# List of Tables



# List of Figures

# Contents

# Naive Bayes for label ranking

Artur Aiguzhinov (artur.aiguzhinov@inescporto.pt)[1,2] and Carlos Soares (csoares@fe.up.pt)[2,3]

[1]FEP & CEF.UP, University of Porto

[2]INESC TEC

[3]FEUP, University of Porto

November 26, 2015

**Abstract**

The problem of learning label rankings is receiving increasing attention from several research communities. A number of common learning algorithms have been adapted for this task, including k-Nearest Neighbours (k-NN) and decision trees. Following this line, we propose an adaptation of the naive Bayes classification algorithm for the label ranking problem. Our main idea lies in the use of similarity between the rankings to replace the concept of probability. We empirically test the proposed method on some metalearning problems that consist of relating characteristics of learning problems to the relative performance of learning algorithms. Our method generally performs better than the baseline indicating that it is able to identify some of the underlying patterns in the data.

# 1 Introduction

Label ranking is an increasingly popular topic in the machine learning literature. It studies the problem of learning a mapping from instances to rankings over a finite number of pre-defined labels. In some sense, it is a variation of the conventional classification problem; however, in contrast to the classification settings, where the objective is to assign examples to a specific class, in label ranking we are interested in assigning a complete preference order of labels to every example (Cheng, Hühn, and Hüllermeier, 2009).

Many different algorithms have been adapted to deal with label ranking such as: decision-trees for label ranking (Cheng et al., 2009), algorithm based on Plackett-Luce model (Cheng, Dembczynski, and Hüllermeier, 2010), pairwise comparison (Hüllermeier, Fürnkranz, Cheng, and Brinker, 2008), and k-NN for label ranking (Brazdil, Soares, and Costa, 2003). Table 5 outlines the recent developments in solving a label ranking problem.

In this paper, we introduce the ranking similarity approach. We propose an adaptation of the naive Bayes (NB) algorithm for label ranking. Despite its limitations, NB is an algorithm with successful results in many applications (Domingos and Pazzani, 1997). Additionally, the Bayesian framework is well understood in many domains. For instance, we apply this method on the problem of predicting the rankings of financial analysts since in the Financial Economics the Bayesian models are widely used (e.g., the Black-Litterman model for active portfolio management (Black and Litterman, 1992)).

The main idea lies in replacing the probabilities in the Bayes theorem with the distance between rankings. This can be done because it has been shown that

there is a parallel between the concepts of distance and likelihood (Vogt, Godden, and Bajorath, 2007). We develop two versions of the algorithm: for discrete and continuous cases.

The paper is organized as follows: Section 2 provides the formalization of the label ranking problem; Section 3 briefly describes the naive Bayes algorithm for classification; Section 4 shows the adaptation of the NB algorithm for label ranking (NB4LR); Section 5 provides some extensions of NB4LR; namely, the scenario of features having a continuous values (5.1) and a case when rankings are part of time series (5.2); Section 6 outlines the datasets used for the experiments; Section 7 presents empirical results; finally, Section 8 concludes with the goals for future work.

## 2 Learning label rankings

The formalization of a label ranking problem is the following (Vembu and Gärtner, 2010). Let $\mathcal{X} \subseteq \{\mathcal{V}_1, \ldots, \mathcal{V}_m\}$ be an instance space of nominal variables, such that $\mathcal{V}_a = \{v_{a,1}, \ldots, v_{a,n_a}\}$ is the domain of nominal variable $a$. Also, let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k\}$ be a set of labels, and $\mathcal{Y} = y_{\mathcal{L}}$ be the output space of all possible total orders[1] over $\mathcal{L}$ defined on the permutation space $y$. The goal of a label ranking algorithm is to learn a mapping $h : \mathcal{X} \to \mathcal{Y}$, where $h$ is chosen from a given hypothesis space $\mathcal{H}$, such that a predefined loss function $\ell : \mathcal{H} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is minimized. The algorithm learns $h$ from a training set $\mathcal{T} = \{x_i, y_i\}_{i \in \{1,\ldots,n\}} \subseteq \mathcal{X} \times \mathcal{Y}$ of $n$ examples, where $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\} \in \mathcal{X}$ and $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,k}\} \in y_{\mathcal{L}}$.

