Adaptive partitioning schemes for bipartite ranking How to grow and prune a ranking tree

Stéphan Clémençon, Marine Depecker, Nicolas Vayatis

Machine Learning manuscript No.
(will be inserted by the editor)

Adaptive partitioning schemes for bipartite ranking
How to grow and prune a ranking tree

Stéphan Clémençon · Marine Depecker · Nicolas Vayatis

Received: date / Accepted: date

Abstract Recursive partitioning methods are among the most popular techniques in machine learning. It is the purpose of this paper to investigate how such an appealing methodology may be adapted to the *bipartite ranking problem*, in order to elaborate a *global* learning method. Following in the footsteps of the TreeRank approach developed in [1], we present tree-structured algorithms designed for learning to rank/order instances based on classification data. Crucial questions concerning practical implementation of the TreeRank algorithm, those related to the splitting rule and the choice of the "right" size for the ranking tree namely, are tackled. From the angle embraced in this paper, splitting is viewed as a cost-sensitive classification task with data-dependent cost, so that, up to straightforward modifications, any classification algorithm may serve as a splitting rule. As for classification, we propose to implement a cost-complexity pruning method after the growing stage, in order to produce a "right-sized" sub- ranking tree with large AUC. In particular, performance bounds are established for pruning schemes inspired by recent work on nonparametric model selection. It is also discussed how to interpret a ranking tree and various simulation studies are eventually presented for illustration purpose.

1 Introduction

The goal of *bipartite ranking* procedures is to order/rank all possible values $x \in \mathcal{X}$ of a random variable X, modeling the available observation for predicting a random binary label $Y \in \{-1, +1\}$ based on a data sample $\{(X_i, Y_i) : \ 1 \leq i \leq n\}$. This generally boils down to build a *scoring function* $s : \mathcal{X} \to \mathbb{R}$ and use the natural order on the real line: one then expects that the higher the observed value $s(X)$ is, the more likely the event "$Y = +1$" should 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 screen-

Stéphan Clémençon
Telecom ParisTech - LTCI UMR Telecom ParisTech/CNRS No. 5141
E-mail: stephan.clemencon@telecom-paristech.fr

ing or anomaly detection in signal processing. However, until now, relatively few algorithms have been specifically elaborated for building a performant scoring function $s(x)$ from training data, the vast majority of ranking methods relies on the *plug-in* approach or consists of combining classifiers in an additive fashion (see [2]). The main difficulty lies in the *global* nature of the ranking problem, whereas, in contradistinction, popular classification rules, such as those obtained through recursive partitioning of the input space $\mathcal{X}$, are based on the concept of *local learning* (see [3]). Indeed, for such classification procedures, the predicted label of a given instance $x \in \mathcal{X}$ depends on the data lying in the subregion of the partition containing x solely, whereas, in contrast, the notion of ranking/ordering would rather involve comparing the subregions to each other, see [4] or [5].

In [1], a specific recursive partitioning method (RP), called *TreeRank* and producing piecewise constant scoring functions, has been thoroughly investigated. In this simple top-down approach, alike the RP, the related ordering is *tree-structured*, in a way that the ranking may be "read from the left to the right" at the bottom of the tree: instances belonging to the same subregion of the RP being tied. In addition, partitioning of the feature space has been related to approximation/estimation of the optimal ROC curve by 2-d splines and it has been established that, under general assumptions, the resulting piecewise linear ROC curve converges to the optimal one not only in the AUC sense but also in *sup-norm*, mimicking the performance of a nonlinear approximation scheme, which may be viewed as a *finite element method* (FEM) with implicit design. As a top-down RP strategy, TreeRank has the same drawback as the popular CART method (see [6]): it may be fooled by an XOR configuration, yielding inappropriate first splits and compromising then the results of the tree growing procedure. Additionally, it is enhanced here by the global nature of the ranking task, while in classification, given the local aspect of the decision rule, a bad start may be nevertheless compensated by growing the tree further at the cost of a certain amount of artificial complexity. In some sense, ranking errors are stacked as one grows the tree and the performance of the TreeRank algorithm is very sensitive to the splitting rule chosen.

It is the primary goal of this paper to propose pragmatic strategies for performing the *Optimization step* of the TreeRank algorithm efficiently, *i.e.* for splitting the cells in such a flexible manner that accurate approximants of bilevel sets of the regression function may be obtained. Partition-based splitting rules, adaptive or not, 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 ranking tree thus produced 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, involving no cross validation or resampling. In addition, some conditions under which such pruning schemes may be shown consistent in the AUC sense are exhibited.

The paper is organized as follows. Notations are first set out in Section 2, while briefly recalling crucial concepts of the bipartite ranking problem together with certain key results of ROC analysis and important properties of scoring rules that are piecewise

constant, as those produced by the algorithms presented in this paper. In Section 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 Section 4, while Section 5 deals with interpretation of tree-based ranking rules with perpendicular splits. Eventually, simulation results are presented in Section 6 for illustration purpose. Technical proofs are deferred to the Appendix.

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 the way we shall implement it.

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

Although it involves the same probabilistic setting, the bipartite ranking problem is less easy to state than the binary classification problem. Based on the observation of i.i.d. examples $\mathcal{D}_n = \{(X_i, Y_i) : \ 1 \leq i \leq n\}$, the goal is here to learn how to order all instances $x \in \mathcal{X}$ in a way that instances X such that $Y = +1$ with largest probability appear on 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 $F^{-1}(t) = \inf\{u \in \mathbb{R} : \ F(u) \geq t\}$ 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:

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

where, by convention, 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, $s(x)$'s ROC curve is nothing else than the graph of the mapping:

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

Remark 1 (Alternative convention) With the convention mentioned above, it is noteworthy that the curve ROC(s, .) is linear-by-parts as soon as $s(X)$'s conditional distributions are discrete. Another usual convention consists in defining ROC(s, .) as the graph of the mapping (2) in all cases. Equipped with this notation, when G_s or H_s are discrete, $s(x)$'s ROC curve 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 $s(x)$ everywhere:

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

We refer to [7] for a rigorous proof based on a standard Neyman-Pearson's 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 $s(x)$, ROC(s, .) is a concave curve as soon as the likelihood ratio $dG_s/dH_s(s(X))$ is monotone.

For notational simplicity, we set $H^* = H_\eta$ and $G^* = G_\eta$ as well as $Q^*(\alpha) = H^{*-1}(1-\alpha)$ for all $\alpha \in (0,1)$. We recall that if $Q^*(0) = \lim_{\alpha \to 0} Q^*(\alpha) < 1$ (*i.e.* $\eta(X)$'s essential supremum is strictly less than 1), H^* and G^* are differentiable and $H^{*\prime}$ is bounded by below by a strictly positive constant on its support, then ROC^* is twice differentiable on $[0,1]$ with bounded derivatives: $\forall \alpha \in [0,1]$, $\mathrm{ROC}^{*\prime}(\alpha) = (1-\mathrm{p})\mathrm{Q}^*(\alpha)/(\mathrm{p}(1-\mathrm{Q}^*(\alpha)))$ and $\mathrm{ROC}^{*\prime\prime}(\alpha) = (1-\mathrm{p})\mathrm{Q}^{*\prime}(\alpha)/(\mathrm{p}(1-\mathrm{Q}^*(\alpha))^2)$. Refer to Corollary 7 and Proposition 8 in [7] for further details.

