ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

An Ensemble Learning Technique for Multipartite Ranking

Stéphan Clémençon[1] and Sylvain Robbiano[2] *

1- LTCI UMR Telecom ParisTech CNRS No. 5141 - Institut Mines Telecom
46 rue Barrault, 75013 Paris -FRANCE

2- University College of London - Dept of Statistical Science
Gower Street, London WC1E 6BT - UK

Abstract. Decision tree induction algorithms, possibly combined with a consensus technique, have been recently successfully extended to multipartite ranking. It is the goal of this paper to address certain aspects of their weakness, instability and lack of smoothness namely, by proposing dedicated ensemble learning strategies. A shown by numerical experiments, bootstrap aggregation combined with a certain amount of feature randomization dramatically improve performance of such ranking methods, in terms of accuracy and robustness both at the same time.

1 Introduction

Although it can be easily formulated and covers a wide variety of applications (*e.g.* medicine, finance, search engines, e-commerce), the multipartite ranking problem is very difficult to solve. The goal is to rank objects, described by a number $d \geq 1$ of attributes/features $X = (X^{(1)}, \ldots, X^{(d)}) \in \mathbb{R}^d$ and which (temporarily hidden) ordinal labels $Y \in \{1, \ldots, K\}$ are assigned to, in the same order as that induced by the labels, on the basis of a training set of labeled examples. In practice, rankings are defined by means of a scoring function $s : \mathbb{R}^d \to \mathbb{R}$, transporting the natural order on the real line onto the feature space and the gold standard for evaluating the ranking performance of $s(x)$ is the ROC manifold, or its usual summary the VUS criterion (VUS standing for *Volume Under the* ROC *Surface*). Even if the *Empirical Risk Minimization* principle, the founding paradigm of statistical learning theory, has been extended to the situation where performance is evaluated by the VUS criterion (see [1]), very few algorithms for general ROC surface optimization have been documented in the statistical and machine-learning literature. Indeed, whereas a variety of approaches have been proposed in the bipartite situation (*i.e.* when $K = 2$), often reducing ranking to pairwise classification (see [2] for instance), K-partite ranking for $K \geq 3$ is generally adressed by decomposing the original ranking task into $K(K-1)/2$ bipartite ranking subproblems, as in [3] or [4]. However, a multipartite ranking algorithm for ROC manifold recursive optimization, called TreeRank Tournament, has been recently introduced in [5], producing a scoring rule described by an oriented binary (ranking) tree, extending the TreeRank procedure originally introduced in the bipartite setup [6]. Though

*This work is partly supported by the Chair 'Machine Learning for Big Data' of Telecom ParisTech.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

proved to be consistent, this method suffers from major drawbacks, just like decision trees for local learning problems (*e.g.* classification or regression), accentuated by the global nature of the multipartite ranking problem: instability and lack of smoothness. It is the main purpose of this article to investigate to which extent the **b**ootstrap **agg**regat**ing** technique (see [7]) combined with feature randomization can improve the performance of TreeRank Tournament, like in [8] for classification/regression trees or in [9] for bipartite ranking trees produced by the TreeRank algorithm.

The article is structured as follows. Basic concepts of multipartite ranking theory are briefly recalled in section 2, together with a short overview of (the few) statistical learning methods proposed to solve the multipartite ranking problem introduced in the literature. Section 3 describes the main ingredients of the approach we promote to increase the accuracy/stability of TreeRank Tournament, while displaying illustrative numerical results, which provides strong empirical evidence of the improvement on the original algorithm.

2 Background - Multipartite Ranking

We start off with recalling key notions related to multipartite ranking. Just like in the *ordinal regression* setup, in the K-partite ranking problem, one has a system consisting of an ordinal random output variable Y taking its values in an ordered set of cardinality K, $\mathcal{Y} = \{1, \ldots, K\}$ say, and an input random vector X, valued in a high-dimensional feature space $\mathcal{X} \subset \mathbb{R}^d$ with $d \geq 1$. Informally, based on a training sample of independent copies of the generic pair (X, Y), the goal is to learn a (measurable) *scoring function* $s : \mathcal{X} \to \mathbb{R}$ in order to rank any new observations $X_1 \ldots, X_N$ with temporarily hidden labels $Y_1, \ldots, Y_N$, so that $s(X_i)$ and Y_i tend to increase or decrease together. Though akin to multiclass classification, multipartite ranking problem cannot be formulated in an "universal" manner, see [4] for details. When it is not empty, we denote $\mathcal{S}^*$ the ensemble of optimal elements for the K-partite ranking problem.

