Alternative Decomposition Techniques for Label Ranking

Massimo Gurrieri*, Philippe Fortemps, and Xavier Siebert

UMons, Rue du Houdain 9, 7000 Mons, Belgium
massimo.gurrieri@umons.ac.be

Abstract. This work focuses on label ranking, a particular task of preference learning, wherein the problem is to learn a mapping from instances to rankings over a finite set of labels. This paper discusses and proposes alternative reduction techniques that decompose the original problem into binary classification related to pairs of labels and that can take into account label correlation during the learning process.

Keywords: Preference Learning, Label Ranking, Reduction Techniques, Machine Learning, Binary Classification.

1 Introduction

Preference learning [1] is gaining increasing attention in data mining and related fields. Preferences can be considered as instruments to support or identify liking or disliking of an object over others in a declarative and explicit way. In particular, the learning and modelling of preferences are being recently investigated in several fields such as knowledge discovery, machine learning, multi-criteria decision making, information retrieval, social choice theory and so on. In a general meaning, preference learning is a non trivial task consisting in inducing predictive preference models from collected empirical data. The most challenging aspect is the possibility of predicting weak or partial orderings of classes (labels), rather than single values (as in supervised classification). For this reason, preference learning can be considered as an extension of conventional supervised learning tasks, wherein the input space can be interpreted as the set of preference contexts (e.g. queries, users) while the output space consists in the preference predictions provided in the form of partial orders, linear orders, top-k lists, etc. Preference learning problems are typically distinguished in three topics [1]: object ranking, instance ranking and label ranking. **Object ranking** consists in finding a ranking function F whose input is a set X of instances characterized by attributes and whose output is a ranking of this set of instances, in the form of a weak order. Such a ranking is typically obtained by giving a score to each $x \in X$ and by ordering instances with respect to these scores. The training process takes as input either partial rankings or pairwise preferences between instances of X. In the context of **instance ranking**, the goal is to find a ranking function

* Corresponding author.

A. Laurent et al. (Eds.): IPMU 2014, Part II, CCIS 443, pp. 464–474, 2014.

F whose input is a set X of instances characterized by attributes and whose output is a ranking of this set (again a weak order on X). However, in contrast with object ranking, each instance x is associated with a class among a set of ordered classes. The output of a such a kind of problem consists in rankings wherein instances labeled with higher classes are preferred (or precede) instances labeled with lower classes. The third learning scenario concerns a set of training instances which are associated with rankings over a finite set of *labels*, i.e. **label ranking** [2, 3, 4, 5, 6, 8]. The main goal in label ranking is to predict weak or partial orderings of labels. This paper is organized as follows. In section 2, we introduce label ranking and existing approaches. In particular, we discuss learning reduction techniques that transform label ranking into binary classification. In section 3, we describe some novel reduction techniques to reduce label ranking to binary classification that are capable of taking into account correlations among labels during the learning process. Finally, in section 4 and 5, we present some experimental results, conclusions and future work, respectively.

2 Label Ranking

In label ranking, the main goal is to predict for any instance x, from an instance space X, a preference relation $\succ_x$: $X \rightarrow L$, where $L = \{\lambda_1; \lambda_2; ...; \lambda_k\}$ is a set of labels or alternatives, such that $\lambda_i \succ_x \lambda_j$ means that instance x prefers label λ_i to label λ_j or, equivalently, λ_i is ranked higher than λ_j. More specifically, we are interested to the case where $\succ_x$ is a total strict order over L, or equivalently, a ranking of the entire set L. This ranking can therefore be identified with a permutation $\pi_x \in \Omega$ (the permutation space of the index set of L), such that $\pi_x(i) < \pi_x(j)$ means that label λ_i is preferred to label λ_j ($\pi_x(i)$ represents the position of label λ_i in the ranking). As in classification, it is possible to associate x to an unknown *probability distribution* $\mathbb{P}(.|x)$ over the set Ω so that $\mathbb{P}(\tau|x)$ is the probability to observe the ranking τ given the instance x. Typically, the prediction quality of a label ranker M is measured by means of its *expected loss* on rankings:

$$\mathbb{E}(D(\tau_x, \tau'_x)) = \mathbb{E}(D(\tau, \tau')|x) \tag{2.1}$$

where $D(.,.)$ is a distance function (between permutations), τ_x is the true value (ground truth) and τ'_x is the prediction made by the model M. Given such a distance metric, the best prediction is: $\tau^* = \arg\min_{\tau' \in \Omega} \sum_{\tau \in \Omega} \mathbb{P}(\tau|x) D(\tau', \tau)$. Spearman's footrule, Kendall's tau and the sum of squared distances are well-known distances between rankings [14, 15, 17]. There are two main groups of approaches to label ranking. On the one hand, decomposition (or learning reduction) methods transform label ranking problem into binary classification [2, 3, 4, 5]. On the other hand, direct methods adapt existing classification algorithms in order to deal with label ranking [6, 9, 10, 16]. This work focuses on decomposition methods because they directly learn binary preferences, i.e. simple statements like $x \succ y$ and allow to build meta-learners, i.e. rankers where any binary classifier can be used as base classifier. For example, a rule-based label ranker has

been recently proposed [2, 3]. In the context of multi-label classification, it has been recently reported [7, 13] that it is crucial to take into account correlations between labels. As a consequence, it seems natural to put this issue into perspective also in label ranking. However, the standard pairwise learning reduction [5] does not take such correlations into account: a separate binary classifier is trained for each pair of labels so that each pair of labels is treated independently from the remaining pairs. In view of this, we propose in this paper alternative decomposition techniques, other than [4, 5], to take into account correlations among labels, while limiting computational complexity.

3 Reduction Framework

In the context of label ranking, the training set is $T = \{(\mathbf{x}, \pi_x)\}$, where $\mathbf{x} = (q_1, q_2, ..., q_l)$ is a vector of l attributes (the feature vector) and π_x is the corresponding target label ranking associated with the instance $\mathbf{x}$. In sections 3.1, 3.2, 3.3 we present three novel pairwise reductions techniques: Nominal decomposition (similar to the one presented in [2, 3]), Dummy Coding decomposition and Classifier Chains (based on the Classifier Chains for Multi-label Classification [7]). In section 3.4 we also discuss a method for the ranking aggregation problem. A probabilistic interpretation of the Classifier Chains for label ranking is finally presented in section 3.5.

3.1 Pairwise Decomposition: Nominal Coding

In this decomposition, each learning instance $(\mathbf{x}, \pi_x) = (q_1, q_2, ..., q_l, \pi_x)$ is transformed into a set of simpler and easier-to-learn instances $\{\mathbf{x}_{1,2}, \mathbf{x}_{1,3}, ..., \mathbf{x}_{i,j}, ...\}$, where the generic instance $\mathbf{x}_{i,j}$ is responsible to convey not only the feature vector $(q_1, q_2, ..., q_l)$ but also information about a specific pair of labels (λ_i,λ_j), according to a given decomposition of π_x into pairwise preferences. The number of pairs is at most (in case of full rankings) $k(k-1)/2$, where $k = |L|$. The learning process associated with this reduction is:

$$\mathbf{x}_{i,j} = (q_1, q_2, ..., q_l, r_{i,j}, d) \tag{3.1}$$

with $i, j \in \{1, 2, ..., k\}, i < j$, $d \in \{-1, +1\}$ and $r_{i,j}$ is a nominal attribute which uniquely identifies the pair (λ_i,λ_j). In this manner, each $\mathbf{x}_{i,j}$ is a learning instance responsible only for the specific pair (λ_i,λ_j). The binary attribute $d \in \{-1; +1\}$ takes into account the preference relation between the two labels (λ_i,λ_j), according to the original ranking π_x. That is, $d = +1$ when λ_i is preferred to λ_j (or ranked higher), otherwise $d = -1$, according to the provided input $(\mathbf{x}, \pi_x)$. For example, the instance $\mathbf{x} = (-1.5, 2.4, 1.6, \lambda_2 \succ \lambda_1 \succ \lambda_3)$ generates the following learning instances: $\mathbf{x}_{1,2} = (-1.5, 2.4, 1.6, r_{1,2}, -1)$, $\mathbf{x}_{1,3} = (-1.5, 2.4, 1.6, r_{1,3}, +1)$, $\mathbf{x}_{2,3} = (-1.5, 2.4, 1.6, r_{2,3}, +1)$. By using this reduction, it is possible to treat the overall pairwise preference information in a single learning set, instead of creating independent learning sets as in [5]. This allows

to process the overall preference information at once in a unique learning set and therefore to learn a model M that takes into account correlations between labels, if any. The classification problem derived from the original label ranking problem can be solved by any binary classifier (e.g. Multilayer Perceptron). Assuming a given base learner whose complexity is $\Phi(l, |X|)$, the complexity of this reduction is $\Phi(l+1, |X| \times p)$, where $p = k(k-1)/2$, since the number of instances is multiplied by p (i.e. a copy of the original instance for every pair of labels). To classify a new instance x', for each pair of labels (λ_i, λ_j), with $i, j \in \{1, 2, ..., k\}, i < j$, the feature vector of the testing instance x' is augmented by adding a variable $r_{i,j}$ one at a time. This allows the model M to predict either $+1$ or -1 for the specific query pair (λ_i, λ_j).