The AUC **criterion.** In practice, the functional performance measure described above is generally summarized by a scalar feature, 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

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

Of course, $\mathcal{S}^*$ corresponds to the set of scoring functions with maximum AUC. We set:

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

The popularity of the AUC criterion mainly arises from the fact that it may be interpreted in a probabilistic manner.

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

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

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

Remark 2 (Optimal AUC) It has been shown in [8] that, when $\eta(X)$'s distribution is continuous, the maximal AUC depends on the dispersion of $\eta(X)$ through the relationship:

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

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 term $\mathbb{P}\{s(X) > s(X') \mid (Y, Y') = (+1, -1)\}$ solely.

2.2 Piecewise constant scoring functions

Here we focus on the simplest scoring functions, namely real-valued *piecewise constant* functions on the feature space $\mathcal{X}$. Any scoring function $s(x)$ of this type, taking $K \geq 1$ distinct values say, 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 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 $\lambda_1 > \ldots > \lambda_K$ 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 $\sigma \in S_K$.

Reciprocally, a partition $\mathcal{P} = \{C_1, \ldots, C_K\}$ including $\#\mathcal{P} = K$ non empty cells combined with a permutation $\sigma \in S_K$ 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 $\mathrm{ROC}(\mathrm{s}_{\mathcal{P},\sigma}, .)$.

Remark 4 (A global learning problem) 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 is useless, nearest neighbors rules are non sense for this problem and cells of $\mathcal{P}$ have to be compared to each other somehow, by means of the permutation $\sigma \in S_{\#\mathcal{P}}$ in the setup described above. For tree-structured partition however, unless otherwise specified, the ordering will be implicit, resulting from the left-right orientation of the underlying tree, see subsection 2.3.

Here is a list of basic properties of piecewise constant scoring functions. In order to formulate them rigorously, we introduce the following notations. We 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 $C \subset \mathcal{X}$. In the following result, the ROC curve of a piecewise constant scoring function and the corresponding AUC are explicited.

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: $\forall k \in \{1,\ \ldots,\ K\}$,*

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

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

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

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

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 notation and definitions are needed.

Definition 4 (Subpartition) Let $\mathcal{P}$ and $\mathcal{P}'$ be two partitions of $\mathcal{X}$. One says that $\mathcal{P}'$ is a *subpartition* of $\mathcal{P}$, when any cell $C' \in \mathcal{P}'$ may be written as the union of cells $C \in \mathcal{P}$. One then writes: $\mathcal{P}' \subset \mathcal{P}$.

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

Theorem 1 *(*AUC optimality*, [9]) Consider a partition of $\mathcal{X}$ with $K \geq 1$ non empty cells: $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$. Let $\sigma_{\mathcal{P}}^* \in S_K$ such that*

$$\frac{\beta(C_{\sigma_{\mathcal{P}}^*(1)})}{\alpha(C_{\sigma_{\mathcal{P}}^*(1)})} \geq \cdots \geq \frac{\beta(C_{\sigma_{\mathcal{P}}^*(K)})}{\alpha(C_{\sigma_{\mathcal{P}}^*(K)})}.$$

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

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

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

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

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.

Before discussing practical methods for generating partitions of the feature space in a data-driven fashion that are specifically tailored for the scoring problem, a few remarks are in order.

Remark 5 (On concavity) It is noteworthy that $\sigma^*_{\mathcal{P}}$ corresponds to permutations $\sigma \in \mathcal{S}_K$ making the piecewise linear curve $\mathrm{ROC}(s_{\mathcal{P},\sigma}, .)$ concave, as $\mathrm{ROC}^*(.)$.

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 the C_k's 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 $\mu(C_k) = p\alpha(C_k) + (1-p)\beta(C_k)$ 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 [9] for a similar result with different notations).

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

$$\mathrm{AUC}^* - \mathrm{AUC}(\mathrm{s}^*_{\mathcal{P}}) \leq \frac{||\eta_{\mathcal{P}}(\mathrm{X}) - \eta(\mathrm{X})||_{\mathrm{L}_1(\mu)}}{\mathrm{p}(\mathrm{p}-1)} + \frac{1}{4\mathrm{p}(1-\mathrm{p})} \sum_{\mathrm{k}=1}^{\mathrm{K}} \mathcal{G}(\mathrm{C_k}),$$

where, for all $k \in \{1, \ldots, K\}$, $\mathcal{G}(C_k) = \mathbb{E}[|\eta(X) - \eta(X')| \cdot \mathbb{I}\{(X, X') \in C_k^2\}]$ 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 $\mathcal{D}_n = \{(X_i, Y_i);\ 1 \leq i \leq n\}$. The relevance of a candidate $s(x)$ is thus evaluated by plotting the empirical version of its ROC curve.

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

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

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

Let $\sigma \in S_n$ be such that $\hat{\alpha}_{\sigma(1)} \leq \ldots \leq \hat{\alpha}_{\sigma(n)}$ 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: $\forall i \in \{1, \ldots, n\}$, $\forall \alpha \in [\hat{\alpha}_{\sigma(i-1)}(s),\ \hat{\alpha}_{\sigma(i)}(s)[$,

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

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

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

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. This 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 $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k+1}$, 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.

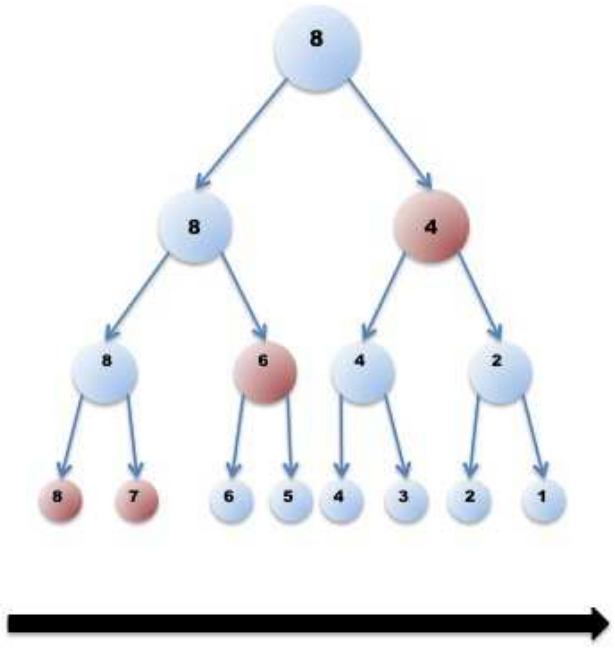


Fig. 1 A tree-structured ranking rule. A score is assigned to each cell. The restriction of these values to the outer leaves of any subtree of the master ranking tree produces a scoring rule which order the corresponding cells according to the left-right orientation.