[1] A total order is a complete, transitive, and asymmetric relation $\succ$ on $\mathcal{L}$, where $\lambda_i \succ \lambda_j$ indicates that $\lambda_i$ precedes $\lambda_j$. In this paper, given $\mathcal{L} = \{A, B, C\}$, we will use the notation $\{A, C, B\}$ and $\{1, 3, 2\}$ interchangeably to represent the order $A \succ C \succ B$.

Furthermore, we define $y_i^{-1} = \{y_{i,1}^{-1}, y_{i,2}^{-1}, \ldots, y_{i,k}^{-1}\}$ as the order of the labels in example $i$. Given that we are focusing on total orders, $y_i^{-1}$ is a permutation of the set $\{1, 2, \ldots, k\}$ where $y_{i,j}^{-1}$ is the rank of label $\lambda_j$ in example $i$.

Unlike classification, where for each instance $x \in \mathcal{X}$ there is an associated class $y_i \in \mathcal{L}$[2], in label ranking problems there is a ranking of the labels associated with every instance $x$ and the goal is to predict it. This is also different from other ranking problems, such as in information retrieval or recommender systems. In these problems the target variable is a set of ratings or binary relevance labels for each item, and not a ranking.

The algorithms for label ranking can be divided into two main approaches: methods that transform the ranking problem into multiple binary problems and methods that were developed or adapted to predict the rankings. An example of the former is the ranking by pairwise comparisons (Hüllermeier et al., 2008). Some examples of algorithms that are specific for rankings are: the predictive clustering trees method (Todorovski, Blockeel, and Dzeroski, 2002), the similarity-based k-Nearest Neighbor for label ranking (Brazdil et al., 2003), the probabilistic k-Nearest Neighbor for label ranking (Cheng et al., 2009) and the linear utility transformation method (Har-Peled, Roth, and Zimak, 2002; Dekel, Manning, and Singer, 2004).

To assess the accuracy of the predicted rankings relative to the corresponding target rankings, a suitable loss function is needed. In this paper we compare two rankings using the Spearman correlation coefficient (Brazdil et al., 2003; Vembu

[2]Here, we use both $y_i$ to represent the target class (label) in classification and the target ranking in label ranking to clarify that they are both the target of the learning problem. We will explicitly state the task we are dealing with when it is not clear from the context.

and Gärtner, 2010):

$$\rho(y, \hat{y}) = 1 - \frac{6 \sum_{j=1}^{k} (y_j - \hat{y}_j)^2}{k^3 - k} \tag{1}$$

where $y$ and $\hat{y}$[3] are, respectively, the target and predicted rankings for a given instance. Two orders with all the labels placed in the same position will have a Spearman correlation of $+1$. Labels placed in reverse order will produce correlation of $-1$. Thus, the higher the value of $\rho$ the more accurate the prediction is compared to target. The loss function is given by the mean Spearman correlation values (Equation (1)) between the predicted and target rankings, across all examples in the dataset:

$$\ell = \frac{\sum_{i=1}^{n} \rho(y_i, \hat{y}_i)}{n} \tag{2}$$

An extensive survey of label ranking algorithms is given by Vembu and Gärtner (2010).

# 3 The Naive Bayes Classifier

We follow Mitchell (1997) to formalize the naive Bayes classifier. In classification, each instance $x_i \in \mathcal{X}$ is binded to class $y_i \in \mathcal{L}$. The task of a learner is to create a classifier from the training set $\mathcal{T}$.The classifier takes a new, unlabeled instance and assigns it to a class (label).

The naive Bayes method classifies a new instance $x_i$ by determining the most

[3]In the following, we will use $y_i$ and $y_i$ interchangeably to represent the target ranking.

probable target value, $c_{MAP}(x_i)$[4], given the attribute values that describe the instance:

$$c_{MAP(x_i)} = \arg\max_{\lambda \in \mathcal{L}} P(\lambda | x_{i,1}, x_{i,2}, \ldots, x_{i,m}) \tag{3}$$

where $x_{i,j}$ is the value of attribute $j$ for instance $i$.

The algorithm is based on the Bayes theorem that establishes the probability of $A$ given $B$ as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4}$$

Thus, the Bayes theorem provides a way to calculate the posterior probability of a hypothesis.

