Mach Learn (2011) 83: 31–69
DOI 10.1007/s10994-010-5190-y

Adaptive partitioning schemes for bipartite ranking

How to grow and prune a ranking tree

Stéphan Clémençon · Marine Depecker · Nicolas Vayatis

Received: 8 July 2009 / Revised: 8 April 2010 / Accepted: 15 April 2010 / Published online: 5 June 2010

Abstract Recursive partitioning methods are among the most popular techniques in machine learning. The purpose of this paper is to investigate how to adapt this methodology to the *bipartite ranking problem*. Following in the footsteps of the TREERANK approach developed in Clémençon and Vayatis (Proceedings of the 2008 Conference on Algorithmic Learning Theory, 2008 and IEEE Trans. Inf. Theory 55(9):4316–4336, 2009), we present tree-structured algorithms designed for learning to rank instances based on classification data. The main contributions of the present work are the following: the practical implementation of the TREERANK algorithm, well-founded solutions to the crucial issues related to the splitting rule and the choice of the "right" size for the ranking tree. From the angle embraced in this paper, splitting is viewed as a cost-sensitive classification task with data-dependent cost. Hence, up to straightforward modifications, any classification algorithm may serve as a splitting rule. Also, we propose to implement a cost-complexity pruning method after the growing stage in order to produce a "right-sized" ranking sub-tree with large AUC. In particular, performance bounds are established for pruning schemes inspired by recent work on nonparametric model selection. Eventually, we propose indicators for variable importance and variable dependence, plus various simulation studies illustrating the potential of our method.

Keywords Scoring algorithms · Ranking trees · AUC · ROC curve · Recursive partitioning · Splitting rules · TreeRank · Pruning · Variable importance

Editor: Hendrik Blockeel.

S. Clémençon · M. Depecker
Telecom ParisTech, LTCI UMR Telecom ParisTech/CNRS No. 5141, 46, rue Barrault,
75634 Paris cedex 13, France

S. Clémençon
e-mail: stephan.clemencon@telecom-paristech.fr

N. Vayatis (✉)
ENS Cachan & UniverSud, CMLA UMR CNRS No. 8536, 61, avenue du Président Wilson,
94235 Cachan cedex, France
e-mail: nicolas.vayatis@gmail.com

1 Introduction

The goal of *bipartite ranking* procedures is to order all possible values $x \in \mathcal{X}$ of a random variable X over a measurable space $\mathcal{X}$. The available output information on each realization X is modeled by a random binary label $Y \in \{-1, +1\}$. Consider the classification dataset $\{(X_i, Y_i) : 1 \le i \le n\}$ obtained by sampling the random pair (X, Y). The scoring approach to ranking binary classification data consists of building a *scoring function* $s : \mathcal{X} \to \mathbb{R}$ which takes higher values when the event "$Y = +1$" is more likely to be observed. This problem arises in a large variety of applications, ranging from the design of search engines in information retrieval to medical diagnosis through credit-risk screening or anomaly detection in signal processing.

Several approaches have been considered in order to develop ranking algorithms under binary label information. Standard methods build a scoring rule based on the *plug-in* approach (such as logistic regression models, see for instance Hastie and Tibshirani 1990). Machine learning methods are mostly based on the maximization of a performance functional, like the AUC criterion, which depends on pairs of observations (refer to RankSVM (Joachims 2002), RankNet (Burges et al. 2005), RankBoost (Freund et al. 2003), RankRLS (Pahikkala et al. 2007)). A natural direction to explore is also the adaptation of decision trees in the spirit of CART (Breiman et al. 1984) for ranking purposes. The number of papers introducing modifications of decision trees is considerable (see for instance Provost and Domingos 2003; Ferri et al. 2003; Flach and Matsubara 2007; Hüllermeier and Vanderlooy 2008, 2009; Yu et al. 2008 and references therein). The main ideas underlying these works are: (i) the use of classification decision trees as estimators of the regression function, also known as *Probabilistic Estimation Trees* (PET), (ii) the choice of a splitting rule adapted to the bipartite ranking problem. Indeed, adapting successful classification or regression methods to ranking may require significant innovations since the ranking problem is of different nature. We point out that popular classification rules are based on the concept of *local learning* (see Friedman 1996). For classification procedures such as those obtained through recursive partitioning of the input space $\mathcal{X}$, the predicted label of a given instance $x \in \mathcal{X}$ only depends on the data lying in the subregion of the partition containing x. In contrast, the notion of ranking/ordering would rather involve comparing the subregions to each other.

Following this line of thought, we have proposed, in our previous work (Clémençon and Vayatis 2008, 2009), a different description of ranking decision trees. We characterize the output of a decision tree algorithm not only by a partition of the feature space and the local properties of the cells composing the partition, but also by a permutation over the cells. The permutation indicates how to rank new observations (points lying in the same cell being ranked equal). These two ingredients (partition and permutation) define a piecewise constant real-valued function, a so-termed scoring rule. We also developed, and thoroughly investigated, a specific recursive partitioning method, called the TreeRank algorithm. This algorithm produces scoring rules in a simple top-down approach. An important contribution of this work also consists in the connection established between the partitioning of the feature space through this algorithm and the approximation/estimation of the optimal ROC curve by splines of order 1. In Clémençon and Vayatis (2009), it was proved that, under general assumptions, the resulting piecewise linear ROC curve converges to the optimal one not only in the AUC sense but also in a stronger sense (with respect to the supremum norm). However, due to the very principle of recursive partitioning, the TreeRank algorithm suffers from the same drawback as the popular CART method (see Breiman et al. 1984): it may be fooled by an XOR configuration, yielding inappropriate first splits and compromising the results of the tree growing procedure. In classification, given the local aspect of the decision rule, a bad start may nevertheless be compensated by growing the tree further at the cost of a certain amount of artificial complexity. With ranking, this drawback may have much more dramatic consequences due to the global nature of the ranking task. In some sense, ranking errors are stacked as one grows the tree and the performance of the TREERANK algorithm is very sensitive to the chosen splitting rule. Recursive splitting is achieved by the means of the optimization of an entropic measure which accounts for AUC maximization on a given cell of the partition induced by the tree. This is called the *Optimization step* of the TREERANK algorithm and it is the critical step both from computational and approximation viewpoints.

The present paper proposes to solve the practical issues inherent to the nature of the TREERANK algorithm. The primary goal of this paper is to propose pragmatic strategies for performing the *Optimization step* of the TREERANK algorithm efficiently. Technically, the question addressed is how to split the cells in a flexible manner, so that accurate approximants of bilevel sets of the regression function may be obtained. Partition-based splitting rules, both fixed and adaptive, are considered for this purpose. We also provide an interpretation of the *Optimization step* as a *cost-sensitive* classification task with a data-dependent cost, equal to the rate of positive instances within the node to split. In this view, TREERANK appears as a recursive implementation of a cost-sensitive version of CART. The question of selecting the final size of the resulting ranking tree is also tackled from the perspective of *model selection* based on complexity penalization pruning. In this respect, two approaches are considered. The cross validation-based selection method of the CART algorithm is first extended to the ranking setup. Expected performance bounds are also established for ranking trees selected through direct minimization of a specific complexity penalized version of the AUC criterion. In addition, conditions under which such pruning schemes are shown to be consistent in the AUC sense are exhibited.

The paper is organized as follows. In Sect. 2, notations are first set out and we briefly recall important concepts of the bipartite ranking problem together with certain key results of ROC analysis. We also list the important properties of piecewise constant scoring rules which are produced by the algorithms presented in this paper. In Sect. 3 we examine how to implement the *Optimization step* of the TREERANK algorithm. Issues related to the selection of the size of the ranking tree are tackled in Sect. 4, while Sect. 5 deals with interpretation of tree-based ranking rules with perpendicular splits. Eventually, implementations were tested on artificial and real data sets and simulation results are presented in Sects. 6 and 7. Detailed proofs are deferred to Appendix A section.

2 Background and preliminaries

We start off with a brief description of the bipartite ranking task and recall key concepts related to this statistical learning problem. We also recall the principles underlying the TREERANK algorithm and state preliminary results in order to give an insight into subsequent implementations.

2.1 The bipartite setup

The probabilistic framework is exactly the same as the one in standard binary classification. We denote by (X, Y) a pair of random variables where $Y \in \{-1, +1\}$ is a binary label and X models some observation for predicting Y, taking its values in a feature space $\mathcal{X} \subset \mathbb{R}^q$ of high dimension. Throughout the paper, $\mathcal{L}$ denotes the joint distribution of (X, Y) and $p =$

$\mathbb{P}\{Y=+1\}$. The probability distribution $\mathcal{L}$ is entirely determined by the pair (μ,η) where μ denotes the marginal distribution of X and $\eta(x)=\mathbb{P}\{Y=+1 \mid X=x\}$, $x\in\mathcal{X}$, is the regression function. We also introduce $G(dx)$ and $H(dx)$, the conditional distributions X given $Y=+1$ and $Y=-1$ respectively. We will assume that these probability measures are equivalent. Observe that, with these notations, $\eta(x)=pdG/dH(x)/(1-p+pdG(x)/dH(x))$ and $\mu(dx)=pG(dx)+(1-p)H(dx)$.

We now state the bipartite ranking problem. Based on the observation of i.i.d. examples $\mathcal{D}_n=\{(X_i,Y_i): 1\le i\le n\}$, the goal here is to learn how to order all instances $x\in\mathcal{X}$ in a way that positive labels appear with large probability on the top in the list. Clearly, the simplest way of defining an order relationship on $\mathcal{X}$ is to transport the natural order on the real line to the feature space through a *scoring rule* $s:\mathcal{X}\to\mathbb{R}$. The notion of ROC curve, which we recall below, provides a functional criterion for evaluating the performance of the ordering induced by such a function. Here and throughout, we denote by $F^{-1}(t)=\inf\{u\in\mathbb{R}: F(u)\ge t\}$ the pseudo-inverse of any cumulative distribution function $F:\mathbb{R}\to\mathbb{R}$ and by $\mathcal{S}$ the set of all scoring functions, *i.e.* the space of real-valued measurable functions on $\mathcal{X}$. The indicator function of any event $\mathcal{E}$ is denoted by $\mathbb{I}\{\mathcal{E}\}$ and the notation $\mathbb{I}_C$ will also be used for denoting the indicator function of any set $C\subset\mathcal{X}$.

Definition 1 (ROC CURVE) Let $s\in\mathcal{S}$. The ROC curve of the scoring function $s(x)$ is the PP-plot given by:

$$t\mapsto(\mathbb{P}\{s(X)\ge t \mid Y=-1\},\mathbb{P}\{s(X)\ge t \mid Y=+1\}), \tag{1}$$

where, by convention, discontinuity points corresponding to possible jumps of the conditional distributions of $s(X)$ given $Y=+1$ and given $Y=-1$ are continuously connected by line segments. We denote by $\alpha\in(0,1)\mapsto \mathrm{ROC}(s,\alpha)$ the resulting curve.

Let $G_s(dx)$ and $H_s(dx)$ denote the conditional distributions of $s(X)$ given $Y=+1$ and given $Y=-1$ respectively, for any $s\in\mathcal{S}$. In the case where these probability distributions are both continuous, the ROC curve of s is the graph of the mapping:

$$\alpha\in[0,1]\mapsto \mathrm{ROC}(s,\alpha)=1-G_s\circ H_s^{-1}(1-\alpha). \tag{2}$$

Remark 1 (ALTERNATIVE CONVENTION) With the convention mentioned above, it is noteworthy that the curve $\mathrm{ROC}(s,.)$ is linear-by-parts as soon as the conditional distributions of $s(x)$ are both discrete. Another usual convention consists of defining $\mathrm{ROC}(s,.)$ as the graph of the mapping (2) in all cases. Equipped with this notation, when G_s or H_s are discrete, the ROC curve of s is piecewise constant.

Optimal ROC *curve.* It is a well-known result in ROC analysis that increasing transforms of the regression function $\eta(x)$ form the class $\mathcal{S}^*$ of optimal scoring functions in the sense that their ROC curve, namely $\mathrm{ROC}^*=\mathrm{ROC}(\eta,.)$, dominates the ROC curve of any other scoring function $s(x)$ uniformly:

$$\forall\alpha\in[0,1[,\quad \mathrm{ROC}(s,\alpha)\le\mathrm{ROC}^*(\alpha).$$

We refer to Clémençon and Vayatis (2009) for a rigorous proof based on a standard Neyman-Pearson argument together with a detailed list of properties of the optimal ROC curve. It is noteworthy that the curve ROC* is *concave*. More generally, for any scoring function

$s(x)$, $\mathrm{ROC}(s,.)$ is a concave curve as soon as the likelihood ratio dG_s/dH_s is a monotone function.

We now set the notations $H^* = H_\eta$ and $G^* = G_\eta$ as well as $Q^*(\alpha) = H^{*-1}(1-\alpha)$ for all $\alpha \in (0,1)$. We recall from Clémençon and Vayatis (2009) that if $Q^*(0) = \lim_{\alpha\to 0} Q^*(\alpha) < 1$, H^* and G^* are differentiable and $H^{*\prime}$ is lower bounded by a strictly positive constant on its support, then the function ROC^* is twice differentiable on $[0,1]$ with bounded derivatives:

$$\forall \alpha \in [0,1], \quad \mathrm{ROC}^{*\prime}(\alpha) = \frac{(1-p)Q^*(\alpha)}{p(1-Q^*(\alpha))}$$

and

$$\mathrm{ROC}^{*\prime\prime}(\alpha) = \frac{(1-p)Q^{*\prime}(\alpha)}{p(1-Q^*(\alpha))^2}.$$

Refer to Corollary 7 and Proposition 8 in Clémençon and Vayatis (2009) for further details.

The AUC *criterion.* In practice, the function-like performance measure described above is generally summarized by a scalar quantity, the *area under the* ROC *curve* (AUC in abbreviated form).

Definition 2 (The AUC criterion) Let $s(x)$ be a scoring function. The *area under its* ROC *curve* is given by

$$\mathrm{AUC}(s) = \int_{\alpha=0}^{1} \mathrm{ROC}(s,\alpha)d\alpha.$$

It is easy to check that the class $\mathcal{S}^*$ of optimal scoring functions corresponds to the set of scoring functions with maximum AUC. We set:

$$\forall s \in \mathcal{S}^*, \quad \mathrm{AUC}^* = \mathrm{AUC}(s).$$

The popularity of the AUC criterion mainly arises from the fact that it may be interpreted in a probabilistic manner, as shown by the following result, whose proof is left to the reader.

Proposition 1 *For any scoring function* $s(x)$, *we have*:

$$\begin{aligned}\mathrm{AUC}(s) &= \mathbb{P}\{s(X) > s(X') \mid (Y,Y') = (+1,-1)\} \\ &\quad + \frac{1}{2}\mathbb{P}\{s(X) = s(X') \mid (Y,Y') = (+1,-1)\},\end{aligned}$$

where (X',Y') *denotes a copy of the pair* (X,Y), *independent from the latter.*

Remark 2 (Optimal AUC) It has been shown in Clémençon et al. (2008) that, when the distribution of $\eta(X)$ is continuous, the maximal AUC depends on the dispersion of $\eta(X)$ through the relationship:

$$\mathrm{AUC}^* = \frac{1}{2} + \frac{\mathbb{E}[|\eta(X) - \eta(X')|]}{4p(1-p)},$$

where X' denotes an independent copy of the r.v. X. The quantity $\mathbb{E}[|\eta(X) - \eta(X')|]$ is known as the *Gini mean difference* of $\eta(X)$, a popular measure of dispersion in statistics. Hence, the more concentrated $\eta(X)$, the more difficult the ranking problem.

Remark 3 (ALTERNATIVE CONVENTION (BIS)) We point out that, with the other convention for ROC curves mentioned in Remark 1, the area under the ROC curve of any scoring function s reduces to the expression $\mathrm{AUC}(s) = \mathbb{P}\{s(X) > s(X') \mid (Y, Y') = (+1, -1)\}$.

2.2 Piecewise constant scoring functions

For reasons related to approximation theory and of computational nature that shall appear clearly in the subsequent analysis, we focus here on the simplest scoring functions, namely real-valued *piecewise constant* functions on the feature space $\mathcal{X}$. We also underline that it is of practical importance in many ranking applications (medical diagnosis, credit-risk screening, marketing) to segment the population in ordered "strata" with distinct features in an interpretable fashion. Any scoring function $s(x)$ of this type, taking $K \geq 1$ distinct values, yields a ranking/ordering of all instances $x \in \mathcal{X}$ entirely characterized by a partition $\mathcal{P}$ counting K nonempty measurable subsets $C_1, \ldots, C_K$, together with a permutation σ in the symmetric group $\mathfrak{S}_K$ of $\{1, \ldots, K\}$.

Definition 3 (($\mathcal{P}, \sigma$)-REPRESENTATION) The $(\mathcal{P}, \sigma)$-representation of a piecewise constant scoring function $s(x)$ taking K distinct values $\lambda_1 > \cdots > \lambda_K$ is given by:

$$s(x) = \sum_{k=1}^{K} \lambda_k \cdot \mathbb{I}\{x \in C_{\sigma(k)}\}, \tag{3}$$

where $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$ is a partition of $\mathcal{X}$ in K non empty cells and $\sigma \in \mathfrak{S}_K$.

