ExactBoost: Directly Boosting the Margin in Combinatorial and Non-decomposable Metrics

Daniel Csillag
daniel.csillag@impa.br
IMPA

Carolina Piazza
cpiazza@princeton.edu
Princeton

Thiago Ramos
thiagorr@impa.br
IMPA

João Vitor Romano
joao.vitor@impa.br
IMPA

Roberto Oliveira
rimfo@impa.br
IMPA

Paulo Orenstein
pauloo@impa.br
IMPA

Abstract

Many classification algorithms require the use of surrogate losses when the intended loss function is combinatorial or non-decomposable. This paper introduces a fast and exact stagewise optimization algorithm, dubbed ExactBoost, that boosts instead to the actual loss function. By developing a novel extension of margin theory to the non-decomposable setting, it is possible to provably bound the generalization error of ExactBoost for many important metrics with different levels of non-decomposability. Through extensive examples, it is shown that such theoretical guarantees translate to competitive empirical performance. In particular, when used as an ensembler, ExactBoost is able to significantly outperform other surrogate-based and exact algorithms available.

1 INTRODUCTION

Several challenging classification tasks involve combinatorial and non-decomposable loss functions (Kar et al., 2014; Gao et al., 2019). A combinatorial metric is one that is computed in terms of indicator functions, while non-decomposable metrics are those that cannot be reduced to a sum of loss functions on each sample point. Since such losses are neither differentiable nor parallelizable, common approaches based on convex optimization or stochastic gradient descent are not readily applicable without resorting to surrogate losses.

Many popular metrics are of this nature. The area under the ROC curve (AUC) is a prime example. Other examples include the Kolmogorov-Smirnov (KS), widely used in the credit industry, and precision at k (P@k), which is usually applied to ranking problems. Generally, the data comes as independent and identically distributed (iid) points $(X_i, y_i)_{i=1}^n$, with features $X_i \in \mathbb{R}^p$ and binary labels $y_i \in \{0, 1\}$, and the goal is to devise algorithms that learn score functions (or classifiers) $S : \mathbb{R}^p \to [-1, 1]$ that correctly distinguish between the two label classes. Let n_0 and n_1 denote the number of labels in each class. These loss functions can be written

$$\widehat{\mathrm{AUC}}(S, y) = 1 - \frac{1}{n_1} \sum_{y_i=1} \frac{1}{n_0} \sum_{y_j=0} \mathbf{1}_{[S(X_i) > S(X_j)]}, \quad (1)$$

$$\widehat{\mathrm{KS}}(S, y) = 1 - \max_{t \in \mathbb{R}} \sum_{i=1}^{n} \rho_i \mathbf{1}_{[S(X_i) \leq t]}, \quad (2)$$

$$\widehat{\mathrm{P@k}}(S, y) = 1 - \frac{1}{n} \sum_{i=1}^{n} y_i \mathbf{1}_{[i \in \mathcal{M}_k]}, \quad (3)$$

where $\rho_i = 1/n_0$ if $y_i = 0$ and $\rho_i = -1/n_1$ if $y_i = 1$, and $\mathcal{M}_k$ denotes the set of indices $i = 1, \ldots, n$ achieving the highest k scores. These three examples display different levels of non-decomposability: AUC relies on pairwise interactions, KS has a global threshold chosen optimally, and P@k also has a global threshold but with no optimality structure. Many other popular loss functions belong to the combinatorial or non-decomposable classes, including F-score and partial AUC.

Boosting is a leading technique to deal with classification problems, though it usually requires the development of surrogate losses for combinatorial and non-decomposable metrics. Still, not using the exact

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

metric of interest often incurs in performance degradation, and the development of surrogate losses with optimality guarantees typically require significant work.

This paper considers, instead, a novel approach that works more generally for losses such as (1), (2) and (3). The procedure, dubbed ExactBoost, is a stagewise optimization algorithm tailored to the exact loss function with a margin condition. While margin theory is readily applicable in the decomposable setting, a novel extension is developed here for non-decomposable losses, yielding provable finite-sample performance guarantees. Given labels $\mathbf{y} = (y_1, \ldots, y_n)$, initial scores $\mathbf{S}_0 = (S_0(X_1), \ldots, S_0(X_n))$, and empirical loss function $\widehat{L} : [-1, 1]^n \times \{0, 1\}^n \to \mathbb{R}$, ExactBoost solves, at iteration $t = 1, \ldots, T$,

$$(\alpha_t, \mathbf{h}_t) = \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L}_\theta(\mathbf{S}_{t-1} + \alpha \mathbf{h}, \mathbf{y}), \tag{4}$$

and sets $\mathbf{S}_t = \mathbf{S}_{t-1} + \alpha_t \mathbf{h}_t$, where $h : \mathbb{R}^p \to \mathbb{R}$ is a base learner (e.g., a stump), $\mathbf{h} = (h(X_1), \ldots, h(X_n))$, $\alpha \geq 0$ is its corresponding weight, and, crucially, $\widehat{L}_\theta$ is a margin-adjusted version of the empirical loss $\widehat{L}$. The combinatorial nature of the losses allows each boosting iteration to be solved relatively quickly. By employing interval arithmetic, ExactBoost is of order $O(pn \log n)$.

While ExactBoost is a competitive standalone estimator, its performance is even better as an ensembler. Using surrogate-based algorithms' predictions as features for ExactBoost allows it to combine them specifically for the chosen loss function, extracting the remaining signal tailored to the loss, akin to transfer learning.

The main contributions of this work are:

- developing an extension of margin theory for combinatorial and non-decomposable losses;
- showing that ExactBoost, a fast margin-adjusted exact stagewise optimization algorithm, has provable optimality bounds on its performance;
- showing that its empirical performance is comparable or superior to general-purpose and other loss-specific algorithms available in the literature;
- demonstrating that, as an ensembler, ExactBoost significantly outperforms other ensembling methods based on surrogate and exact losses.

Related Work. Boosting algorithms for combinatorial and non-decomposable losses (Kar et al., 2014; Gao et al., 2019) typically employ surrogate metrics, as is the case with Gradient Boosting (Friedman, 2001) and AdaBoost (Freund and Schapire, 1997). Both use approximations of the loss that lead to fast algorithms that are generally sensitive to misclassification error (Bartlett et al., 2006). Still, some loss in performance may follow from not using the exact metric of interest (Cortes and Mohri, 2003; Fathony and Kolter, 2020). Recently, there have been efforts to find better surrogates to popular combinatorial losses (Ferri et al., 2002; Joachims, 2005; Boyd et al., 2012; Agarwal, 2013; Kar et al., 2014; Lyu and Ying, 2018; Tasche, 2018; Engilberge et al., 2019; Pfetsch and Pokutta, 2020; Grabocka et al., 2020; Jiang et al., 2020; Adam et al., 2020), trading off speed for a more accurate loss function. There has also been interest in developing heavily constrained approaches that use the exact loss function (Li et al., 2014; Fang and Chen, 2019). ExactBoost, instead, relies on a novel and general extension of the margin theory for non-decomposable losses (Zhai et al., 2013; Schapire and Freund, 2013) to obtain empirical error bounds, such as in Schapire et al. (1998); Bartlett and Mendelson (2002); Koltchinskii and Panchenko (2002), not previously available in this setting.

Organization. Section 2 introduces the ExactBoost algorithm. Section 3 collects theoretical guarantees about ExactBoost for representative losses, both as a standalone classifier and as an ensembler. Section 4 displays the performance of the algorithms on different datasets and compares it to the performance of traditional classifiers and loss-specific optimizers. Finally, Section 5 concludes the paper.

2 OVERVIEW OF EXACTBOOST

Consider data $(X_1, y_1), \ldots, (X_n, y_n) \sim \mathcal{D}$ independently, with $X_i \in \mathbb{R}^p$ features and $y_i \in \{0, 1\}$ labels, and an empirical loss $\widehat{L} : [-1, 1]^n \times \{0, 1\}^n \to [0, 1]$ that is invariant under rescaling and translation in its first argument, such as (1), (2) and (3). The goal is to find a score function $S : \mathbb{R}^p \to [-1, 1]$ where higher scores $S(X_i)$ indicate higher likelihood of $y_i = 1$. It will be assumed that, after t rounds, a score has the form

$$S_t(X_i) = \sum_{r=1}^{t} w_r h_r(X_i), \tag{5}$$

with $w_r \geq 0$, $\sum_{r=1}^{t} w_r = 1$ and $h_r \in \mathcal{H}$, where $\mathcal{H}$ is a set of base learners. For stagewise minimization of the empirical loss, one solves $(\alpha_{t+1}, \mathbf{h}_{t+1}) = \operatorname{argmin}_{\alpha \geq 0, h \in \mathcal{H}} \widehat{L}(\mathbf{S}_t + \alpha \mathbf{h}, \mathbf{y})$ and updates the score via $S_{t+1} = (S_t + \alpha_{t+1} h_{t+1})/(1 + \alpha_{t+1})$, where the denominator ensures the weights sum to one, as in (5).