3.2 Pairwise Decomposition: Dummy Coding

To avoid the use of nominal attributes, another reduction is based on the dummy coding so that ones and zeros are added to the feature space in order to convey the preference information about pairs of labels. Similarly as in (3.1), each learning instance $(\mathbf{x}, \pi_x) = (q_1, q_2, ..., q_l, \pi_x)$ is transformed into a set of instances $\{\mathbf{x}_{1,2}, \mathbf{x}_{1,3}, ..., \mathbf{x}_{i,j}, ...\}$. While in the previous reduction the feature space was augmented by one (a nominal attribute identifying a pair of labels), in this reduction scheme the feature space is augmented by exactly p binary attributes, where only one attribute is set to 1, the others being set to 0. The learning process associated with this reduction is:

$$\mathbf{x}_{i,j} = (q_1, q_2, ..., q_l, r_{1,2}, ..., r_{i,j}, ...r_{k-1,k}, d) \tag{3.2}$$

with $i, j \in \{1, 2, ..., k\}, i < j$, $d \in \{-1, +1\}$ and $r_{v,z} = 1$ if $v = i \wedge z = j$, 0 otherwise. Since for every pair (i, j) only one variable $r_{i,j}$ is set to 1, the corresponding learning instance $\mathbf{x}_{i,j}$ is responsible for that specific pair. For example, the instance: $\mathbf{x} = (-1.5, 2.4, 1.6, \lambda_2 \succ \lambda_1 \succ \lambda_3)$ generates the following learning instances: $\mathbf{x}_{1,2} = (-1.5, 2.4, 1.6, 1, 0, 0, -1)$, $\mathbf{x}_{1,3} = (-1.5, 2.4, 1.6, 0, 1, 0, +1)$, $\mathbf{x}_{2,3} = (-1.5, 2.4, 1.6, 0, 0, 1, +1)$. Assuming a given base learner whose complexity is $\Phi(l, |X|)$, the complexity of this reduction is $\Phi(l + p, |X| \times p)$, where $p = k(k-1)/2$, since p binary attributes are added to each instance $\mathbf{x}_{i,j}$, while the number of instances is multiplied by p (i.e. a copy of the original instance for each pair of labels). The classification of a new instance x' is similar to the nominal decomposition.

3.3 Pairwise Decomposition: Classifier Chains

In this section we present another learning reduction technique which is based on Classifier Chains for multi-label classification [7, 13]. The proposed reduction scheme involves $p = k(k-1)/2$ binary classifiers, each binary classifier being responsible for learning and predicting the preference for a specific pair, given preference relations on previous pairs of labels, in a *chaining* scheme. In this way all previous pairs of labels are treated as additional attributes to model

conditional dependence between a given pair of labels and all preceding pairs. The most interesting aspect of this reduction scheme is that it is possible to propagate preference information about pairs of labels between all classifiers throughout the chain, enabling thus to take correlations among labels into account. Moreover, there is a gain in complexity w.r.t. the previous reductions because the size of the learning set does not change at each iteration. The set of binary classifiers (the chain) $h = (h_1, h_2, ..., h_p)$ is used to model a global label ranker where each classifier h_j is trained with