Using Equation (4), we can rewrite Equation (3) as

$$\begin{aligned} c_{MAP(x_i)} &= \arg\max_{\lambda \in \mathcal{L}} \frac{P(x_{i,1}, x_{i,2}, \ldots, x_{i,m}|\lambda)P(\lambda)}{P(x_{i,1}, x_{i,2}, \ldots, x_{i,m})} \\ &= \arg\max_{\lambda \in \mathcal{L}} P(x_{i,1}, x_{i,2} \ldots x_{i,m}|\lambda)P(\lambda) \end{aligned} \tag{5}$$

Computing the likelihood $P(x_{i,1}, x_{i,2}, \ldots, x_{i,m}|\lambda)$ is very complex and requires large amounts of data, in order to produce reliable estimates. Therefore, the naive Bayes classifier makes one simple, hence, naive, assumption that the attribute values are conditionally independent from each other. This implies that the probability of observing the conjunction $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$ is the product of the probabilities for the individual attributes: $P(x_{i,1}, x_{i,2}, \ldots, x_{i,m}|\lambda) = \prod_{j=1}^{m} P(x_{i,j}|\lambda)$. Substituting this expression into Equation (5), we obtain the naive Bayes classi-

[4] $MAP$ – Maximum A Posteriori

fier:

$$c_{nb}(x_i) = \arg\max_{\lambda \in \mathcal{L}} P(\lambda) \prod_{j=1}^{m} P(x_{i,j}|\lambda) \tag{6}$$

# 4 Adapting NB to Ranking

Consider the classic problem of the "play/no play" tennis based on weather conditions. The naive Bayes classification algorithm can be successfully applied to this problem (Mitchell, 1997, chap. 6). For illustration purposes, we extend this example application to the label ranking setting by replacing the target with a ranking on the preferences of a golf player regarding three golf courts on different days (Table 1). The last three columns represent the ranks of the golf courts A, B and C.

As described earlier, the difference between classification and label ranking lies in the target variable, $y$. Therefore, to adapt NB for ranking we have to adapt the parts of the algorithm that depend on the target variable, namely:

- prior probability, $P(y)$
- conditional probability, $P(x|y)$

The adaptation should take into account the differences in nature between label rankings and classes. For example, if we consider label ranking as a classification problem, then the prior probability of ranking $\{A, B, C\}$ on the data given in Table 1 is $P(\{A, B, C\})$ = 3/6 = 0.5, which is quite high. On the other hand, the probability of $\{A, C, B\}$ is quite low, $P(\{A, C, B\})$ = 1/6 = 0.17. However, taking into account the stochastic nature of these rankings (Cheng et al., 2009), it is

intuitively clear that the observation of $\{A, B, C\}$ increases the probability of observing $\{A, C, B\}$ and vice-versa. This affects even rankings that are not observed in the available data. For example, the case of unobserved ranking $\{B, A, C\}$ in Table 1 would not be entirely unexpected in the future considering a similar observed ranking $\{B, C, A\}$.

One approach to deal with stochastic nature characteristic of label rankings is to use ranking distributions, such as the Mallows model (e.g., (Lebanon and Lafferty, 2002; Cheng et al., 2009)). Alternatively, we may consider that the intuition described above is represented by varying similarity between rankings.

Similarity-based label ranking algorithms have two important properties:

- they assign non-zero probabilities even for rankings which have not been observed. This property is common to distribution-based methods;
- they are based on the notion of similarity between rankings, which also underlies the evaluation measures that are commonly used. Better performance is naturally expected by aligning the algorithm with the evaluation measure.

Similarity and probability are different concepts and, in order to adapt NB for label ranking based on the concept of similarity, it is necessary to relate them. A parallel has been established between probabilities and the general Euclidean distance measure (Vogt et al., 2007). This work shows that maximizing the likelihood is equivalent to minimizing the distance (i.e., maximizing the similarity) in a Euclidean space. Although not all assumptions required for that parallel hold when considering distance (or similarity) between rankings, given that the naive Bayes algorithm is known to be robust to violations of its assumptions, we propose a

similarity-based adaptation of NB for label ranking.