This approach produces competitive results on test data in many settings. However, to attenuate overfitting with combinatorial and non-decomposable losses, a margin-adjusted loss $\widehat{L}_\theta$ is justified. Consider

$$\widehat{L}_\theta(\mathbf{S}, \mathbf{y}) = \widehat{L}(\mathbf{S} - \theta \mathbf{y}, \mathbf{y}), \tag{6}$$

where $\theta > 0$ is a margin parameter (though the P@k case is slightly more subtle, see Theorem 3). That way, scores for positive labels are artificially decreased, forcing the algorithm to increase the confidence when correctly classifying samples (since losses are translation-invariant, this is equivalent to imposing high confidence on negative cases). This simple adjustment is crucial to provide optimality bounds on the generalization performance of the resulting algorithm (see Section 3).

Now, consider the optimization program (4). While ExactBoost and its guarantees hold for general sets of base learners $\mathcal{H}$, in practice learners beyond stumps (e.g., trees of higher depths) do not yield significant improvements and can be much more costly computationally. Thus, take $\mathcal{H}$ to be the set of stumps:

$$\mathcal{H} = \left\{ \pm \mathbf{1}_{[X_{(j)} \leq \xi]} \pm \mathbf{1}_{[X_{(j)} > \xi]} \ : \ \xi \in \mathbb{R}, j \in [p] \right\}, \quad (7)$$

with $X_{(j)}$ denoting the jth feature of X.

Since the losses are invariant under rescaling and translation of the first argument, ExactBoost must pick

$$\begin{aligned}(\alpha_t, \mathbf{h}_t) &= \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L}_\theta \left(\frac{1}{1+\alpha} \mathbf{S}_{t-1} + \frac{\alpha}{1+\alpha} \mathbf{h}, \mathbf{y} \right) \\ &= \operatorname*{argmin}_{\alpha, \mathbf{h}} \widehat{L} \left(\mathbf{S}_{t-1} - \theta \mathbf{y} + \alpha(\mathbf{h} - \theta \mathbf{y}), \mathbf{y} \right).\end{aligned}$$

Let $\tilde{h}(X) = \tilde{a}\mathbf{1}_{[X_{(j)} \leq \xi]} + \tilde{b}\mathbf{1}_{[X_{(j)} > \xi]} - (|\tilde{b} - \tilde{a}|/2)\theta y$, a function parametrized by $\tilde{a}, \tilde{b}, \xi \in \mathbb{R}$ and $j \in [p]$. Note

$$\begin{aligned}\tilde{h}(X) - \frac{\tilde{a} + \tilde{b}}{2} &= \frac{|\tilde{b} - \tilde{a}|}{2} \left(\pm \mathbf{1}_{[X_{(j)} \leq \xi]} \pm \mathbf{1}_{[X_{(j)} > \xi]} - \theta y \right) \\ &= \alpha(a \mathbf{1}_{[X_{(j)} \leq \xi]} + b \mathbf{1}_{[X_{(j)} > \xi]} - \theta y) \\ &= \alpha(h(X) - \theta y),\end{aligned}$$

where $a, b \in \{-1, 1\}$ and $\alpha \geq 0$. Thus, the program (4) is iteratively solved by picking (ξ_t, j_t, a_t, b_t) via

$$\begin{aligned}\min_{\xi \in \mathbb{R}, j \in [p], \tilde{a}, \tilde{b} \in \mathbb{R}} \widehat{L} \Big(\mathbf{S}_{t-1} + \tilde{a}\mathbf{1}_{[X_{(j)} \leq \xi]} + \tilde{b}\mathbf{1}_{[X_{(j)} > \xi]} \\ - \left(1 + (|\tilde{b} - \tilde{a}|)/2 \right) \theta \mathbf{y}, \mathbf{y} \Big),\end{aligned} \quad (8)$$

then setting $\mathbf{S}_t = \mathbf{S}_{t-1} + a_t \mathbf{1}_{[X_{(j_t)} \leq \xi_t]} + b_t \mathbf{1}_{[X_{(j_t)} > \xi_t]}$. Note the discrete nature of combinatorial loss functions allows (8) to be solved by only considering a finite set of ξ, $\tilde{a}$ and $\tilde{b}$: for ξ, it suffices to look at the unique values of feature $X_{(j)}$ for $j = 1, \ldots, p$, and for a and b the unique values of $S(X_i)$, for $i = 1, \ldots, n$. Other values of ξ, a and b do not yield different training losses.

The resulting algorithm is called ExactBoost, as it is based on the exact loss function provided rather than a surrogate loss. To avoid overfitting, subsampling is used (see Subsection 3.4 for theoretical guarantees). Finally, randomized runs of the algorithm are averaged, similar in spirit to random forests, and can be trivially parallelized. Algorithm 1 includes the full pseudocode. It takes as input an initial set of scores, which could for instance be scores trained by other learning models.

Algorithm 1 ExactBoost

function ExactBoost(data $(\mathbf{X}, \mathbf{y})$, initial scores S_0, margin θ, iterations T, estimator runs E)
  for $e \in \{1, \ldots, E\}$ **do**
    $S_e \leftarrow S_0$
    for $t \in \{1, \ldots, T\}$ **do**
      $\mathbf{X}^s, \mathbf{y}^s \leftarrow$ subsample $\mathbf{X}, \mathbf{y}$
      for $j \in \{1, \ldots, p\}$ **do**
        $\widehat{L}(h) \leftarrow \widehat{L}_\theta(S_e(\mathbf{X}^s_{(j)}) + h(\mathbf{X}^s_{(j)}), \mathbf{y}^s)$
        $h_j \leftarrow \operatorname{argmin}_h \widehat{L}(h)$ ▷ Algorithm 2
      end for
      $h \leftarrow \operatorname{argmin}_{h_j} \widehat{L}_\theta(S_e(\mathbf{X}^s) + h_j(\mathbf{X}^s), \mathbf{y}^s)$
      $S'_e \leftarrow S_e + h$
      if $\widehat{L}_\theta(S'_e(\mathbf{X}), \mathbf{y}) \leq \widehat{L}_\theta(S_e(\mathbf{X}), \mathbf{y})$ **then**
        $S_e \leftarrow S'_e$
        $S_e \leftarrow (S_e - \min S_e)/(\max S_e - \min S_e)$
      end if
    end for
  end for
  return mean$(S_1, \ldots, S_E)$
end function

Algorithm 2 Iterative Minimization

function Minimize(loss $\hat{L}_\theta$, data $\mathbf{X}_{(j)}$, labels $\mathbf{y}$, scores S, margin θ)
  $\Xi \leftarrow [\min \mathbf{X}_{(j)}, \max \mathbf{X}_{(j)}]$
  $A \leftarrow [-1, 1]$; $B \leftarrow [-1, 1]$
  for $k \in \{1, \ldots, c\}$ **do**
    $l_\star \leftarrow +\infty$
    for bisections $(\Xi^{(b)}, A^{(b)}, B^{(b)})$ **do**
      for $i \in \{1, \ldots, n\}$ **do**
        $s \leftarrow S + A^{(b)} \mathbf{1}_{[\mathbf{X}_{(j)} \leq \Xi^{(b)}]} + B^{(b)} \mathbf{1}_{[\mathbf{X}_{(j)} > \Xi^{(b)}]}$
        $\mathbf{s}_i \leftarrow \underline{s}$ if $y_i = 0$ otherwise $\overline{s}$
      end for
      if $\widehat{L}_\theta(\mathbf{s}, \mathbf{y}) < l_\star$ **then**
        $l_\star \leftarrow \widehat{L}_\theta(\mathbf{s}, \mathbf{y})$
        $\Xi_\star \leftarrow \Xi^{(b)}$; $A_\star \leftarrow A^{(b)}$; $B_\star \leftarrow B^{(b)}$
      end if
    end for
  end for
  $I_\Xi \leftarrow \{\underline{\Xi_\star}, \overline{\Xi_\star}\}$; $I_A \leftarrow \{\underline{A_\star}, \overline{A_\star}\}$; $I_B \leftarrow \{\underline{B_\star}, \overline{B_\star}\}$
  $S(a, b, \xi) \leftarrow S + a\mathbf{1}_{[\mathbf{X}_{(j)} \leq \xi]} + b\mathbf{1}_{[\mathbf{X}_{(j)} > \xi]}$
  $(\xi_\star, a_\star, b_\star) \leftarrow \operatorname*{argmin}_{\xi \in I_\Xi, a \in I_A, b \in I_B} \widehat{L}_\theta(S(a, b, \xi), \mathbf{y})$
  return $S + a_\star \mathbf{1}_{[\mathbf{X}_{(j)} \leq \xi_\star]} + b_\star \mathbf{1}_{[\mathbf{X}_{(j)} > \xi_\star]}$
end function

In order to efficiently solve (8), we use an interval arithmetic (IA)-based algorithm: We use the usual IA notations and operations, see Hickey et al. (2001); e.g., $Z = [\underline{Z}, \overline{Z}]$ is an interval, $F(Z) = [\underline{F(Z)}, \overline{F(Z)}]$ means $\underline{F(Z)}$ is the lower bound for the interval $F(Z)$, and $U + V = [\underline{U}, \overline{U}] + [\underline{V}, \overline{V}] = [\underline{U} + \underline{V}, \overline{U} + \overline{V}]$. Let

$\odot$ denote elementwise multiplication. The optimization algorithm, whose pseudocode is presented in Algorithm 2, takes the form of a bisection-like iterative procedure: we first assign intervals A, B and Ξ as the search domain, and then, c times, we halve them as follows: for each possible way to halve A, B and Ξ, compute the IA lower bound in the subinterval, $\widehat{L}(S'(A^{(b)}, B^{(b)}, \Xi^{(b)}), \mathbf{y}) = \widehat{L}(\mathbf{y} \odot S'(A^{(b)}, B^{(b)}, \Xi^{(b)}) + (1 - \mathbf{y}) \odot S'(A^{(b)}, B^{(b)}, \Xi^{(b)}), \mathbf{y})$ (this follows directly from the IA definitions applied to our losses), and pick the one with the lowest IA lower bound as the new search domain. Since each step halves the search domain and gives an extra bit of numerical precision, c is fixed as the precision of the floating-point type. The Supplementary Material includes more details.