$$\mathbf{x} = (q_1, q_2, ..., q_l, r_1, r_2, ..., r_{j-1}, d) \tag{3.3}$$

as a learning instance, $r_1, r_2, ..., r_{j-1}$ being the values (either $+1$ meaning $\succ$, or -1 meaning $\prec$) on the $j-1$ previous pairs of labels provided by the ranking π_x and according to the chosen order of decomposition. The attribute $d \in \{-1; +1\}$ is the preference information about the jth pair of labels also according to π_x and to the order of decomposition. It should be noticed that a default or a random order of labels can be considered in the decomposition of the label set L. Assuming a given base learner whose complexity is $\Phi(l, |X|)$, the complexity of each single classifier h_i is $\Phi(l + c_i, |X|)$, where $1 \leq c_i \leq p$, since c_i attributes (binary variables) are added (at most p, in the last classifier) to each instance. For example, if $|L| = 3$ and the order decomposition is $\{(2,1), (2,3), (3,1)\}$, the chain consists in (h_1, h_2, h_3) and the input instance $x = (-1.5, 2.4, 1.6, \lambda_2 \succ \lambda_1 \succ \lambda_3)$ is used as a learning instance in the following way: $h_1 \leftarrow (-1.5, 2.4, 1.6, +1)$, $h_2 \leftarrow (-1.5, 2.4, 1.6, +1, +1)$, $h_3 \leftarrow (-1.5, 2.4, 1.6, +1, +1, -1)$. The classification of a new instance x' is performed in the following way. The classifier h_1 predicts the value (either $+1$ or -1) for the first pair of labels, according to the given decomposition order. Afterwards, the feature vector of x' is augmented with the prediction on the first pair of label and the classifier h_2 predicts the value of the second pair of labels (by testing the feature vector of x' augmented by the previous prediction). In an iterative way the classifier h_j predicts the value of the jth pair using the feature vector augmented by all previous predictions provided by $(h_1, h_2, ..., h_{j-1})$. Since the order of labels could have an impact on the prediction accuracy, we also consider the ensemble scheme proposed in [7, 13]. In this manner, it is possible to avoid not only the bias due to a single (default or random) order of labels but also the effect of error propagation along the chain in case the first classifiers perform poorly. The main idea is to train T classifier chains (typically $T = 10$) where each classifier is given a random label order and moreover, each classifier is trained on a random selection of learning instances *sampled with replacement* (typically 75% of the learning set) in order to reduce time complexity without loss in prediction quality. It should be noticed that to avoid the use of an ensemble of classifier chains, some heuristics could be used to select the most appropriate order. Such heuristics are currently under study.

3.4 Ranking Generation Process

The reduction techniques (3.1), (3.2) and (3.3) require an additional step to provide a final ranking for a testing instance x'. The final ranking should be as much as possible consistent with the preference relations $\succ_{x'}$ on each pair of labels learned during the classification process. However, this is not trivial [2, 3, 4, 11, 14, 15] since the resulting preference relation is total, asymmetric, irreflexive but not transitive, in general. The underlying problem is how to find a consensus between the pairwise predictions in order to obtain a linear order? This is related to the well-known NP-hard Kemeny optimal rank aggregation problem [14, 15]. A natural choice, at least in this context, for solving the ranking aggregation problem is the *Net Flow Score* procedure [2, 3, 11] whose complexity is $O(k^2)$, where k is the number of labels. This procedure allows to obtain a ranking by ordering labels according to their net flow scores. These scores can be computed by using estimations of conditional probabilities on pairs of labels (good estimations can be provided for example by neural networks [18]) and are defined as follows. Let us define:

$$\Gamma^+_{(i,j)} = \mathbb{P}(\lambda_i \succ_{x'} \lambda_j) = \mathbb{P}(d = +1|x') \tag{3.4}$$

as the probability that for the instance x' label λ_i is ranked higher (preferred to) than λ_j. Each label λ_i is then evaluated by means of the following score:

$$S(i) = \sum_{j \neq i} (\Gamma^+_{(i,j)} - \Gamma^+_{(j,i)}), \tag{3.5}$$

where $\Gamma^+_{(i,j)}$ is given by (3.4). The final ranking is obtained by ordering labels according to decreasing values of (3.5), so that the higher the score, the higher the preference in the ranking: $S(i) > S(j) \Leftrightarrow \tau_i < \tau_j$. It is possible to prove, in a similar way as proved in [5], that the *Net Flow Score* procedure, as defined in (3.5), minimizes the expected loss (2.1), according to the sum of squared rank distance. This means that, if correct posterior probabilities can be obtained (or at least good estimations thereof), it is possible to find an optimal ranking by simply ordering labels according to scores (3.5). Even though the net flow score procedure does not provide in general optimal rankings w.r.t. the Kendall distance, empirically it provides good performances (see section 4).

