, is an adaptation of Random Forests for label ranking.

There is not much work on descriptive pattern mining of label rankings and preference data. We address this shortcoming with two additional contributions, *Pairwise Association Rules* and *Exceptional Preferences Mining* (Section 1.3.4), which are two rule-based methods.

Most empirical studies on label rankings are based on a set of benchmark datasets, in the KEBI Data Repository [26]. These were generated from other datasets which were not original label ranking problems. Given the process of transformation used, it is unclear whether these datasets are useful to assess the quality of label ranking methods. Thus, the final contribution of this thesis are two swap randomization techniques for the label ranking task (Section 1.3.5). The proposed methods were used to investigate the usefulness of the available label ranking datasets.

1.3.1 Label Ranking Association Rules

Association Rules mining is used to discover interesting relationships between attributes in large databases [2]. An association rule has the form $A \rightarrow B$,

meaning that when the set of values A is observed in the data, there is a chance of observing B.

Although association rules were originally developed for descriptive tasks, their success has quickly lead to their adaptation for prediction problems. The motivation for adapting Association Rules (AR) for classification is that, a classification rule model built from such an unrestrained set of rules, can potentially be more accurate than the ones using a greedy search approach [97].

Label Ranking Association Rules [33] were proposed as a predictive approach for label ranking [36]. The main adaptations to the original algorithm were on the *support* and *confidence* measures, which were modified to take into account the similarity between rankings.

The method proposed originally to mine LRAR has a parameter. Such parameter, works as a threshold that determines what should and should not be considered a sufficiently similar pair of rankings, in order to be covered by the same rule. However, the impact of that parameter in the results was not investigated originally. In Chapter 2, we consolidate the original work by discussing results of the analysis on the values of this parameter. The type of questions we investigate is, whether there is a rule of thumb to select its value or it is data-specific.

1.3.2 Discretization

As in any machine learning task, data preparation is essential for the development of accurate label ranking models. For instance, some algorithms are unable to deal with numeric variables, such as the basic versions of Naive Bayes and Association Rules [102, 4], in which case numeric variables should be discretized beforehand.

While there has been a significant development of learning algorithms for label ranking in recent years, there are not many pre-processing methods specifically for this task. Following the adaptation of Association Rules for Label Ranking, the development of a suitable discretization method was paramount. Without such a method, it would not be possible to adequately analyze data with numerical variables.

Discretization, from a general point of view, is the process of partitioning a given interval into a set of discrete sub-intervals. It is normally used to split continuous intervals into two or more sub-intervals which can then be

treated as nominal values. When we transform continuous intervals into discrete sub-intervals, regardless of the splits taken, generally leads to a loss of information [60]. In theory, a good discretization should have a good balance between the loss of information and the number of partitions [90].

Discretization methods are typically organized into two groups, *supervised* and *unsupervised*, depending on whether or not they involve the target variable, respectively. In prediction problems, supervised methods usually produce more useful discretizations than unsupervised methods [46].

The difference in nature between the target variable in classification and label ranking problems implies that supervised discretization methods developed for classification are not suitable for LR. For this reason, two methods, based on a well-known supervised discretization approach for classification, were proposed as part of this PhD research. The original method, *Minimum Description Length Partition* (MDLP) [54], uses a measure of entropy from information theory, known as *Shannon entropy* [54].

The first proposed approach, *Minimum Description Length Partition for Ranking* (MDLP-R) [40] (Chapter 3), uses a ranking entropy measure based on the similarities between rankings. This ranking entropy is the equivalent of the Shannon entropy for label ranking problems. A simpler and improved measure of entropy was latter proposed and implemented in a new method, EDiRa (Entropy-based Discretization for Ranking) [39] (Chapter 3).

1.3.3 Tree-based models

Tree-based models are popular for a number of reasons, including how they can clearly express information about the problem, because their structure is relatively easy to interpret even for people without a background in learning algorithms. They have been used in classification [111], regression [20] and also label ranking [120, 26] tasks.

On the other hand, ensemble methods, which use multiple learning algorithms, usually compensate some loss in interpretability with significant accuracy improvements [19]. One of the most popular approaches are ensembles of trees, such as Random Forests [19].

Our contributions concerning the development of tree-based models for label ranking are a new variant of decision trees and the adaptation of the random forests algorithm for this task.

Entropy Ranking Trees Decision trees, like ID3 [111], grow in a top-down recursive partitioning scheme that iteratively splits data into smaller subsets [102]. This splits are performed such that each node divides the data into increasingly more homogeneous subsets, in terms of the target variable. The search for the best split point tries to optimize a given splitting criterion, such as the *information gain* [102]. Information gain measures the difference in entropy between the previous and current state relatively to a target variable.

By implementing the previously proposed *ranking entropy* measure (Section 3) in the splitting process, we proposed a novel ranking tree approach, Entropy Ranking Trees [35] (Chapter 4). The goal is to obtain leaf nodes that contain examples with target rankings as homogeneous as possible.

Label Ranking Forests Adapting Random Forests to label ranking comes in a natural way based on any decision trees approach for label ranking. Motivated by the success of Random Forests in terms of improved accuracy for classification and regression problems [13], we proposed a Random Forest approach for label ranking, Label Ranking Forests [32] (Chapter 4).

1.3.4 Descriptive mining for label ranking

Preference learning approaches can benefit from the analysis of descriptive methods [57]. In label ranking, only recently, a few descriptive approaches for mining label ranking data have been proposed [70, 122]. In [70], the authors suggest an approach using association rules that search for patterns exclusively in rankings (i.e. the independent variables are ignored). In [122], a *ranked tiling* approach to search for patterns in the ranking scores, i.e. ranks, is suggested.