By using Algorithm 2 to solve (8), ExactBoost has a runtime complexity of order $O(pn\log(n))$ and a space complexity of order $O(n)$. Thus, it can scale well even to large datasets, as shown in Section 4.

3 THEORETICAL RESULTS

This section develops a theory of generalization for ExactBoost under margin-type conditions. It shows, in particular, that the population error of an ExactBoost's score S can be upper bounded by the sum of a margin-adjusted sample error of S plus an error depending on $\mathcal{H}$. Crucially, the latter is controlled uniformly over S and only depends on the class of functions $\mathcal{H}$. Thus, if a method has a margin-adjusted training loss that is sufficiently small relative to θ, then it generalizates well. When $\mathcal{H}$ is the set of stumps (7), for example, one can allow for a number of features that is nearly as large as an exponential in the number of positive and negative examples.

The theoretical results are based on the representative losses (1), (2) and (3), which display different levels of non-decomposability. While this affects the guarantees for each loss slightly differently, the proof techniques allow for generalization to other non-decomposable losses, as pointed out below. Importantly, the margin adjustment on each loss is essentially the same.

The results below extend to non-decomposable losses previous work in obtaining empirical bounds for classification tasks (Schapire et al., 1998; Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002). The results presented here differ in spirit from those obtained via surrogate losses (Agarwal, 2013; Kar et al., 2015). Surrogate metrics can provide upper bounds of the desired loss but often lack a natural quantitative interpretation. The theorems below, on the other hand, show that minimizing a margin-adjusted empirical loss leads, with high probability, to a small population loss.

Notation. Assume $\mathcal{D}$ is a probability distribution over pairs $(X, y) \in \mathbb{R}^p \times \{0, 1\}$, and let $\mathcal{D}_0$ (respectively, $\mathcal{D}_1$) denote the conditional distribution of X when $y = 0$ (respectively, 1). When unambiguous, $\mathcal{D}$ might also denote the marginal distribution of X. The data is $(X_i, y_i)_{i=1}^n \sim \mathcal{D}$ iid, and, conditionally on the number n_1 of indices i with $y_i = 1$ (and also defining $n_0 \coloneqq n - n_1$), the subsamples $\mathbf{X}_1 \coloneqq (X_i : i \in [n], y_i = 1)$ and $\mathbf{X}_0 \coloneqq (X_i : i \in [n], y_i = 0)$ are iid from $\mathcal{D}_1$ and $\mathcal{D}_0$. Score functions $S : \mathbb{R}^p \to [-1, 1]$ are convex combinations of elements in a family of measurable functions $\mathcal{H} : \mathbb{R}^p \to [-1, 1]$. Let $\{\sigma_i\}_{i=1}^n$ be iid uniform over ± 1 and independent from data. Define the Rademacher complexities of $\mathcal{H}$ with respect to $\mathcal{D}$, $\mathcal{D}_0$ and $\mathcal{D}_1$:

$$\mathcal{R}_n(\mathcal{H}) \coloneqq \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \sigma_i h(X_i)\right]\right]$$

$$\mathcal{R}_{n,y}(\mathcal{H}) \coloneqq \mathbb{E}_{\mathcal{D}_y}\left[\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}} \frac{1}{n_y}\sum_{i\,:\,y_i=y} \sigma_i h(X_i)\right]\right],$$

for $y \in \{0, 1\}$. Note $\mathcal{R}_{n,y}(\mathcal{H})$ is defined conditionally on n_y, the number of examples with label y. When n_y equals zero, we set $\mathcal{R}_{n,y}(\mathcal{H}) = 1$ by convention. Note $\mathcal{R}_n(\mathcal{H}) = O(\sqrt{\log p/n})$ when $\mathcal{H}$ is as in (7).

3.1 Margin result for AUC loss

The AUC loss, for $(X, X') \sim \mathcal{D}_1 \times \mathcal{D}_0$, and its θ-margin-adjusted version are given by:

$$\mathrm{AUC}(S) \coloneqq 1 - \Pr\{S(X) > S(X')\},$$

$$\widehat{\mathrm{AUC}}_\theta(S) \coloneqq 1 - \frac{1}{n_1}\sum_{i:y_i=1}\frac{1}{n_0}\sum_{j:y_j=0} \mathbf{1}_{[S(X_i)-\theta>S(X_j)]}.$$

Note $\widehat{\mathrm{AUC}}_\theta(S)$ is one minus the area under the curve when one subtracts θ from the scores of 1-labelled samples. Because AUC relies on pairwise interactions, it is not readily decomposable over each sample point. Still, the U-statistic structure of this loss allows for the following result.

Theorem 1. *Given $\theta > 0$, $\delta \in (0, 1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1, 1]$, the following holds with probability at least $1 - \delta$: for all score functions $S : \mathbb{R}^p \to [-1, 1]$ obtained as convex combinations of the elements of $\mathcal{H}$,*

$$\mathrm{AUC}(S) \le \widehat{\mathrm{AUC}}_\theta(S) + \frac{4}{\theta}\zeta_{\mathrm{AUC}}(\mathcal{H}) + \sqrt{\frac{2\log(1/\delta)}{\min\{n_0, n_1\}}},$$

where $\zeta_{\mathrm{AUC}}(\mathcal{H}) = \mathcal{R}_{\min\{n_0,n_1\},0}(\mathcal{H}) + \mathcal{R}_{\min\{n_0,n_1\},1}(\mathcal{H})$.

Theorem 1 holds conditionally on $n_0, n_1 > 0$, which will hold with very high probability unless $\mathcal{D}$ is too imbalanced towards $y = 0$ or $y = 1$. When $\mathcal{H}$ is

given by (7), the theorem implies, for constant δ, that the score S produced by the algorithm satisfies $\mathrm{AUC}(S) \leq \widehat{\mathrm{AUC}}_\theta(S) + o(1)$ with high probability when $\min\{n_0, n_1\} \gg \theta^{-2} \log p$. Theorem 1 can be extended to similar pairwise losses.

3.2 Margin result for KS loss

For a score S, the KS loss and its margin-adjusted sample version are defined as:

$$\mathrm{KS}(S) = 1 - \sup_{t \in \mathbb{R}} \Big(\Pr_{X \sim \mathcal{D}_0} \{S(X) \leq t\} - \Pr_{X \sim \mathcal{D}_1} \{S(X) \leq t\} \Big),$$

$$\widehat{\mathrm{KS}}_\theta(S) = 1 - \max_{t \in \mathbb{R}} \left(\frac{1}{n_0} \sum_{i \,:\, y_i = 0} \mathbf{1}_{[S(X_i) \leq t]} - \frac{1}{n_1} \sum_{i \,:\, y_i = 1} \mathbf{1}_{[S(X_i) - \theta \leq t]} \right),$$

where, by convention, $\widehat{\mathrm{KS}}_\theta(S) = 1$ if $n_1 = 0$ or $n_0 = 0$.

Theorem 2. *Given $\theta > 0$, $\delta \in (0,1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, the following holds with probability at least $1 - \delta$: for all score functions $S : \mathbb{R}^p \to [-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$,*

$$\mathrm{KS}(S) \leq \widehat{\mathrm{KS}}_\theta(S) + \frac{8}{\theta} \zeta_{\mathrm{KS}}(\mathcal{H}) + \sqrt{\frac{\log(2/\delta)}{2}} \left(\frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}} \right),$$

where $\zeta_{\mathrm{KS}}(\mathcal{H}) = \mathcal{R}_{n_0,0}(\mathcal{H}) + \mathcal{R}_{n_1,1}(\mathcal{H}) + n_0^{-1/2} + n_1^{-1/2}$.

Thus a score that achieves a small margin-adjusted KS loss will, with high probability, have good performance on the population, if we condition on $n_0, n_1 > 0$. Similarly to Theorem 1, when the base learners are stumps, we obtain

$$\mathrm{KS}(S) \leq \widehat{\mathrm{KS}}_\theta(S) + C \sqrt{\frac{\theta^{-2}(1 + \log p) + \log(2/\delta)}{\min\{n_0, n_1\}}}.$$

Thus for constant δ, good training performance on the margin-adjusted loss leads to good generalization whenever $\theta^{-2} \log p \ll \min\{n_0, n_1\}$.