2.3 The TreeRank approach

We now recall the specific method proposed in [1] for adaptively generating a tree-structured partition of the feature space $\mathcal{X}$ in ordered cells. Precisely, the piecewise constant scoring rule it outputs is described by a *master ranking tree*, each of whose terminal leaves corresponds to a unique cell of the partition, ordering of the cells being simply obtained by perusing the terminal leaves from the left to the right at the bottom of the tree.

Assume that a training data set $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ of n independent copies 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 $\mathcal{X}$, on which attainable partitions are based. Let $D \geq 1$ be fixed.

TreeRank Algorithm

1. **Initialization.** Set $C_{0,0} = \mathcal{X}$.
2. **Iterations.** For $d = 0, \ldots, D-1$ and $k = 0, \ldots, 2^d - 1$:
   (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:
   $$C_{d+1,2k} = \operatorname*{arg\,max}_{C \in \mathcal{C},\ C \subset C_{d,k}} \widehat{\Lambda}_{d,k+1}(C) \ .$$
   Then, set $C_{d+1,2k+1} = C_{d,k} \setminus C_{d+1,2k}$.
   (b) (Update.) Set
   $$\begin{aligned} \alpha_{d+1,2k+1} &= \alpha_{d,k} + \widehat{\alpha}(C_{d+1,2k}) \\ \beta_{d+1,2k+1} &= \beta_{d,k} + \widehat{\beta}(C_{d+1,2k}) \end{aligned}$$
   and
   $$\begin{aligned} \alpha_{d+1,2k+2} &= \alpha_{d,k+1} \\ \beta_{d+1,2k+2} &= \beta_{d,k+1} \ . \end{aligned}$$
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 $\mathrm{ROC}(\mathrm{s_D}, .)$, namely the broken line $\widehat{\mathrm{ROC}}(s_D, .)$ that connects the knots $\{(\alpha_{D,k}, \beta_{D,k}) : \ k = 0, \ldots, 2^D\}$, and the following estimate of $\mathrm{AUC}(\mathrm{s_D})$:
   $$\widehat{\mathrm{AUC}}(s_D) = \int_{\alpha=0}^{1} \widehat{\mathrm{ROC}}(s_D, \alpha) d\alpha = \frac{1}{2} + \frac{1}{2} \sum_{k=0}^{2^{D-1}-1} \widehat{\Lambda}_{D-1,k+1}(C_{D,2k}).$$

Remark 6 (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 set $C_{d+1,2k} = C_{d,k}$ and $C_{d+1,2k+1} = \emptyset$.

Remark 7 (On concavity (bis).) 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 is not necessarily concave, see Proposition 21 in [7]. 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, *cf* Remark 5. If the $C_{d,k}$'s are built by aggregating elementary subsets, such as cubes of a grid partition of the feature space $\mathcal{X}$ say (see subsection 3.2), concavity is of course guaranteed. However, this property is not satisfied in general, when candidates are produced recursively, by applying a simple cutting rule at each step to the current node, see Section 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. The latter may be described as follows, one may refer to Section D in [7] 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 $\{(\alpha^*_{1,k}, \mathrm{ROC}^*(\alpha^*_{1,\mathrm{k}})) : \ \mathrm{k} \in \{0,1,2\}\}$ with minimum L_1-distance to the target curve ROC^*, or, equivalently, with maximum AUC. We point out that this is also the best interpolant with two linear pieces in terms of sup-norm, see Proposition 20 in [7] 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 interpolant thus obtained. At step D, a tree-structured subdivision $\alpha^*_{D,0} = 0 < \alpha^*_{D,1} < \ldots < \alpha^*_{D,2^D} = 1$ of the unit interval has then been produced, yielding a linear-by-parts interpolant 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 $\forall k \in \{0, \ldots, 2^d\}$,

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

where

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

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:

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

Like the subdivision $\{\alpha^*_{D,k} : \ k = 0, \ldots, 2^D\}$ of the unit interval, this partition is obtained recursively through the procedure described above and is thus related to a tree-structure as well: $\forall d \geq 0$, $\forall k \in \{0, \ldots, 2^d\}$, $C^*_{d,k}$ splits into $C^*_{d,2k}$ 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 [7] 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 Section VI of [7] 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 $\mathcal{P}_D = \{C_{D,k} : \ k = 0, \ldots, 2^D - 1\}$ close to the ideal partition $\mathcal{P}^*_D$.

3 Splitting for Ranking

In this section, we focus on practical implementation of the *Optimization step* of the TreeRank algorithm. In a preliminary fashion, we precisely set the goals of the splitting rule from the perspective of AUC maximization and 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 $\{x \in \mathcal{X} : \ \eta(x) \geq 1/2\} \bigcup \{x \in \mathcal{X} : \ \eta(x) < 1/2\}$, 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 revealed 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:*

$$\mathrm{AUC(s)} = \frac{1}{2} + \frac{1}{2}\left(\beta(\mathrm{C}) - \alpha(\mathrm{C})\right) \leq \mathrm{AUC}(\mathrm{s}^*_1). \tag{5}$$

More precisely, the following identity holds:

$$\mathrm{AUC}(\mathrm{s}^*_1) - \mathrm{AUC(s)} = \frac{1}{2\mathrm{p}(1-\mathrm{p})} \cdot \mathbb{E}[|\eta(\mathrm{X}) - \mathrm{p}| \cdot \mathbb{I}\{\mathrm{X} \in \mathrm{C}^* \Delta \mathrm{C}\}], \tag{6}$$

where Δ denotes the symmetric difference between sets.
In addition, we have

$$\mathrm{AUC}(\mathrm{s}^*_1) = \frac{1}{2\mathrm{p}(1-\mathrm{p})} \mathbb{E}\left[\max\{(1-\mathrm{p})\eta(\mathrm{X}),\ \mathrm{p}(1-\eta(\mathrm{X}))\}\right]. \tag{7}$$

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 $\mathrm{AUC}(\mathrm{s}_1^*)$. In addition, when the optimal ROC curve is differentiable and strictly concave (see subsection 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 $\mathrm{ROC}(\mathrm{s}_1^*)$ is the broken line defined by the point of ROC^* where the tangent has a slope equal to 1, see Fig. 2. We point out that, under the set of assumptions listed in subsection 2.1, $(1-p)/p$ always belongs to $[\mathrm{ROC}^{*\prime}(1), \mathrm{ROC}^{*\prime}(0)]$, since this condition amounts to suppose 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 Section C in [7] for further details.

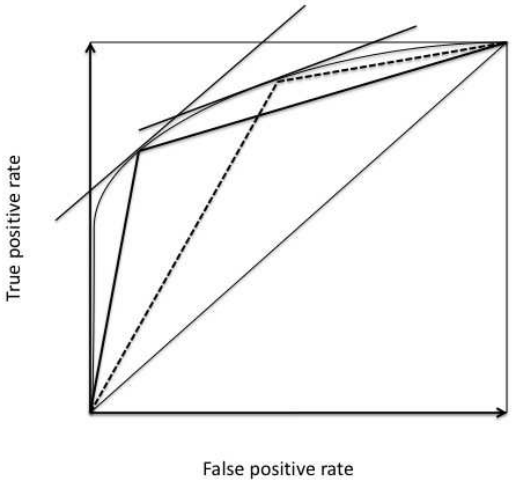


Fig. 2 ROC curves: optimal binary scoring function (solid broken line) *vs.* Bayes classifier (dotted broken line) in a situation where $p > 1/2$.