3.5 Probabilistic Classifier Chains

In the context of multi-label classification, a Bayes-optimal probabilistic classifier chains has been recently discussed [13]. In this section, we discuss a probabilistic classifier chains for Label Ranking which relies on the same idea. Given a decomposition order of the label set into pairs, a permutation π can be identified in a unique way with a binary vector $(y^\pi_1, ..., y^\pi_p) \in \{-1, +1\}^p$ so that

$y_i^\pi = +1 \Leftrightarrow \lambda_j \succ \lambda_v$ while $y_i^\pi = -1$ otherwise. In this manner, the probability of a permutation π is equivalent to the probability of its associated vector $(y_1^\pi, ..., y_p^\pi)$ and by means of the chain rule as in a Bayesian network:

$$\mathbb{P}(\pi|x') = \mathbb{P}(y_1^\pi, ..., y_p^\pi|x') = \mathbb{P}(y_1^\pi|x') * \prod_{i=2}^{p} \mathbb{P}(y_i^\pi|x', y_1^\pi, ..., y_{i-1}^\pi). \tag{3.6}$$

The chaining procedure (3.3) allows to learn a probabilistic classifier f_i, $i = 1, ..., p$ for each pair of labels, where $p = k*(k-1)/2$. This classifier predicts, for the ith pair of labels $y_i = (\lambda_j, \lambda_v)$, either $+1$ meaning $\lambda_j \succ \lambda_v$ or -1 meaning that $\lambda_v \succ \lambda_j$, according to the probability distribution learnt by the classifier f_i. By knowing for each pair of labels its (conditional) probability, it is possible to compute the (conditional) probability of π. By means of (3.6), it is therefore possible to rank all possible permutations for x' w.r.t. their probabilities so that an optimal prediction is given by:

$$\pi^* = \arg\max_{\pi \in S} \left[\mathbb{P}(y_1^\pi|x') * \prod_{i=2}^{p} \mathbb{P}(y_i^\pi|x', y_1^\pi, ..., y_{i-1}^\pi)\right] \tag{3.7}$$

As a result, this probabilistic formulation is well-tailored for the subset 0/1 loss function [7, 13]:

$$L(\pi, \pi') = \mathbb{1}_{\pi \neq \pi'}. \tag{3.8}$$

Interestingly, it can easily be proved that the optimal prediction for the associated **risk** minimization problem is given by: $\pi^* = \arg\max_{\pi \in S} \mathbb{P}(\pi|x')$. Moreover, the classifier chains presented in section 3.3 can be considered as a deterministic approximation of (3.6), as similarly pointed out in [13]. While the probabilistic approach evaluates all possible permutations, the classifier chain provides, in general, a suboptimal prediction gradually obtained at each iteration of the chaining scheme by using:

$$\mathbf{h}_j(x') = \arg \max_{r_j \in \{-1,+1\}} \mathbb{P}(r_j|x', r_1, ..., r_{j-1}). \tag{3.9}$$

As a consequence, the chaining scheme (3.9) does not provide in general neither an optimal solution w.r.t the subset 0/1 loss function nor a linear order. Interestingly, the probabilistic approach (3.6) does not require any *ranking aggregation algorithm* since it directly evaluates permutations. However, a label ranker well-tailored for the subset 0/1 loss is probably not reasonable in this context given that it is a quite severe loss (even a slightly different prediction gets the highest penalty). Nevertheless, a method well-tailored for the subset 0/1 loss function should exibit good performances w.r.t. other loss functions (Kendall's tau distance, Spearman's footrule, etc.). On the other hand, the cost in terms of computational complexity is very high: in case of k labels, $k!$ permutations have to be evaluated, which imposes an upper limit of $k \approx 6$ labels. Nevertheless, it should be noticed that a stop criterion could be applied in order to reduce time

complexity. If $p*$ is the probability associated with an initial permutation π_0, the evaluation of another permutation π_k can be stopped at the jth iteration as soon as:

$$\mathbb{P}(y_1^{\pi_k}|x') * \prod_{i=2}^{j} \mathbb{P}(y_i^{\pi_k}|x', y_1^{\pi_k}, ..., y_{i-1}^{\pi_k}) < p^* \tag{3.10}$$

This stop criterion allows to discard not only the permutation π_k but also all permutations wherein the first j pairs share the same values as π_k. This criterion could considerably reduce the complexity during the testing process. As in section 3.3, an ensemble of probabilistic classifier chains can be considered.