3.3 Margin result for P@k loss

For the precision at k loss, given a score $S : \mathbb{R}^p \to [-1,1]$ and $\alpha \in (0,1)$, let $t_\alpha(S)$ denote its $(1-\alpha)$-quantile under the population distribution and $\widehat{t}_\alpha(S)$ the sample version,

$$t_\alpha(S) \coloneqq \inf \{t \in \mathbb{R} : \Pr\{S(X) \leq t\} \geq 1 - \alpha\}$$

$$\widehat{t}_\alpha(S) \coloneqq \inf \left\{ t \in \mathbb{R} : \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[S(X_i) \leq t]} \geq 1 - \alpha \right\}.$$

The precision at k loss of S (for parameter α) and its margin-adjusted sample version are

$$\mathrm{P@k}_\alpha(S) \coloneqq 1 - \Pr\{y = 1, S(X) \geq t_\alpha(S)\},$$

$$\widehat{\mathrm{P@k}}_\theta(S) \coloneqq 1 - \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{[y_i = 1, S(X_i) - \theta \geq \widehat{t}_\alpha(S)]}.$$

Informally, $\widehat{\mathrm{P@k}}_\theta(S)$ is the sample precision at k when 1-labelled examples have their scores reduced by θ after the threshold $\widehat{t}_\alpha(S)$ has been computed. Similarly to the KS loss, P@k is non-decomposable due to a global threshold $\widehat{t}_\alpha(S)$, but the lack of optimality structure makes proving the next result much more involved.

Theorem 3. *Given $\theta > 0$, $\delta \in (0,1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, define*

$$\bar{\eta}_n(\mathcal{H}) \coloneqq \sqrt{4\mathcal{R}_n(\mathcal{H}) + \frac{4}{\sqrt{n}}} + \sqrt{\frac{\log(3/(\delta - \delta^2))}{n}},$$

Assume $\theta > 2\bar{\eta}_n(\mathcal{H})$ and $\Pr(\min\{n_0, n_1\} > 0) \geq 1 - \delta$. Then the following holds with probability $\geq 1 - \delta$: if $\delta' \coloneqq \delta - \delta^2$, then for all score functions $S : \mathbb{R}^p \to [-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$, it holds

$$\mathrm{P@k}(S) \leq \widehat{\mathrm{P@k}}_\theta(S) + \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} + \bar{\eta}_n(\mathcal{H}) + \sqrt{2\frac{\log(3/\delta')}{n_1}} + \sqrt{\frac{\log(3/\delta')}{2n}}.$$

The proof techniques of the theorem above can be generalized to other combinatorial losses that use a restricted sample, such as partial AUC.

3.4 Subsampling

Subsampling can help ExactBoost avoid overfitting. The next proposition is helpful in controlling its impact in the optimization procedure for some losses.

Proposition 1. *Let $\widehat{L}$ be either the $\widehat{\mathrm{AUC}}$ or the $\widehat{\mathrm{KS}}$ loss. Consider a subset of indices $I = I_0 \cup I_1 \subset [n]$ chosen independently and uniformly at random with equal number of positive and negative cases, $|I_0| = |I_1| = k$. Let h_R be the optimal stump over the reduced sample $\{(X_j, y_j)\}_{j \in I}$ and score S and h_* the optimal stump over the entire sample $\{(X_i, y_i)\}_{i \in [n]}$. Then,*

$$\mathbb{E}[\widehat{L}(S + h_R)] \leq \widehat{L}(S + h_*) + \frac{e}{k},$$

where the expectation is over the choice of I.

Hence, using random subsets of observations in ExactBoost with balanced proportions of positive and negative examples leads to an expected error close to the optimal one.

3.5 Ensembling

Since minimizing the margin-adjusted empirical loss can generalize to the population loss, it is natural to investigate whether ExactBoost can also provide a good ensembling technique for other classifiers. Indeed, for some losses, it is possible to guarantee that the empirical loss of the ensembler is smaller than the empirical loss of each ensembler member.

Denote the vector of scores for the ith data point by $Z_i \coloneqq (S_1(X_i), S_2(X_i), \ldots, S_M(X_i))^T \in \mathbb{R}^M$, M being the number of models, and train ExactBoost over a modified dataset $(Z_i, y_i)_{i=1}^n$. The next proposition shows that the training set performance of ExactBoost over $(Z_i, y_i)_{i=1}^n$ using either the KS or P@k metrics is always at least as good as that of the the best score function over $(X_i, y_i)_{i=1}^n$.

Proposition 2. *Let $\widehat{L}$ be either the $\widehat{\text{KS}}$ or the $\widehat{\text{P@k}}$ loss. Consider the score $S_* : \mathbb{R}^M \to \mathbb{R}$ obtained by ExactBoost over the dataset $(Z_i, y_i)_{i=1}^n$ with initial score $S_0 \equiv 0$. Then:*

$$\widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \min_{1\leq m\leq M} \widehat{L}_{(X_i,y_i)_{i=1}^n}(S_m),$$

where $\widehat{L}_{(Z_i,y_i)_{i=1}^n}(\cdot)$ and $\widehat{L}_{(X_i,y_i)_{i=1}^n}(\cdot)$ denote the loss over the ensemble and the original data.

Section 4 shows that, in practice, ensembling with ExactBoost leads to better results than ensembling with other surrogate-based algorithms. The fact that the inputs for the ensembler can be trained with surrogate-based methods attenuates overfitting, and speeds up ExactBoost by reducing the set of original features p to the number of models M.

4 EXPERIMENTS

To test its performance, ExactBoost is compared against 10 exact and surrogate-based algorithms, on 30 heterogeneous datasets, over three different losses. For ease of presentation, results of 10 representative datasets are shown in the main paper; the rest are in the Supplementary Material.

Datasets. Table 1 displays the main characteristics of each dataset, which span economic, medical, radar, financial and ecological applications, and range from balanced to imbalanced. Sources for the data can be found in the Supplementary Material.

Surrogate benchmarks. ExactBoost is compared to various standard learning algorithms: AdaBoost, k-nearest neighbors, logistic regression and random forest (via their Scikit-Learn implementation in Pedregosa et al. (2011)), gradient boosting (via XGBoost, see Chen and Guestrin (2016)) and a 4-layer connected neural net (via TensorFlow, see Abadi et al. (2015)).

Dataset	Observations	Features	Positives
a1a	1605	119	24.6%
german	1000	20	70.0%
gisette	6000	5000	50.0%
gmsc	150000	10	6.7%
heart	303	21	45.9%
ionosphere	351	34	64.1%
liver-disorders	145	5	37.9%
oil-spill	937	49	4.4%
splice	1000	60	48.3%
svmguide1	3089	4	35.3%

Table 1: Dataset properties.

Exact benchmarks. Several algorithms that specifically optimize the performance metric are considered. For KS, the baseline is DMKS (Fang and Chen, 2019), and, for P@k, the baseline is TopPush (Li et al., 2014). For AUC, the baseline is RankBoost (Freund et al., 2003), a boosting algorithm shown to optimize the AUC under certain conditions in Cortes and Mohri (2003).

Dataset	RankBoost	DMKS	TopPush
a1a	55.90×	102.78×	0.82×
german	23.98×	1.28×	0.88×
gisette	OOT	55.68×	0.02×
gmsc	OOT	22.89×	0.08×
heart	3.32×	19.00×	5.25×
ionosphere	3.97×	3.48×	2.69×
liver-disorders	1.91×	6.36×	12.53×
oil-spill	5.93×	7.92×	2.18×
splice	49.78×	1.19×	1.20×
svmguide1	220.05×	1.88×	4.27×

Table 2: Timings of various exact algorithms vs ExactBoost (above 1× indicates ExactBoost is faster). TopPush is fast but much less precise; see Table 3.

Hyperparameters. Hyperparameters were fixed throughout the experiments. Baseline models were trained with the package-provided hyperparameters; see the Supplementary Material. Aided by experimental evidence on held-out datasets, ExactBoost uses as default $E = 250$ runs, $T = 50$ rounds, subsampling of 20% and margin of $\theta = 0.05$. See Subsection 4.1 for further discussions.

Computational allowance, environment and code. Experiments were run with four Intel Xeon E5-4650 CPUs with 2.60 GHz, 64 threads, and 810 GB of RAM. Code to reproduce figures and tables can be found at `https://github.com/dccsillag/exactboost`. Methods had at most 5 days to run on each dataset.

4.1 Effect of Hyperparameters on ExactBoost

ExactBoost has two main hyperparameters that control overfitting: the margin θ and the number of runs averaged E (see Algorithm 1). Figure 1 shows how the margin affects the test error for the AUC, KS and P@k losses in three different datasets. Generally, though not always, the loss decreases with small positive margins, but becomes increasing once the margin is too large.

To consider the number of runs to be averaged, Figure 2 displays the train and test KS loss landscape, as well as ExactBoost's trajectory averaging over $E = 1, 2, 10, 100$ and 250 runs, for the `heart` dataset. The plot uses UMAP (McInnes and Healy, 2018) to reduce the dimensionality of an ExactBoost run to 2D and colors the corresponding KS loss of each point in the mapping. The Supplementary Material includes more details. As E grows, the train and test display lower KS values and smoother trajectories.