The available label ranking mining approaches focus exclusively on the target ranking, and do not relate its values to the values of the independent variables. However, we believe that much valuable information can be extracted by taking both into account. For example, consider we discover that in 80% of the cases sushi A is preferred to sushi B. By taking independent variables into account, we might actually find that females prefer sushi B to sushi A, but males, which represent 80% of the population, prefer sushi A to sushi B. For that reason, we propose two approaches for mining label ranking data.

Exceptional Preferences Mining In Chapter 5, we propose an approach for finding deviating patterns in label rankings, in the context of Subgroup Discovery [88], referred to as Exceptional Preferences Mining. The aim of Subgroup Discovery is to discover subgroups for which the target shows an unusual distribution, as compared to the overall population in the data [88].

In the context of label ranking, we need to determine to what extent the subgroups show different preferences, and whether any of these preferences are in conflict with the average behavior. To that end, we developed three quality measures, *Pairwise*, *Labelwise* and *Norm*. Each of them strives to find subgroups where the preference relations are exceptional from slightly different perspectives.

The *Pairwise* measure identifies subgroups with strong deviating preferences between pairs of labels. The *Labelwise* measure identifies subgroups where at least one particular label is exceptionally under- or over-appreciated. Finally, the *Norm* quality measure will give more relevance to subgroups where several, or all, labels deviate strongly.

Pairwise Association Rules Association rules use a set of descriptors to represent meaningful subsets of the data [69], hence providing an easy interpretation of the patterns mined. We propose an approach that decomposes rankings into pairwise comparisons and then looks for meaningful associations rules of the form:

$$A \rightarrow \{\lambda_a \succeq \lambda_b \vee \lambda_a \perp \lambda_b \ \vee \lambda_a = \lambda_b | \lambda_a, \lambda_b \in \mathcal{L}\}$$

which we refer as Pairwise Association Rules (Chapter 2). [4]

1.3.5 Label Ranking Data

Due to the lack of benchmark LR datasets, 16 semi-synthetic datasets were adapted from multi-class and regression datasets from the UCI repository and Statlog project [26]. For each multi-class problem, an LR dataset (referred to as *type A* problem) was created by training a Naive Bayes and the target was replaced with a ranking based on the probability score of each

[4] For similar reasons, Label Ranking Association Rules can also be used for mining label ranking data. However, the fact that these search exclusively for complete ranking patterns, can be seen as a limitation.

class. Additionally, for each regression problem, the ranking target was created based on the values of a set of selected numerical attributes (*type B* problems).

This set of 16 datasets has been used by the majority and the contributions in the Label Ranking field [28, 27, 116, 64]. However, it is unclear if the type B datasets contain any meaningful relations between the target rankings and independent variables. Additionally, the rankings in type A problems represent the preferences of an agent, which in this case is the naive Bayes classifier. Therefore, the bias in these algorithms seems too strongly defined and, thus, their ability to represent real world distributions of data is questionable.

In many data mining applications, *swap randomizations* techniques are used together with statistical tests to validate the significance of findings [62]. Using a similar concept, we can investigate the usefulness of type B datasets. For this purpose, we propose two *swap randomization* methods specific for the label ranking datasets, *ranking permutations* and *labelwise permutations*.

Ranking permutations Randomly permuting the rankings is a natural adaptation of the methods used in classification [63]. By doing so, we want to test the *strength* of the relation between independent variables and targets in the data. After the permutation, because we break this relation, we can measure how the label ranking learners behave and compare with the results on the original data. If the differences are not significant, we can conclude that there is no real relation between independent variables and targets.

Labelwise permutations In [19], each attribute was permuted at a time to measure the impact of variables for prediction, in terms of misclassification rate. We propose a similar method by applying the same concept to each individual label (Chapter 6). We define *labelwise permutation* as the process of permuting the ranks of a specific label. This enables us to test if the amount of information in the independent variables about the rank of the selected label is significant. By comparison with the original data (without permutations), statistical significance tests can be used to assess the relevance of each label.

The number of benchmark datasets for label ranking is still relatively small. A final contribution of this project is the adaptation from a multivariate regression problem into a label ranking dataset (Chapter 5). We adapted

the dataset from the COIL 1999 Competition Data, taken from the UCI Repository [96], concerning the frequencies of algae populations in different environments, which we refer to as Algae.

1.4 Thesis outline

This thesis is presented as a series of papers in the form of self-contained chapters. These are either papers that have been published or that have been submitted for publication. The dissertation consists of 6 chapters following this introductory chapter.

Chapter 2, Preference Rules [37], presents an empirical study on Label Ranking Association Rules and Pairwise Association Rules. This paper, which has been submitted to the *Information Fusion* journal, is an extension of previous work, Mining Association Rules for Label Ranking [36].

Chapter 3, Entropy-based discretization methods for ranking data [39], presents a supervised approach to discretize datasets with target rankings. This chapter, which is published in the *Information Sciences* journal, is based on preliminary work published in the proceedings of the Discovery Science 2013 conference, Singapore [40].

In Chapter 4, Label Ranking Forests [32], we can find a successful adaption of ensembles of trees for label ranking problems, which has been published in the *Expert Systems* journal. This work is an extension to the preliminary work published in EPIA 2015, in which Entropy Ranking Trees, were proposed [35].

Chapter 5, Exceptional Preferences Mining [34], proposes an approach to look for exceptional behavior in label ranking datasets. This paper is published in the proceedings of the Discovery Science 2016 conference held in Bari, Italy.

Chapter 6, Permutation Tests for Label Ranking [38], presents a smaller contribution where, semi-synthetic datasets used in Label Ranking community, where evaluated with different tests. This chapter is published in the local proceedings of the *BENELUX conference on artificial intelligence 2015.*

Finally, Chapter 7, gives an overview of the main contributions and findings in this PhD dissertation.