4 Experimental Setup and Results

This section is devoted to experimentations that we conducted to evaluate the performances of the proposed methods in terms of predictive accuracy. The data sets used in this paper were taken from the KEBI Data Repository[1]. The evaluation measures used in this study are the *Kendall's tau* and the *Spearman Rank Correlation coefficient* [3, 4, 16, 17]. Performance of the methods was estimated by using a cross-validation study (10-fold). We compared the standard pairwise comparison (SD) [5] (note that the Net Flow score is used for the rank aggregation issue) with the proposed reductions: the nominal decomposition (ND), the dummy coding decomposition (DD), random classifier chains (CD) and ensembled classifier chains (ECD) (the voting procedure for the final ranking is also based on the Net Flow Score procedure). In this experiment, we used Multilayer Perceptron (MLP) and Radial Basis Function (RBF) as base classifiers, both with default parameters, which generally provide good estimations of posterior probabilities [18]. All experiments were run on a 64-bit machine, allowing up to 4 GB RAM of heap memory size for larger datasets. Results w.r.t. the probabilistic classifiers chains (PCD) and ensembled probabilistic classifiers chains (EPCD) are not yet available. Tables 1 and 2 show the performances of the five classifiers in terms of Kendall's tau and Spearman's Rank correlation with MLP and RBF as base classifiers, respectively. Following the Friedman Test described in [12], we found that in both cases the null-hypothesis is rejected at a significance level of 1%. According to the post-hoc Nemenyi test [12], the significant difference in average ranks of the classifiers is 1.760 at a significant level of 5% and 1.587 at a significant level of 10%. At a significance level of 5%, ECD outperforms SD and ND when using MLP as base classifier, while the post-hoc test is not powerful enough to establish any other statistical difference. At a significance level of 10%, ECD also outperforms CC. When using RBF as base classifier, at a significant level of 5%, ECD outperforms ND and DD while SD outperforms ND and DD. Moreover, CD outperforms ND.

[1] See `http://www.uni-marburg.de/fb12/kebi/research/repository`

Table 1. Comparison of reduction techniques with MLP as base classifier

Kendall tau	SD	ND	DD	CD	ECD
IRIS	.973+-.045 (4)	.964+-.055 (5)	**.991+-.017** (1)	.982+-.021 (2)	.977+-.029 (3)
GLASS	.880+-.064 (2)	.860+-.079 (5)	.865+-.062 (4)	.878+-.055 (3)	**.888+-.056** (1)
WINE	.929+-.048 (4)	.925+-.040 (5)	**.939+-.059** (1)	.936+-.036 (2.5)	.936+-.048 (2.5)
VEHICLE	.875+-.028 (4)	.877+-.023 (5)	.877+-.023 (3)	.892+-.026 (2)	**.893+-.020** (1)
VOWEL	**.910+-.014** (1.5)	.825+-.038 (5)	.861+-.022 (4)	.888+-.027 (3)	**.910+-.014** (1.5)
STOCK	.830+-.013 (5)	.874+-.010 (3)	.868+-.009 (4)	.905+-.010 (2)	**.914+-.017** (1)
CPU	.443+-.011 (4)	.472+-.015 (3)	.479+-.009 (2)	.431+-.024 (5)	**.487+-.014** (1)
BODYFAT	.229+-.054 (4)	.241+-.065 (3)	**.272+-.042** (1)	.150+-.081 (5)	.243+-.072 (2)
DDT	.062+-.040 (5)	.103+-.027 (3)	.120+-.022 (2)	.069+-.041 (4)	**.123+-.022** (1)
HOUSING	.641+-.032 (5)	.712+-.040 (2)	.699+-.032 (3)	.667+-.061 (4)	**.721+-.034** (1)
AUTORSHIP	.858+-.023 (5)	.929+-.016 (3)	.915+-.015 (4)	.937+-.010(2)	**.941+-.015** (1)
WISCONSIN	**.583+-.039** (1)	.108+-.111 (5)	.294+-.125 (4)	.451+-.014 (3)	.573+-.031 (2)
Av. Rate	3.70	3.91	2.75	3.12	1.50