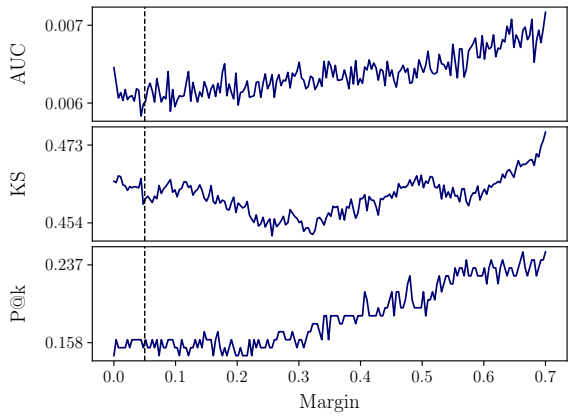

Figure 1: Effect of margin on ExactBoost's test performance on `svmguide1`, `gmsc` and `splice`. The vertical line shows the default $\theta = 0.05$. There are gains with small margins; the performance degrades with large θ.

4.2 ExactBoost vs exact and surrogate benchmarks

The performance of ExactBoost as an estimator is investigated via its 5-fold cross-validated test error. Table 3 shows that ExactBoost is generally better than loss-specific alternatives. In particular, the table includes comparisons to additional exact models available in the literature, such as SVMPerf (Joachims, 2005), which directly optimizes for multivariate performance metrics such as P@k, and plugin logistic (Koyejo et al., 2014; Dembczyński et al., 2017), a fast hybrid method that uses the metric of interest, say AUC, to pick the optimal threshold for logistic regression using a separate data fold. Figure 3 shows that ExactBoost also has good performance against surrogate benchmarks. Full results are included in the Supplementary Material.

In terms of timings, Table 2 shows that ExactBoost scales well even to large datasets. Note it is faster than other exact alternatives, and while TopPush can be faster, it is generally much less precise (see Table 3).

4.3 ExactBoost as an ensembler

In the experiments below, 5-fold cross-validation is used to compare ExactBoost against other ensemblers. Six base models were used: AdaBoost, k-nearest neighbors, logistic regression, neural network, random forest and XGBoost. These models were trained on training folds, and their predictions on test folds were used as features for the ensemble models.

Table 4 shows the results of using different surrogate and exact models as ensemblers. The surrogate ensemblers were AdaBoost, logistic regression, neural network, random forest and XGBoost, while the exact benchmarks were given by RankBoost (for AUC), DMKS (for KS) and TopPush (for P@k).

ExactBoost is generally the best ensembler available. In fact, it is able to match or overcome the performance of the best base model available and is robust to noisy features coming from poorly performing base models. This is particularly attractive because, given the discrete nature of combinatorial losses, it is often the case that the best performing model changes from dataset to dataset. ExactBoost's success can be interpreted as transfer learning: it is able to better combine high-signal features trained with surrogate losses by considering the exact metric of interest.

5 CONCLUSION

This paper introduced ExactBoost, a stagewise boosting algorithm that directly optimizes combinatorial and non-decomposable losses. By a novel extension of the notion of margin to this setting, it is possible to give finite-sample bounds on the generalization error of the algorithm for popular loss functions with varying levels of non-decomposability. The margin extension and the underlying proof techniques should apply broadly and we anticipate that similar results can be derived for other important non-decomposable losses. Also, while ExactBoost uses stumps as its base learners, it is straightforward to extend it (and its theoretical guarantees) to more general learners, such as trees of higher depth, though that entails a higher computational price. The theoretical results presented also cover subsampling and ensembling techniques.

The empirical results above show that ExactBoost al-

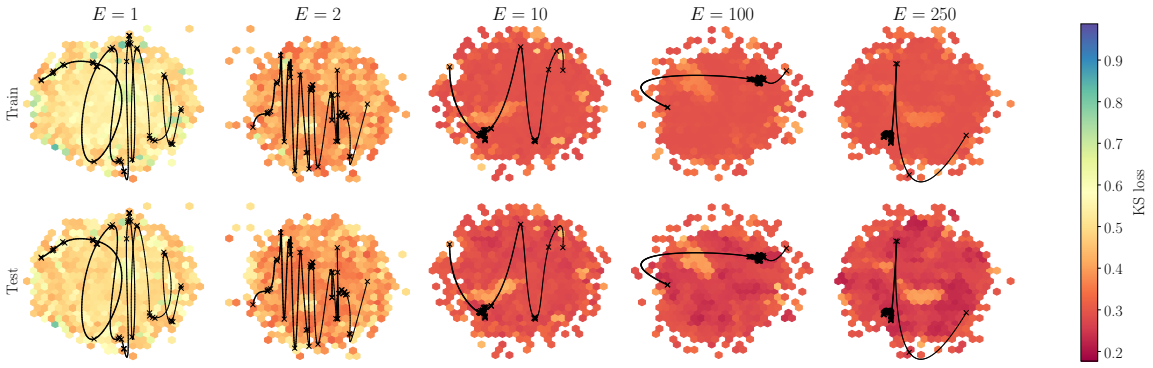

Figure 2: KS loss landscape visualizations via UMAP highlighting ExactBoost's optimization trajectories, which go from left to right. More averaged runs E lead to better train and test losses.

	AUC			KS		P@k		
Dataset	ExactBoost	RankBoost	Plugin Logistic	ExactBoost	DMKS	ExactBoost	TopPush	SVMPerf
a1a	$\mathbf{0.11 \pm 0.0}$	0.13 ± 0.0	0.20 ± 0.0	$\mathbf{0.37 \pm 0.0}$	$\mathbf{0.37 \pm 0.0}$	0.26 ± 0.1	0.29 ± 0.1	$\mathbf{0.22 \pm 0.1}$
german	$\mathbf{0.23 \pm 0.0}$	0.24 ± 0.0	0.28 ± 0.0	$\mathbf{0.53 \pm 0.0}$	0.55 ± 0.0	$\mathbf{0.11 \pm 0.0}$	0.26 ± 0.1	0.21 ± 0.0
gisette	$\mathbf{0.01 \pm 0.0}$	OOT	0.03 ± 0.0	0.09 ± 0.0	$\mathbf{0.06 \pm 0.0}$	0.02 ± 0.0	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$
gmsc	$\mathbf{0.21 \pm 0.0}$	OOT	0.38 ± 0.0	$\mathbf{0.44 \pm 0.0}$	0.45 ± 0.0	$\mathbf{0.52 \pm 0.0}$	0.96 ± 0.0	0.85 ± 0.0
heart	$\mathbf{0.09 \pm 0.0}$	0.13 ± 0.0	0.19 ± 0.0	0.30 ± 0.0	$\mathbf{0.28 \pm 0.0}$	$\mathbf{0.04 \pm 0.1}$	0.13 ± 0.1	$\mathbf{0.04 \pm 0.1}$
iono	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	0.17 ± 0.0	$\mathbf{0.13 \pm 0.0}$	0.28 ± 0.0	$\mathbf{0.03 \pm 0.0}$	0.15 ± 0.1	0.16 ± 0.1
liver	$\mathbf{0.22 \pm 0.1}$	0.32 ± 0.1	0.35 ± 0.1	$\mathbf{0.45 \pm 0.1}$	0.50 ± 0.1	$\mathbf{0.23 \pm 0.1}$	0.47 ± 0.2	0.33 ± 0.2
oil-spill	$\mathbf{0.09 \pm 0.1}$	$\mathbf{0.09 \pm 0.1}$	0.39 ± 0.1	$\mathbf{0.25 \pm 0.1}$	0.45 ± 0.1	$\mathbf{0.52 \pm 0.3}$	0.96 ± 0.1	1.00 ± 0.0
splice	0.04 ± 0.0	$\mathbf{0.02 \pm 0.0}$	0.21 ± 0.0	$\mathbf{0.16 \pm 0.0}$	0.36 ± 0.0	$\mathbf{0.03 \pm 0.0}$	0.12 ± 0.0	0.10 ± 0.0
svmg1	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.05 ± 0.0	$\mathbf{0.06 \pm 0.0}$	0.09 ± 0.0	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$	0.03 ± 0.0

Table 3: Evaluation of exact benchmarks. OOT indicates the time budget of 5 days was exceeded. ExactBoost has the best performance for all metrics: it is faster and uses less memory than RankBoost and DMKS (see Table 2), and much more accurate than Plugin Logistic and TopPush.

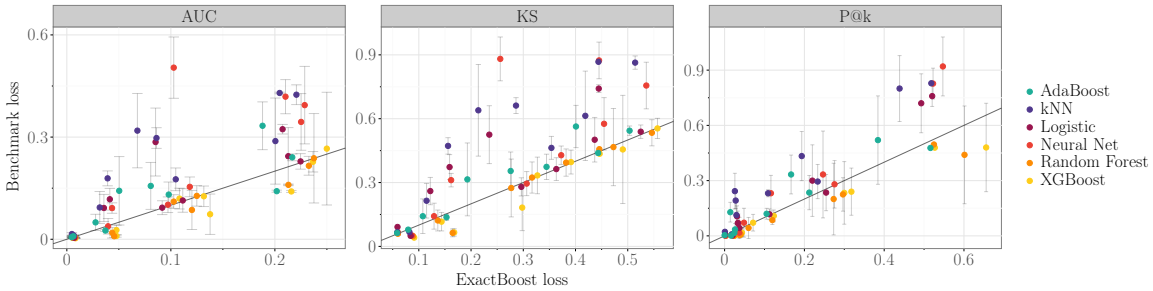

Figure 3: Test error for ExactBoost vs surrogate methods as estimators. Each point represents a dataset from Table 1. Alternatives are generally worse than ExactBoost or statistically indistinguishable.

gorithm is a fast, competitive classifier and an even better ensembler. It is scalable both in terms of speed and memory usage, and it is able to outperform other loss-specific algorithms previously introduced in the literature, as well as traditional surrogate alternatives. More broadly, ExactBoost shows promising results as an ensembler of traditional machine learning classifiers and prompts additional work on algorithms that can further interweave surrogate-based and loss-specific iterations to combine speed and accuracy.