Bipartite ranking as a collection of imbricated binary scoring problems. In the following we propose data-driven procedures for constructing a binary scoring function with AUC close to $\mathrm{AUC}(\mathrm{s}_1^*)$. In a "fractal manner", when running the TreeRank algorithm, such a procedure will be iteratively applied to the subsample lying in each cell C of the current tree-structured partition. Indeed, 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:

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

Equipped with this notation, we have indeed: $\forall j \geq 0$, $\forall k \in \{0, \ldots, 2^j - 1\}$,

$$C^*_{j+1,2k} = \underset{C \subset C^*_{j,k}}{\arg\max}\, \mathrm{AUC}(2 \cdot \mathbb{I}_{\mathrm{C}} + \mathbb{I}_{\mathrm{C}^*_{\mathrm{j,k}} \setminus \mathrm{C}} \mid \mathrm{C}^*_{\mathrm{j,k}}).$$

We underline that, within the TreeRank approach, bipartite ranking boils down to solve a collection of "nested" binary scoring problems, in contrast to the Rankboost method developped by [2], that 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*.

Partition-based splitting rule

1. (Input.) Data $\mathcal{D}_n = \{(X_i, Y_i) : \ 1 \leq i \leq n\}$ in the region $\mathcal{X}$, partition $\{C_1, \ldots, C_K\}$ of $\mathcal{X}$ with $K \geq 1$.
2. ("Concavification" step.) Compute $\sigma \in S_K$ such that:
$$\frac{\widehat{\beta}(C_{\sigma(1)})}{\widehat{\alpha}(C_{\sigma(1)})} \geq \ldots \geq \frac{\widehat{\beta}(C_{\sigma(K)})}{\widehat{\alpha}(C_{\sigma(K)})},$$
where $\widehat{\alpha}(.)$ and $\widehat{\beta}(.)$ denote the empirical false and true positive rates based on the sample $\mathcal{D}_n$.
3. (Merging step.) $\forall k \in \{1, \ldots, K\}$, set $L_k = \bigcup_{l \leq k} C_{\sigma(l)}$ and compute the entropic measure $\widehat{\Lambda}(k) = \widehat{\beta}(L_k) - \widehat{\alpha}(L_k)$. Let
$$k^* = \underset{1 \leq k \leq K}{\arg\max} \left\{ \widehat{\beta}(L_k) - \widehat{\alpha}(L_k) \right\}.$$
4. (Output.) Form the leaves:
$$L = L_{k^*} \text{ and } R = L \setminus \mathcal{X}.$$

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

Proposition 4 *Let $\mathcal{P} = \{C_k\}_{1 \leq k \leq K}$ 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 subset $C \subset \mathcal{X}$ formed by the union of some cells in $\mathcal{P}$, we have:*

$$\widehat{\mathrm{AUC}}(s) \leq \widehat{\mathrm{AUC}}(\widehat{s}^*),$$

where $s = 2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X} \setminus C}$.

This simply results from Theorem 1 applied to the empirical distribution of the (X_i, Y_i)'s, details are omitted. We point out that, although there are $2^K - 2$ different binary scoring functions that may be built from $\mathcal{P}$, this result shows that the empirical optimum may be attained in $O(K \log K)$ operations by means of an efficient sorting algorithm.

Uniform partitions. Rather than translating and/or rescaling the input vector X, we suppose, for simplicity, that $\mathcal{X} = [0,1]^q$ in the subsequent analysis 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 $0 \leq k_l < 2^j$ for all $l \in \{1, \ldots, q\}$. Note that the partition has cardinality $\#\mathcal{P}(j) = 2^{jq}$. We denote by $\widehat{L}_j$ the subregion L output when implementing 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. Provided it is regular enough, it is reasonable to expect that the level set $\{x \in \mathcal{X} : \ \eta(x) \geq p\}$ may be accurately estimated from a collection of such cubes when the latter 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 $j \geq 1$, denote by $\mathcal{P}_{2,j}$ the collection of partitions of $\mathcal{X}$ made of two non empty sets, obtained by union 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 $c < \infty$ such that for all $\delta \in (0,1)$, we have with probability at least $1 - \delta$: for $n \geq 1$ large enough and for all $j \geq 1$,*

$$\mathrm{AUC}(\mathrm{s}_1^*) - \mathrm{AUC}(\widehat{\mathrm{s}}_{\mathrm{j(n)}}^*) \leq \mathrm{c} \cdot \frac{2^{\mathrm{jq}}}{\sqrt{\mathrm{n}}} + \left\{ \mathrm{AUC}(\mathrm{s}_1^*) - \max_{\mathrm{s} \in \mathcal{S}_{\mathcal{P}_{2,\mathrm{j}}}} \mathrm{AUC}(\mathrm{s}) \right\}. \tag{8}$$

Remark 8 (Bias, smoothness assumptions and model selection) Classically, under smoothness assumptions on the level set $\{x \in \mathcal{X} : \ \eta(x) \geq p\}$, 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:

$$\mathrm{AUC}(\mathrm{s}_1^*) - \mathrm{AUC}(\mathrm{s}) \leq \frac{||\mathrm{d}\mu/\mathrm{dx}||_\infty}{2\mathrm{p}(1-\mathrm{p})} \cdot \lambda(\mathrm{C}^* \Delta \mathrm{C}),$$

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 η's continuity), 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 [10]. 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 [9].

Remark 9 (On faster rates of convergence) Neglecting the bias component, which boils down to assume that C^* belongs to the collection of subset candidates $\mathcal{C}$, faster rates of convergence may be attained, as in the classification setting, except that, here, it is the behavior of $\eta(x)$ in the vicinity of p that describes the complexity

of the problem. Under the following extension of Massart's noise condition, stipulating that there exists some constant $c > 0$ such that

$$|\eta(X) - p| \geq c \ \text{ a.s. },$$

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

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

for some $a \in [0, 1]$. Following line by line the argument in [11], 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 $\eta(X)$ cumulative df. 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 get a rate bound of order $n^{-2/3}$.

Remark 10 (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 7.

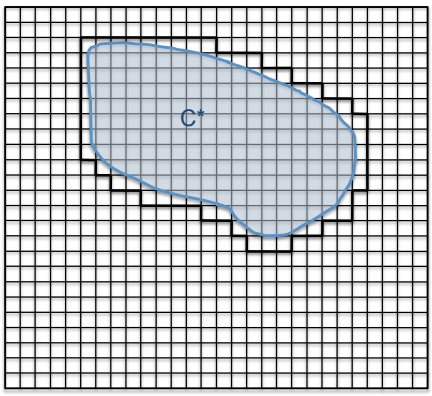


Fig. 3 Approximating a regression level set in $2 - d$ using a uniform grid: the thick broken line delineates the collection of hypercubes that intersect the target subset.