Reciprocally, a partition $\mathcal{P} = \{C_1, \ldots, C_K\}$ including $\#\mathcal{P} = K$ non empty cells combined with a permutation $\sigma \in \mathfrak{S}_K$ defines a scoring function with $(\mathcal{P}, \sigma)$-representation:

$$s_{\mathcal{P},\sigma}(x) = \sum_{k=1}^{K} (K - k + 1) \cdot \mathbb{I}\{x \in C_{\sigma(k)}\}.$$

The ordering induced by (3) is entirely characterized by the pair $(\mathcal{P}, \sigma)$, in the sense that its ROC curve coincides with $\mathrm{ROC}(s_{\mathcal{P},\sigma}, .)$.

We emphasize the fact that ranking/scoring is a global learning problem. Indeed, in contrast to binary classification, where a decision rule may be immediately derived from a partition $\mathcal{P}$ of the feature space alone, through a majority-voting scheme, the bipartite ranking problem is of global nature. The local properties of the regression function on a given cell alone are useless, nearest neighbors rules make no sense for this problem and cells of $\mathcal{P}$ have to be compared to each other somehow, by means of the permutation $\sigma \in \mathfrak{S}_{\#\mathcal{P}}$ in the setup described above.

Now set:

$$\alpha(C) = \mathbb{P}\{X \in C \mid Y = -1\},$$
$$\beta(C) = \mathbb{P}\{X \in C \mid Y = +1\},$$

for any a measurable subset $C \subset \mathcal{X}$. The next proposition particularizes the ROC curve and the AUC for a piecewise constant scoring function. Its proof is straightforward and thus omitted.

Proposition 2 *Let $s(x)$ be a piecewise constant scoring function with $(\mathcal{P}, \sigma)$-representation $s(x) = \sum_{k=1}^{K} \lambda_k \cdot \mathbb{I}\{x \in C_{\sigma(k)}\}$.*

(i) *The* ROC *curve of the scoring function* $s(x)$ *is the broken line that connects the knots* $\{(\alpha_k(s), \beta_k(s)) : 0 \leq k \leq K\}$, *where*: $\forall k \in \{1, \dots, K\}$,

$$\alpha_k(s) = \sum_{l=1}^{k} \alpha(C_{\sigma(l)}) \quad \textit{and} \quad \beta_k(s) = \sum_{l=1}^{k} \beta(C_{\sigma(l)}),$$

and $\alpha_0(s) = \beta_0(s) = 0$ *by convention.*

(ii) *The* AUC *of the scoring function* $s(x)$ *is given by*:

$$\text{AUC}(s) = \frac{1}{2} \sum_{k=0}^{K-1} (\alpha_{k+1}(s) - \alpha_k(s)) \cdot (\beta_{k+1}(s) + \beta_k(s)) . \tag{4}$$

Optimal permutations. The next result describes the best scoring function in the AUC sense among all piecewise constant scoring functions that may be represented by means of a given partition $\mathcal{P}$. In order to state it precisely, further notations and definitions are needed.

Definition 4 (SUBPARTITION) Let $\mathcal{P}$ and $\mathcal{P}'$ be two partitions of $\mathcal{X}$. The partition $\mathcal{P}'$ is a *subpartition* of $\mathcal{P}$, when any cell $C' \in \mathcal{P}'$ may be written as a union of cells $C \in \mathcal{P}$. The following notation will be used: $\mathcal{P}' \subset \mathcal{P}$.

We denote by $\mathcal{S}_{\mathcal{P}}$ the set of scoring functions with a $(\mathcal{P}, \sigma)$-representation for some $\sigma \in \mathfrak{S}_{\#\mathcal{P}}$.

Theorem 1 (AUC OPTIMALITY, Clémençon and Vayatis 2009) *Consider a partition of* $\mathcal{X}$ *with* $K \geq 1$ *non empty cells*: $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$. *Let* $\sigma^*_{\mathcal{P}} \in \mathfrak{S}_K$ *such that*

$$\frac{\beta(C_{\sigma^*_{\mathcal{P}}(1)})}{\alpha(C_{\sigma^*_{\mathcal{P}}(1)})} \geq \dots \geq \frac{\beta(C_{\sigma^*_{\mathcal{P}}(K)})}{\alpha(C_{\sigma^*_{\mathcal{P}}(K)})}.$$

Then, $s^*_{\mathcal{P}}(x) = s_{\mathcal{P}, \sigma^*_{\mathcal{P}}}(x)$ *maximizes the* AUC *over* $\bigcup_{\mathcal{P}' \subset \mathcal{P}} \mathcal{S}_{\mathcal{P}'}$:

$$\text{AUC}(s^*_{\mathcal{P}}) = \max_{s \in \mathcal{S}_{\mathcal{P}'}, \mathcal{P}' \subset \mathcal{P}} \text{AUC}(s).$$

In the case where the cells are equivalent with respect to the false positive rate, i.e. $\forall k \in \{1, \dots, K\}$: $\alpha(C_k) = 1/K$, *we also have*

$$\forall \alpha \in [0, 1], \quad \text{ROC}(s, \alpha) \leq \text{ROC}(s^*_{\mathcal{P}}, \alpha),$$

for all $s \in \mathcal{S}_{\mathcal{P}'}$, $\mathcal{P}' \subset \mathcal{P}$. *The latter result also holds when cells are equivalent with respect to the true positive rate.*

It is noteworthy that $\sigma^*_{\mathcal{P}}$ corresponds to the permutation in $\mathfrak{S}_K$ which renders the piecewise linear curve ROC$(s_{\mathcal{P},\sigma}, .)$ concave.

On plug-in ranking rules. To any partition $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$ of $\mathcal{X}$ also correspond piecewise constant approximants of the regression function, which may serve as scoring functions. For instance, $\eta_{\mathcal{P}}(x) = \sum_{k=1}^{K} p\beta(C_k)/\mu(C_k) \cdot \mathbb{I}\{x \in C_k\}$ is the best approximant among functions that are constant on each cell C_k of the partition in the $L_2(\mu)$-sense, *i.e.* $\|\eta_{\mathcal{P}}(X) - \eta(X)\|^2_{L_2(\mu)} = \min_{s \in \mathcal{S}_{\mathcal{P}}} \mathbb{E}[(s(X) - \eta(X))^2]$. It follows from the fact that $\mu(C_k) = (1-p)\alpha(C_k) + p\beta(C_k)$ for all k that the *plug-in* scoring function $\eta_{\mathcal{P}}(x)$ yields the same ranking as $s^*_{\mathcal{P}}(x)$. Hence, as a scoring function, the approximant $\eta_{\mathcal{P}}(x)$ of the regression function is optimal in the AUC sense among all scoring rules in $\bigcup_{\mathcal{P}' \subset \mathcal{P}} \mathcal{S}_{\mathcal{P}'}$.

The next proposition relates the deficit of AUC for the scoring function $s^*_{\mathcal{P}}(x)$ to the $L_1(\mu)$-error of the corresponding plug-in estimator $\eta_{\mathcal{P}}(x)$ (see Corollary 9 in Clémençon and Vayatis 2009 for a similar result with different notations).

Proposition 3 *Assume that $\eta(X)$ has a continuous distribution. Then, for any partition $\mathcal{P} = \{C_k\}_{1 \le k \le K}$ of $\mathcal{X}$ with $K \ge 2$ non empty cells, we have*:

$$\mathrm{AUC}^* - \mathrm{AUC}(s^*_{\mathcal{P}}) \le \frac{\|\eta_{\mathcal{P}}(X) - \eta(X)\|_{L_1(\mu)}}{p(1-p)} + \frac{1}{4p(1-p)} \sum_{k=1}^{K} \mathcal{G}(C_k),$$

where, for all $k \in \{1, \dots, K\}$, $\mathcal{G}(C_k) = \mathbb{E}[|\eta(X) - \eta(X')| \cdot \mathbb{I}\{(X, X') \in C_k^2\}]$ denotes the Gini mean difference of $\eta(X)$ with the expectation restricted to the domain $\{(X, X') \in C_k \times C_k\}$.

Empirical ROC *curve and* AUC. From a practical perspective, the selection of a scoring function $s(x)$ is based on training data $\mathcal{D}_n = \{(X_i, Y_i);\ 1 \le i \le n\}$. The relevance of a candidate $s(x)$ is thus evaluated by plotting the empirical version of its ROC curve.

We set: $\forall i \in \{1, \dots, n\}$,

$$\hat{\alpha}_i(s) = \frac{1}{n_-} \sum_{j / Y_j = -1} \mathbb{I}\{s(X_j) \ge s(X_i)\},$$

$$\hat{\beta}_i(s) = \frac{1}{n_+} \sum_{j / Y_j = +1} \mathbb{I}\{s(X_j) \ge s(X_i)\},$$

where $n_+ = \sum_{i \le n} \mathbb{I}\{Y_i = +1\} = n - n_-$.

Let $\sigma \in S_n$ be such that $\hat{\alpha}_{\sigma(1)}(s) \le \dots \le \hat{\alpha}_{\sigma(n)}(s)$ and set $\hat{\alpha}_{\sigma(0)}(s) = \hat{\beta}_{\sigma(0)}(s) = 0$ by convention. The empirical ROC curve of $s(x)$ is the piecewise linear function given by: $\forall i \in \{1, \dots, n\}$, $\forall \alpha \in [\hat{\alpha}_{\sigma(i-1)}(s), \hat{\alpha}_{\sigma(i)}(s)[$,

$$\widehat{\mathrm{ROC}}(s, \alpha) = \frac{\hat{\beta}_{\sigma(i)}(s) - \hat{\beta}_{\sigma(i-1)}(s)}{\hat{\alpha}_{\sigma(i)}(s) - \hat{\alpha}_{\sigma(i-1)}(s)} \cdot \big(\alpha - \hat{\alpha}_{i-1}(s)\big) + \hat{\beta}_{i-1}(s).$$

By definition, the empirical AUC of $s(x)$ is the area under its empirical ROC curve:

$$\begin{aligned}
\widehat{\mathrm{AUC}}(s) &= \int_{\alpha=0}^{1} \widehat{\mathrm{ROC}}(s, \alpha) d\alpha \\
&= \frac{1}{n_+ n_-} \sum_{i / Y_i = +1} \sum_{j / Y_j = -1} \mathbb{I}\{s(X_i) > s(X_j)\} \\
&\quad + \frac{1}{2 n_+ n_-} \sum_{i / Y_i = +1} \sum_{j / Y_j = -1} \mathbb{I}\{s(X_i) = s(X_j)\},
\end{aligned}$$

the latter expression being the empirical version of the identity stated in Proposition 1.

All results established when considering true ROC curves extend to their empirical versions, replacing G, H and p by their counterparts calculated from the sample $\mathcal{D}_n$. In particular, given a partition $\mathcal{P} = \{C_k\}_{1\leq k\leq K}$ of the feature space $\mathcal{X}$, the ordering of the cells with maximum empirical AUC corresponds to permutations $\widehat{\sigma}^*$ such that,

$$\frac{\widehat{\beta}(C_{\widehat{\sigma}^*(1)})}{\widehat{\alpha}(C_{\widehat{\sigma}^*(1)})} \geq \cdots \geq \frac{\widehat{\beta}(C_{\widehat{\sigma}^*(K)})}{\widehat{\alpha}(C_{\widehat{\sigma}^*(K)})},$$

where for all measurable subset $C \subset \mathcal{X}$:

$$\widehat{\alpha}(C) = \frac{1}{n_-}\sum_{i=1}^{n} \mathbb{I}\{X_i \in C, Y_i = -1\},$$

$$\widehat{\beta}(C) = \frac{1}{n_+}\sum_{i=1}^{n} \mathbb{I}\{X_i \in C, Y_i = +1\},$$

which correspond respectively to the empirical false positive rate and the empirical true positive rate of a classifier predicting $+1$ on the set C.

It renders the empirical ROC curve concave and corresponds to the same ranking induced by the estimator of the regression function

$$\widehat{\eta}_{\mathcal{P}}(x) = \sum_{k=1}^{K} \frac{n_+\widehat{\beta}(C_k)}{n_-\widehat{\alpha}(C_k) + n_+\widehat{\beta}(C_k)} \cdot \mathbb{I}\{x \in C_k\},$$

meaning that $\widehat{\eta}_{\mathcal{P}} = \arg\max_{s\in\mathcal{S}_{\mathcal{P}}} \widehat{\mathrm{AUC}}(s)$.

Tree-structured ranking rules. The present article focuses on a specific family of piecewise constant scoring rules, those defined by *binary ranking trees* namely. Consider first a complete, left-right oriented, rooted binary tree $\mathcal{T}_D$, with finite depth $D \geq 1$. Every nonterminal node (d, k) of $\mathcal{T}_D$, with $d \in \{0, \ldots, D-1\}$ and $k \in \{0, \ldots, 2^d - 1\}$, corresponds to a subset $C_{d,k} \subset \mathcal{X}$ and has two descendants: a *left sibling* corresponding to a subset $C_{d+1,2k} \subset C_{d,k}$ and a *right sibling* associated to $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$, with $C_{0,0} = \mathcal{X}$ for the root node by convention. In the sequel, we call such a (complete) ranking tree a *master ranking tree*.

This way, any subtree $\mathcal{T} \subset \mathcal{T}_D$ acts as a ranking rule, by scanning its outer leaves from left to right. In particular, the resulting order corresponds to the one induced by the scoring function:

$$s_{\mathcal{T}}(x) = \sum_{(d,k):\ \text{terminal nodes of } \mathcal{T}} (2^D - 2^{D-d}k) \cdot \mathbb{I}\{x \in C_{d,k}\}.$$

The score $s_{\mathcal{T}}(x)$ may be computed in a top-down fashion, through a sequence of binary rules. At the root node, the score is initially set to 2^D and at each subsequent internal node (d, k) of $\mathcal{T}$, the current score remains unchanged if x moves to the left child, while one substracts $2^{D-(d+1)}$ to it if x moves to the right child.

2.3 The TreeRank approach

Assume that a training data set $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ of n independent samples of the pair (X, Y) is available. For notational convenience, we set $\alpha_{d,0} = \beta_{d,0} = 0$ and $\alpha_{d,2^d} = \beta_{d,2^d} = 1$ for all $d \geq 0$. We suppose that we are given a class $\mathcal{C}$ of subsets of $\mathcal{X}$,

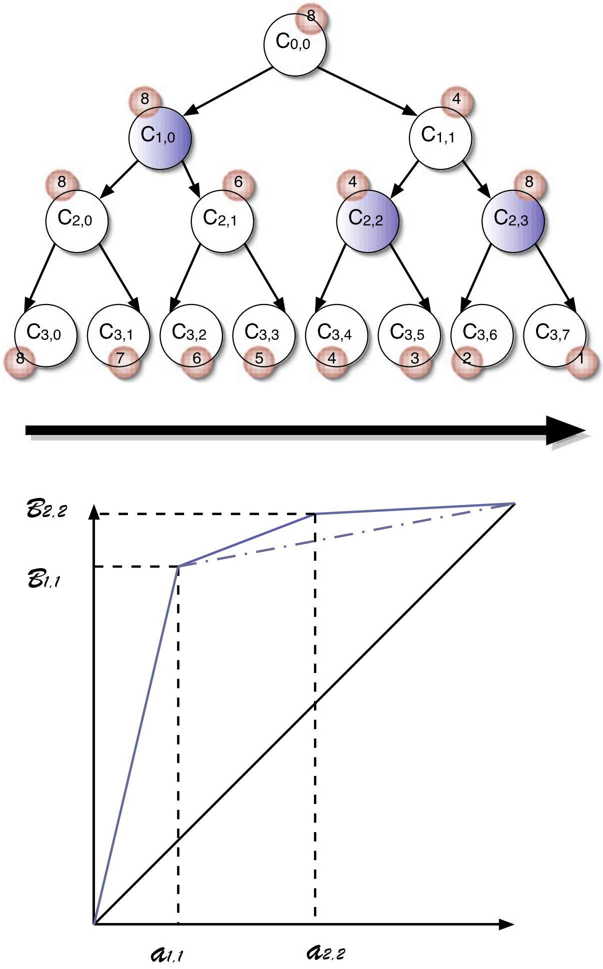


Fig. 1 (Color online) A tree-structured ranking rule (*top*) and the ROC curve of a subtree (*bottom*). A score (*red circles*) is assigned to each cell. The restriction of these values to the outer leaves of any subtree of the master ranking tree (*blue circles*) produces a scoring rule which orders the corresponding cells according to the left-right orientation

on which attainable partitions are based. Let $D \geq 1$ be fixed. We now recall the specific method called TreeRank which was proposed in Clémençon and Vayatis (2008), and further studied in Clémençon and Vayatis (2009), for adaptively generating a tree-structured partition of the feature space $\mathcal{X}$ in ordered cells $\{C_{D,k} \,:\, k = 0, \ldots, 2^D - 1\}$. Precisely, the piecewise constant scoring rule it outputs is described by a *master ranking tree*. Each one of the terminal leaves of the tree corresponds to a unique cell of the partition. The ordering of the cells is simply obtained by perusing the terminal leaves from the left to the right at the bottom of the tree (see Fig. 1). The pseudo-code is described in Fig. 2.