Acknowledgements

We are grateful to Stone Pagamentos for a project that inspired some of the ideas in this work.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P.,

Loss	Dataset	ExactBoost	AdaBoost	Logistic	Neural Net	Rand. For.	XGBoost	Exact Bench.
AUC	a1a	$\mathbf{0.13 \pm 0.0}$	0.17 ± 0.0	0.14 ± 0.0	0.15 ± 0.0	0.27 ± 0.1	0.28 ± 0.1	0.16 ± 0.0
	german	$\mathbf{0.23 \pm 0.0}$	0.32 ± 0.0	0.24 ± 0.0	0.50 ± 0.1	0.33 ± 0.0	0.35 ± 0.0	0.30 ± 0.1
	gisette	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.03 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	gmsc	0.15 ± 0.0	$\mathbf{0.14 \pm 0.0}$	0.31 ± 0.0	0.46 ± 0.0	0.42 ± 0.0	0.41 ± 0.0	0.15 ± 0.0
	heart	$\mathbf{0.12 \pm 0.0}$	0.18 ± 0.1	$\mathbf{0.12 \pm 0.0}$	0.23 ± 0.1	0.19 ± 0.0	0.23 ± 0.1	0.15 ± 0.0
	iono.	$\mathbf{0.04 \pm 0.0}$	0.05 ± 0.0	0.07 ± 0.0	0.07 ± 0.0	0.07 ± 0.0	0.09 ± 0.0	0.05 ± 0.0
	liver	$\mathbf{0.30 \pm 0.1}$	0.34 ± 0.1	0.34 ± 0.1	0.34 ± 0.1	0.38 ± 0.0	0.38 ± 0.0	0.38 ± 0.1
	oil-spill	$\mathbf{0.17 \pm 0.1}$	0.19 ± 0.1	0.29 ± 0.2	0.46 ± 0.1	0.38 ± 0.1	0.35 ± 0.2	0.19 ± 0.1
	splice	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$	0.08 ± 0.0	0.05 ± 0.0	0.04 ± 0.0	0.04 ± 0.0	0.02 ± 0.0
	svmg1	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.03 ± 0.0	0.04 ± 0.0	0.01 ± 0.0
KS	a1a	$\mathbf{0.37 \pm 0.1}$	0.44 ± 0.1	0.40 ± 0.1	0.41 ± 0.1	0.54 ± 0.1	0.57 ± 0.1	0.49 ± 0.1
	german	$\mathbf{0.50 \pm 0.1}$	0.68 ± 0.1	0.53 ± 0.1	0.89 ± 0.1	0.66 ± 0.0	0.69 ± 0.1	0.53 ± 0.1
	gisette	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	0.07 ± 0.0	0.07 ± 0.0	0.06 ± 0.0	$\mathbf{0.04 \pm 0.0}$	0.10 ± 0.0
	gmsc	$\mathbf{0.43 \pm 0.0}$	0.44 ± 0.0	0.73 ± 0.0	0.95 ± 0.0	0.85 ± 0.0	0.83 ± 0.0	0.46 ± 0.0
	heart	$\mathbf{0.34 \pm 0.1}$	0.38 ± 0.1	0.37 ± 0.1	0.52 ± 0.1	0.38 ± 0.1	0.46 ± 0.1	0.40 ± 0.0
	iono.	$\mathbf{0.13 \pm 0.1}$	0.18 ± 0.1	0.18 ± 0.1	0.17 ± 0.1	0.15 ± 0.1	0.19 ± 0.1	0.27 ± 0.1
	liver	$\mathbf{0.53 \pm 0.1}$	0.60 ± 0.2	0.59 ± 0.2	0.61 ± 0.1	0.76 ± 0.1	0.76 ± 0.0	0.60 ± 0.2
	oil-spill	$\mathbf{0.33 \pm 0.2}$	$\mathbf{0.33 \pm 0.2}$	0.47 ± 0.2	0.89 ± 0.1	0.76 ± 0.2	0.69 ± 0.3	0.63 ± 0.3
	splice	$\mathbf{0.06 \pm 0.0}$	0.09 ± 0.0	0.28 ± 0.0	0.21 ± 0.0	0.09 ± 0.0	0.09 ± 0.0	0.28 ± 0.0
	svmg1	$\mathbf{0.06 \pm 0.0}$	0.08 ± 0.0	$\mathbf{0.06 \pm 0.0}$	$\mathbf{0.06 \pm 0.0}$	0.07 ± 0.0	0.07 ± 0.0	$\mathbf{0.06 \pm 0.0}$
P@k	a1a	$\mathbf{0.22 \pm 0.1}$	0.34 ± 0.1	0.28 ± 0.1	0.32 ± 0.1	0.34 ± 0.2	0.40 ± 0.1	0.29 ± 0.1
	german	$\mathbf{0.13 \pm 0.0}$	0.16 ± 0.1	$\mathbf{0.13 \pm 0.0}$	0.33 ± 0.0	0.20 ± 0.0	0.21 ± 0.1	0.18 ± 0.0
	gisette	0.01 ± 0.0	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$	0.02 ± 0.0	0.02 ± 0.0	0.01 ± 0.0
	gmsc	0.51 ± 0.0	$\mathbf{0.48 \pm 0.0}$	0.74 ± 0.1	0.88 ± 0.0	0.65 ± 0.1	0.62 ± 0.0	0.96 ± 0.0
	heart	0.07 ± 0.1	0.19 ± 0.1	$\mathbf{0.06 \pm 0.0}$	0.19 ± 0.1	0.23 ± 0.1	0.29 ± 0.1	0.14 ± 0.2
	iono.	$\mathbf{0.03 \pm 0.0}$	0.04 ± 0.1	0.05 ± 0.0	0.06 ± 0.1	0.09 ± 0.1	0.10 ± 0.1	0.10 ± 0.1
	liver	$\mathbf{0.27 \pm 0.2}$	0.33 ± 0.2	0.33 ± 0.2	0.40 ± 0.3	0.40 ± 0.2	0.33 ± 0.2	0.30 ± 0.2
	oil-spill	$\mathbf{0.44 \pm 0.2}$	0.72 ± 0.2	0.84 ± 0.2	0.92 ± 0.1	0.72 ± 0.2	0.68 ± 0.3	0.68 ± 0.2
	splice	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.01 \pm 0.0}$	0.04 ± 0.0	0.04 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	0.05 ± 0.0
	svmg1	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.01 ± 0.0	0.05 ± 0.0	0.05 ± 0.0	$\mathbf{0.00 \pm 0.0}$

Table 4: Evaluation of ensemblers. The exact benchmarks are RankBoost (AUC), DMKS (KS) and TopPush (P@k). ExactBoost is generally the best performer (and top 2 in all cases, for all losses).

Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Adam, L., Mácha, V., Šmídl, V., and Pevnỳ, T. (2020). General framework for binary classification on top samples. *arXiv preprint arXiv:2002.10923*.

Agarwal, S. (2013). Surrogate regret bounds for the area under the roc curve via strongly proper losses. In *Conference on Learning Theory*, pages 338–353. PMLR.

Bartlett, P. and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Boyd, S., Cortes, C., Mohri, M., and Radovanovic, A. (2012). Accuracy at the top. In *Advances in neural information processing systems*, pages 953–961.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Cortes, C. and Mohri, M. (2003). AUC optimization vs. error rate minimization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 313–320, Cambridge, MA, USA. MIT Press.

Dembczyński, K., Kotłowski, W., Koyejo, O., and Natarajan, N. (2017). Consistency analysis for binary classification revisited. In *International Conference on Machine Learning*, pages 961–969. PMLR.

Engilberge, M., Chevallier, L., Pérez, P., and Cord, M. (2019). Sodeep: a sorting deep net to learn ranking loss surrogates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10792–10801.

Fang, F. and Chen, Y. (2019). A new approach for credit scoring by directly maximizing the kolmogorov–smirnov statistic. *Computational Statistics & Data Analysis*, 133:180–194.

Fathony, R. and Kolter, Z. (2020). Ap-perf: Incorporating generic performance metrics in differentiable

learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4130–4140. PMLR.

Ferri, C., Flach, P., and Hernández-Orallo, J. (2002). Learning decision trees using the area under the ROC curve. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 139–146.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4(null):933–969.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232.

Gao, X., Zhang, H., Panahi, A., and Arodz, T. (2019). Differentiable combinatorial losses through generalized gradients of linear programs. *arXiv preprint arXiv:1910.08211*.