In the following description, we will retain the probabilistic terminology (e.g., prior probability) from the original algorithm, even though it does not apply for similarity functions. However, in the mathematical notation, we will use the subscript $_{LR}$ to distinguish the concepts. Despite the abuse, we believe this makes the algorithm easier to understand.

We start by defining $\mathcal{S}$ as a similarity matrix between the target rankings in a training set, i.e. $\mathcal{S}_{n\times n} = \rho(y_i, y_j)$. The prior probability of a label ranking is given by:

$$P_{LR}(y) = \frac{\sum_{i=1}^{n} \rho(y, y_i)}{n} \tag{7}$$

We say that the prior probability is the mean of similarity of a given rankings to all the others. We measure similarity using the Spearman correlation coefficient (Equation (1)). Equation (7) shows the average similarity of one ranking relative to others. The greater the similarity between two particular rankings, the higher is the probability that the next unobserved ranking will be similar to the known ranking. Take a look at panel A of Table 2 with the calculated prior probability for the unique rankings. We also added a column with prior probabilities considering the rankings as one class ($P(y)$). As stated above, the ranking $\{A, C, B\}$, due to its similarity to the other two rankings, achieves a higher probability (0.708)[1].

The similarity of rankings based on the value $i$ of attribute $a$, ($v_{a,i}$), or conditional probability of label rankings, is:

$$P_{LR}(v_{a,i}|y) = \frac{\sum_{i:x_{i,a}=v_{a,i}} \rho(y, y_i)}{|\{i : x_{i,a} = v_{a,i}\}|} \tag{8}$$

[1]Since we measure $P_{LR}$ as a similarity between rankings, it would not sum to one as the in case of probability for classification.

Panel B of Table 2 demonstrates the logic behind the conditional probabilities based on similarity. Notice that there are no examples with $Outlook = Sunny$ and a target ranking of $\{A, C, B\}$; thus, $P(Outlook = Sunny|\{A, C, B\}) = 0.000$. However, in the similarity approach, the probability of $\{A, C, B\}$ depends on the probability of similar rankings, yielding $P_{LR}(Outlook = Sunny|\{A, C, B\}) = 0.412$.

Applying Equation (6), we get the estimated posterior probability of ranking $y$:

$$P_{LR}(y|x_i) = P_{LR}(y) \prod_{a=1}^{m} P_{LR}(x_{i,a}|y) = \tag{9}$$
$$= \frac{\sum_{j=1}^{n} \rho(y, y_j)}{n} \left[ \prod_{a=1}^{m} \frac{\sum_{j:x_{j,a}=x_{i,a}} \rho(y, y_j)}{|\{j : x_{j,a} = x_{i,a}\}|} \right]$$

The similarity-based adaptation of naive Bayes for label ranking will output the ranking with the higher $P_{LR}(y|x_i)$ value:

$$\hat{y} = \arg\max_{y \in y_{\mathcal{L}}} P_{LR}(y|x_i) = \tag{10}$$
$$= \arg\max_{y \in y_{\mathcal{L}}} P_{LR}(y) \prod_{a=1}^{m} P_{LR}(x_{i,a}|y)$$

# 5 Naive Bayes for label ranking: special cases

## 5.1 Continuous case

The naive Bayes algorithm for label ranking mentioned above requires nominal variables in order to calculate the probabilities. In this section we extend the

adaptation for the continuous case.

We propose to modify conditional label ranking probability by utilizing Gaussian distribution of the independent variables; thus, applying traditional normal distribution approach. The naive Bayes for classification with continuous variables was implemented in Bouckaert (2005). We apply the same logic for conditional probability of label rankings and Equation (8) for the discrete case transforms to the continuous one as:

$$P_{LR}(x_i|y) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{\frac{(x_i-\mu_y)^2}{2\sigma_y^2}} \tag{11}$$

where $\mu_y$ and $\sigma_y^2$ weighted mean and weighted variance for LR, defined as follows:

$$\mu_y = \frac{\sum_{i=1}^{n} \rho(y, y_i) x_i}{\sum_{i=1}^{n} \rho(y, y_i)}; \qquad \sigma_y^2 = \frac{\sum_{i=1}^{n} \rho(y, y_i)(x_i - \mu_y)^2}{\sum_{i=1}^{n} \rho(y, y_i)} \tag{12}$$