Remark 4 (On stopping rules) One may consider continuing to split the nodes until either the number of data points within a cell has reached a minimum number specified *a priori*, or else splitting yields no improvement in the empirical AUC sense. From a practical perspective, in both cases one then sets: $C_{d+1,2k} = C_{d,k}$ and $C_{d+1,2k+1} = \emptyset$.

Remark 5 (On concavity) We point out that, unless the collection $\mathcal{C}$ of subset candidates is *union stable* (*i.e.* $\forall (C, C') \in \mathcal{C}^2$, $C \cup C' \in \mathcal{C}$), the empirical curve $\widehat{\mathrm{ROC}}(s_D, .)$ output by

```
TreeRank Algorithm
1. (Initialization) Set $C_{0,0} = \mathcal{X}$.
2. (Iterations) For $d = 0, \dots, D-1$ and $k = 0, \dots, 2^d - 1$:
   (a) (Optimization step) Set the entropic measure:
           $\widehat{\Lambda}_{d,k+1}(C) = (\alpha_{d,k+1} - \alpha_{d,k})\hat{\beta}(C) - (\beta_{d,k+1} - \beta_{d,k})\hat{\alpha}(C)$.
       Find the best subset $C_{d+1,2k}$ of rectangle $C_{d,k}$ in the AUC sense:
           $C_{d+1,2k} = \mathop{\arg\max}_{C \in \mathcal{C},\ C \subset C_{d,k}} \widehat{\Lambda}_{d,k+1}(C)$.
       Then, set $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$.
   (b) (Update) Set
           $\alpha_{d+1,2k+1} = \alpha_{d,k} + \widehat{\alpha}(C_{d+1,2k})$,
           $\beta_{d+1,2k+1} = \beta_{d,k} + \widehat{\beta}(C_{d+1,2k})$,
       and
           $\alpha_{d+1,2k+2} = \alpha_{d,k+1}$,
           $\beta_{d+1,2k+2} = \beta_{d,k+1}$.
3. (Output) After $D$ iterations, get the piecewise constant scoring function:
       $s_D(x) = \sum_{k=0}^{2^D-1} (2^D - k)\, \mathbb{I}\{x \in C_{D,k}\}$,
   together with an estimate of the curve $\mathrm{ROC}(s_D, .)$, namely the broken line $\widehat{\mathrm{ROC}}(s_D, .)$ that connects the knots $\{(\alpha_{D,k}, \beta_{D,k}) : k = 0, \dots, 2^D\}$, and the following estimate of $\mathrm{AUC}(s_D)$:
       $\widehat{\mathrm{AUC}}(s_D) = \int_{\alpha=0}^{1} \widehat{\mathrm{ROC}}(s_D, \alpha)\, d\alpha = \frac{1}{2} + \frac{1}{2} \sum_{k=0}^{2^{D-1}-1} \widehat{\Lambda}_{D-1,k+1}(C_{D,2k})$.
```

Fig. 2 Pseudo-code for the TreeRank algorithm

TreeRank is not necessarily concave (see Proposition 21 in Clémençon and Vayatis 2009). If it is not, one should notice that the rankings induced by $s_D(x)$ and the plug-in estimator $\widehat{\eta}_{\mathcal{P}_D}(x)$ based on the partition $\mathcal{P}_D = \{C_{D,k} : 0 \leq k \leq 2^D - 1\}$ are not the same. If the cells $C_{d,k}$ are built by aggregating elementary subsets, such as cubes of a grid partition of the feature space $\mathcal{X}$ (see Sect. 3.2), then it is easy to see that concavity is guaranteed. However, when candidates are produced recursively by applying a simple splitting rule at each step to the current node, this property is generally not satisfied (see Sect. 3).

The TreeRank algorithm produces an empirical ROC curve that mimics the piecewise linear approximant of the optimal ROC curve obtained through an adaptive nonlinear partitioning scheme of the unit interval. We describe this approximation scheme below. We also refer to Sect. D in Clémençon and Vayatis (2009) for further details.

Adaptive piecewise linear approximation of ROC*. As initial approximant, we start with the main diagonal $\beta = \alpha$ of the ROC space corresponding the subdivision $\alpha^*_{0,0} = 0 < \alpha^*_{(0,1)} = 1$. At the next step, the approximation is refined by adding a point $\alpha^*_{1,1}$ between $\alpha^*_{1,0} = \alpha^*_{0,1}$ and $\alpha^*_{1,2} = \alpha^*_{0,1}$ in the meshgrid, in order to produce a broken line, connecting the knots $\{(\alpha^*_{1,k}, \mathrm{ROC}^*(\alpha^*_{1,k})) : k \in \{0, 1, 2\}\}$ with minimum L_1-distance to the target curve ROC*, or, equivalently, with maximum AUC. We point out that this is also the best interpolator with two linear pieces in terms of sup-norm, see Proposition 20 in Clémençon and Vayatis (2009) and additionally that the point $(\alpha^*_{1,1}, \mathrm{ROC}^*(\alpha^*_{1,1}))$ added to the meshgrid corresponds to the point of ROC* at which the tangent has the same slope as the straight line passing through $(\alpha^*_{0,0}, \mathrm{ROC}^*(\alpha^*_{0,0}))$ and $(\alpha^*_{0,1}, \mathrm{ROC}^*(\alpha^*_{0,1}))$. The procedure is then iterated: one adds a point $\alpha^*_{2,1}$ between $\alpha^*_{2,0} = \alpha^*_{1,0}$ and $\alpha^*_{2,2} = \alpha^*_{1,1}$ and another one, $\alpha^*_{2,3}$, between $\alpha^*_{2,2} = \alpha^*_{1,1}$ and $\alpha^*_{2,4} = \alpha^*_{1,2}$ in order to maximize the AUC of the interpolator thus obtained. At step D, a tree-structured subdivision $\alpha^*_{D,0} = 0 < \alpha^*_{D,1} < \cdots < \alpha^*_{D,2^D} = 1$ of the unit interval has then been produced, yielding a linear-by-parts interpolator with $2^D + 1$ pieces. The resulting curve may be viewed as the ROC curve of a scoring function, namely the piecewise constant function:

$$s^*_D(x) = \sum_{k=0}^{2^D-1} (2^D - k) \cdot \mathbb{I}\{x \in C^*_{D,k}\},$$

where the $C^*_{d,k}$'s are the specific bilevel sets of the regression function defined recursively by: $C^*_{0,0} = \mathcal{X}$ and $\forall d \geq 0$, $\Delta^*_{d,0} = 0$, $\Delta^*_{d,2^d} = 1$ and $\forall k \in \{0, \dots, 2^d\}$,

$$C^*_{d,k} = \{x \in \mathcal{X} : \Delta^*_{d,k+1} \leq \eta(x) < \Delta^*_{d,k}\},$$

where

$$\Delta^*_{d+1,2k+1} = \frac{p\beta(C^*_{d,k})}{\mu(C^*_{d,k})} \quad \text{and} \quad \Delta^*_{d+1,2k} = \Delta^*_{d,k}.$$

With the notations previously set out, we have $s^*_D(x) = s^*_{\mathcal{P}^*_D}(x)$ where $\mathcal{P}^*_D$ is the partition of the feature space given by:

$$\mathcal{P}^*_D = \{C^*_{D,k} : k = 0, \dots, 2^D - 1\}.$$

Like the subdivision $\{\alpha^*_{D,k} : k = 0, \dots, 2^D\}$ of the unit interval, this partition is obtained recursively through the procedure described above and is thus related to a tree-structure as well: $\forall d \geq 0$, $\forall k \in \{0, \dots, 2^d\}$, $C^*_{d,k}$ splits into $C^*_{d+1,2k}$ and $C^*_{d+1,2k+1}$. Hence, the TREERANK algorithm may be viewed as a statistical version of this recursive partitioning scheme, which adaptively search for a collection of $\eta(x)$'s bilevel sets in order to optimize the ROC curve. However, the *Optimization step*, which consists in splitting in a nearly optimal fashion each cell of the current partition based on labeled data lying in it, is not described in a specific manner. Indeed, the convergence rate analysis of TREERANK in Clémençon and Vayatis (2009) has been carried out under the assumption that the class $\mathcal{C}$ of cell candidates includes all the $C^*_{d,k}$'s. Therefore, it is very unlikely that simple rules, such as the one which consists in searching for the best *perpendicular split* at each step in the spirit of the original CART methodology, can produce cells close to the bilevel sets $C^*_{d,k}$, except in very specific cases (refer to Sect. VI of Clémençon and Vayatis 2009 for illustrative examples). It is the main goal of the subsequent analysis to specify possible flexible strategies for splitting regions of the feature space, in order to generate partitions $\mathcal{P}_D = \{C_{D,k} : k = 0, \dots, 2^D - 1\}$ close to the ideal partition $\mathcal{P}^*_D$.

3 Splitting for ranking

In this section, we focus on the practical implementation of the *Optimization step* of the TREERANK algorithm. We first set out the goals of the splitting rule from the perspective of AUC maximization and we underline the difference with the standard classification task. Eventually, the ranking splitting rule is interpreted as a *cost-sensitive* classification splitting rule with a data-dependent cost.

3.1 Binary scoring rule *vs.* classification rule

In the classification setup, partitioning techniques aim at splitting the feature space into two halves, ideally as $\{x \in \mathcal{X} : \eta(x) \geq 1/2\} \cup \{x \in \mathcal{X} : \eta(x) < 1/2\}$, by means of a majority voting scheme in each cell of the partition. It is noteworthy that, as a binary scoring function, the Bayes classifier $x \in \mathcal{X} \mapsto 2 \cdot \mathbb{I}\{\eta(x) \geq 1/2\} - 1$ is suboptimal regarding the AUC criterion, except in very specific cases, as shown by the next result.

Lemma 1 (OPTIMAL BINARY SCORING FUNCTIONS) *Let* $p = \mathbb{P}(Y = +1)$ *and consider the* (*binary*) *scoring function* $s_1^*(x) = 2 \cdot \mathbb{I}\{x \in C^*\} + \mathbb{I}\{x \in \mathcal{X} \setminus C^*\}$ *with* $C^* = \{\eta(x) \geq p\}$. *Let* $C \subset \mathcal{X}$ *be an arbitrary measurable subset and set* $s = 2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X} \setminus C}$. *We then have*:

$$\mathrm{AUC}(s) = \frac{1}{2} + \frac{1}{2}(\beta(C) - \alpha(C)) \leq \mathrm{AUC}(s_1^*). \tag{5}$$

More precisely, *the following identity holds*:

$$\mathrm{AUC}(s_1^*) - \mathrm{AUC}(s) = \frac{1}{2p(1-p)} \cdot \mathbb{E}[|\eta(X) - p| \cdot \mathbb{I}\{X \in C^* \Delta C\}], \tag{6}$$

where Δ *denotes the symmetric difference between sets.*

In addition, *we have*

$$\mathrm{AUC}(s_1^*) = \frac{1}{2p(1-p)} \mathbb{E}[\max\{(1-p)\eta(X), p(1-\eta(X))\}]. \tag{7}$$

This result shows that, unless the two sets $\{\eta \geq 1/2\}$ and $\{\eta \geq p\}$ coincide up to a μ-negligible set, the AUC of the Bayes classifier is strictly smaller than $\mathrm{AUC}(s_1^*)$. In addition, when the optimal ROC curve is differentiable and strictly concave (see Sect. 2.1 above), the ROC curve of the Bayes classifier is determined by the knot $(\alpha, \mathrm{ROC}^*(\alpha))$, where ROC^* has a tangent with slope $(1-p)/p$, whereas $\mathrm{ROC}(s_1^*)$ is the broken line defined by the point of ROC^* where the tangent has a slope equal to 1, see Fig. 3. We point out that, under the set of assumptions listed in Sect. 2.1, $(1-p)/p$ always belongs to $[\mathrm{ROC}^{*\prime}(1), \mathrm{ROC}^{*\prime}(0)]$, since this condition amounts to saying that p lies between the essential infimum and supremum of $\eta(X)$ and we have $\mathbb{E}[\eta(X)] = p$. Refer to Remark 5 of Sect. C in Clémençon and Vayatis (2009) for further details.

Bipartite ranking as a collection of imbricated binary scoring problems. We propose data-driven procedures for constructing a binary scoring function with AUC close to $\mathrm{AUC}(s_1^*)$. When running the TREERANK algorithm, such a procedure will be iteratively applied, in a "fractal" manner, to the subsample lying in each cell C of the current tree-structured partition. Indeed, let us introduce the conditional AUC restricted to the cell C:

$$\begin{aligned}\mathrm{AUC}(s \mid C) = {} & \mathbb{P}\{s(X) > s(X') \mid (Y, Y') = (+1, -1), (X, X') \in C\} \\ & + \frac{1}{2}\mathbb{P}\{s(X) = s(X') \mid (Y, Y') = (+1, -1), (X, X') \in C\}.\end{aligned}$$

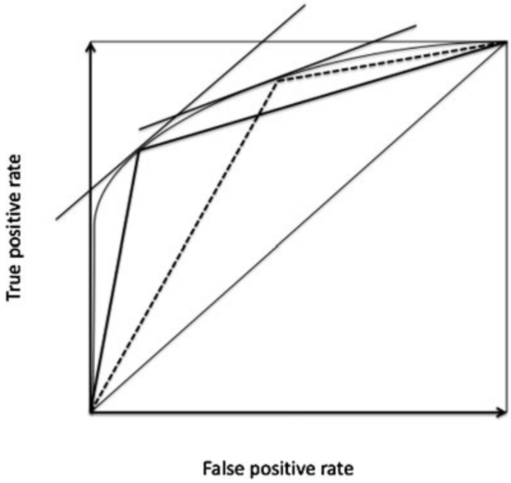


Fig. 3 ROC curves: optimal binary scoring function (*solid broken line*) *vs.* Bayes classifier (*dotted broken line*) in a situation where $p > 1/2$

It suffices to observe that, conditioned upon the event $X \in C$, the AUC of the scoring function $s = 2 \cdot \mathbb{I}_{C'} + \mathbb{I}_{C \setminus C'}$ where $C' \subset C$ is given by:

$$\begin{aligned}\mathrm{AUC}(s \mid C) &= \frac{1}{2} + \frac{1}{2}\left(\frac{\beta(C')}{\beta(C)} - \frac{\alpha(C')}{\alpha(C)}\right)\\ &= \frac{1}{2}\left(1 + \frac{\alpha(C)\beta(C') - \beta(C)\alpha(C')}{\alpha(C)\beta(C)}\right).\end{aligned}$$

This observation illustrates that, within the TreeRank approach, bipartite ranking boils down to solving a collection of "nested" binary scoring problems, in contrast to the RankBoost method developped by Freund et al. (2003), which consists of combining binary scoring rules in an additive fashion.

3.2 Partition-based splitting rule

We now describe a simple strategy for building a nearly optimal binary scoring function based on a partition of the feature space specified *a priori*.

As shown by the next result, the procedure described in Fig. 4 determines the binary scoring function, constant on each cell of the initial partition $\mathcal{P} = \{C_k\}_{1 \le k \le K}$, that has maximum empirical AUC.

Proposition 4 *Let $\mathcal{P} = \{C_k\}_{1 \le k \le K}$ be a partition of the space $\mathcal{X}$ and denote by $\widehat{s}^*(x) = 2 \cdot \mathbb{I}\{x \in L\} + \mathbb{I}\{x \in R\}$ the scoring function determined by the partition-based splitting rule based on $\mathcal{P}$ and the sampling data $\mathcal{D}_n$. Then, for any binary scoring rule $s = 2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X} \setminus C}$, where the subset $C \subset \mathcal{X}$ is formed by a union of cells in $\mathcal{P}$, we have*:

$$\widehat{\mathrm{AUC}}(s) \le \widehat{\mathrm{AUC}}(\widehat{s}^*).$$

This proposition simply results from Theorem 1 applied to the empirical distribution of the (X_i, Y_i)'s, the details are omitted.

```
PARTITION-BASED SPLITTING RULE
1. (INPUT) Cell: $C \subset \mathcal{X}$
   Data points in $C$: $\mathcal{D}_n(C) = \{(X_i, Y_i) : 1 \le i \le n, X_i \in C\}$
   Partition of $C$: $\{C_1, \ldots, C_K\}$ with $K \ge 1$.
2. ("CONCAVIFICATION" STEP) Compute $\sigma \in \mathfrak{S}_K$ such that:
       $\frac{\widehat{\beta}(C_{\sigma(1)})}{\widehat{\alpha}(C_{\sigma(1)})} \ge \cdots \ge \frac{\widehat{\beta}(C_{\sigma(K)})}{\widehat{\alpha}(C_{\sigma(K)})}$,
   where $\widehat{\alpha}(.)$ and $\widehat{\beta}(.)$ denote the empirical false and true positive rates based on the sample $\mathcal{D}_n(C)$.
3. (MERGING STEP) $\forall k \in \{1, \ldots, K\}$, set $L_k = \bigcup_{l \le k} C_{\sigma(l)}$ and compute the entropic measure $\widehat{\Lambda}(k) = \widehat{\beta}(L_k) - \widehat{\alpha}(L_k)$. Let
       $k^* = \arg\max_{1 \le k \le K} \{\widehat{\Lambda}(k)\}$.
4. (OUTPUT) Form the leaves:
       $L = L_{k^*}$ and $R = C \setminus L$.
```

Fig. 4 Pseudo-code for the partition-based splitting rule

Uniform partitions. Assume, for simplicity, that $\mathcal{X} = [0, 1]^q$ and consider subpartitions of the partition $\mathcal{P}(j)$ made of dyadic cubes of side length 2^{-j}, *i.e.* of subsets of the form $\prod_{l=1}^{q} [k_l/2^j, (k_l + 1)/2^j$ [where $0 \le k_l < 2^j$ for all $l \in \{1, \ldots, q\}$. Note that the partition has cardinality $\#\mathcal{P}(j) = 2^{jq}$. We denote by $\widehat{L}_j = L$ the output of the partition-based splitting rule from $\mathcal{P}(j)$ and by $\widehat{s}_j^*(x) = 2 \cdot \mathbb{I}\{x \in \widehat{L}_j\} + \mathbb{I}\{x \in \widehat{R}_j\}$ the related binary scoring function. It is reasonable to expect that the level set $\{x \in \mathcal{X} : \eta(x) \ge p\}$ may be accurately estimated from a collection of such cubes when the boundary of the set is sufficiently smooth and the sidelength 2^{-j} is chosen small enough. This is formalized by the next result.