Grabocka, J., Scholz, R., and Schmidt-Thieme, L. (2020). Learning surrogate losses.

Hickey, T., Ju, Q., and van Emden, M. (2001). Interval arithmetic: From principles to implementation. *J. ACM*, 48:1038–1068.

Jiang, Q., Adigun, O., Narasimhan, H., Fard, M. M., and Gupta, M. (2020). Optimizing black-box metrics with adaptive surrogates. In *International Conference on Machine Learning*, pages 4784–4793. PMLR.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 377–384, New York, NY, USA. Association for Computing Machinery.

Kar, P., Narasimhan, H., and Jain, P. (2014). Online and stochastic gradient methods for non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pages 694–702.

Kar, P., Narasimhan, H., and Jain, P. (2015). Surrogate functions for maximizing precision at the top. In *International Conference on Machine Learning*, pages 189–198. PMLR.

Koltchinskii, V. and Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Ann. Statist.*, 30(1):1–50.

Koyejo, O., Natarajan, N., Ravikumar, P., and Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics. In *NIPS*, volume 27, pages 2744–2752. Citeseer.

Li, N., Jin, R., and Zhou, Z.-H. (2014). Top rank optimization in linear time. In *Advances in neural information processing systems*, pages 1502–1510.

Lyu, S. and Ying, Y. (2018). A univariate bound of area under ROC. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 43–52. AUAI Press.

McDiarmid, C. (1998). *Concentration*, pages 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg.

McInnes, L. and Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pfetsch, M. and Pokutta, S. (2020). Ipboost–nonconvex boosting via integer programming. In *International Conference on Machine Learning*, pages 7663–7672. PMLR.

Schapire, R. E. and Freund, Y. (2013). Boosting: Foundations and algorithms. *Kybernetes*.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5):1651–1686.

Tasche, D. (2018). A plug-in approach to maximising precision at the top and recall at the top. *CoRR*, abs/1804.03077.

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Zhai, S., Xia, T., Tan, M., and Wang, S. (2013). Direct 0-1 loss minimization and margin maximization with boosting. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 872–880. Curran Associates, Inc.

Supplementary Material: ExactBoost: Directly Boosting the Margin in Combinatorial and Non-decomposable Metrics

A Proofs and technical results

Subsection A.1 collects some preliminary or technical results, while Subsection A.2 has the proofs for all the results presented in the paper.

A.1 Technical Rresults

We present a general theoretical framework that we apply to obtain the margin results in Section A.2.

Let $Z_1, \ldots, Z_m$ be an iid sample from a probability distribution $\mathcal{D}_Z$ over a feature space $\mathcal{Z}$ (with suitable σ-field). Given a family of measurable functions $\mathcal{G}$ from $\mathcal{Z}$ to $\mathbb{R}$, its (averaged) Rademacher complexity is defined as

$$\mathcal{R}_m(\mathcal{G}) \coloneqq \mathbb{E}_{Z_1,\ldots,Z_m \sim \mathcal{D}_Z} \mathbb{E}_\sigma \sup_{g \in \mathcal{G}} \frac{\sum_{i=1}^m \sigma_i g(Z_i)}{m},$$

where the $\sigma_1, \ldots, \sigma_m$ are iid uniform over $\{-1, +1\}$ and independent of the Z_i. We assume implicitly throughout this section that that the families $\mathcal{G}$ we consider is nice enough that the supremum is measurable and integrable. A fundamental property of $\mathcal{R}_m(\mathcal{G})$ is the *symmetrization inequality*: if all functions $g \in \mathcal{G}$ are integrable,

$$\mathbb{E}_{Z_1,\ldots,Z_m \sim \mathcal{D}_Z} \sup_{g \in \mathcal{G}} \frac{\sum_{i=1}^m \mathbb{E}_{Z \sim \mathcal{D}_Z} g(Z) - g(Z_i)}{m} \leq 2\mathcal{R}_m(\mathcal{G}). \tag{9}$$

A.1.1 Empirical vs. cumulative distribution functions

We now note a "margin-type" result relating population and empirical cumulative distribution functions of elements of $\mathcal{G}$. It essentially follows from (Koltchinskii and Panchenko, 2002, Theorem 1).

Lemma 1. *With the above notation, assume further that the functions in $\mathcal{G}$ are bounded by 1 in absolute value. Given $\eta > 0$, the inequality below holds with probability at least $1 - \delta$:*

$$\forall g \in \mathcal{G}, t \in \mathbb{R} : \Pr_{Z \sim \mathcal{D}_Z} \{g(Z) \leq t\} \leq \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[g(Z_i) \leq t + \eta]} + \frac{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}}{\eta} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Similarly, the following holds with probability at least $1 - \delta$:

$$\forall g \in \mathcal{G}, t \in \mathbb{R} : \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[g(Z_i) \leq t]} \leq \Pr_{Z \sim \mathcal{D}_Z} \{g(Z) \leq t + \eta\} + \frac{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}}{\eta} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Proof. We only prove the first of these results, as the second one is similar. Define:

$$\Delta \coloneqq \sup_{g \in \mathcal{G}, t \in \mathbb{R}} \left(\Pr_{Z \sim \mathcal{D}_Z} \{g(Z) \leq t\} - \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[g(Z_i) \leq t + \eta]} \right).$$

Since $\|g\|_\infty \leq 1$ for all $g \in \mathcal{G}$, the term inside the brackets is equal to 0 for $t \geq 1$ and at most 0 for $t \leq -1$. In particular, the supremum defining Δ is nonnegative and achieved for some $t \in [-1, 1]$.

Now consider $\phi_\eta : \mathbb{R} \to [0,1]$ defined by:

$$\phi_\eta(x) \coloneqq \begin{cases} 1, & x \leq 0; \\ 1 - \frac{x}{\eta}, & 0 < x \leq \eta; \\ 0, & x > \eta. \end{cases} \quad (x \in \mathbb{R}).$$

Then we see at once that $\mathbf{1}_{[g(Z_i) \leq t+\eta]} \geq \phi_\eta(g(Z_i) - t) \geq \mathbf{1}_{[g(Z_i) \leq t]}$, so that, for any $g \in \mathcal{G}$ and $t \in [-1,1]$,

$$\Pr_{Z \sim \mathcal{D}_Z}\{g(Z) \leq t\} - \mathbf{1}_{[g(Z_i) \leq t+\eta]} \leq \mathbb{E}_{Z \sim \mathcal{D}_Z} \phi_\eta(g(Z) - t) - \phi_\eta(g(Z_i) - t).$$

Therefore,

$$\Delta \leq \Delta^* \coloneqq \sup_{g \in \mathcal{G},\, t \in [-1,1]} \left(\mathbb{E}_{Z \sim \mathcal{D}} \phi_\eta(g(Z) - t) - \frac{1}{m} \sum_{i=1}^m \phi_\eta(g(Z_i) - t) \right).$$

We now consider Δ^*. The symmetrization inequality (9) implies that

$$\mathbb{E}\, \Delta^* \leq 2\mathcal{R}_m(\tilde{\mathcal{G}}), \tag{10}$$

where $\tilde{\mathcal{G}}$ is the family of all functions of the form $\phi_\eta(g(\cdot) - t) - \phi_\eta(0)$ where $g \in \mathcal{G}$ and $t \in [-1,1]$. Note also that ϕ_η is $1/\eta$-Lipschitz. Using items 4 and 5 of (Bartlett and Mendelson, 2002, Theorem 12), we see that:

$$\mathcal{R}_m(\tilde{\mathcal{G}}) \leq 2\, \frac{\mathcal{R}_m(\mathcal{G}) + \frac{1}{\sqrt{m}}}{\eta}. \tag{11}$$

This bounds $\mathbb{E}\, \Delta^*$. To obtain a concentration inequality, notice that the random variable Δ^* is a function of independent random variables $Z_1, \ldots, Z_n$, and that changing the value of one of the Z_i will change the value of Δ^* by at most $1/m$ in absolute value. McDiarmid's inequality implies:

$$\Pr\left\{ \Delta^* - \mathbb{E}\, \Delta^* \leq \sqrt{\frac{\log(1/\delta)}{2m}} \right\} \geq 1 - \delta.$$

Combining this with (10) and (11) finishes the proof. □

The following corollary of Lemma 1 will also be useful. It may be viewed as a high-probability uniform bound for the Levy distance between empirical and population cdf's of $g \in \mathcal{G}$.