The LeafRank 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 be naturally chosen depending on the data. A possible strategy could consist of implementing the rule described above from the partition adaptively generated by TreeRank based on a simple splitting criterion.

LeafRank Algorithm

1. (Input.) Data $\{(X_i, Y_i) : \ 1 \leq i \leq n\}$ in the region $\mathcal{X}$, depth $d \geq 1$.
2. (Growing step.) Run TreeRank with a naive splitting rule at depth d, yielding a ranking tree with terminal leaves:
$$C_{d,k}, \ \ k = 0, \ldots, 2^d - 1.$$
3. (Partition-based splitting rule.) Apply the partition-based splitting rule from the partition $\mathcal{P}_d = \{C_{d,k} : \ 0 \leq k < 2^d\}$.

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 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 [12], 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 *([12]) 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:

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

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

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

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 $\{2 \cdot \mathbb{I}_C + \mathbb{I}_{\mathcal{X}\setminus C} : \ C \in \mathcal{C}\}$.

Weighted ERM Algorithm

1. (Input.) Data $\{(X_i, Y_i) : \ 1 \leq i \leq n\}$ lying in the region $\mathcal{X}$, class $\mathcal{C}$ of subset candidates.
2. (Asymmetry factor.) Compute the number of positive instances lying in the region $\mathcal{X}$: $n_+ = \sum_{i=1}^n \mathbb{I}\{Y_i = +1\}$. Take $\omega = n_+/n$ as asymmetry factor.
3. (Weighted ERM.) Compute the *weighted empirical risk minimizer*:

$$L = \arg\min_{C \in \mathcal{C}} \widehat{\mathcal{L}}_\omega(C)$$

and set $R = \mathcal{X} \setminus L$.

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 be then viewed as a recursive implementation of the weighted CART growing procedure, in which the weight is locally updated at each iteration, chosen as the rate of positive instances within the cell to split. Figure 4 below illustrates this view. This AUC splitting procedure could be refined by applying a pruning procedure to the classification tree obtained, see [6] or [13] for instance. SVM or recent procedures like *Bagging* or *Boosting* can also be considered in order to improve efficiency of the *Optimization step*.

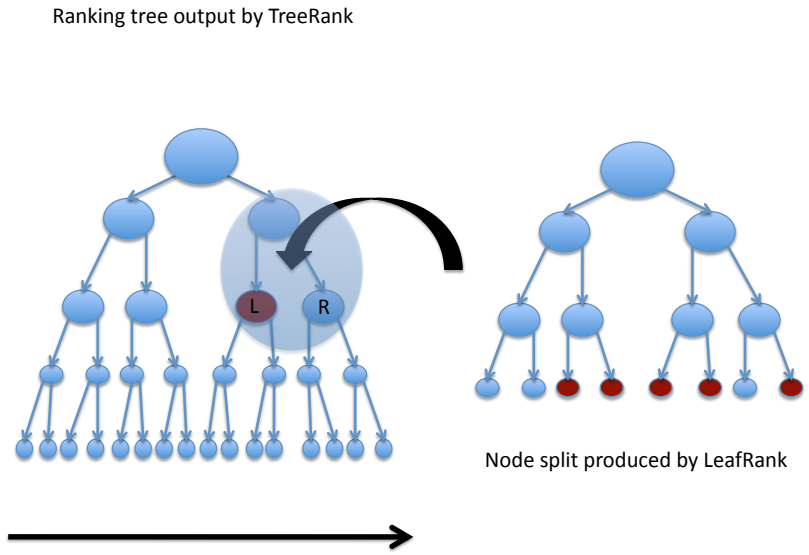


Fig. 4 Schematic of the TreeRank algorithm, where each split is obtained through the LeafRank procedure: a naive TreeRank implementation, followed by concavification and merging steps. The final ranking is read at the bottom of the tree from the left to the right.

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 $\mathcal{P}(\mathcal{T})$, 2^D namely. If the depth D is chosen too small, the ROC curve associated to the ranking tree produced will not permit to mimic ROC*'s variability, 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 [13]. 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 $\mathcal{P}(\mathcal{T}) = \{C_{D,k},\ 0 \leq k < 2^D\}$ by 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 $0 \leq d \leq D$ and $0 \leq k < 2^D$, 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 $d \in \{0, \ldots, D\}$ and $k \in \{0, \ldots, 2^D - 1\}$, 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* subsection 2.2)

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

Indeed one may check that the ordering defined by $S_{\mathcal{P}(\mathcal{T}(\omega))}$ coincides with the one determined by the tree $\mathcal{T}(\omega)$ when left-right oriented, see Fig. 5. 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

$$\omega^* = \arg\max_{\omega} \mathrm{AUC}(\mathrm{S}_{\mathcal{P}(\mathcal{T}(\omega))}), \tag{11}$$

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 $\mathrm{AUC}(\mathrm{S}_{\mathcal{P}(\mathcal{T}(\omega))})$ by an estimate

$$\begin{aligned} \widehat{\mathrm{AUC}}'(S_{\mathcal{P}(\mathcal{T}(\omega))}) = {} & \frac{1}{n'_+ n'_-} \sum_{i:\, Y_i=+1} \sum_{j:\, Y_j=-1} \mathbb{I}\{S_{\mathcal{P}(\mathcal{T}(\omega))}(X_i) > S_{\mathcal{P}(\mathcal{T}(\omega))}(X_j)\} \\ & + \frac{1}{2} \frac{1}{n'_+ n'_-} \sum_{i:\, Y_i=+1} \sum_{j:\, Y_j=-1} \mathbb{I}\{S_{\mathcal{P}(\mathcal{T}(\omega))}(X_i) = S_{\mathcal{P}(\mathcal{T}(\omega))}(X_j)\}, \end{aligned} \tag{12}$$

based on a dataset $\mathcal{D}'_{n'} = \{(X'_1, Y'_1), \ldots, (X'_{n'}, Y'_{n'})\}$ 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 one will obtain $\mathcal{T}(\omega^*) = \mathcal{T}$. However, in many applications, there is insufficient data to split them into two large enough separate subsets and all available data are used in the training stage. We next propose two approaches for model selection in this situation.

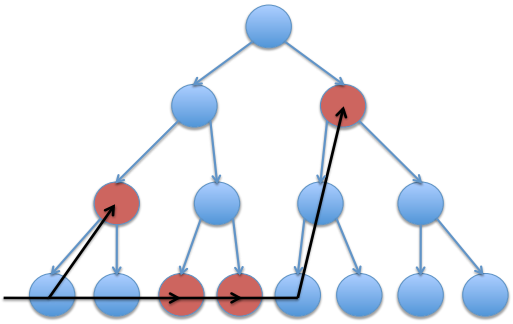


Fig. 5 A pruned ranking tree: terminal nodes are colored red, top ranked cells are closest to the bottom left corner of the tree.