ROC **analysis.** Let $\mathcal{S}$ be the set of all measurable scoring functions. For any $s \in \mathcal{S}$, we denote by $F_{s,k}(t) = \mathbb{P}\{s(X) \leq t \mid Y = k\}$ the cdf of the r.v. $s(X)$ given $Y = k$, for $1 \leq k \leq K$. For simplicity, we only consider the case $K = 3$ here (the present analysis can be straightforwardly extended to the general K-partite context). We denote $W^{-1}(u) = \inf\{t \in]-\infty, +\infty] : W(t) \geq u\}$, $u \in [0, 1]$, the generalized inverse of any cdf $W(t)$ on $\mathbb{R}$. Equipped with these notations, the ROC surface can be then viewed as the graph of a function $(\alpha, \gamma) \in (0, 1)^2 \mapsto \mathrm{ROC}_s(\alpha, \gamma)$, where $\mathrm{ROC}_s(\alpha, \gamma) = \left(F_{s,2} \circ F_{s,3}^{-1}(1-\gamma) - F_{s,2} \circ F_{s,1}^{-1}(\alpha)\right)_+$, at points (α, γ) such that $F_{s,1} \circ F_{s,1}^{-1}(\alpha) = \alpha$ and $F_{s,3} \circ F_{s,3}^{-1}(1-\gamma) = 1-\gamma$, with $u_+ = \max(u, 0)$ for any $u \in \mathbb{R}$. When we fix $\gamma = 1$ (resp. $\alpha = 0$), we recover the ROC curve associated to the problem 1 vs 2 (resp. 2 vs 3) $\alpha \in [0, 1] \mapsto \mathrm{ROC}_{1,2}(s, \alpha)$, defined by $\mathrm{ROC}_{1,2}(s, \alpha) = 1 - F_{s,2} \circ F_{s,1}^{-1}(1-\alpha)$ (up to a coordinate transform $(\alpha, \beta) \in [0, 1]^2 \mapsto (1-\alpha, \beta)$. As proved in [4], the ROC surface of $\mathcal{S}^*$'s elements, ROC^* say, is concave and dominates everywhere that of any other scoring function s. This functional criterion can also be summarized

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

by the *Volume Under the* ROC *Surface* $\mathrm{VUS(s)} \stackrel{\text{def}}{=} \int\int \mathrm{ROC_s}(\alpha,\gamma)\mathrm{d}\alpha\mathrm{d}\gamma$. The formula below (see [10]) permits to interpret it as the "rate of concording 3-tuples":

$$\begin{aligned}\mathrm{VUS(s)} &= \mathbb{P}\{s(X_1) < s(X_2) < s(X_3)\} + \frac{1}{2}\mathbb{P}\{s(X_1) = s(X_2) < s(X_3)\} \\ &+ \frac{1}{2}\mathbb{P}\{s(X_1) < s(X_2) = s(X_3)\} + \frac{1}{6}\mathbb{P}\{s(X_1) = s(X_2) = s(X_3)\} \quad (1)\end{aligned}$$

where X_k's denote independent r.v.'s with the distribution X given $Y = k$. Statistical versions of the ROC surface and of the VUS criterion are obtained by replacing the class distributions by their empirical counterparts. We may now quantitatively rephrase the ranking task. The goal is to build, from training data, a scoring function s whose ROC surface is "as close as possible" to ROC^*. Although many measures of "closeness" can be considered, the L_1 case is of special interest $d_1(s, s^*) = \int\int |\mathrm{ROC}^*(\alpha,\gamma) - \mathrm{ROC}_s(\alpha,\gamma)| d\alpha d\gamma = \mathrm{VUS}^* - \mathrm{VUS(s)}$, mainly because minimization of $d_1(s, s^*)$ is clearly equivalent to maximization of VUS(s).

Using TreeRank **for multipartite ranking.** TreeRank is a recursive algorithm for the problem of bipartite ranking. It produces an oriented partition of the feature space $\mathcal{X}$, defining thus a ranking for which elements of a same cell being viewed as ties. The process is described by a left-to-right oriented binary tree structure, termed *ranking tree*. The root node represents the whole feature space $\mathcal{C}_{0,0} = \mathcal{X}$ and each *internal node* (j, k) with $j < J$ and $k \in \{0, \ldots, 2^j - 1\}$ corresponds to a subset $\mathcal{C}_{j,k} \subset \mathcal{X}$, whose left and right siblings respectively depict disjoint subsets $\mathcal{C}_{j+1,2k}$ and $\mathcal{C}_{j+1,2k+1}$ such that $\mathcal{C}_{j,k} = \mathcal{C}_{j+1,2k} \cup \mathcal{C}_{j+1,2k+1}$. We call the splitting rule LeafRank and its goal is to maximize the area under the ROC curve. Therefore, it happens that this problem boils down to solve a cost-sensitive binary classification problem so TreeRank can be viewed as a weighted version of CART, see subsection 3.3 in [11] for further details.