Theorem 2 (DYADIC SPLITTING RULE) *For all $j \ge 1$, denote by $\mathcal{P}_{2,j}$ the collection of partitions of $\mathcal{X}$ made of two non empty sets, obtained as unions of dyadic cubes of side length 2^{-j}. Suppose that $p \in [\underline{p}, \bar{p}]$ with $0 < \underline{p} < \bar{p} < 1$. There exists a constant $c < \infty$ depending on $\underline{p}$ and $\bar{p}$ such that for all $\delta \in (0, 1)$, we have with probability at least $1 - \delta$: for $n \ge 1$ large enough and for all $j \ge 1$,*

$$\mathrm{AUC}(s_1^*) - \mathrm{AUC}(\widehat{s}_j^*) \le c \cdot \frac{2^{jq}}{\sqrt{n}} + \left\{ \mathrm{AUC}(s_1^*) - \max_{s \in \mathcal{S}_{\mathcal{P}_{2,j}}} \mathrm{AUC}(s) \right\}. \tag{8}$$

Remark 6 (BIAS, SMOOTHNESS ASSUMPTIONS AND MODEL SELECTION) Under smoothness assumptions on the level set $C^* = \{x \in \mathcal{X} : \eta(x) \ge p\}$, it is possible to control the bias term. Indeed, in the case where μ has a bounded density with respect to Lebesgue measure λ on $\mathbb{R}^d$, by virtue of Lemma 1, we have:

$$\mathrm{AUC}(s_1^*) - \mathrm{AUC}(s) \le \frac{\|d\mu/dx\|_\infty}{2p(1-p)} \cdot \lambda(C^* \Delta C),$$

for any $s = 2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X} \setminus C}$ with $C \in \mathcal{P}_{2,j}$. When the boundary ∂C^* is of finite perimeter $per(\partial C^*) < \infty$ (which is the case if $\eta(x)$ is of bounded variation, the boundary being then

$\partial C^* = \{x \in \mathcal{X} : \eta(x) = p\}$ by virtue of the continuity of η), the bias term is bounded by $\min_{C \in \mathcal{P}_{2,j}} \lambda(C^* \Delta C) \leq c \cdot per(\partial C^*) 2^{-jq}$, for some constant $c < \infty$, see Proposition 9.7 in Mallat (1990). Then, choosing the level of resolution $j = j(n)$ so that $2^{j(n)} \sim n^{1/(4q)}$ as $n \to \infty$ yields a rate bound of order $n^{-1/4}$ in (8). Faster generalization bounds may be established under more restrictive assumptions involving a regularity parameter θ of ∂C^*, such as its *box dimension*. Although the optimal choice for j would then depend on θ, a standard fashion of nearly achieving the optimal rate of convergence is to perform model selection, adding an adequate penalty term to the empirical AUC criterion, see Clémençon and Vayatis (2009).

Remark 7 (ON FASTER RATES OF CONVERGENCE) As in the classification setting, faster rates of convergence may be attained. The difference lies here in that the complexity of the problem is related to the behavior of $\eta(x)$ in the vicinity of p (instead of 1/2). Under the following extension of Massart's noise condition, stipulating that there exists some constant $c > 0$ such that, almost surely,

$$|\eta(X) - p| \geq c,$$

a rate bound of order $O(n^{-1})$ can be obtained using concentration results involving the variance of the AUC deficit. The proof can be derived as in the classification setup. We point out that this condition is incompatible with the regularity conditions for the curve ROC* listed in Sect. 2.1, insofar as it entails that G^* and H^* both jump at p. It is possible to weaken the condition by considering a modified version of Tsybakov's noise condition:

$$\mathbb{P}\{|\eta(X) - p| \leq t\} \leq M \cdot t^{\frac{a}{1-a}}$$

for some $a \in [0, 1]$. Following the argument in Tsybakov (2004), this leads to a rate of order $n^{1/(2-a)}$. Observe that this condition may be rewritten as:

$$F^*(p + t) - F^*(p - t) \leq M \cdot t^{\frac{a}{1-a}},$$

where $F^* = pG^* + (1 - p)H^*$ denotes the cumulative distribution function of $\eta(X)$. Therefore, if it is assumed that G^* and H^* are differentiable with bounded derivatives and $H^{*\prime} > 0$, one necessarily has $a = 1/2$ and gets a rate bound of order $n^{-2/3}$.

Remark 8 (A UNION STABLE COLLECTION OF CANDIDATES) By construction, the collection $\mathcal{P}_{2,j}$ is union stable. Hence, in the case where the *Optimization step* is implemented by means of the partition-based splitting rule from the $\mathcal{P}_{2,j}$, the empirical ROC curve $\widehat{\mathrm{ROC}}(s_D, .)$ output by TREERANK is concave and s_D yields the same ranking of the $C_{D,k}$'s as the plug-in scoring rule $\widehat{\eta}_{\mathcal{P}_D}$, see Remark 5.

The LEAFRANK *splitting algorithm.* As soon as the dimension q of the feature space $\mathcal{X}$ is large, one faces significant computational problems when using uniform partitions. In this case, the partition on which the split is based should naturally be chosen depending on the data. The strategy we propose is to start with the partition adaptively generated by TREERANK based on a simple splitting criterion and then implement the splitting rule described above (see Fig. 5).

Even though the implementation of TREERANK is implemented from a naive splitting rule such as the one based on perpendicular splits, one may expect that the partition produced is sufficiently rich to form a good approximant of the set $\{x \in \mathcal{X} : \eta(x) \geq p\}$ by the union of

```
LEAFRANK ALGORITHM

1. (INPUT) Cell: $C \subset \mathcal{X}$
   Data points in $C$: $\mathcal{D}_n(C) = \{(X_i, Y_i) : 1 \leq i \leq n, X_i \in C\}$
   Depth $d \geq 1$.
2. (GROWING STEP) Run TREERANK with a naive splitting rule at depth $d$, yielding a
   ranking tree with terminal leaves:
       $C_{d,k}, \quad k = 0, \dots, 2^d - 1.$
3. (PARTITION-BASED SPLITTING RULE) Apply the partition-based splitting rule from
   the partition $\mathcal{P}_d = \{C_{d,k} : 0 \leq k < 2^d\}$. Find $\sigma$ and $k^*$ as before.
4. (OUTPUT) Form the leaves:
       $L = L_{k^*} \quad \text{and} \quad R = C \setminus L.$
```

Fig. 5 Pseudo-code for the LEAFRANK algorithm

certain cells, if the depth d is chosen large enough. Alike the resolution level j for dyadic partitions, the parameter d rules the complexity of the splitting rule. The subsequent analysis provides a remarkable interpretation of this procedure.

3.3 A cost-sensitive classification problem with data-dependent cost

Here we show that the *Optimization step* of the TREERANK algorithm may be interpreted as a 'weighted' or 'cost-sensitive' classification problem, where the cost depends on the data lying in the node to split, through the local empirical rate of positive instances.

Following in the footsteps of Clémençon and Vayatis (2008), the level set $\{\eta(x) \geq p\}$ may be viewed as the solution of a *weighted classification problem*. Define the weighted classification error:

$$\mathcal{L}_\omega(C) = 2p(1-\omega)\,(1-\beta(C)) + 2(1-p)\omega\,\alpha(C),$$

with $\omega \in (0, 1)$ being the asymmetry factor. Its empirical counterpart is given by:

$$\widehat{L}_\omega(C) = \frac{2\omega}{n}\sum_{i=1}^{n}\mathbb{I}\{Y_i = -1, X_i \in C\} + \frac{2(1-\omega)}{n}\sum_{i=1}^{n}\mathbb{I}\{Y_i = +1, X_i \notin C\}.$$

Proposition 5 (Clémençon and Vayatis 2008) *The optimal set for this error measure is* $C_\omega^* = \{x \ : \ \eta(x) > \omega\}$. *We have indeed, for all* $C \subset \mathcal{X}$:

$$\mathcal{L}_\omega(C_\omega^*) \leq \mathcal{L}_\omega(C).$$

More precisely, the excess risk for an arbitrary set C *can be written*:

$$\mathcal{L}_\omega(C) - \mathcal{L}_\omega(C_\omega^*) = 2\mathbb{E}\left[\mid \eta(X) - \omega \mid \cdot \mathbb{I}\{X \in C \Delta C_\omega^*\}\right].$$

The optimal error is given by:

$$\mathcal{L}_\omega(C_\omega^*) = 2\mathbb{E}[\min\{\omega(1-\eta(X)), (1-\omega)\eta(X)\}].$$

```
WEIGHTED ERM ALGORITHM

1. (INPUT) Cell: $C \subset \mathcal{X}$
   Data points in $C$: $\mathcal{D}_n(C) = \{(X_i, Y_i) : 1 \le i \le n, X_i \in C\}$
   Class $\mathcal{C}$ of candidate subsets.
2. (ASYMMETRY FACTOR) Compute:
    i. The number of positive instances in $C$: $n_+ = \sum_{i=1}^{n} \mathbb{I}\{X_i \in C, Y_i = +1\}$
   ii. The total number of instances in $C$: $n_C = \sum_{i=1}^{n} \mathbb{I}\{X_i \in C\}$.
   Take $\omega = n_+/n_C$ as the asymmetry factor.
3. (WEIGHTED ERM) Compute the weighted empirical risk minimizer:
   $$L = \arg\min_{\widetilde{C} \in \mathcal{C}} \widehat{\mathcal{L}}_\omega(\widetilde{C})$$
   and set $R = C \setminus L$.
```

Fig. 6 Pseudo-code for the WEIGHTED ERM algorithm

As shown by the Proposition above, when choosing $\omega = p$, the optimal set is given by $C^* = \{x \in \mathcal{X} : \eta(x) \ge p\}$. In addition, we point out that, in this case, the weighted classification error may be expressed as:

$$\mathcal{L}_p(C) = 4p(1-p)\{1 - \mathrm{AUC}(s)\}, \tag{9}$$

where $s(x) = 2 \cdot \mathbb{I}\{x \in C\} + \mathbb{I}\{x \in \mathcal{X} \setminus C\}$.

As the theoretical proportion of positive instances within the sample is unknown, an empirical counterpart of the weighted classification error $\mathcal{L}_p(C)$ can be obtained by replacing p by $\hat{p} = n_+/n$:

$$\widehat{\mathcal{L}}_{\hat{p}}(C) = 4\hat{p}(1-\hat{p})\left\{1 - \widehat{\mathrm{AUC}}(s)\right\}.$$

This leads to consider the *weighted empirical risk minimizer* over a class $\mathcal{C}$ of candidate sets, or equivalently the empirical AUC maximizer over the corresponding set of binary scoring functions $\{2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X} \setminus C} : C \in \mathcal{C}\}$ (see Fig. 6).

The interpretation of the splitting issue for the purpose of AUC maximization as a cost-sensitive classification problem sheds some light on possible ways of performing the *Optimization step*. Indeed, from any binary classification algorithm a practical splitting rule for empirical AUC maximization may be straightforwardly derived. In particular, when using the LEAFRANK routine with perpendicular splits for performing the *Optimization step*, the TREERANK algorithm may then be viewed as a recursive implementation of the weighted CART growing procedure (see Fig. 7), in which the weight is locally updated at each iteration, chosen as the rate of positive instances within the cell to split. This AUC splitting procedure could be refined by applying a pruning procedure to the classification tree obtained, see Breiman et al. (1984) or Nobel (2002) for instance.

In Sect. 2.3, we mentioned that the heuristics underlying the TREERANK algorithm rely on the recursive construction of estimates of bi-level sets of the regression function with the very levels being adaptively chosen through the approximation scheme applied to the ROC curve. Indeed, it is known from Clémençon and Vayatis (2010) that the performance of bipartite ranking methods crucially depends on their ability to capture the geometry of level set boundaries of the regression function η. For instance, scoring methods deriving

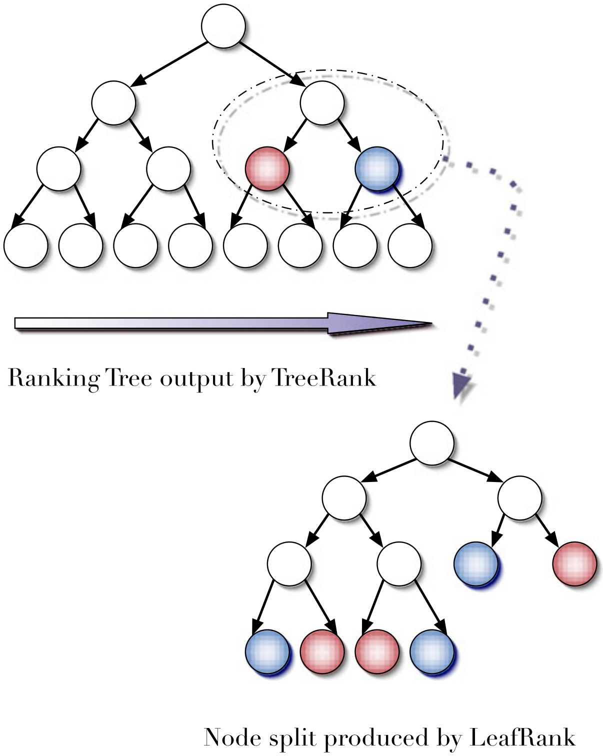

Ranking Tree output by TreeRank
Node split produced by LeafRank

Fig. 7 Schematic of the TREERANK algorithm, where each split is obtained through the LEAFRANK procedure: a naive TREERANK implementation, followed by concavification and merging steps. The final ranking is read at the *bottom* of the tree from the *left* to the *right*

from linear logistic regression or linear discriminant analysis only account for decision sets made of shifted affine subspaces. From a practical perspective, and in the absence of prior knowledge of the form of these level sets, other strategies should be developed. We claim that the TREERANK approach provides a powerful tool for the approximation of complex level sets as it relies on the approximation/estimation of a finite collection of level sets which are adaptively selected through the procedure and the fine choice of the classification method used within the LEAFRANK splitting rule. With the analysis of the splitting rule depicted above, the potential, in terms of approximation capacity, of the tree-based ranking rules we propose increases dramatically. Indeed, this interpretation of the splitting rule conveys a great amount of flexibility to the method since any classification algorithm could be used for the *Optimization step* and it makes possible to try various splitting strategies in order to determine which one is the most adapted to the data at hand.

4 Merging the cells—how to prune a ranking tree

Based on a training dataset $\mathcal{D}_n$, the TREERANK procedure with fixed depth D allows for growing a *master ranking tree* $\mathcal{T} = \mathcal{T}_n$ with $2^{D+1} - 1$ nodes, *i.e.* a binary tree, left-right oriented and whose terminal leaves correspond to the cells of a partition $\mathcal{P}(\mathcal{T}_n)$ of the feature space $\mathcal{X}$, ordered according to $\mathcal{T}_n$'s orientation. The complexity of the resulting ranking rule may be naturally described by the number of cells of the partition $\mathcal{P}(\mathcal{T})$ which is equal to 2^D. If the depth D is chosen too small, the ROC curve associated to the ranking tree produced will not permit to mimic the variability of the optimal curve ROC*, while if it is too large, the ranking tree produced may clearly overfit the data. It is the purpose of this section to investigate possible ways of optimally choosing the size of the ranking tree. From a practical perspective, the design of the ranking tree is done in two steps, as for binary classification (Nobel 2002). One first grows a large ranking tree $\mathcal{T}$ in a "greedy" fashion, and then, using a *cost-complexity pruning scheme*, one selects a certain (tree-structured) ordered subpartition of $\mathcal{P}(\mathcal{T}) = \{C_{D,k}, 0 \le k < 2^D\}$ by the means of a 'bottom-up' search strategy through the tree-structure $\mathcal{T}$ on which the $C_{d,k}$'s are aligned. One naturally hopes that the expected AUC of the resulting scoring function is larger than the one of $s_D(x)$.