4.1 A cross-validation based procedure

We start off by adapting the pruning method proposed by [6] 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 $\widehat{\mathrm{AUC}}(S_{\mathcal{P}(\mathcal{T}(\omega))})$ a *linear* complexity term that penalizes large ranking trees. Thus, with

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

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:

$$\omega_\lambda^* = \arg\max_\omega \widehat{\mathrm{CPAUC}}(S_{\mathcal{P}(\mathcal{T}(\omega))}, \lambda).$$

It remains to choose λ, which issue we now discuss. 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 $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_m = \infty$ such that*

$$root = \mathcal{T}(\omega_{\lambda_m}^*) \subseteq \ldots \subseteq \mathcal{T}(\omega_{\lambda_1}^*) \subseteq \mathcal{T}(\omega_{\lambda_0}^*) = \mathcal{T},$$

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 [6], see also [14].

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 [15], [13] or [16]. 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:

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

where $\mathrm{pen}\,(K, n)$ 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

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

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

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

with $K \in \{1, \ldots, 2^D\}$, see Proposition 6 in subsection 8.5. As shown in [8] (see also [17]), 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 Section 3 and yielding different, nonlinear this time, penalties for model selection.

$\mathbf{O_1}$: Splits are obtained through the LeafRank procedure with at most k perpendicular cuts, $k \geq 1$.

$\mathbf{O_2}$: 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 $\mathbf{O_1}$ rule, then set: $\forall (K,k) \in \mathbb{N}^{*2}$,

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

(*ii*) *If splits are optimized using the* $\mathbf{O_2}$ *rule, then set:* $\forall (K,j) \in \mathbb{N}^{*2}$,

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

Then, the expected deficit of AUC *of the ranking sub-tree maximizing the complexity-penalized area under the* ROC *curve is bounded as follows:*

$$\mathrm{AUC}^* - \mathbb{E}[\mathrm{AUC}(\mathrm{S}_{\mathcal{P}(\mathcal{T}(\widetilde{\omega}_{\mathrm{n}}^*))})] \leq \inf_{1\leq \mathrm{K}\leq 2^{\mathrm{D}}} \mathcal{B}(\mathrm{K},\mathrm{n}), \tag{14}$$

where

$$\mathcal{B}(K,n) = cst \cdot pen(K,n) + \left\{\mathrm{AUC}^* - \sup_{\mathrm{s}\in\mathcal{S}_{\mathrm{T}}(\mathrm{K})} \mathrm{AUC}(\mathrm{s})\right\}.$$

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.

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* $\mathbb{E}[\mathrm{AUC}(\mathrm{S}_{\mathcal{T}_{\mathrm{n}}(\omega_{\mathrm{n}})})] \to \mathrm{AUC}^*$, *as* $n \to \infty$. *Assume in addition that:*
(*i*) *if* $\mathcal{T}_n$ *is grown through the* $\mathbf{O_1}$ *splitting rule with* $k = k(n)$ *axis-parallel splits, then*

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

(*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*

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

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

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

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}, $j = j(n) \geq 1$, *and depth* D_n. *If in addition, as* $n \to \infty$, $j(n) \to \infty$ *and the sizes of the cells* $\{C_k^{(n)} : k = 0, \ldots, 2^{D_n+1}-1\}$ *of the related partition* $\mathcal{P}(\mathcal{T}_n)$ *uniformly shrink to zero in the sense that* $\max_{0\leq k<2^{D_n+1}} \mu(C_k^{(n)}) \to 0$, *then the pruned ranking trees* $\mathcal{T}_n(\widetilde{\omega}_n^*)$ *obtained from* $\mathcal{T}_n$ *are* AUC *consistent.*

Remark 11 (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 [8].

Remark 12 (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.

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. 4 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, here we 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 split 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 [6] 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 $X = (X^{(1)}, \ldots, X^{(d)})$. 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: $\forall j \in \{1, \ldots, d\}$,

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

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 $X^{(1)}, \ldots, X^{(q)}$ 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 $X = (X^{(1)}, \ldots, X^{(q)})$ corresponding to a given subset of indexes $I_0 \subset \{1, \ldots, q\}$. Denote by $I_1 = \{1, \ldots, q\} \setminus I_0$ 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 XI_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 subsection 8.2 in [18] 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 subsection 4.1 with $N = 10$.

6.1 First example - Mixtures of uniform distributions

The artificial data sample represented in Fig. 6a has been generated as follows. We have split the unit square $\mathcal{X} = [0,1]^2$ into four quarters: $\mathcal{X}_1 = [0,1/2]^2$, $\mathcal{X}_2 = [1/2,1] \times [0,1/2]$, $\mathcal{X}_3 = [1/2,1]^2$ 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

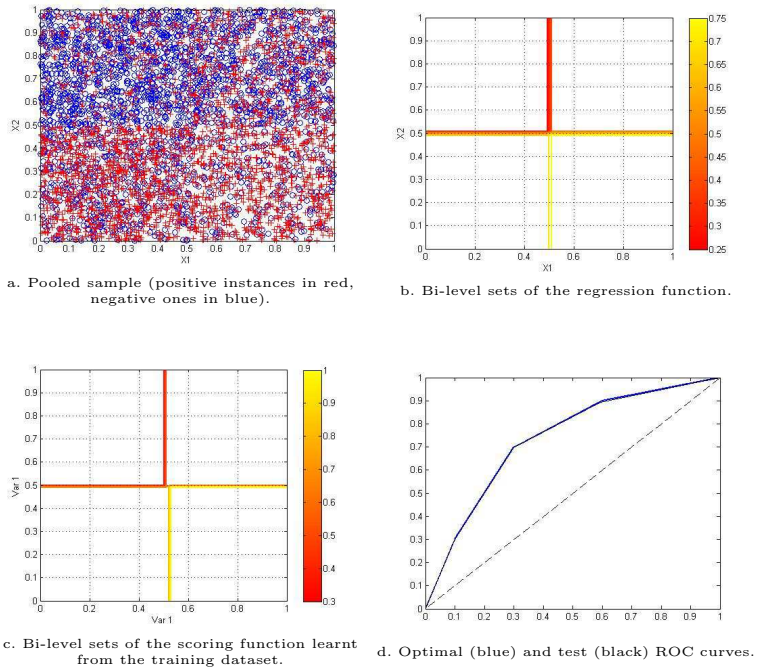
$$H(dx) = 0.2 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.1 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_3} + 0.4 \cdot \mathcal{U}_{\mathcal{X}_4},$$
$$G(dx) = 0.4 \cdot \mathcal{U}_{\mathcal{X}_1} + 0.3 \cdot \mathcal{U}_{\mathcal{X}_2} + 0.2 \cdot \mathcal{U}_{\mathcal{X}_3} +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

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. 6. In the growing stage, splits have been obtained through the LeafRank method by constraining the number of terminal nodes to be less than 5.



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. 6 First example - Mixtures of uniform distributions

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.