Spearman rank correlation	SD	ND	DD	CD	ECD
IRIS	.980+-.033 (4)	.973+-.041 (5)	**.993+-.013** (1)	.986+-.016 (2)	.983+-.022 (3)
GLASS	.908+-.059 (2)	.891+-.078 (5)	.891+-.072 (4)	.900+-.059 (2)	**.918+-.057** (1)
WINE	.944+-.042 (4)	.943+-.030 (5)	**.954+-.044** (1)	.952+-.027 (2)	.949+-.042 (3)
VEHICLE	.901+-.026 (4)	.880+-.031 (5)	.902+-.021 (3)	.913+-.025 (2)	**.916+-.018** (1)
VOWEL	**.956+-.009** (1.5)	.900+-.031 (5)	.928+-.014 (4)	.930+-.022 (3)	**.956+-.008** (1.5)
STOCK	.902+-.008 (5)	.931+-.005 (3)	.928+-.005 (4)	.947+-.007 (2)	**.953+-.011** (1)
CPU	.520+-.011 (4)	.536+-.017 (3)	.543+-.013 (2)	.475+-.028 (5)	**.547+-.017** (1)
BODYFAT	.297+-.062 (4)	.315+-.078 (3)	**.347+-.047** (1)	.196+-.098 (5)	.317+-.090 (2)
DDT	.071+-.044 (5)	.119+-.029 (3)	.135+-.028 (2)	.076+-.050 (4)	**.141+-.024** (1)
HOUSING	.751+-.028 (4)	.802+-.040 (2)	.791+-.029 (3)	.742+-.059 (5)	**.807+-.035** (1)
AUTORSHIP	.895+-.018 (5)	.954+-.011 (3)	.942+-.015 (4)	.958+-.007 (2)	**.963+-.010** (1)
WISCONSIN	**.737+-.040** (1)	.158+-.155 (5)	.400+-.161 (4)	.589+-.021 (3)	.728+-.035 (2)
Av. Rate	3.66	3.91	2.75	3.12	1.54

Table 2. Comparison of reduction techniques with RBF as base classifier

Kendall tau	SD	ND	DD	CD	ECD
IRIS	.968+-.044 (3)	.808+-.063 (5)	.844+-.057 (4)	.982+-.035 (2)	**.986+-.020** (1)
GLASS	.876+-.049 (2)	.687+-.083 (5)	.707+-.053 (4)	.852+-.051 (3)	**.885+-.040** (1)
WINE	.958+-.050 (3)	.749+-.011 (5)	.828+-.061 (4)	**.977+-.033** (1)	.973+-.038 (2)
VEHICLE	.808+-.034 (3)	.544+-.041 (5)	.548+-.095 (4)	.816+-.017 (2)	**.830+-.025** (1)
VOWEL	**.815+-.015** (1)	.309+-.044 (5)	.313+-.040 (4)	.728+-.030 (3)	.786+-.019 (2)
STOCK	**.861+-.013** (1)	.527+-.072 (5)	.609+-.098 (4)	.841+-.016 (3)	.853+-.010 (2)
CPU	**.429+-.017** (1)	.245+-.034 (5)	.254+-.009 (4)	.400+-.012 (2)	.397+-.009 (3)
BODYFAT	.174+-.077 (2)	.127+-.046 (4)	.125+-.053 (5)	.143+-.038 (3)	**.179+-.053** (1)
DDT	.126+-.030 (3)	.114+-.034 (4)	.130+-.045 (2)	.112+-.027 (5)	**.144+-.018** (1)
HOUSING	.659+-.019 (2)	.441+-.111 (4)	.438+-.087 (5)	.642+-.049 (3)	**.665+-.037** (1)
AUTORSHIP	.935+-.017 (3)	.517+-.073 (5)	.584+-.103 (4)	**.936+-.013** (1.5)	**.936+-.009** (1.5)
WISCONSIN	.459+-.030 (2)	.198+-.089 (4)	.177+-.071 (5)	.427+-.047 (3)	**.465+-.037** (1)
Av. Rate	2.16	4.75	4.08	2.54	1.45