## 5.2 Time series of rankings

The time dependent label ranking (TDLR) problem takes the intertemporal dependence between the rankings into account. That is, rankings that are similar to the most recent ones are more likely to appear. To capture this, we propose the weighted TDLR prior probability:

$$P_{TDLR}(y_t) = \frac{\sum_{t=1}^{n} w_t \rho(y, y_t)}{\sum_{t=1}^{n} w_t} \tag{13}$$

where $w_t = \{w_1, \ldots, w_n\} \rightarrow \mathbf{w}$ is the vector of weights calculated from the exponential function $\mathbf{w} = b^{\frac{1-\{n\}_1^t}{n}}$. Parameter $b \in \{1 \ldots \infty\}$ sets the degree of the

"memory" for the past rankings, i.e., the larger $b$, the more weight is given to the most recent rankings.

As for the conditional label ranking probability, the equation for the weighted mean (Equation (11)) becomes:

$$\mu(x_{t,m}|y_t) = \frac{\sum_{t=1}^{n} w_t \rho(y, y_t) x_{t,m}}{\sum_{t=1}^{n} \rho(y, y_t)} \tag{14}$$

and variance:

$$\sigma_w^2(x_{t,m}|y) = \frac{\sum_{i=1}^{n} w_t \rho(y, y_t)[x_{t,m} - \mu(x_{t,m}|y)]^2}{\sum_{i=1}^{n} \rho(y, y_t)} \tag{15}$$

# 6 Data

Given the novelty of the label ranking problem, it is hard to find the available data for LR experiments. We follow the convention to use the KEBI data set[2]. The description of individual set presented in Table 3. Notice that there are two types of data. The explanation of each of them is given in Cheng et al. (2009):

> [Type (A) data sets:] 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) ... [Type (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

[2] https://www.uni-marburg.de/fb12/kebi/research/repository/

ordered by size.

# 7 Experiment Results

We empirically test the proposed adaptation of the naive Bayes algorithm for learning label rankings.

Given that the attributes in the datasets are numerical and the NB algorithm is for symbolic attributes, they must be discretized. We used a simple equal-width binning method using 10 bins. We also perform the experiments on continuous data applying the modified naive Bayes for continuous case outlined in 5.1. In addition, we compare the results with the state-of-the-art LR algorithm developed in Brazdil et al. (2003); namely, k-NN (k=3) for label ranking.

The baseline is a simple method based on the mean rank of each label over all training examples (Brazdil, Soares, Giraud-Carrier, and Vilalta, 2009).

$$\hat{y}_j^{-1} = \frac{\sum_{i=1}^{n} y_{i,j}^{-1}}{n} \tag{16}$$

where $y_{i,j}^{-1}$ is the rank of label $\lambda_j$ on dataset $i$. The final ranking is obtained by ordering the mean ranks and assigning them to the labels accordingly. This ranking is usually called the *default ranking*, in parallel to the default class in classification.

The performance of the label ranking methods was estimated using a methodology that has been used previously for this purpose (Brazdil et al., 2003). It is based on 10-fold cross validation. The accuracy of the rankings predicted by methods was evaluated by comparing them to the target rankings (i.e., the rank-

ings based on the observed performance of the algorithms) using the Spearman's correlation coefficient (Equation (1)). The code for all the examples in this paper has been written in R (R Development Core Team, 2008).

The results of the experiments are presented in Table 4. We report that the naive Bayes for label ranking for continuous case exhibits the maximum number of datasets for which it out-performed the baseline and is competitive with the state-of-the-art. The bold numbers in the table represent the value of the average Spearman correlation across 10-folds that are higher than that of the baseline.

For the continuous case, the naive Bayes for label ranking algorithm outperformed the baselines in 13 datasets out of 17. For the discretized case the number of outperformed datasets is 12. The state-of-the-art algorithm outperformed the baseline in continuous and discrete scenarios in 14 and 12 respectively. Observe, that given the different nature of datasets (types A or B) and different variable classes (continuous vs. nominal) both label ranking algorithms exhibit relatively great performance in predicting the rankings.