To the best of our knowledge, two approaches for building consistent multipartite ranking rules in the VUS sense have been documented in the literature. The first one consists in reducing the K-partite problem to $K-1$ (or $K(K-1)/2$) bipartite subproblems, see [3] for instance. In [4], such a method is shown VUS consistent as soon as the bipartite procedure used to solve each subproblem is AUC consistent, just like the TreeRank algorithm studied at length in [11]. Alternatively, the technique proposed in [5], called TreeRank Tournament, corresponds to a recursive procedure for VUS maximization, producing a tree-structured scoring function. In each cell, the LeafRank algorithm is run for the problems $1vs2$ and $2vs3$ and we choose the splitting rule that maximize the VUS (Tournament step). It is the purpose of the next section to show how a bagging procedure combined with randomization can improve the performance of this multipartite ranking method.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

3 Bootstrap Aggregation and Randomization

We now describe the general approach we propose to improve performance and stability of a learning algorithm in the multipartite ranking context. Two components involved in the ensemble multipartite ranking procedure we promote here: bagging and randomization. By definition, a learning algorithm $\mathbf{S}$ is a mapping from the set of all possible training sample $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ of independent copies of the random pair (X, Y), *i.e.* the product space $(\mathcal{X} \times \{1, \ldots, K\})^n$ to the set of scoring functions. Its performance related to a given sample $\mathcal{D}_n$ is evaluated through the quantity $\mathrm{VUS}(\mathbf{S}(\mathcal{D}_n, .))$. The instability of the algorithm $\mathbf{S}(.,.)$ can be measured by the quantity $\mathbf{Instab}_n(\mathbf{S})$ equals to $\mathbb{E}[d_\tau(\mathbf{R}((\mathbf{S}(\mathcal{D}_n, X_1''), \ldots, \mathbf{S}(\mathcal{D}_n, X_N'')), \mathbf{R}((\mathbf{S}(\mathcal{D}_n', X_1''), \ldots, \mathbf{S}(\mathcal{D}_n', X_N''))))]$, where $\mathcal{D}_n$ and $\mathcal{D}_n'$ are two independent training samples, independent from the unlabelled i.i.d. sample $\{X_1'', \ldots, X_N''\}$ of the marginal distribution $F(dx)$, while $\mathbf{R}(s(X_1''), \ldots, s(X_N''))$ denotes the rank vector related to the vector $(s(X_1''), \ldots, s(X_N''))$ and d_τ the Kendall tau distance. Concerning feature randomization, we recall that it can be applied at two levels in the algorithms considered. At the level of the global ranking tree, TreeRank in the consensus approach and TreeRank Tournament can be implemented using a subset of $F_T \leq d$ features chosen at random, while each LeafRank recursion can be implemented by means of a subset of F_L features chosen among the F_T features chosen for fitting the ranking tree.

Algorithm

Input. Number $B \geq 1$ of bootstrap samples, feature randomization parameter $\theta = (F_T, F_L)$. Training sample $\mathcal{D}_n$. Randomized learning algorithm $\mathbf{S}((\mathcal{D}, \theta), .)$. Unlabeled vector $(X_1, \ldots, X_N)$ to be ordered.

1. From the original data $\mathcal{D}_n$, generate $B \geq 1$ bootstrap samples $\mathcal{D}_n^{*(1)}, \ldots, \mathcal{D}_n^{*(B)}$ by drawing with replacement
2. Build $B \geq 1$ scoring functions using $\mathbf{S}((\mathcal{D}_n^{*(b)}, \theta), .)$, $b = 1, \ldots, B$.

Output. Aggregated ranking rule

$$\left(\bar{\mathbf{S}}_B(X_1), \ldots, \bar{\mathbf{S}}_B(X_n)\right) = \frac{1}{B}\sum_{b=1}^{B} \mathbf{R}((\mathbf{S}(\mathcal{D}_n^{*(b)}, X_1''), \ldots, \mathbf{S}(\mathcal{D}_n^{*(b)}, X_N''))).$$

Numerical Experiments We investigate the impact of the aggregation with resampling and feature randomization on the performance of TreeRank Tournament. We illustrate the methodology promoted in this paper by implementing it on a real data set, the *Cardiotocography Data Set* considered in [12] namely. The data have been collected as follow: 2126 fetal cardiotocograms (CTG's in abbreviated form) have been automatically processed and 20 diagnostic features measured. The CTG's have been next analyzed by three expert obstetricians