Spearman rank correlation	SD	ND	DD	CD	ECD
IRIS	.976+-.033 (3)	.856+-.047 (5)	.883+-.042 (4)	.986+-.026 (2)	**.990+-.015** (1)
GLASS	.910+-.041 (2)	.739+-.084 (4)	.735+-.043 (5)	.886+-.048 (3)	**.922+-.032** (1)
WINE	.965+-.044 (3)	.803+-.085 (5)	.865+-.051 (4)	**.983+-.025** (1)	.980+-.029 (2)
VEHICLE	.842+-.029 (3)	.613+-.051 (5)	.621+-.098 (4)	.853+-.017 (2)	**.868+-.023** (1)
VOWEL	**.895+-.012** (1)	.347+-.055 (4)	.345+-.044 (5)	.808+-.030 (3)	.874+-.015 (2)
STOCK	**.923+-.009** (1)	.631+-.079 (5)	.706+-.111 (4)	.908+-.012 (3)	.918+-.006 (2)
CPU	**.480+-.019** (1)	.304+-.042 (5)	.309+-.009 (4)	.448+-.012 (2)	.442+-.012 (3)
BODYFAT	.212+-.095 (2)	.162+-.061 (5)	.166+-.074 (4)	.180+-.051 (3)	**.227+-.076** (1)
DDT	.144+-.036 (3)	.127+-.039 (5)	.149+-.045 (2)	.128+-.036 (4)	**.163+-.023** (1)
HOUSING	.760+-.022 (2)	.529+-.123 (5)	.531+-.095 (4)	.738+-.041 (3)	**.761+-.031** (1)
AUTORSHIP	.958+-.012 (3)	.631+-.059 (5)	.696+-.078 (4)	**.959+-.009** (2)	**.960+-.006** (1)
WISCONSIN	.605+-.036 (2)	.281+-.126 (5)	.250+-.097 (4)	.569+-.051 (3)	**.608+-.043** (1)
Av. Rate	2.16	4.83	4	2.58	1.41

5 Conclusions

In this paper, we introduced alternative decomposition techniques for Label Ranking, closely related to the standard decomposition method [5], but that can take correlations among labels into account. We mainly investigated three

decompositions that transform label ranking into binary classification and that allow to create meta learners. In particular, we adapted the classifier chains and its ensembled version for multi-label classification [7, 13] to label ranking and showed that the ensemble of classifier chains outperforms all others decomposition methods in a statistically significant way. In order to increase accuracy of classifier chains, some heuristics to determine the most appropriate label order are currently under study. Furthermore, probabilistic interpretations of the classifier chains and the ensembled version have also been introduced, though experimental results have not been provided due to their extremily high computational complexity. In particular the probabilistic classifier chains minimizes the subset 0/1 loss function in expectation. Another important result concerns the Net Flow Score procedure that provides a good approximation algorithm to the ranking aggregation problem.

References

1. Fürnkranz, J., Hüllermeier, E. (eds.): Preference Learning. Springer (2010)
2. Gurrieri, M., Siebert, X., Fortemps, P., Greco, S., Słowiński, R.: Label Ranking: A New Rule-Based Label Ranking Method. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part I. CCIS, vol. 297, pp. 613–623. Springer, Heidelberg (2012)
3. Gurrieri, M., Siebert, X., Fortemps, P., Slowinski, R., Greco, S.: Reduction from Label Ranking to Binary Classification. In: DA2PL 2012 From Multiple Criteria Decision Aid to Preference Learning, pp. 3–13. UMONS (Université de Mons), Mons (2012)
4. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification for multiclass classificatin and ranking. In: Advances in Neural Information Processing Systems, pp. 785–792 (2002)
5. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label Ranking by learning pairwise preference. Artif. Intell. 172(16-17), 1897–1916 (2008)
6. Cheng, W., Hühn, J., Hüllermeier, E.: Decision Tree and Instance-Based Learning for Labele Ranking. In: Proc. ICML 2009, International Conference on Machine Learning, Montreal, Canada (2009)
7. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011)
8. Gärtner, T., Vembu, S.: Label Ranking Algorithms: A Survey. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning. Springer (2010)
9. Dekel, O., Manning, C.D., Singer, Y.: Log-linear models for label ranking. In: Advances in Neural Information Processing Systems, vol. 16 (2003)
10. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems, vol. 14 (2001)
11. Bouyssou, D.: Ranking methods based on valued preference relations: A characterization of the net flow method. European Journal of Operational Research 60(1), 61–67 (1992)
12. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Cheng, W., Hüllermeier, E., Dembczynski, K.J.: Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 279–286 (2010)

14. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation revisited (2001)
15. Schalekamp, F., van Zuylen, A.: Rank Aggregation: Together We're Strong. In: ALENEX, pp. 38–51 (2009)
16. de Sá, C.R., Soares, C., Jorge, A.M., Azevedo, P., Costa, J.: Mining Association Rules for Label Ranking. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 432–443. Springer, Heidelberg (2011)
17. Diaconis, P., Graham, R.L.: Spearman's footrule as a measure of disarray. Journal of the Royal Statistical Society, Series B (Methodological) 39, 262–268 (1977)
18. Hung, M.S., Hu, M.Y., Shanker, M.S., Patuwo, B.E.: Estimating posterior probabilities in classification problems with neural networks. International Journal of Computational Intelligence and Organizations 1(1), 49–60 (1996)