# 8 Conclusion

In this paper we presented an adaptation of the naive Bayes algorithm for label ranking that is based on similarities of the rankings taking advantage of a parallel that can be established between the concepts of likelihood and distance. We tested the new algorithm on label ranking datasets and conclude that it consistently outperforms a baseline method and is competitive with the state-of-the-art.

A number of issues remain open, which we plan to address in the future. Firstly, we are currently working on creating new datasets for ranking applications

in different areas, including finance (e.g., predicting the rankings of the financial analysts based on their recommendations). These new datasets will enable us to better understand the behavior of the proposed algorithm. In addition, we assume that target rankings are total orders. In practice, this is often not true (Cheng et al., 2010; Brazdil et al., 2003). We plan to address the problem of partial orders in the future. Finally, we plan to compare the new method with existing ones.

# References


Aiguzhinov, A., C. Soares, and A. Serra (2010). A similarity-based adaptation of naive Bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In B. Pfahringer, G. Holmes, and A. Hoffmann (Eds.), *Discovery Science*, Volume 6332 of *Lecture Notes in Computer Science*, pp. 16–26. Springer Berlin Heidelberg.

Black, F. and R. Litterman (1992). Global portfolio optimization. *Financial Analysts Journal 48*(5), 28–43.

Bouckaert, R. R. (2005). Naive Bayes classifiers that perform well with continuous variables. In *AI 2004: Advances in Artificial Intelligence*, Volume 3339 of *Lecture Notes in Computer Science*, pp. 85–116.

Brazdil, P., C. Soares, and J. Costa (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning 50*(3), 251–277.

Brazdil, P., C. Soares, C. Giraud-Carrier, and R. Vilalta (2009). *Metalearning Applications to Data Mining*. Springer.

Cheng, W., K. Dembczynski, and E. Hüllermeier (2010). Label ranking methods based on the Plackett-Luce model. In J. Fürnkranz and T. Joachims (Eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Omnipress.

Cheng, W., J. Hühn, and E. Hüllermeier (2009). Decision tree and instance-based learning for label ranking. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, pp. 161–168. ACM.

Cheng, W. and E. Hüllermeier (2009). A new instance-based label ranking approach using the Mallows model. In W. Yu, H. He, and N. Zhang (Eds.), *Advances in Neural Networks – ISNN 2009*, Volume 5551 of *Lecture Notes in Computer Science*, pp. 707–716. Springer Berlin Heidelberg.

de Sá, C., C. Soares, A. Jorge, P. Azevedo, and J. Costa (2011). Mining association rules for label ranking. In J. Huang, L. Cao, and J. Srivastava (Eds.), *Advances in Knowledge Discovery and Data Mining*, Volume 6635 of *Lecture Notes in Computer Science*, pp. 432–443. Springer Berlin Heidelberg.

Dekel, O., C. Manning, and Y. Singer (2004). Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16*.

Domingos, P. and M. Pazzani (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning 29*(2), 103–130.

Grbovic, M., N. Djuric, and S. Vucetic (2012). Learning from pairwise preference data using gaussian mixture model. *Preference Learning: Problems and Applications in AI*, 33.

Gurrieri, M., X. Siebert, P. Fortemps, S. Greco, and R. Słowiński (2012). Label ranking: A new rule-based label ranking method. In S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. Yager (Eds.), *Advances on Computational Intelligence*, Volume 297 of *Communications in Computer and Information Science*, pp. 613–623. Springer Berlin Heidelberg.

Har-Peled, S., D. Roth, and D. Zimak (2002). Constraint classification: A new approach to multiclass classification. In N. Cesa-Bianchi, M. Numao, and R. Reischuk (Eds.), *Algorithmic Learning Theory*, Volume 2533 of *Lecture Notes in Computer Science*, pp. 365–379. Springer Berlin Heidelberg.

Hüllermeier, E., J. Fürnkranz, W. Cheng, and K. Brinker (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence 172*(2008), 1897–1916.

Lebanon, G. and J. Lafferty (2002). Cranking: Combining rankings using conditional probability models on permutations. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 370. Morgan Kaufmann Publishers Inc.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ribeiro, G., W. Duivesteijn, C. Soares, and A. Knobbe (2012). Multilayer perceptron for label ranking. In A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2012*, Volume 7553 of *Lecture Notes in Computer Science*, pp. 25–32. Springer Berlin Heidelberg.