In the following subsections, we propose two approaches for pruning a ranking tree. In order to describe them precisely, we introduce further definitions and notations. For $0 \le d \le D$ and $0 \le k < 2^D$, to each cell $C_{d,k}$, one assigns a scalar weight $\omega(C_{d,k})$ in a way that the following constraints are both satisfied.

(i) (KEEP-OR-KILL) For all $d \in \{0, \dots, D\}$ and $k \in \{0, \dots, 2^D - 1\}$, the weight $\omega(C_{d,k})$ belongs to $\{0, 1\}$.
(ii) (HEREDITY) If $\omega(C_{d,k}) = 1$, then for each cell $C_{d',k'}$ such that $C_{d,k} \subset C_{d',k'}$, we have $\omega(C_{d',k'}) = 1$.

Any collection of weights ω obeying these two constraints will be said *admissible* and determines the nodes of a subtree $\mathcal{T}(\omega)$ of the original tree $\mathcal{T}$. A cell $C_{d,k}$ is said *terminal* when $\omega(C_{d,k}) = 1$ and $\omega(C_{d',k'}) = 0$ for any cell $C_{d',k'} \subset C_{d,k}$. Terminal cells correspond to the outer leaves of the tree $\mathcal{T}(\omega)$ and form a partition $\mathcal{P}(\mathcal{T}(\omega))$ of the feature space $\mathcal{X}$. Given two admissible sequences of weights ω_1 and ω_2, $\mathcal{P}(\mathcal{T}(\omega_1))$ is a subpartition of $\mathcal{P}(\mathcal{T}(\omega_2))$, see Definition 4, if and only if $\{C_{d,k} : \omega_1(C_{d,k}) = 0\} \subset \{C_{d,k} : \omega_2(C_{d,k}) = 0\}$, one will then write $\mathcal{T}(\omega_1) \subseteq \mathcal{T}(\omega_2)$. The pruning stage consists of selecting those terminal leaves, *i.e.* an admissible collection of weights ω, and of building the scoring function (*cf.* Sect. 2.2)

$$s_\omega(x) = \sum_{C_{d,k} \in \mathcal{P}(\mathcal{T}(\omega))} (2^D - 2^{D-d}k) \cdot \mathbb{I}\{x \in C_{d,k}\}. \tag{10}$$

Indeed one may check that the ordering defined by s_ω coincides with the one determined by the tree $\mathcal{T}(\omega)$ when left-right oriented, see Fig. 8. In the ideal case where the class distributions G and H are known, the best sub-ranking tree in the AUC sense is described by

$$\omega^* = \arg\max_\omega \mathrm{AUC}(s_\omega), \tag{11}$$

where the maximum is taken over all admissible collections of weights ω. Of course, the class distributions are not available in practice and one must replace $\mathrm{AUC}(s_\omega)$ by an estimate

$$\begin{aligned} \widehat{\mathrm{AUC}}'(s_\omega) = {} & \frac{1}{n'_+ n'_-} \sum_{i:Y_i=+1} \sum_{j:Y_j=-1} \mathbb{I}\{s_\omega(X_i) > s_\omega(X_j)\} \\ & + \frac{1}{2} \frac{1}{n'_+ n'_-} \sum_{i:Y_i=+1} \sum_{j:Y_j=-1} \mathbb{I}\{s_\omega(X_i) = s_\omega(X_j)\}, \end{aligned} \tag{12}$$

based on a dataset $\mathcal{D}'_{n'} = \{(X'_1, Y'_1), \dots, (X'_{n'}, Y'_{n'})\}$ formed of i.i.d. copies of the pair (X, Y), where $n'_+ = \sum_{i=1}^n \mathbb{I}\{Y'_i = +1\} = n' - n_-$. Ideally, $\mathcal{D}'_{n'}$ should be chosen independent from the training dataset $\mathcal{D}_n$ used for growing the ranking tree $\mathcal{T}$. If one takes the same dataset for both the growing and pruning procedures, the estimator (12) will then naturally tend to overestimate the ranking performance of the largest ranking trees and it is very likely that

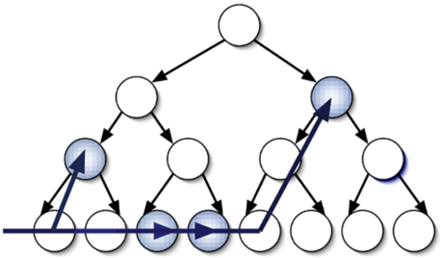


Fig. 8 (Color online) A pruned ranking tree: terminal nodes are in *blue*, top ranked cells are closest to the *bottom left* corner of the tree

one will obtain $\mathcal{T}(\omega^*) = \mathcal{T}$. However, in many applications, there is an insufficient amount of data to split it into two large enough separated subsets and all available data are used in the training stage. We next propose two approaches for model selection in this situation.

4.1 A cross-validation based procedure

We start off by adapting the pruning method proposed by Breiman et al. (1984) for the original CART algorithm in the classification setup in order to prune ranking trees. The idea is to add to the optimistic training performance estimate $\widehat{\mathrm{AUC}}(s_\omega)$ a *linear* complexity term that penalizes large ranking trees. Thus, with

$$\widehat{\mathrm{CPAUC}}(s_\omega, \lambda) = \widehat{\mathrm{AUC}}(s_\omega) - \lambda \cdot \#\mathcal{P}(\mathcal{T}(\omega)), \tag{13}$$

where $\lambda \geq 0$ is a tuning parameter governing the trade-off between training performance *vs.* model complexity, one seeks the subtree achieving the maximal complexity-penalized empirical AUC:

$$\omega^*_\lambda = \arg\max_\omega \widehat{\mathrm{CPAUC}}(s_\omega, \lambda).$$

It remains to choose λ and we now discuss this issue. The next theorem first shows that there exists a finite nested sequence of sub-ranking trees of the original ranking tree $\mathcal{T}$ containing all $\mathcal{T}(\omega^*_\lambda)$, $\lambda \geq 0$.

Theorem 3 *For a given ranking tree $\mathcal{T}$, there exists a finite increasing sequence of constants $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_m = \infty$ such that*

$$root = \mathcal{T}(\omega^*_{\lambda_m}) \subseteq \cdots \subseteq \mathcal{T}(\omega^*_{\lambda_1}) \subseteq \mathcal{T}(\omega^*_{\lambda_0}) = \mathcal{T},$$

and: $\forall j \in \{1, \ldots, m\}$, $\forall \lambda \in [\lambda_{j-1}, \lambda_j[$,

$$\mathcal{T}(\omega^*_\lambda) = \mathcal{T}(\omega^*_{\lambda_j}).$$

The proof is omitted since it is entirely similar to the one of Theorem 3.10 in Breiman et al. (1984), see also Ripley (1996).

In order to compute the $\mathcal{T}(\omega^*_\lambda)$'s, it suffices to successively collapse the internal node that produces the smallest per-node decrease in terms of empirical AUC and continue until the root is obtained. Estimation of $\lambda \in \{\lambda_j\}_{0\leq j\leq m}$ is achieved by N-fold cross validation: one picks the value $\widehat{\lambda}$ that maximizes the cross-validated AUC. The selected ranking tree is then $\mathcal{T}(\omega^*_{\widehat{\lambda}})$.

4.2 Complexity regularization—structural AUC maximization

Nonparametric model selection procedures have been successfully developed in the statistical learning setup for binary classification, see Massart (2006), Nobel (2002) or Boucheron et al. (2005). In addition to the pruning method described in the preceding subsection, we also propose a similar strategy for selecting a sub- ranking tree $\mathcal{T}(\omega)$ in a data-driven fashion and with largest possible AUC. Here the pruning scheme consists of maximizing:

$$\widetilde{\mathrm{CPAUC}}(s_\omega) = \widehat{\mathrm{AUC}}(s_\omega) - pen(\#\mathcal{P}(\mathcal{T}(\omega)), n),$$

where $pen(K, n)$ is a fixed and explicit penalty term, so that no resampling or cross-validation is required by the selection procedure. We set $\widetilde{s}_n^* = s_{\mathcal{P}(\mathcal{T}(\widetilde{\omega}_n^*))}$ with

$$\widetilde{\omega}_n^* = \underset{\omega \text{ admissible}}{\arg\max}\ \widetilde{\mathrm{CPAUC}}(s_\omega).$$

Classically, the key to an adequate choice for the penalty term lies in establishing a distribution-free bound for the quantity:

$$\mathbb{E}\left[\sup_{\omega:\#\mathcal{P}(\mathcal{T}(\omega))=K} |\widehat{\mathrm{AUC}}(s_\omega) - \mathrm{AUC}(s_\omega)|\right],$$

with $K \in \{1, \ldots, 2^D\}$, see Proposition 6 in Appendix A.5. As shown in Clémençon et al. (2008) (see also Clémençon et al. 2005), bounds for the uniform deviation between the AUC and its empirical counterpart over a collection of scoring functions can be proved by noticing that the empirical AUC may be expressed as a U-statistic (up to a multiplicative factor) and applying results of the theory of U-processes.

In the subsequent analysis, we consider two situations, corresponding to distinct ways of performing the *Optimization step* in the growing stage among those mentioned in Sect. 3 and yielding different, nonlinear this time, penalties for model selection.

O$_1$: Splits are obtained through the LeafRank procedure with at most κ perpendicular cuts, $\kappa \geq 1$.

O$_2$: The feature space is $\mathcal{X} = [0, 1]^q$ and splits are obtained through the partition-based rule from the collection of dyadic cubes $\prod_{m=1}^q [k_m 2^{-J}, (k_m + 1)2^{-J})$ with $0 \leq k_m < 2^J$ for all $m \in \{1, \ldots, q\}$.

The following proposition describes the performance of the scoring rule $\widetilde{S}_n^*$ based on structural AUC maximization in each of these situations.

Proposition 6 (Oracle inequalities) *Suppose that the proportion p belongs to an interval $[\underline{p}, \bar{p}]$ with $0 < \underline{p} < \bar{p} < 1$ and for all $K \in \{1, \ldots, 2^D\}$ and $n \geq 1$ the penalty term is picked as follows, depending on the strategy chosen for performing the Optimization step.*

(i) *If splits are optimized using the* **O**$_1$ *rule, then set*: $\forall (K, \kappa) \in \mathbb{N}^{*2}$,

$$pen(K, n) = \frac{1}{\underline{p}(1-\bar{p})}\sqrt{32\frac{\log(16((n+1)q)^{2K\kappa}) + K}{n}}.$$

(ii) *If splits are optimized using the* $\mathbf{O}_2$ *rule, then set*: $\forall(K,J)\in\mathbb{N}^{*2}$,

$$pen(K,n)=\frac{1}{\underline{p}(1-\bar{p})}\sqrt{\frac{\log(4K^{2Jq})+K}{2n}}.$$

Then, there exists a positive constant C such that the expected deficit of AUC *of the ranking sub-tree maximizing the complexity-penalized area under the* ROC *curve is bounded as follows*:

$$\mathrm{AUC}^*-\mathbb{E}\big(\mathrm{AUC}(\widetilde{s}_n^*)\big)\le\inf_{1\le K\le 2^D}\left\{C\cdot pen(K,n)+\left\{\mathrm{AUC}^*-\sup_{\omega:\#\mathcal{P}(\mathcal{T}(\omega))=K}\mathrm{AUC}(s_\omega)\right\}\right\}. \tag{14}$$

On AUC *consistency of sub-ranking trees.* The next results are immediate corollaries of Proposition 6, they reveal that under mild assumptions, AUC-consistent sub-ranking trees do exist. Its proof is left to the reader.

Corollary 1 (CONSISTENCY) *Suppose that assumptions of Proposition 6 are fulfilled and that there exists a sequence* $\mathcal{T}_n(\omega_n)$ *of subtrees of the master ranking trees* $\mathcal{T}_n$ *produced by* TREERANK *such that* $\mathbb{E}[\mathrm{AUC}(s_{\omega_n})]\to\mathrm{AUC}^*$, *as* $n\to\infty$. *Assume in addition that*:

(i) *if* $\mathcal{T}_n$ *is grown through the* $\mathbf{O}_1$ *splitting rule with* $\kappa=\kappa(n)$ *axis-parallel splits, then*

$$\kappa(n)\cdot\mathbb{E}\left[\#\mathcal{P}(\mathcal{T}_n(\omega_n))\right]=o(n/\log n)\quad\text{as } n\to\infty,$$

(ii) *if* $\mathcal{T}_n$ *is grown through the* $\mathbf{O}_2$ *splitting rule based on dyadic hypercubes of side length* 2^{-J} *with* $J=J(n)$, *then*

$$\mathbb{E}\left[\#\mathcal{P}(\mathcal{T}_n(\omega_n))\right]=o(n)\quad\text{and}\quad J(n)=o(n/\log n)\quad\text{as } n\to\infty.$$

Then, the scoring rule based on structural AUC *maximization is* AUC *consistent*:

$$\lim_{n\to\infty}\mathbb{E}\left[\mathrm{AUC}\left(\widetilde{s}_n^*\right)\right]=\mathrm{AUC}^*.$$

In the $\mathbf{O}_2$ case, it follows from Proposition 3 that, under additional constraints on the size of the cells of the master ranking tree output by TREERANK, AUC consistency of the pruning procedure can be proved by means of classical approximation results.

Proposition 7 *Suppose that assumptions of Proposition 6 are satisfied and that the master ranking tree* $\mathcal{T}_n$ *is grown through the TreeRank algorithm with the* $\mathbf{O}_2$ *splitting rule based on dyadic hypercubes of side length* 2^{-J}, $J=J(n)\ge 1$, *and depth* D_n. *If in addition, as* $n\to\infty$, $J(n)\to\infty$ *and the sizes of the cells* $\{C_k^{(n)}:k=0,\dots,2^{D_n+1}-1\}$ *of the related partition* $\mathcal{P}(\mathcal{T}_n)$ *uniformly shrink to zero in the sense that* $\max_{0\le k<2^{D_n+1}}\mu(C_k^{(n)})\to 0$, *then the pruned ranking trees* $\mathcal{T}_n(\widetilde{\omega}_n^*)$ *obtained from* $\mathcal{T}_n$ *are* AUC *consistent.*

Remark 9 (EXTENSIONS TO MORE GENERAL SPLITS) Here, we have studied structural AUC maximization in two situations, corresponding to simple ways of performing the growing stage: in the $\mathbf{O}_2$ case, selection occurs over a finite number of models so that complexity is simply described by the cardinality of the collection considered, whereas, in the $\mathbf{O}_1$

case, the final scoring rule is selected among a collection of models of which complexity is described by shattering coefficients in a combinatorial fashion. More sophisticated splitting rules could be naturally considered, leading to more complex collections of scoring functions. We point out that, in some cases, explicit penalties, involving (conditional) Rademacher averages, could be deduced from the very general bounds for the supremum of U-processes established in Clémençon et al. (2008).

Remark 10 (ALTERNATIVE PRUNING SCHEMES) When data are not that expensive, one may consider using a different dataset for the pruning stage. In such a case, bounds on the expected AUC performance of complexity-based pruning schemes for ranking trees can be established via similar arguments. Owing to space limitations, details are omitted here.

Remark 11 (MODEL SELECTION FOR BIPARTITE RANKING) In Clémençon and Vayatis (2009), a model selection procedure has also been considered in the bipartite ranking context. Although its analysis has been carried out using the same type of inequalities for U-statistics, we highlight the fact that it is of a very different nature than the methods on which we focus here. Indeed, related penalties are based on smoothness assumptions for the regression function and selection operates on a collection of partitions fixed in advance.

5 Interpreting a ranking tree

Beyond the fact that they permit to handle missing data in a straightforward manner (by assigning to a partially observed instance x the empirical mean of each unobserved component within the cell where it currently lies) in the training stage or for prediction, a crucial advantage of decision trees concerns interpretability. Indeed, a ranking tree may be easily visualized in two dimensions, see Fig. 7, and the related scoring function may be described through a chain of simple rules. In various applications, such as medical diagnosis or credit-risk screening for instance, it is essential to interpret the "rank/score" $s(x)$ and determine which attributes contribute the most to its variation (provided an adequate measure of variability of the rank is given, see the discussion below). In the case where the ranking tree is obtained through axis-parallel splits, we now propose some monitoring tools for interpreting ranking trees.

5.1 Variable relative importance

When using the LEAFRANK procedure with perpendicular splits for performing the *Optimization step* in the growing stage, each internal node N of the resulting ranking tree $\mathcal{T}$ is splitted according to a sub-tree t_N with perpendicular cuts providing a binary scoring rule $s_{t_N}(x)$.