6.2 Second example - conditional Gaussian distributions

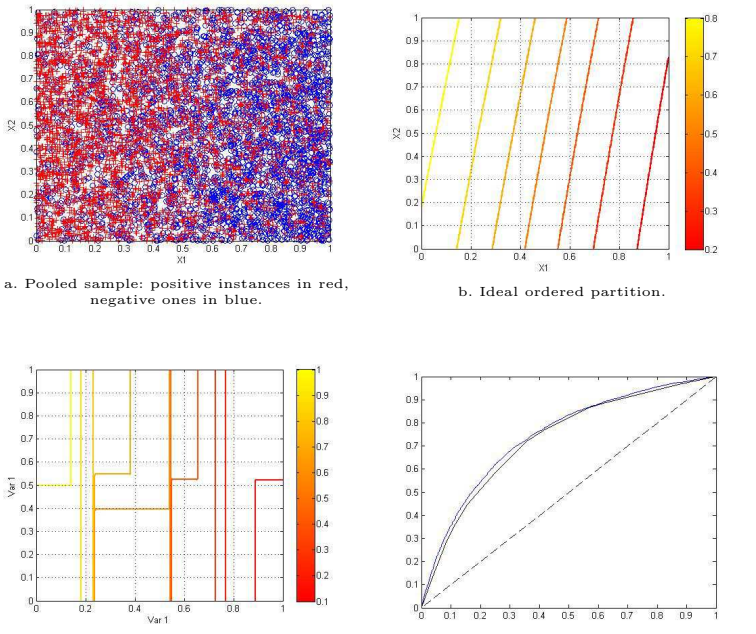
Considering a q-dimensional Gaussian random vector Z, drawn as $\mathcal{N}(m, \Gamma)$, and a borelian set $C \subset \mathbb{R}^q$ weighted by $\mathcal{N}(m, \Gamma)$, we denote by $\mathcal{N}_C(m, \Gamma)$ the conditional distribution of Z given $Z \in C$. Equipped with this notation, the class 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. 7a, while the level sets of the regression function related to the approximation scheme mimicked by TreeRank are represented in Fig. 7b. For comparison purpose, the level sets of the piecewise scoring function output by the learning method are displayed in Fig. 7c and its test ROC curve is plotted in Fig. 7d, together with the optimal one.



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

b. Ideal ordered partition.

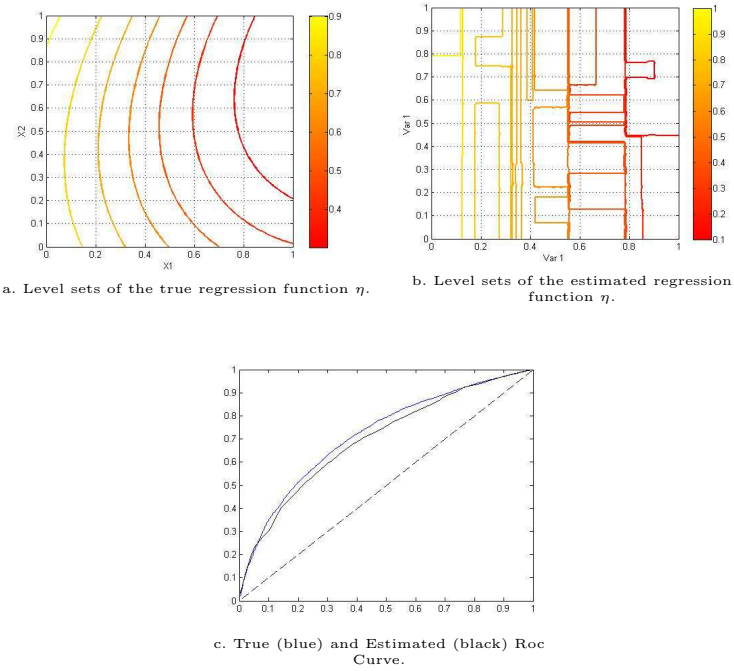
c. Ordered partition learnt from the training dataset.

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

Fig. 7 Second example - Mixture of conditional Gaussian distributions.

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. 7, results are still satisfactory when taking Gaussian with closer centers.



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. 8 Results on the tougher gaussian mixture model

7 Conclusion

Summarize what we achieved in this paper and enhance limitations due to the hierarchical structure and the pileup of errors. Announcement for further simulation studies, "ranking pursuit" in order to overcome the error pileup phenomenon, the bagging of ranking trees for guaranteeing more stability.

8 Technical Proofs

8.1 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 $\sigma \in S_K$, fix $k \in \{1, \ldots, K-1\}$ and let $\tau_k \in S_K$ 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*

$$\mathrm{AUC}(\mathrm{s}_{\mathcal{P},\sigma}) \geq \mathrm{AUC}(\mathrm{s}_{\mathcal{P},\sigma\circ\tau_\mathrm{k}}).$$

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

$$\mathrm{AUC}(\mathrm{s}_{\mathcal{P},\sigma}) - \mathrm{AUC}(\mathrm{s}_{\mathcal{P},\sigma\circ\tau_\mathrm{k}}) = \frac{1}{2}\left\{\beta(\mathrm{C}_{\sigma(\mathrm{k}+1)})\alpha(\mathrm{C}_{\sigma(\mathrm{k})}) - \beta(\mathrm{C}_{\sigma(\mathrm{k})})\alpha(\mathrm{C}_{\sigma(\mathrm{k}+1)})\right\},$$

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, $k \in \{1, \ldots K-1\}$, the proof of the first part of the theorem immediately follows from the lemma stated above. The second part straightforwardly results from Eq. (4) in Proposition 2.

8.2 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 $\mathcal{P} = \{C_k\}_{1\leq k\leq K}$ with $K \geq 2$ non empty cells, we have: $\forall s \in \mathcal{S}_{\mathcal{P}}$,*

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

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:

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

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

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

Combined to Lemma 3, this establishes the desired bound.

8.3 Proof of Lemma 1

Observe first that:

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

Now the lemma results from the fact that:

$$\begin{aligned} 2p(1-p)\{\mathrm{AUC}(\mathrm{s}_1^*)-\mathrm{AUC(s)}\} &= \mathbb{E}[(1-p)\eta(X)\cdot(\mathbb{I}\{X\in C^*\}-\mathbb{I}\{X\in C\})] \\ &\quad+\mathbb{E}[p(1-\eta(X))\cdot(\mathbb{I}\{X\notin C^*\}-\mathbb{I}\{X\notin C\})] \\ &= \mathbb{E}[|\eta(X)-p|\cdot\mathbb{I}\{X\in C\Delta C^*\}]. \end{aligned}$$

8.4 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: $\forall j\geq 1$,

$$\widetilde{L}_j^* = \arg\max_{C\in\mathcal{C}_j}\{\beta(C)-\alpha(C)\}$$

as well as

$$\widehat{L}_j^* = \arg\max_{C\in\mathcal{C}_j}\{\beta(\widehat{C}_j)-\alpha(\widehat{C}_j)\}.$$

We denote the related binary scoring functions by:

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

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

$$\begin{aligned} \mathrm{AUC}(\mathrm{s}_1^*)-\mathrm{AUC}(\widetilde{\mathrm{s}}_\mathrm{j}^*) &= \{\mathrm{AUC}(\mathrm{s}_1^*)-\mathrm{AUC}(\widetilde{\mathrm{s}}_\mathrm{j}^*)\}+\{\mathrm{AUC}(\widetilde{\mathrm{s}}_\mathrm{j}^*)-\widehat{\mathrm{AUC}}(\widetilde{\mathrm{s}}_\mathrm{j}^*)\} \\ &\quad+\{\widehat{\mathrm{AUC}}(\widetilde{s}_j^*)-\widehat{\mathrm{AUC}}(\widehat{s}_j^*)\}+\{\widehat{\mathrm{AUC}}(\widehat{s}_j^*)-\mathrm{AUC}(\widehat{\mathrm{s}}_\mathrm{j}^*)\} \\ &\leq \mathrm{AUC}(\mathrm{s}_1^*)-\mathrm{AUC}(\widetilde{\mathrm{s}}_\mathrm{j}^*)+2\sup_{\mathrm{s}\in\mathcal{S}_{\mathcal{P}_{2,\mathrm{j}}}}|\widehat{\mathrm{AUC}}(\mathrm{s})-\mathrm{AUC(s)}|. \end{aligned}$$

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 section 5.6 of [19]) 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 $p \in [\underline{p}, \bar{p}]$.

8.5 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 $\mathcal{S}_T(K)$:

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

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:* $\forall (K,k) \in \mathbb{N}^{*2}$,

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

(ii) If splits are optimized using the $\mathbf{O_2}$ *rule and the penalization is chosen accordingly, then:* $\forall (K,j) \in \mathbb{N}^{*2}$,

$$\mathbb{P}\left\{ \sup_{s \in \mathcal{S}_T(K)} \mathrm{AUC(s)} - \mathrm{AUC}(\widetilde{\mathrm{S}}^*_{\mathrm{n}}) \geq \epsilon \right\} \leq 4K2^{jq} e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/8} + e^{-n\underline{p}^2(1-\bar{p})^2\epsilon^2/2}.$$

Proof We follow the argument of [20], see also Section 18.1 in [21]. Write: $\forall \epsilon > 0$, $\forall K \geq 1$,

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

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

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

Turning to the second term, observe that

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

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

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

• We first place ourselves in the situation $\mathbf{O_1}$, where *Optimization steps* are performed using at most k perpendicular splits. We follow the approach developed in [8] 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 [8]).

Lemma 5 *([8]) 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:

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

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: $\forall\epsilon$, $\forall n\geq 1$,

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

Thus, for n large enough, we have

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

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

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

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

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

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 lenght 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 section 5.6 of [19]) 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

$$\mathrm{AUC}^* - \mathbb{E}\left[\mathrm{AUC}(\widetilde{\mathrm{S}}_\mathrm{n}^*)\right] = \inf_{\mathrm{K}\geq 1}\{(\mathrm{AUC}^* - \sup_{\mathrm{s}\in\mathcal{S}_\mathrm{T}(\mathrm{K})} \mathrm{AUC}(\mathrm{s})) + (\sup_{s\in\mathcal{S}_T(K)} \mathrm{AUC}(\mathrm{s}) - \mathbb{E}[\mathrm{AUC}(\widetilde{\mathrm{S}}_\mathrm{n}^*)])\}.$$

Therefore,

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

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

8.6 Proof of Proposition 7

We place ourselves in the $\mathbf{O_2}$ case. Given Corollary 1, it suffices to show that

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

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 $\widetilde{\mathcal{T}_n}$ 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 10. By virtue of Proposition 3, we thus have:

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

where $\mathcal{G}(C_{n,k}) = \mathbb{E}\left[|\eta(X) - \eta(X')| \cdot \mathbb{I}\{(X,X') \in C_{n,k}^2\}\right]$. 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 Eq. (19) vanishes as $n \to \infty$. As Theorem 6.1's argument in [21] ensures that the term on the left hand side also goes to 0 as $n \to \infty$, the result is then proved.

References

1. Clémençon, S., Vayatis, N.: Tree-structured ranking rules and approximation of the optimal ROC curve. In: ALT '08: Proceedings of the 2008 conference on Algorithmic Learning Theory (2008)
2. Freund, Y., Iyer, R.D., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research **4**, 933–969 (2003)
3. Friedman, J.: Local learning based on recursive covering. Tech. Report, Dept. of Statistics, Stanford University, Stanford, CA 94305 (1996)
4. P. Flach, E.T.M.: A simple lexicographic ranker and probability estimator. In: Proceedings of the 18th European Conference on Machine Learning. Joost N. Kok, Jacek Koronacki, Ramon Lopez de Mantaras, Stan Matwin, Dunja Mladenic, Andrzej Skowron, (eds.), pp. 575–582 (September 2007)
5. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning **52**(3), 199–215 (2003)
6. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks (1984)
7. Clémençon, S., Vayatis, N.: Tree-based ranking rules. Tech. Rep. hal-00268068, HAL (2008). URL /hal.archives-ouvertes.fr/hal-00268068/fr/
8. Clémençon, S., Lugosi, G., Vayatis, N.: Ranking and empirical risk minimization of U-statistics. The Annals of Statistics **36**(2), 844–874 (2008)
9. Clémençon, S., Vayatis, N.: On partitioning rules for bipartite ranking. In: AISTATS '09: Proceedings of the 2009 conference on Artificial Intelligence and Statistics (2009)
10. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press (1990)
11. Tsybakov, A.: Optimal aggregation of classifiers in statistical learning. Annals of Statistics **32**(1), 135–166 (2004)
12. Clémençon, S., Vayatis, N.: Overlaying classifiers: a practical approach for optimal ranking. In: NIPS '08: Proceedings of the 2008 conference on Advances in neural information processing systems (2008)
13. Nobel, A.: Analysis of a complexity-based pruning scheme for classification trees. IEEE Transactions on Information Theory **48**(8), 2362–2368 (2002)
14. Ripley, B.: Pattern Recognition and Neural Networks. Cambridge University Press (1996)
15. Massart, P.: Concentration inequalities and model selection. Lecture Notes in Mathematics. Springer (2006)
16. Boucheron, S., Bousquet, O., Lugosi, G.: Theory of Classification: A Survey of Some Recent Advances. ESAIM: Probability and Statistics **9**, 323–375 (2005)
17. Clémençon, S., Lugosi, G., Vayatis, N.: Ranking and scoring using empirical risk minimization. In: P. Auer, R. Meir (eds.) Proceedings of COLT 2005, *Lecture Notes in Computer Science*, vol. 3559, pp. 1–15. Springer (2005)
18. Friedman, J.: Greedy Function Approximation: a Gradient Boosting Machine. IMS Reitz Lecture, 1999. Annals of Statistics **6**, 393–425 (2001)
19. Serfling, R.: Approximation theorems of mathematical statistics. John Wiley & Sons (1980)
20. Lugosi, G., Zeger, K.: Concept learning using complexity regularization. IEEE Transactions on Information Theory **42**(1), 48–54 (1996)
21. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer (1996)