Todorovski, L., H. Blockeel, and S. Dzeroski (2002). Ranking with predictive clustering trees. In T. Elomaa, H. Mannila, and H. Toivonen (Eds.), *Machine Learning: ECML 2002*, Lecture Notes in Computer Science, pp. 444–455. Springer Berlin Heidelberg.

Vembu, S. and T. Gärtner (2010). Label ranking algorithms: A survey. In J. Fürnkranz and E. Hüllermeier (Eds.), *Preference Learning*, pp. 45–64. Springer.

Vogt, M., J. Godden, and J. Bajorath (2007). Bayesian interpretation of a distance function for navigating high-dimensional descriptor spaces. *Journal of Chemical Information and Modeling 47*(1), 39–46.

Table 1: Example of preferences of a golf player

The table shows preferences of a golf player for the golf courts based on weather conditions during the 6 days.

| Day | Outlook | Temperature | Humidity | Wind | Ranks | | |
|---|---|---|---|---|---|---|---|
| | | | | | A | B | C |
| 1 | Sunny | Hot | High | Weak | 1 | 2 | 3 |
| 2 | Sunny | Hot | High | Strong | 2 | 3 | 1 |
| 3 | Overcast | Hot | High | Weak | 1 | 2 | 3 |
| 4 | Rain | Mild | High | Weak | 1 | 3 | 2 |
| 5 | Rain | Mild | High | Strong | 1 | 2 | 3 |
| 6 | Sunny | Mild | High | Strong | 3 | 2 | 1 |

Table 2: Comparison of probabilities

The table shows the comparison of values of prior (panel A) and conditional (panel B) probabilities of golf courts rankings from Table 1 as a classification ($P$) and as a label ranking ($P_{LR}$) problem.

**Panel A: prior probability**

| $y$ | | | $P(y)$ | $P_{LR}(y)$ |
|---|---|---|---|---|
| A | B | C | 0.500 | 0.667 |
| B | C | A | 0.167 | 0.542 |
| A | C | B | 0.167 | 0.708 |

**Panel B: conditional probability**

| $y$ | | | $P(Outlook = Sunny\|y)$ | $P_{LR}(Outlook = Sunny\|y)$ |
|---|---|---|---|---|
| A | B | C | 0.333 | 0.312 |
| B | C | A | 1.000 | 0.615 |
| A | C | B | 0.000 | 0.412 |

Table 3: Description of the label ranking dataset

The table depicts the data used in the label ranking experiments. Type A datasets is based on the naive Bayes classifier. Type B is from the regression data.

| Datasets | type | Instances | Features | Labels |
|---|---|---|---|---|
| authorship | A | 841 | 70 | 4 |
| glass | A | 214 | 9 | 6 |
| iris | A | 150 | 4 | 3 |
| segment | A | 2310 | 18 | 7 |
| vehicle | A | 846 | 18 | 4 |
| vowel | A | 528 | 10 | 11 |
| wine | A | 178 | 13 | 3 |
| bodyfat | B | 252 | 7 | 7 |
| cpu-small | B | 8192 | 6 | 5 |
| housing | B | 506 | 6 | 6 |
| stock | B | 950 | 5 | 5 |
| wisconsin | B | 194 | 16 | 16 |
| cold | - | 2465 | 24 | 4 |
| diau | - | 2465 | 24 | 7 |
| dtt | - | 2465 | 24 | 4 |
| heat | - | 2465 | 24 | 6 |
| spo | - | 2465 | 24 | 11 |

Table 4: Results of Label Ranking experiments on KEBI datasets

The table depicts the results of label ranking experiments applied on KEBI dataset sorted by type of the dataset. We use 10-fold cross validation. Bold fonts means the algorithm outperformed the baseline.