Following in the footsteps of the heuristics proposed in Breiman et al. (1984) for tree-based classification, a measure of relevance in predicting the "cost-sensitive" classifier $s_t(x)$ corresponding to such a sub-tree t can be proposed for each component of the input vector $X = (X^{(1)}, \dots, X^{(d)})$. For each node m of the sub-tree, denote by $v(m)$ the index of the component serving as *split variable* and by $\Delta\widehat{\mathrm{AUC}}(m)$ the gain in terms of empirical AUC induced by this particular split. In this respect, recall that, if the cell $C \subset \mathcal{X}$ corresponding to node m has left child C', one may write $\Delta\widehat{\mathrm{AUC}}(m) = \{\widehat{\alpha}(C)\widehat{\beta}(C') - \widehat{\beta}(C)\widehat{\alpha}(C')\}/2$. We set: $\forall j \in \{1, \dots, d\}$,

$$\mathcal{I}_j(t) = \sum_{m:\text{ internal nodes of } t} \left(\Delta\widehat{\mathrm{AUC}}(m)\right)^2 \cdot \mathbb{I}\{v(m) = j\}.$$

At the level of the global ranking tree, the squared relative importance of component $X^{(j)}$ is obtained by summing over all $\mathcal{T}$'s internal nodes:

$$\mathcal{I}_j = \sum_{N:\ \text{internal nodes of } \mathcal{T}} \mathcal{I}_j(t_N).$$

We point out that the computation of relative importance indicators is straightforward, since it only involves quantities that are computed when fitting the ranking tree.

5.2 Partial dependence plots

After sorting the attributes $X^{(1)}, \dots, X^{(q)}$ according to their relevance, the next step to take is to quantify the dependence of the scoring model on each of them. Consider a subvector X^{I_0} of the input vector $X = (X^{(1)}, \dots, X^{(q)})$ corresponding to a given subset of indexes $I_0 \subset \{1, \dots, q\}$. Denote by $I_1 = \{1, \dots, q\} \setminus I_0$ the complement set. Rather than renumbering the components, suppose that $X = (X^{I_0}, X^{I_1})$. In order to gain insight into the way the ranking defined by the stepwise scoring function $s(x)$ depends on the set of components X^{I_0}, one may investigate the variability of the *partial dependence function* $s(x^{I_0} \mid I_1) = \mathbb{E}[s(x^{I_0}, X^{I_1})]$, through its statistical counterpart

$$x^{I_0} \mapsto \widehat{s}(x^{I_0} \mid I_1) = \frac{1}{n} \sum_{i=1}^{n} s(x^{I_0}, X_i^{I_1}),$$

which can be visualized when $\# I_0 = 2$. One may refer to Appendix A.2 in Friedman (2001) for a discussion on the relevance of partial dependence plots and further details on computational aspects in the case of a tree-structured piecewise-constant function.

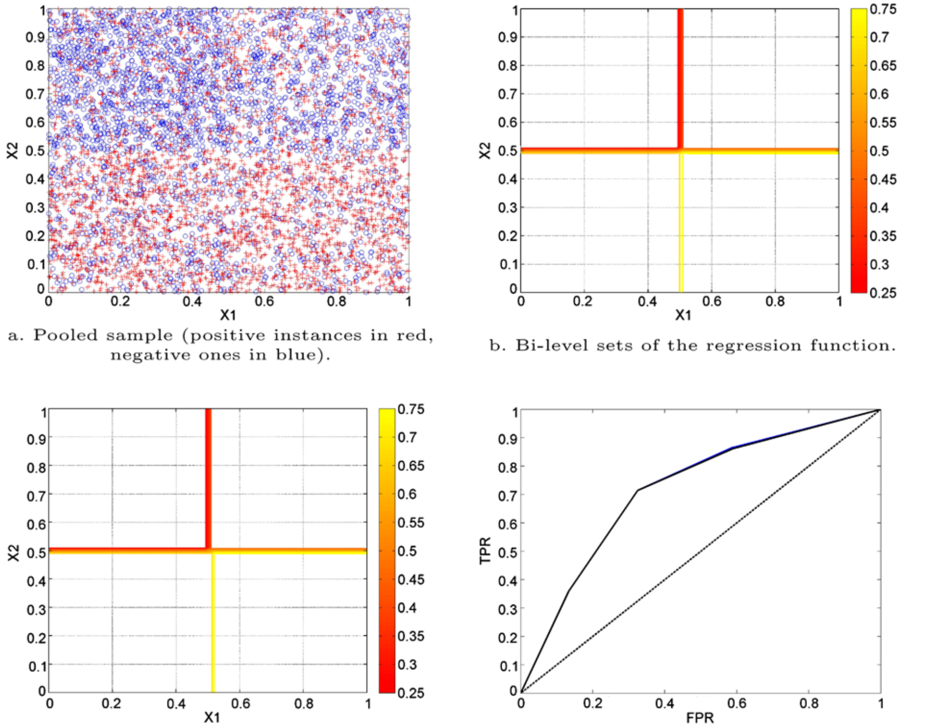
6 Numerical experiments

In order to illustrate some of the ideas developed throughout the article, we now present a few simulation results. In this respect, two bi-dimensional toy models have been considered. The first one involves mixtures of uniform distributions, so that the target curve ROC* has exactly the same form as the estimate produced by TREERANK (*i.e.* linear-by-parts), while conditional Gaussian distributions with different covariance matrices are considered in the second one, yielding level sets with quadratic frontiers.

In both examples, we take $p = 1/2$. From an empirical perspective, the impact of the order of magnitude of the proportion of positive instances among the pooled sample will be investigated in a forthcoming paper, entirely devoted to a systematic comparison of various ranking methods over a number of datasets. Here, in each example, the artificial data simulated are split into a training sample, used for the growing and pruning stages both at the same time, and a test sample, used for plotting the "test ROC curve". The master ranking tree is grown by means of the LEAFRANK procedure with perpendicular splits (each split is built from less than 5 terminal nodes) and next pruned via the N-fold cross-validation procedure described in Sect. 4.1 with $N = 10$.

6.1 First example—mixtures of uniform distributions

The artificial data sample represented in Fig. 9a has been generated as follows. We have split the unit square $\mathcal{X} = [0, 1]^2$ into four quarters: $\mathcal{X}_1 = [0, 1/2]^2$, $\mathcal{X}_2 = [1/2, 1] \times [0, 1/2]$,



a. Pooled sample (positive instances in red, negative ones in blue).

b. Bi-level sets of the regression function.

c. Bi-level sets of the scoring function learnt from the training dataset.

d. Optimal (blue) and test (black) ROC curves.

Fig. 9 First example—mixtures of uniform distributions

$\mathcal{X}_3 = [1/2, 1]^2$ and $\mathcal{X}_4 = [0, 1/2] \times [1/2, 1]$. Denoting by $\mathcal{U}_C$ the uniform distribution on a measurable set $C \subset \mathcal{X}$, the class distributions are given by

$$H(dx) = 0.2 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.1 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_3} + 0.4 \cdot \mathcal{U}_{\mathcal{X}_4},$$

$$G(dx) = 0.4 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.2 \cdot \mathcal{U}_{\mathcal{X}_3} + 0.1 \cdot \mathcal{U}_{\mathcal{X}_4}.$$

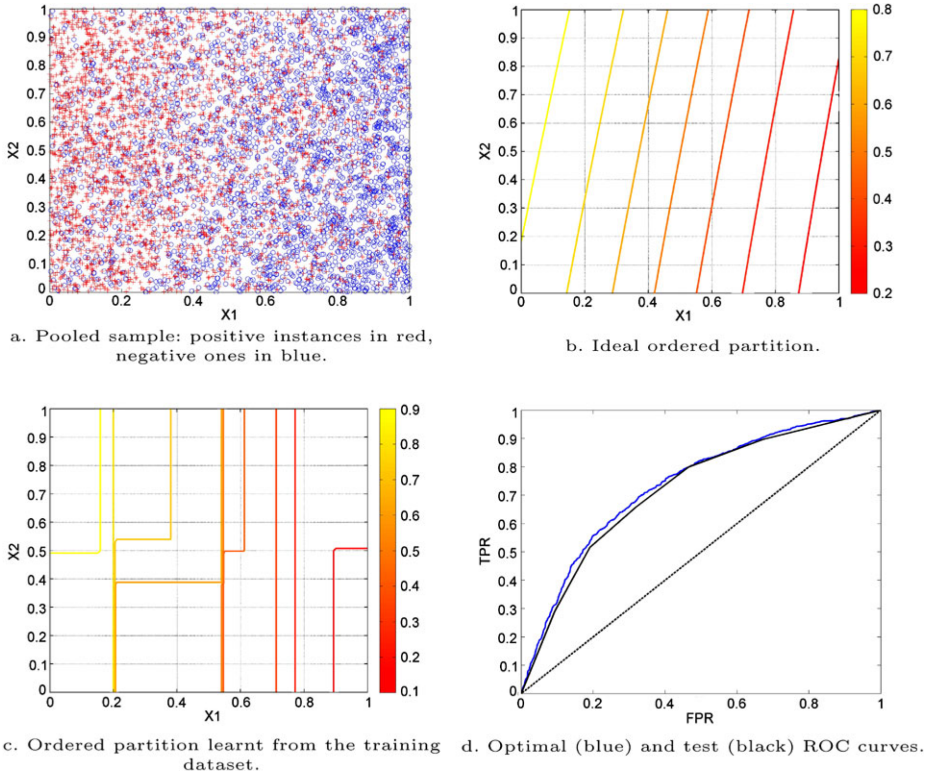
In this setup, optimal scoring functions are piecewise constant, like the regression function

$$\eta = 0.7 \cdot \mathbb{I}_{\mathcal{X}_1} + 0.75 \cdot \mathbb{I}_{\mathcal{X}_2} + 0.4 \cdot \mathbb{I}_{\mathcal{X}_3} + 0.2 \cdot \mathbb{I}_{\mathcal{X}_4},$$

leading to a linear-by-parts optimal ROC curve.

Results produced by the TreeRank algorithm, followed by a cross-validation based pruning procedure are displayed in Fig. 9. Note that the display (b) shows the sets of the form $\{x : a < \eta(x) < b\}$ while the display (c) shows the sets of the form $\{x : a < \hat{s}(x) < b\}$ where $\hat{s}$ is the scoring rule given by the algorithm. In the growing stage, splits have been obtained through the LeafRank method by constraining the number of terminal nodes to be less than 5.

In spite of the simplicity of this first example, it is comforting to observe that the four bi-level sets of η are almost perfectly retrieved by the algorithm, so that the test ROC curve and the optimal one can hardly be distinguished.



a. Pooled sample: positive instances in red, negative ones in blue.

b. Ideal ordered partition.

c. Ordered partition learnt from the training dataset.

d. Optimal (blue) and test (black) ROC curves.

Fig. 10 Second example—mixture of conditional Gaussian distributions

6.2 Second example—conditional Gaussian distributions

Considering a q-dimensional Gaussian random vector Z, drawn as $\mathcal{N}(m,\Gamma)$, and a Borelian set $C\subset\mathbb{R}^q$ weighted by $\mathcal{N}(m,\Gamma)$, we denote by $\mathcal{N}_C(m,\Gamma)$ the conditional distribution of Z given $Z\in C$. Equipped with this notation, the class distributions used in this example can be written as:

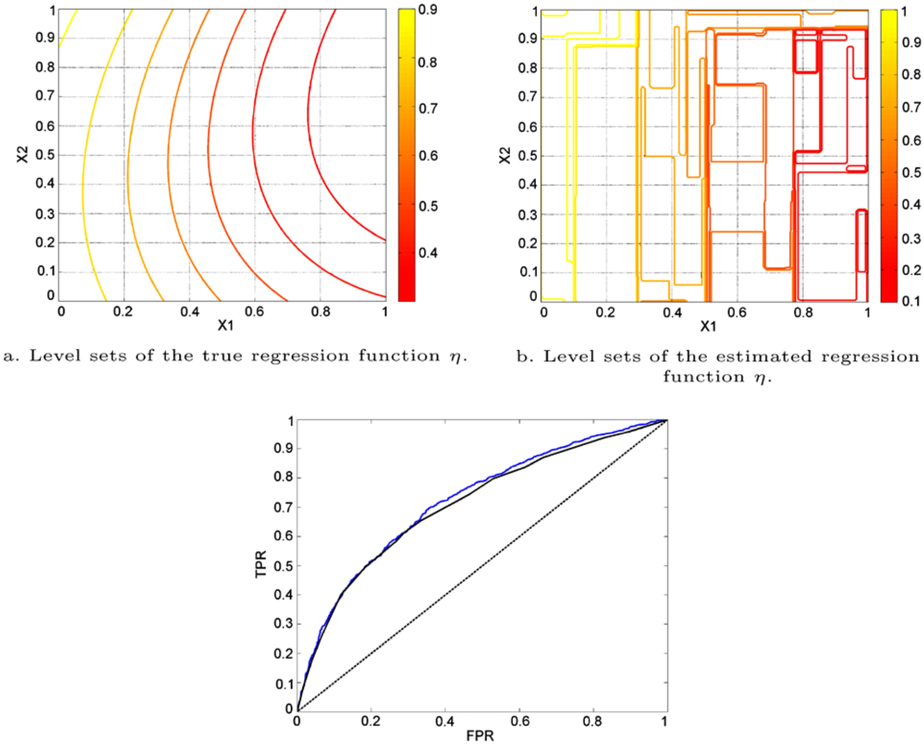
$$H(dx)=\mathcal{N}_{[0,1]^2}\left(\begin{pmatrix}2\\0.5\end{pmatrix},\begin{pmatrix}1&0.25\\0.25&1.15\end{pmatrix}\right),$$

$$G(dx)=\mathcal{N}_{[0,1]^2}\left(\begin{pmatrix}-1\\0.5\end{pmatrix},\begin{pmatrix}1&0.15\\0.15&1.25\end{pmatrix}\right).$$

When $p=1/2$, the regression function is then given by:

$$\eta(x)=\frac{1.02\cdot\exp(0.02x_1^2+0.05x_2^2-3.08x_1+0.53x_2-0.11x_1x_2+1.32)}{1+1.02\cdot\exp(0.02x_1^2+0.05x_2^2-3.08x_1+0.53x_2-0.11x_1x_2+1.32)}.$$

The simulated dataset is plotted in Fig. 10(a), while the level sets of the regression function related to the approximation scheme mimicked by TreeRank are represented in Fig. 10(b). For comparison purpose, the level sets of the piecewise scoring function out-



a. Level sets of the true regression function η.

b. Level sets of the estimated regression function η.

c. True (blue) and Estimated (black) Roc Curve.

Fig. 11 Results on the tougher Gaussian mixture model

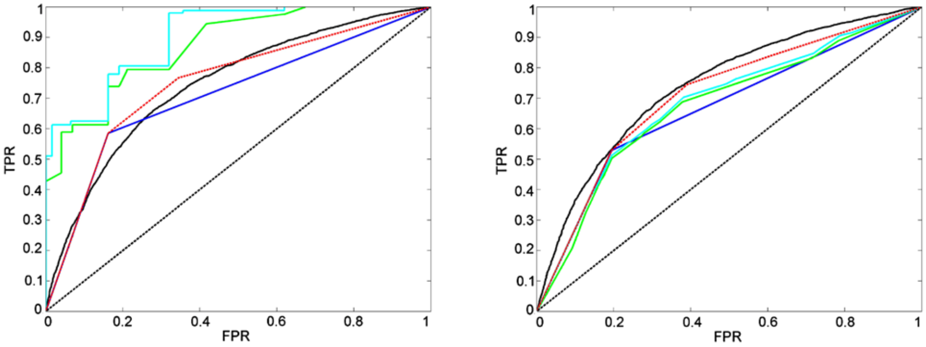
put by the learning method are displayed in Fig. 10(c) and its test ROC curve is plotted in Fig. 10(d), together with the optimal one.

Although the frontiers of the target level sets of η are quadratic, they look almost linear, due to the scale effect caused by the large distance between the centers of the two normal distributions. However, this does not suffice for explaining the performance of the scoring function in terms of ROC curve. Indeed, as shown by the example represented in Fig. 10, results are still satisfactory when taking Gaussian with closer centers.

We have also considered another example with two Gaussian distributions where the frontier between the two classes is more difficult to approximate with partitions made of orthogonal splits. Figure 11 reveals that performance still is satisfactory based on the ROC curves and the estimated level sets attempt to follow the geometry of the true level sets.