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

and a consensus ordinal label has been then assigned to each of them, depending on the degree of anomaly observed: 1 for "normal", 2 for "suspect" and 3 for "pathologic". The performance is measure through several criteria : the mean of the empirical VUS, the standard deviation of the empirical VUS, the instability ($Instab_\tau$) and the ΔEnv that is the difference between the max and the min of the VUS. We evaluate the performance of the methods through 5 replications of a 5 fold cross-validation procedure. The evaluated procedures are the following:

- TreeRank Tournament ("TRT" in the table) the version without aggregation nor randomization, with one master tree of max depth 20 and LeafRank max depth is 5.
- Bagging TreeRank Tournament ("TRT bagg" in the table) the aggregated version of the previous one with $B = 20, 50, 100$ bootstrap samples.
- TreeRank Tournament Forest ("TRT Forest" in the table), the forest version with $B = 20, 50$ bootstrap samples and $F_T = F_L = 10$ and $F_T = F_L = 5$.

$B =$ "$-$" means that we skip the bootstrap samples so we learn 1 tree. We also compare these strategies with the aggregation of TreeRank ("AggTR" in the table) learn for each pair of labels $1 \leq i < j \leq K$(see [4] for a detailed analysis of this procedure), with bootstrap samples of size $B = 1$(single Tree), $20, 50$.

Results and comments Results are presented in Table 1. The main conclusion is that Bagging with $B = 20$ always improves hugely the plain TreeRank Tournament. TreeRank tournament outperforms the aggregation of all the TreeRank scoring function learned on each pair of labels. Moreover one can see that the features randomization at the tree level coupled with the bagging improves the performance. One can see that the randomization at the node level degrades a lot the accuracy or needs more bagging to catch up.

References

[1] S. Rajaram and S. Agarwal. Generalization bounds for k-partite ranking. In *NIPS Workshop on Learning to Rank*, 2005.

[2] S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and empirical risk minimization of U-statistics. *Ann. Statist.*, 36:844–874, 2008.

[3] J. Fürnkranz, E. Hüllermeier, and S. Vanderlooy. Binary decomposition methods for multipartite ranking. In *ECML PKDD '09*, 2009.

[4] S. Clémençon, S. Robbiano, and N.Vayatis. Ranking data with ordinal labels: Optimality and pairwise aggregation. *Machine Learning*, 91(1):67–104, 2013.

[5] S. Clémençon and S. Robbiano. The TreeRank Tournament for Multipartite Ranking. *To appear in Journal of Nonparametric Statistics*, 2014.

[6] S. Clémençon and N. Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, 2009.

[7] L. Breiman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.

[8] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Table 1: Performance for the cardiotocography dataset

Method	F_T	F_L	B	$\overline{\text{VUS}}$	σ	$Instab_\tau$	ΔEnv
AggTR	-	-	-	0.6949	0.0669	0.1486	0.3104
AggTR	-	-	20	0.7877	0.0378	0.2000	0.1538
AggTR	-	-	50	0.7866	0.0376	0.1733	0.1562
TRT bagg	-	-	-	0.7986	0.0429	0.1243	0.2135
TRT bagg	-	-	20	0.8170	0.0503	0.1133	0.1757
TRT bagg	-	-	50	0.8195	0.0576	0.1733	0.2268
TRT forest	10	10	-	0.7805	0.0553	0.1200	0.1859
TRT forest	10	10	20	0.8342	0.0434	0.1933	0.1611
TRT forest	10	10	50	0.8349	0.0416	0.1667	0.1706
TRT forest	5	5	-	0.7053	0.0647	0.1015	0.2280
TRT forest	5	5	20	0.8151	0.0447	0.1667	0.1642
TRT forest	5	5	50	0.830	0.042	0.113	0.179
TRT forest	10	5	-	0.615	0.123	0.173	0.424
TRT forest	10	5	20	0.7784	0.0493	0.1600	0.2160
TRT forest	10	5	50	0.7906	0.0414	0.1733	0.1496

[9] S. Clémençon, M. Depecker, and N. Vayatis. Ranking Forests. *Journal of Machine Learning Research*, 43(1):31–69, 2011.

[10] B.K. Scurfield. Multiple-event forced-choice tasks in the theory of signal detectability. *Journal of Mathematical Psychology*, 40:253–269, 1996.

[11] S. Clémençon, M. Depecker, and N. Vayatis. Adaptive partitioning schemes for bipartite ranking. *Machine Learning*, 43(1):31–69, 2011.

[12] A. Frank and A. Asuncion. UCI machine learning repository, 2010.