| Datasets | type | baseline | nbr.cont | nbr.disc | knn.cont | knn.disc |
|---|---|---|---|---|---|---|
| authorship | A | 0.643 | 0.365 | **0.665** | **0.955** | **0.936** |
| glass | A | 0.698 | 0.695 | **0.764** | **0.901** | **0.846** |
| iris | A | 0.150 | **0.817** | **0.82** | **0.973** | **0.896** |
| segment | A | 0.470 | **0.756** | **0.742** | **0.978** | **0.952** |
| vehicle | A | 0.216 | **0.611** | **0.656** | **0.889** | **0.845** |
| vowel | A | 0.250 | **0.747** | **0.405** | **0.947** | **0.875** |
| wine | A | 0.346 | **0.781** | **0.483** | **0.942** | **0.892** |
| bodyfat | B | -0.074 | **0.178** | **0.077** | **0.196** | **0.175** |
| cpu-small | B | 0.259 | **0.343** | **0.315** | **0.504** | 0.126 |
| housing | B | 0.069 | **0.604** | **0.557** | **0.819** | **0.628** |
| stock | B | 0.075 | **0.666** | **0.414** | **0.962** | **0.823** |
| wisconsin | B | -0.026 | **0.555** | **0.184** | **0.601** | **0.454** |
| cold | - | 0.050 | **0.091** | 0.02 | **0.087** | **0.052** |
| diau | - | 0.259 | 0.157 | 0.251 | 0.19 | 0.179 |
| dtt | - | 0.124 | **0.143** | 0.105 | 0.09 | 0.087 |
| heat | - | 0.035 | **0.054** | 0.029 | **0.056** | 0.033 |
| spo | - | 0.204 | 0.113 | 0.182 | 0.11 | 0.099 |

Table 5: Summary of models

| Category | Label ranking methods | Description | References |
|---|---|---|---|
| *Decomposition*: The LR problem is decomposed into small, simpler sub-problems (binary classification problems) that, on average, achieve the great performance in experiments but requires an ensemble of binary models | Constraint classification (CC) | Turns the LR problem into single binary classification problem in an extended space and learns LR model from the classifier | Har-Peled et al. (2002) |
| | Log-linear model (LL) | Learns the utility function for each individual label | Dekel et al. (2004) |
| | Pairwise comparison (RPC) | Directly models individual preferences (without estimating utility function). An extension of pairwise classification | Hüllermeier et al. (2008) |
| *Probabilistic*: leverages statistical probability models to develop LR methods. Good: provides the measure of reliability of prediction. Bad: requires storing the all training data in memory | Instance-base (Mallows). | Distance-based probability model that defines the probability of ranking according to its distance to a center ranking. | Cheng and Hüllermeier (2009) |

*... continued next page*

Table 5: (continued)

| Category | Label ranking methods | Description | References |
|---|---|---|---|
| | Decision trees | Similar to conventional decision tree learning. The difference is that the split criterion is at inner nodes and different criterion for stopping the recursing partitioning. | Cheng et al. (2009) |
| | Instance-base (Plackett-Luce) | The probability is based on the scores of unassigned labels. | Cheng et al. (2010) |
| | Generalize linear models | | Cheng et al. (2010) |
| | Gaussian mixture model | The model consist of mixtures defined by prototypes which are associated with preference judgment for each pair of labels. | Grbovic, Djuric, and Vucetic (2012) |
| *Similarity*: replaces probability with similarity between the rankings. Minimizing the distance is equivalent to maximizing the likelihood (maximizing the similarity). Good: assigns non-zero probabilities that are not observed in data. Bad: shows moderate predicting accuracy | Naive Bayes | Adaptation of naive Bayes for classification. Adapts the prior and conditional probabilities in the realm of LR | Aiguzhinov, Soares, and Serra (2010) |

*... continued next page*

Table 5: (continued)

| Category | Label ranking methods | Description | References |
|---|---|---|---|
| | Association rules | Adaptation of APRIORI. The goal is to discover frequent pairs of attributes associated with a ranking | de Sá, Soares, Jorge, Azevedo, and Costa (2011) |
| | Multilayer perception | Adaptation of MLP. Adapts the error functions that guide the back-propagation leaning process and the method to generate a ranking from the output layer. | Ribeiro, Duivesteijn, Soares, and Knobbe (2012) |
| | Rank distance | The LR model learns rankings from the nearest neighbor | Brazdil et al. (2003) |
| | Rule-based | The learning approach is based on reduction technique | Gurrieri, Siebert, Fortemps, Greco, and Słowiński (2012) |