6.3 Influence of the number of terminal nodes and pruning

We now propose to focus on the sensitivity with respect to complexity parameters of the TreeRank algorithm with node splitting achieved by LeafRank. We consider the simulation example introduced previously with Gaussian distributions and evaluate performance on two fixed samples, one for training (500 points) and one for testing (10000 points). Denote by $\#\mathcal{P}_{TRK}$ and $\#\mathcal{P}_{LRK}$ the cardinality of partitions (or the number of terminal nodes) obtained respectively by TreeRank and LeafRank. We applied our procedure in four distinct situations:



a.Learning Roc Curves.
True:black, (i):blue, (ii):green
(iii):cyan, (iv):dotted red

b.Test Roc Curves.
True:black, (i):blue, (ii):green
(iii):cyan, (iv):dotted red

AUC	(i)	(ii)	(iii)	(iv)	True
(Trk, Lrk)	(3, 3)	(15, 15)	(∞, ∞)	(CV,CV)	
Learning	0.71	0.88	0.91	0.75	0.76
Test	0.67	0.67	0.66	0.71	0.74

Fig. 12 Gaussian mixtures modelization

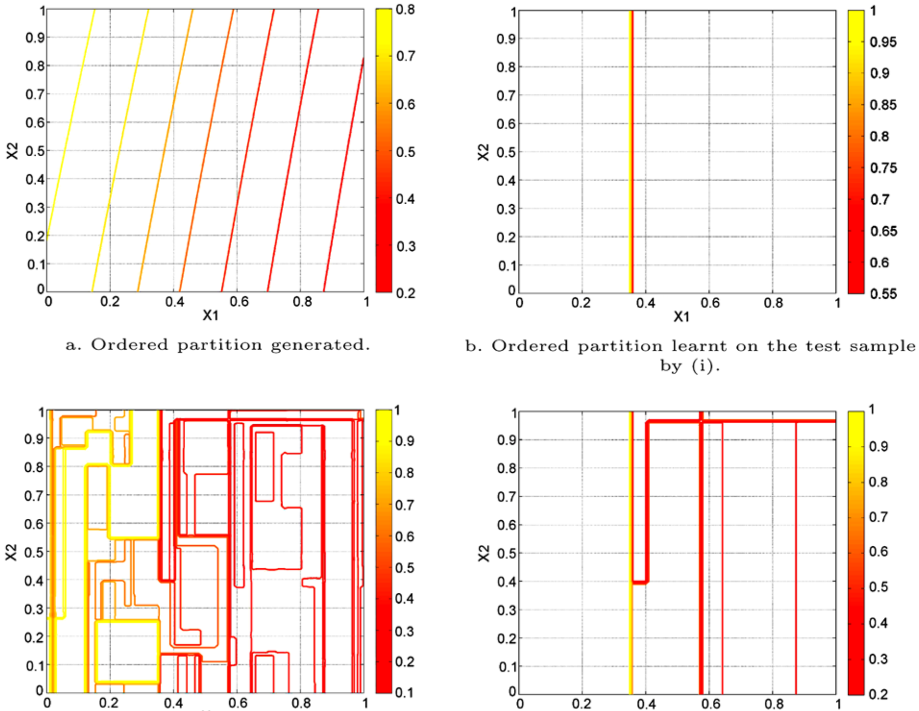
- (i) $\#\mathcal{P}_{TRK} \leq 3$ and $\#\mathcal{P}_{LRK} \leq 3$ (severe pruning—low complexity)
- (ii) $\#\mathcal{P}_{TRK} \leq 15$ and $\#\mathcal{P}_{LRK} \leq 15$ (mild pruning—intermediate complexity)
- (iii) $\#\mathcal{P}_{TRK} \leq \infty$ and $\#\mathcal{P}_{LRK} \leq \infty$ (no pruning—very high complexity)
- (iv) the master ranking tree and the splitting trees are both pruned using 10 fold-cross-validation.

The results are summarized in Fig. 12. We observe some amount of overfitting in the case of (ii) and (iii), while the choice (i) leads to underfitting, and cross-validation seems to lead to the more stable ranking rule. We point out that the difference between the different pruning levels is overwhelming when observing the resulting level sets (Fig. 13).

7 Comparison of TreeRank with other methods on real data sets

The aim of this section is to establish a first comparison between different ranking methods on real data. The TreeRank procedure previously described will be compared to a classical plug-in approach, the logistic regression, and to the RankBoost procedure proposed by Freund et al. (2003), an algorithm based on the combination of weak rank learners. We considered two data sets: (i) Breast Cancer data set from the UCI Repository—569 observations, 63% of positive instances, (ii) Credit Scoring data set (available at http://www.cmla.ens-cachan.fr/fileadmin/Membres/vayatis/Files/Datasets/ReviewCreditScoringDataset.txt)—216 observations, 50% of positive instances.

In these experiments, the training sample is about 80% of the observations both data sets. The plots in Fig. 14 display the results based on a random test sample. The RankBoost procedure was run with $T = 30$ weak learners (step functions), the model selection procedure used for the logistic regression is based on the Bayesian Information Criterion and the size of the master tree and of the subtrees obtained through the TreeRank procedure, have been optimally selected by a pruning stage based on 8-fold cross validation.



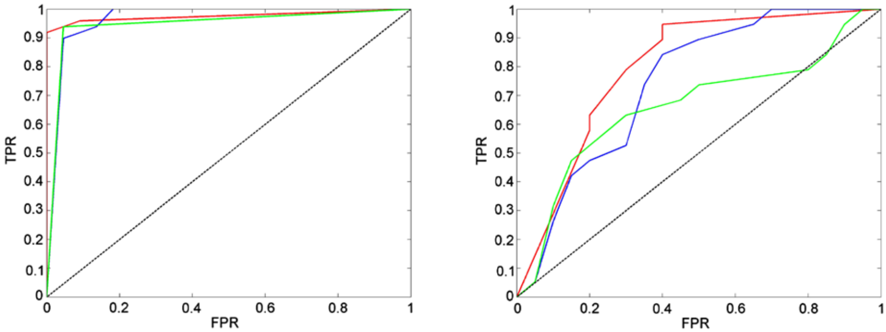
a. Ordered partition generated.

b. Ordered partition learnt on the test sample by (i).

c. Ordered partition learnt on the test sample by (iii).
Same solour code as previously

d. Ordered partition learnt on the test sample by (iv).

Fig. 13 Gaussian mixtures levels set estimation



a. Example of test ROC curves for the Breast Cancer data.
RankBoost (blue), TreeRank (red), Logistic Regression (green)

c. Example of test ROC curves for the Credit Scoring data set.
RankBoost (blue), TreeRank (red), Logistic Regression (green)

Average Test AUC	TreeRank	RankBoost	Logistic Regression
Breast Cancer data	0.923 ± 0.048	0.967 ± 0.012	0.856 ± 0.185
Credit Scoring data	0.733 ± 0.060	0.788 ± 0.053	0.718 ± 0.064

Fig. 14 Experimental results on two real data sets

In order to study the variability of these methods, 50 bootstrap replications of the original sample were generated. The table in Fig. 14 give the mean and the standard deviation of the AUC computed using the 50 bootstrap replications.

These two examples show that TreeRank and RankBoost clearly surpass the logistic regression implementation both in terms of average performance and robustness. The RankBoost algorithm leads to slightly better results but this is not surprising as it benefits of the nice properties of aggregation. We expect to boost the performance of TreeRank by using bagging and other aggregation techniques.

8 Conclusion

In the present paper, two major issues related to the implementation of the TreeRank approach proposed in Clémençon and Vayatis (2009) for bipartite ranking have been addressed, namely *splitting* and *pruning*. We described the interpretation of the splitting task as a cost-sensitive classification problem (with a cost locally depending on the data within the cell to split). This observation paves the way for considering a wide variety of techniques for performing the *Optimization step*. We also carried out a theoretical analysis of pruning strategies, providing hints on how to choose the right size for the ranking trees produced. We thus developed concrete algorithms for nonparametric scoring of high dimensional data with strong arguments for their practical and theoretical interest. A variety of tree-based procedures can then be considered with many possible options for splitting and pruning. One of the key ideas for splitting relies on a recursive call of a naive version of the TreeRank algorithm proposed in a previous work (Clémençon and Vayatis 2008). Two pruning methods are described and analyzed in the paper: penalties based on structural AUC maximization splitting and cross-validation techniques. However, complexity-based penalties cannot be used straightforwardly since further studies are needed to calibrate the numerical constants involved. We refer to a recent paper by Arlot (2009) where resampling strategies are depicted in the classification setup in order to address this issue. The simulation study we performed reveals the potential but also the limitations of these scoring algorithms. The main drawbacks of trees are their instability and those due to the hierarchical structure of the ranking trees (pileup of error when growing the tree). Future work will be devoted to the design of search and aggregation strategies in order to overcome these limitations.

Acknowledgements We thank the referees and the Action Editor for their constructive suggestions.

Appendix A: Proofs

A.1 Proof of Theorem 1

The proof is based on the next lemma.

Lemma 2 *Let $\mathcal{P} = \{C_k\}_{1\leq k\leq K}$ be a partition with $K \geq 2$ non empty cells. Consider $\sigma \in \mathfrak{S}_K$, fix $k \in \{1, \ldots, K-1\}$ and let $\tau_k \in \mathfrak{S}_K$ be the transposition exchanging k and $k+1$.Then, if $(\sigma(k) - \sigma(k+1)) \cdot (\sigma^*_{\mathcal{P}}(k) - \sigma^*_{\mathcal{P}}(k+1)) > 0$, we have*

$$\mathrm{AUC}(s_{\mathcal{P},\sigma}) \geq \mathrm{AUC}(s_{\mathcal{P},\sigma\circ\tau_k}).$$

Proof Without any restrictions, one may suppose that $\sigma(k)-\sigma(k+1)$ and $\sigma^*_{\mathcal{P}}(k)-\sigma^*_{\mathcal{P}}(k+1)$ are both nonnegative. It follows from the expression of the AUC stated in Proposition 2 that

$$\mathrm{AUC}(s_{\mathcal{P},\sigma})-\mathrm{AUC}(s_{\mathcal{P},\sigma\circ\tau_k})=\frac{1}{2}\left\{\beta(C_{\sigma(k+1)})\alpha(C_{\sigma(k)})-\beta(C_{\sigma(k)})\alpha(C_{\sigma(k+1)})\right\},$$

and the latter quantity is negative by definition of $\sigma^*_{\mathcal{P}}$. □

Observing that any permutation σ may be decomposed as $\sigma^*_{\mathcal{P}}\circ\tau$, where τ is a compound of a finite number of transpositions τ_k, $k\in\{1,\dots,K-1\}$, the proof of the first part of the theorem immediately follows from the lemma stated above. The second part straightforwardly results from (4) in Proposition 2.

A.2 Proof of Proposition 3

We first establish the following preliminary result.

Lemma 3 *Suppose that the r.v. $\eta(X)$ has a continuous distribution. Then, for any partition $\mathcal{P}=\{C_k\}_{1\le k\le K}$ with $K\ge 2$ non empty cells, we have: $\forall s\in\mathcal{S}_{\mathcal{P}}$,*

$$\mathrm{AUC}^*-\mathrm{AUC}(s)=\frac{\mathbb{E}[|\eta(X)-\eta(X')|]\mathbb{I}\{(X,X')\in\Gamma_s\}}{2p(1-p)}+\frac{1}{4p(1-p)}\sum_{k=1}^{K}\mathcal{G}(C_k),$$

where $\Gamma_s=\{(x,x')\in\mathcal{X}^2:(\eta(x)-\eta(x'))\cdot(s(x)-s(x'))<0\}$.

Proof Notice first that, for all scoring function s:

$$\begin{aligned}\mathrm{AUC}(s)&=\mathbb{P}\{s(X)>s(X')\mid(Y,Y')=(+1,-1)\}+\frac{1}{2}\mathbb{P}\{s(X)=s(X')\mid(Y,Y')=(+1,-1)\}\\&=-\frac{1}{2}\mathbb{P}\{s(X)=s(X')\mid(Y,Y')=(1,-1)\}+1-\frac{L(s)}{2p(1-p)},\end{aligned}\tag{15}$$

where $L(s)=\mathbb{P}\{(s(X)-s(X'))\cdot(Y-Y')<0\}$. As $L(s)$ may be expressed as the expectation of $\eta(X)(1-\eta(X'))\mathbb{I}\{s(X)<s(X')\}+(1-\eta(X))\eta(X')\mathbb{I}\{s(X)>s(X')\}$ and $\eta(X)$ has a continuous distribution, one may check that

$$\begin{aligned}L(s)-L(\eta)=\mathbb{E}\left[|\eta(X)-\eta(X')|\mathbb{I}\{(X,X')\in\Gamma_s\}\right]+\frac{1}{2}\mathbb{E}[\mathbb{I}\{s(X)=s(X')\}|\eta(X)-\eta(X')|]\\-\mathbb{P}\{s(X)=s(X'),(Y,Y')=(1,-1)\}.\end{aligned}$$

Observe in addition that, when $s(x)$ admits a $(\mathcal{P},\sigma)$-representation, one may write the second term on the right hand side of the equation above as $\frac{1}{2}\sum_{C\in\mathcal{P}}\mathbb{E}(|\eta(X)-\eta(X')|\mathbb{I}\{(X,X')\in C^2\})$, which eventually concludes the proof. □

Now, observe that: if $(X,X')\in\Gamma_s$, then

$$|\eta(X)-\eta(X')|\le|\eta(X)-\widehat{\eta}(X)|+|\eta(X')-\widehat{\eta}(X')|.$$

Combined to Lemma 3, this establishes the desired bound.

A.3 Proof of Lemma 1

Observe first that:

$$\begin{aligned}p(1-p)\{2\mathrm{AUC}(s)-1\} &= p(1-p)\{\beta(C)-\alpha(C)\}\\ &= \mathbb{E}[(1-p)\eta(X)\cdot\mathbb{I}\{X\in C\}+p(1-\eta(X))\cdot\mathbb{I}\{X\notin C\}]\\ &\quad -p(1-p).\end{aligned}$$

Now the lemma results from the fact that:

$$\begin{aligned}2p(1-p)\{\mathrm{AUC}(s_1^*)-\mathrm{AUC}(s)\} &= \mathbb{E}[(1-p)\eta(X)\cdot(\mathbb{I}\{X\in C^*\}-\mathbb{I}\{X\in C\})]\\ &\quad +\mathbb{E}[p(1-\eta(X))\cdot(\mathbb{I}\{X\notin C^*\}-\mathbb{I}\{X\notin C\})]\\ &= \mathbb{E}[|\eta(X)-p|\cdot\mathbb{I}\{X\in C\Delta C^*\}].\end{aligned}$$

A.4 Proof of Theorem 2

For any $j\geq 1$, define $\mathcal{C}_j$ the collection of (non empty) subsets of $\mathcal{X}$ that may be formed from the 2^{jq} dyadic cubes of side length 2^{-j}, except $\mathcal{X}=[0,1]^q$ itself. Denote also by $\mathcal{P}_{2,j}$ the set partitions of $\mathcal{X}$ formed of two (non empty) elements of $\mathcal{C}_j$. We set: $\forall j\geq 1$, $\widetilde{L}_j^*$ the true left cell based on the partition $\mathcal{C}_j$ of the set $\mathcal{X}$ and $\widehat{L}_j^*$ the empirical counterpart. We denote the related binary scoring functions by:

$$\widetilde{s}_j^*(x)=2\cdot\mathbb{I}\{x\in\widetilde{L}_j^*\}-1\quad\text{and}\quad\widehat{s}_j^*(x)=2\cdot\mathbb{I}\{x\in\widehat{L}_j^*\}-1.$$

Classically, we bound the deficit of AUC by the sum of a bias component and a variance term:

$$\begin{aligned}\mathrm{AUC}(s_1^*)-\mathrm{AUC}(\widehat{s}_j^*) &= \{\mathrm{AUC}(s_1^*)-\mathrm{AUC}(\widetilde{s}_j^*)\}+\{\mathrm{AUC}(\widetilde{s}_j^*)-\widehat{\mathrm{AUC}}(\widetilde{s}_j^*)\}\\ &\quad +\{\widehat{\mathrm{AUC}}(\widetilde{s}_j^*)-\widehat{\mathrm{AUC}}(\widehat{s}_j^*)\}+\{\widehat{\mathrm{AUC}}(\widehat{s}_j^*)-\mathrm{AUC}(\widehat{s}_j^*)\}\\ &\leq \mathrm{AUC}(s_1^*)-\mathrm{AUC}(\widetilde{s}_j^*)+2\sup_{s\in\mathcal{S}_{\mathcal{P}_{2,j}}}|\widehat{\mathrm{AUC}}(s)-\mathrm{AUC}(s)|.\end{aligned}$$

Considering the variance term, we first express the empirical $\widehat{\mathrm{AUC}}(s)$ as:

$$\widehat{\mathrm{AUC}}(s)=\frac{n(n-1)}{2n_+n_-}\widehat{U}_n(s),$$

where

$$\widehat{U}_n(s)=\frac{2}{n(n-1)}\sum_{1\leq i<j\leq n}h_s((X_i,Y_i),(X_j,Y_j))$$

is a U-statistic of order 2 with bounded symmetric kernel

$$h_s((x_1,y_1),(x_2,y_2))=\mathbb{I}\{(y_1-y_2)(s(x_1)-s(x_2))>0\}+\frac{1}{2}\mathbb{I}\{s(x_1)=s(x_2),y_1\neq y_2\}$$

and expectation $U(s) = 2p(1-p)\mathrm{AUC}(s)$. By applying the version of Hoeffding's exponential inequality for U-statistics stated in Theorem A of Sect. 5.6 of Serfling (1980) combined with the union bound, one gets that, for all $\delta \in (0,1)$, with probability larger than $1-\delta$: $\forall n \geq 1$,

$$\sup_{s\in\mathcal{S}_{\mathcal{P}_{2,j}}} \left|\widehat{U}_n(s) - U(s)\right| \leq \sqrt{\frac{\log(\delta/(2\#\mathcal{P}_{2,j}))}{2n}}.$$

The desired bound then follows by noticing that

$$\left|\widehat{\mathrm{AUC}}(s) - \mathrm{AUC}(s)\right| \leq \frac{1}{2\underline{p}(1-\bar{p})}\left|\widehat{U}_n(s) - U(s)\right| + \frac{1}{2}\left\{\left|\frac{1}{p} - \frac{n}{n_+}\right| + \left|\frac{1}{1-p} - \frac{n}{n-n_+}\right|\right\}$$

and applying the standard Hoeffding probability inequality in order to control the fluctuations of n_+/n around $p \in [\underline{p}, \bar{p}]$.

A.5 Proof of Proposition 6

In order to prove the desired oracle inequality, we first establish the lemma below. Let $K \geq 1$, we denote by $\mathbf{P}_T(K)$ the collection of all tree-structured partitions of the feature space $\mathcal{X} \subset \mathbb{R}^q$ with $K \geq 1$ non empty cells and by $\mathcal{S}_T(K) = \bigcup_{\mathcal{P}\in\mathbf{P}_T(K)} \mathcal{S}_{\mathcal{P}}$ the set of piecewise constant scoring functions associated to such partitions. We also introduce the empirical AUC maximizer over $\mathcal{S}_T(K)$:

$$\widehat{s}^*_{n,K} = \arg\max_{s\in\mathcal{S}_T(K)} \widehat{\mathrm{AUC}}(s).$$

Lemma 4 *Assume that the hypotheses of Proposition 6 are fulfilled.*

(i) *If splits are optimized using the $\mathbf{O}_1$ rule and the penalization is chosen accordingly, then*: $\forall(K,\kappa) \in \mathbb{N}^{*2}$,

$$\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)} \mathrm{AUC}(s) - \mathrm{AUC}(\widetilde{s}^*_n) \geq \epsilon\right\}$$
$$\leq 16\left((n+1)q\right)^{2K\kappa} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/512} + e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128}.$$

(ii) *If splits are optimized using the $\mathbf{O}_2$ rule and the penalization is chosen accordingly, then*: $\forall(K,J) \in \mathbb{N}^{*2}$,

$$\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)} \mathrm{AUC}(s) - \mathrm{AUC}(\widetilde{s}^*_n) \geq \epsilon\right\} \leq 4K^{2^{Jq}} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/8} + e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/2}.$$

Proof We follow the argument of Lugosi and Zeger (1996), see also Sect. 18.1 in Devroye et al. (1996). Write: $\forall\epsilon > 0$, $\forall K \geq 1$,

$$\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)} \mathrm{AUC}(s) - \mathrm{AUC}(\widetilde{s}^*_n) \geq \epsilon\right\}$$
$$\leq \mathbb{P}\left\{\sup_{l\geq 1} \widetilde{\mathrm{CPAUC}}(\widetilde{s}^*_{n,l}) - \mathrm{AUC}(\widetilde{s}^*_n) \geq \frac{\epsilon}{2}\right\}$$

$$+\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\sup_{l\geq 1}\widetilde{\mathrm{CPAUC}}(\widehat{s}^*_{n,l})\geq\frac{\epsilon}{2}\right\}.$$

Therefore, the first term on the right hand side of the inequality above may be rewritten and bounded as follows:

$$\begin{aligned}
&\mathbb{P}\left\{\widetilde{\mathrm{CPAUC}}(\widetilde{s}^*_n)-\mathrm{AUC}(\widetilde{s}^*_n)\geq\frac{\epsilon}{2}\right\}\\
&\quad\leq\mathbb{P}\left\{\inf_{l\geq 1}\left\{\widetilde{\mathrm{CPAUC}}(\widehat{s}^*_{n,l})-\mathrm{AUC}(\widehat{s}^*_{n,l})\right\}\geq\frac{\epsilon}{2}\right\}\\
&\quad\leq\sum_{l\geq 1}\mathbb{P}\left\{\left|\mathrm{AUC}(\widehat{s}^*_{n,l})-\widehat{\mathrm{AUC}}(\widehat{s}^*_{n,l})\right|\geq\frac{\epsilon}{2}+pen(l,n)\right\}\\
&\quad\leq\sum_{l\geq 1}\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\left|\mathrm{AUC}(s)-\widehat{\mathrm{AUC}}(s)\right|\geq\frac{\epsilon}{2}+pen(l,n)\right\}.
\end{aligned}\tag{16}$$

Turning to the second term, observe that

$$\begin{aligned}
&\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\sup_{l\geq 1}\widetilde{\mathrm{CPAUC}}(\widehat{s}^*_{n,l})\geq\frac{\epsilon}{2}\right\}\\
&\quad\leq\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\widetilde{\mathrm{CPAUC}}(\widehat{s}^*_{n,K})\geq\frac{\epsilon}{2}\right\}\\
&\quad\leq\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\widehat{\mathrm{AUC}}(\widehat{s}^*_{n,K})\geq\frac{\epsilon}{4}\right\}\\
&\quad\leq\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\left|\widehat{\mathrm{AUC}}(s)-\mathrm{AUC}(s)\right|\geq\frac{\epsilon}{4}\right\},
\end{aligned}\tag{17}$$

since we assumed $pen(K,n)\leq\epsilon/4$.

In both cases, we are thus lead to establish a sharp bound for the tail probability of $\sup_{s\in\mathcal{S}_T(K)}|\widehat{\mathrm{AUC}}(s)-\mathrm{AUC}(s)|$.

We first place ourselves in the situation $\mathbf{O}_1$, where *Optimization steps* are performed using at most κ perpendicular splits. We follow the approach developed in Clémençon et al. (2008) in the context of empirical "ranking risk" minimization. We recall the following lemma, based on Hoeffding's representation of U-statistics (see Lemma A1 in Clémençon et al. 2008).

Lemma 5 (Clémençon et al. 2008) *Let $q_\tau:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be real-valued functions indexed by $\tau\in T$ where T is some set. If $X_1,\ldots,X_n$ are i.i.d. then for any convex nondecreasing function ψ,*

$$\mathbb{E}\left[\psi\left(\sup_{\tau\in T}\frac{1}{n(n-1)}\sum_{i\neq j}q_\tau(X_i,X_j)\right)\right]\leq\mathbb{E}\left[\psi\left(\sup_{\tau\in T}\frac{1}{\lfloor n/2\rfloor}\sum_{i=1}^{\lfloor n/2\rfloor}q_\tau(X_i,X_{\lfloor n/2\rfloor}+i)\right)\right],$$

assuming the suprema are measurable and the expected values exist.

The n-th VC shattering coefficient of the class $\mathcal{A} = \bigcup_{\mathcal{P} \in \mathbf{P}_T(K)} \{A \times B : (A, P) \in \mathcal{P}^2\}$ of subsets of $\mathcal{X} \times \mathcal{X}$ is thus bounded as follows:

$$S(\mathcal{A}, n) \leq ((n+1)q)^{2K\kappa}.$$

Combined with Vapnik-Chervonenkis inequality and the lemma above applied to the collection of kernels $\{h_s - U(s)\}_{s \in \mathcal{S}_T(K)}$, this yields: $\forall \epsilon$, $\forall n \geq 1$,

$$\mathbb{P}\left\{\sup_{s \in \mathcal{S}_T(K)} |\widehat{U}_n(s) - U(s)| \geq \epsilon\right\} \leq 8\,((n+1)q)^{2K\kappa}\, e^{-n\epsilon^2/32}.$$

Thus, for n large enough, we have

$$\mathbb{P}\left\{\sup_{s \in \mathcal{S}_T(K)} |\widehat{\mathrm{AUC}}(s) - \mathrm{AUC}(s)| \geq \epsilon\right\} \leq 16\,((n+1)q)^{2K\kappa}\, e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/32}, \tag{18}$$

the extra multiplicative factor in the AUC bound above accounting for the fluctuations of the empirical rate of positive instances among the pooled sample around the proportion p for n large enough. Combined with (16), we get

$$\begin{aligned}
&\mathbb{P}\left\{\widetilde{\mathrm{CPAUC}}(\widetilde{s}_n^*) - \mathrm{AUC}(\widetilde{s}_n^*) \geq \frac{\epsilon}{2}\right\}\\
&\quad \leq \sum_{l=1}^{2^D} 16\,((n+1)q)^{2K\kappa}\, e^{-n\underline{p}^2(1-\bar{p})^2(\frac{\epsilon}{2}+pen(K,n))^2/32}\\
&\quad \leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128} \sum_{l=1}^{2^D} 16\,((n+1)q)^{2\kappa}\, e^{-n\underline{p}^2(1-\bar{p})^2 pen(K,n)^2/32}\\
&\quad \leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128} \sum_{l \geq 1} e^{-K}\\
&\quad \leq e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/128},
\end{aligned}$$

by replacing $pen(K, n)$ by its explicit expression. Combining (18) with (17), we obtain

$$\mathbb{P}\left\{\sup_{s \in \mathcal{S}_T(K)} \mathrm{AUC}(s) - \sup_{l \geq 1} \widetilde{\mathrm{CPAUC}}(\widehat{s}_{n,l}^*) \geq \frac{\epsilon}{2}\right\} \leq 16\,((n+1)q)^{2K\kappa}\, e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/512}.$$

The first assertion of the lemma is thus proved.

Suppose now that $\mathcal{X} = [0, 1]^q$ and cells are obtained as unions of dyadic cubes of side length 2^{-J}, $J \in \mathbb{N}$. In the situation $\mathbf{O}_2$, it suffices to observe that a version of Hoeffding's inequality for U-statistics (see Theorem A in Sect. 5.6 of Serfling 1980) combined with the union bound and the fact that $\#\{h_s : s \in \mathcal{S}_T(K)\} \leq K^{2^{Jq}}$ gives us: $\forall \epsilon$, $\forall n \geq 1$,

$$\mathbb{P}\left\{\sup_{s \in \mathcal{S}_T(K)} \left|\widehat{U}_n(s) - U(s)\right| \geq \epsilon\right\} \leq 2K^{2^{Jq}} e^{-2n\epsilon^2},$$

and for n large enough:

$$\mathbb{P}\left\{\sup_{s\in\mathcal{S}_T(K)}\left|\widehat{\mathrm{AUC}}(s)-\mathrm{AUC}(s)\right|\geq\epsilon\right\}\leq 4K^{2^{Jq}}e^{-2n\underline{p}^2(1-\bar{p})^2\epsilon^2}.$$

The remainder of the argument is omitted, since it is completely similar to the one in the $\mathbf{O}_1$ situation. □

We have

$$\begin{aligned}&\mathrm{AUC}^*-\mathbb{E}\left[\mathrm{AUC}(\widetilde{s}_n^*)\right]\\&\quad=\inf_{K\geq 1}\left\{\left(\mathrm{AUC}^*-\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)\right)+\left(\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\mathbb{E}[\mathrm{AUC}(\widetilde{s}_n^*)]\right)\right\}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\left(\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\mathbb{E}[\mathrm{AUC}(\widetilde{s}_n^*)]\right)^2\\&\qquad\leq u+\int_{t=u}^{\infty}\mathbb{P}\left\{\left(\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)-\mathrm{AUC}(\widetilde{s}_n^*)\right)^2>t\right\}dt.\end{aligned}$$

Now, the oracle inequalities for the expected deficit of AUC follow by integrating the tail bounds stated in Lemma 4, taking $u=C(pen(K,n))^2$.

A.6 Proof of Proposition 7

We consider the $\mathbf{O}_2$ case. Given Corollary 1, it suffices to show that

$$\lim_{n\to\infty}\sup_{s\in\mathcal{S}_{\mathcal{T}_n}}\mathrm{AUC}(s)=\mathrm{AUC}^*.$$

Let $\{C_{D_n,k}\}_{0\leq k<2^{D_n}}$ be the cells of the partition $\mathcal{P}_{D_n}$ corresponding to the master ranking tree $\mathcal{T}_n$ output by TreeRank and s_{D_n} the related scoring function. Recall that, in the $\mathbf{O}_2$ situation, s_{D_n} and $\widehat{\eta}_{\mathcal{P}_{D_n}}$ produce the same ranking, *cf.* Remark 8. By virtue of Proposition 3, we thus have:

$$\mathrm{AUC}^*-\sup_{s\in\mathcal{S}_T(K)}\mathrm{AUC}(s)\leq\frac{\mathbb{E}[|\eta(X)-\widehat{\eta}_{\mathcal{P}_{D_n}}(X)|]}{2p(1-p)}+\frac{1}{4p(1-p)}\sum_{k=1}^{D_n}\mathcal{G}(C_{n,k}),\tag{19}$$

where $\mathcal{G}(C_{n,k})=\mathbb{E}[|\eta(X)-\eta(X')|\cdot\mathbb{I}\{(X,X')\in C_{n,k}^2\}]$. Observe that

$$\sum_{k=0}^{D_n-1}\mathcal{G}(C_{n,k})\leq\sum_{k=0}^{D_n-1}\mu(C_{n,k})^2\leq\max_{0\leq k<D_n}\mu(C_{n,k}).$$

It follows from the stipulated assumptions and the bound above that the term on the right hand side of (19) vanishes as $n\to\infty$. As the argument of Theorem 6.1 in Devroye et al. (1996) ensures that the term on the left hand side also goes to 0 as $n\to\infty$, the result is then proved.

References

Arlot, S. (2009). Model selection by resampling techniques. *Electronic Journal of Statistics*, *3*, 557–624.

Boucheron, S., Bousquet, O., & Lugosi, G. (2005). Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, *9*, 323–375.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth and Brooks.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. In *ACM international conference proceeding series: Vol. 119. Proceedings of the 22nd international conference on machine learning* (pp. 89–96). New York: ACM.

Clémençon, S., & Vayatis, N. (2008). Tree-structured ranking rules and approximation of the optimal ROC curve. In *ALT '08: Proceedings of the 2008 conference on algorithmic learning theory*.

Clémençon, S., & Vayatis, N. (2008). Overlaying classifiers: a practical approach for optimal ranking. In *NIPS '08: Proceedings of the 2008 conference on advances in neural information processing systems*.

Clémençon, S., & Vayatis, N. (2009). On partitioning rules for bipartite ranking. *Journal of Machine Learning Research: Proceedings of AISTATS '09*

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

Clémençon, S., & Vayatis, N. (2010). Overlaying classifiers: a practical approach for optimal scoring. *Constructive Approximation*. doi:10.1007/s00365-010-9084-9

Clémençon, S., Lugosi, G., & Vayatis, N. (2005). Ranking and scoring using empirical risk minimization. In P. Auer, & R. Meir (Eds.), *Lecture notes in computer science: Vol. 3559. Proceedings of COLT 2005* (pp. 1–15). Berlin: Springer.

Clémençon, S., Lugosi, G., & Vayatis, N. (2008). Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, *36*(2), 844–874.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Berlin: Springer.

Ferri, C., Flach, P., & Hernández-Orallo, J. (2003). Improving the AUC of probabilistic estimation trees. In N. Lavrac, D. Gamberger, L. Todorovski, & H. Blockeel (Eds.), *Proceedings of the 14th European conference on machine learning (ECML 2003)*, Cavtat-Dubrovnik, Croatia (pp. 121–132). Berlin: Springer.

Flach, P., & Matsubara, E. T. (2007). A simple lexicographic ranker and probability estimator. In J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladenic, & A. Skowron (Eds.), *Proceedings of the 18th European conference on machine learning* (pp. 575–582). Berlin: Springer.

Freund, Y., Iyer, R. D., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, *4*, 933–969.

Friedman, J. (1996). *Local learning based on recursive covering*. Tech. Report, Dept. of Statistics, Stanford University, Stanford, CA 94305.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *6*, 393–425. IMS Reitz Lecture, 1999.

Hastie, T., & Tibshirani, R. (1990). *Generalized linear models*. New York: Chapman & Hall/CRC.

Hüllermeier, E., & Vanderlooy, S. (2008). *An empirical and formal analysis of decision trees for ranking*. Technical Report Computer Science Series 56. Philipps Universität Marburg.

Hüllermeier, E., & Vanderlooy, S. (2009). Why fuzzy decision trees are good rankers. *IEEE Transactions on Fuzzy Systems*. *17*(6), 1233–1244.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD'02: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 133–142). New York: ACM.

Lugosi, G., & Zeger, K. (1996). Concept learning using complexity regularization. *IEEE Transactions on Information Theory*, *42*(1), 48–54.

Mallat, S. (1990). *A wavelet tour of signal processing*. San Diego: Academic Press.

Massart, P. (2006). *Concentration inequalities and model selection. Lecture notes in mathematics*. Berlin: Springer.

Nobel, A. (2002). Analysis of a complexity-based pruning scheme for classification trees. *IEEE Transactions on Information Theory*, *48*(8), 2362–2368.

Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., & Salakoski, T. (2007). Learning to rank with pairwise regularized least-squares. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval* (pp. 27–33).

Provost, F., & Domingos, P. (2003). Tree induction for probability-based ranking. *Machine Learning*, *52*(3), 199–215.

Ripley, B. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Serfling, R. (1980). *Approximation theorems of mathematical statistics*. New York: Wiley.

Tsybakov, A. (2004). Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, *32*(1), 135–166.

Yu, P., Wan, W., & Lee, P. (2008). Analyzing ranking data using decision tree. In *Proceedings of the EMCL/PKDD'08 workshop on preference learning*.