Corollary 1. *In the setting of Lemma 1, let*

$$\bar{\eta}_m(\mathcal{G}) \coloneqq \sqrt{4\mathcal{R}_m(\mathcal{G}) + \frac{4}{\sqrt{m}}} + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

Then either of the following statements holds with probability at least $1 - \delta$*:*

$$\forall g \in \mathcal{G}\, \forall t \in \mathbb{R} : \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[g(Z_i) \leq t]} \leq \Pr_{Z \sim \mathcal{D}_Z}\{g(Z) \leq t + \bar{\eta}_m(\mathcal{G})\} + \bar{\eta}_m(\mathcal{G});$$

and

$$\forall g \in \mathcal{G}\, \forall t \in \mathbb{R} : \Pr_{Z \sim \mathcal{D}_Z}\{g(Z) \leq t\} \leq \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[g(Z_i) \leq t + \bar{\eta}_m(\mathcal{G})]} + \bar{\eta}_m(\mathcal{G}).$$

Proof. Apply both parts of Lemma 1 with $\delta/2$ replacing δ and $\eta = \bar{\eta}_m(\mathcal{G})$. □

A.1.2 Rademacher complexities and U-statistic-type sums of indicators

When we consider the AUC metric, we will need a "U-statistic" result for families $\mathcal{G}$. Let $\mathcal{D}'_Z$ be another probability distribution over $\mathcal{Z}$ and $Z'_1, \ldots, Z'_{m'} \sim \mathcal{D}'_Z$ be an iid sample of size m' from that distribution which is independent from $Z_1, \ldots, Z_m$. We let $\mathcal{R}'_{m'}(\mathcal{G})$ denote the Rademacher complexity of $\mathcal{G}$ with respect to the new sample size m' and the new distribution $\mathcal{D}'_Z$.

Lemma 2. *With the above definitions and notation, let $\eta > 0$ and $\delta \in (0,1)$ be given. Let $m_{\min} \coloneqq \min\{m, m'\} > 0$. Then the following holds with probability at least $1-\delta$: for all $g \in \mathcal{G}$,*

$$\Pr_{(Z,Z') \sim \mathcal{D}_Z \times \mathcal{D}_{Z'}} \{g(Z) \leq g(Z')\} \leq \frac{1}{m\,m'} \sum_{i=1}^{m} \sum_{i'=1}^{m'} \mathbf{1}_{[g(Z_i) < g(Z'_{i'}) + \eta]} + 4\,\frac{\mathcal{R}_{m_{\min}}(\mathcal{G}) + \mathcal{R}'_{m_{\min}}(\mathcal{G})}{\eta} + \sqrt{\frac{\log(1/\delta)}{m_{\min}}}.$$

Proof. The rough outline of this proof is similar to that of Lemma 1. We replace indicators by the function ϕ_η; apply symmetrization to bound the expectation of a supremum; and use McDiarmid's inequality to prove concentration. The key difference is at the symmetrization step, where we need to circumvent the fact that we are considering a U-statistic (rather than an iid sum).

Let ϕ_η be as in the proof of Lemma 1, and define

$$\Delta^* \coloneqq \sup_{g \in \mathcal{G}} \left(\mathbb{E}_{(Z,Z') \sim \mathcal{D}_Z \times \mathcal{D}_{Z'}} \phi_\eta(g(Z) - g(Z')) - \frac{1}{m\,m'} \sum_{i=1}^{m} \sum_{i'=1}^{m'} \phi_\eta(g(Z_i) - g(Z'_{i'})) \right).$$

The reasoning in the previous proof shows that:

$$\sup_{g \in \mathcal{G}} \left(\Pr_{(Z,Z') \sim \mathcal{D}_Z \times \mathcal{D}_{Z'}} \{g(Z) \leq g(Z')\} - \frac{1}{m\,m'} \sum_{i=1}^{m} \sum_{i'=1}^{m'} \mathbf{1}_{[g(Z_i) \leq g(Z'_{i'}) + \eta]} \right) \leq \Delta^*. \tag{12}$$

Our proof focuses on controlling Δ^*. We first notice a concentration property. Notice that Δ^* is a function of independent variables Z_i and $Z'_{i'}$. Since $\|\phi_\eta\|_\infty = 1$, changing one of the Z_i will change Δ^* by at most $1/m$ in absolute value, and changing a $Z'_{i'}$ will only change Δ^* by at most $1/m'$. Applying McDiarmid's inequality McDiarmid (1998), we obtain:

$$\Pr\left\{ \Delta^* - \mathbb{E}\,\Delta^* \leq \sqrt{\frac{\log(1/\delta)}{2\left(\frac{1}{\frac{1}{m} + \frac{1}{m'}}\right)}} \right\} \geq 1 - \delta,$$

so that in particular

$$\Pr\left\{ \Delta^* - \mathbb{E}\,\Delta^* \leq \sqrt{\frac{\log(1/\delta)}{m_{\min}}} \right\} \geq 1 - \delta. \tag{13}$$

We now need to bound $\mathbb{E}\,\Delta^*$ in terms of Rademacher complexities. The main difficulty is that Δ^* is not an iid sum, and the symmetrization inequality (9) does not apply directly. However, one can use an averaging argument to obtain an upper bound for the expectation in terms of an iid sum.

The argument is as follows. Let $\mathcal{I}$ be the set of all pairs (S, f), where $S \subset [m]$ has size $m_{\min}$ and $f : S \to [m']$ is a one-to-one function (note that such (S, f) exist because $m_{\min} = \min\{m, m'\}$). By symmetry, we see that for all $(i, i') \in [m] \times [m']$,

$$\frac{\#\{(S,f) \in \mathcal{I} : i \in S,\, f(i) = i'\}}{\#\mathcal{I}} = \frac{m_{\min}}{m\,m'}.$$

Therefore,

$$\frac{1}{m\,m'} \sum_{i=1}^{m} \sum_{i'=1}^{m'} \phi_\eta(g(Z_i) - g(Z'_{i'})) = \frac{1}{\#\mathcal{I}} \sum_{(S,f) \in \mathcal{I}} \sum_{i \in S} \frac{\phi_\eta(g(Z_i) - g(Z'_{f(i)}))}{m_{\min}}.$$

Now plug the above into the definition of Δ^*, and obtain:

$$\Delta^* = \sup_{g\in\mathcal{G}} \left(\frac{1}{\#\mathcal{I}} \sum_{(S,f)\in\mathcal{I}} \sum_{i\in S} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z') - g(Z)) - \phi_\eta(g(Z_i) - g(Z'_{f(i)}))}{m_{\min}} \right).$$

That is, Δ^* is the supremum of an average over $(S,f)\in\mathcal{I}$. The corresponding "average of suprema" is at least as large, so

$$\Delta^* \leq \frac{1}{\#\mathcal{I}} \sum_{(S,f)\in\mathcal{I}} \sup_{g\in\mathcal{G}} \left(\sum_{i\in S} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z) - g(Z')) - \phi_\eta(g(Z_i) - g(Z'_{f(i)}))}{m_{\min}} \right).$$

Crucially, *all terms in the sum over $(S,f)\in\mathcal{I}$ have the same distribution.* In particular, all terms in the RHS of the preceding display have the same expectation. Considering the case where $S = [m_{\min}]$ and $f(i) = i$ for each $i \in [m_{\min}]$, we conclude:

$$\mathbb{E}\,\Delta^* \leq \mathbb{E} \sup_{g\in\mathcal{G}} \left(\sum_{i=1}^{m_{\min}} \frac{\mathbb{E}_{(Z,Z')\sim\mathcal{D}_Z\times\mathcal{D}'_Z} \phi_\eta(g(Z) - g(Z')) - \phi_\eta(g(Z_i) - g(Z'_i))}{m_{\min}} \right).$$

The pairs $\{(Z_i, Z'_i)\}_{i=1}^{m_{\min}}$ are i.i.d, and we can now apply symmetrization inequality (9). Letting

$$\tilde{\mathcal{G}} \coloneqq \{ \text{ all functions of the form "}(z,z')\in\mathcal{Z}\times\mathcal{Z} \mapsto \phi_\eta(g(z) - g(z')) - \phi_\eta(0)\text{" w/ } g\in\mathcal{G}\},$$

we obtain:

$$\mathbb{E}\,\Delta^* \leq\ 2\,\mathbb{E} \sup_{\tilde{g}\in\tilde{\mathcal{G}}} \left(\sum_{i=1}^{m_{\min}} \frac{\sigma_i\, \tilde{g}(Z_i, Z'_i)}{m_{\min}} \right)$$

where the σ_i are iid uniform over ± 1 and independent from the Z_i and Z'_i. As in the proof of Lemma 1, we observe that ϕ_η is $1/\eta$-Lipschitz, and apply item 5 of (Bartlett and Mendelson, 2002, Theorem 12) to obtain:

$$\mathbb{E}\,\Delta^* \leq\ \frac{4}{\eta}\,\mathbb{E} \sup_{g\in\mathcal{G}} \left(\sum_{i=1}^{m_{\min}} \frac{\sigma_i\,(g(Z_i) - g(Z'_i))}{m_{\min}} \right) \leq \frac{4\mathcal{R}_{m_{\min}}(\mathcal{G}) + 4\mathcal{R}'_{m_{\min}}(\mathcal{G})}{\eta}.$$

Combining this bound with (13) and (12) gives the Lemma. □

A.1.3 Other auxiliary results

Proposition 3. *If $\mathcal{H}$ consists of binary functions with VC dimension bounded by d, then $\mathcal{R}_n(\mathcal{H}) \leq C\sqrt{d/n}$ and $\mathcal{R}_{n,y}(\mathcal{H}) \leq C\sqrt{d/n_y}$ (conditionally on $n_y > 0$) for some universal, distribution-independent constant $C > 0$. If $\mathcal{H} = \mathsf{Stumps}$ consists of all stumps over $\mathbb{R}^p$ with coefficients in $[-1,1]$, then $\mathcal{R}_n(\mathsf{Stumps}) \leq C\sqrt{\log p/n}$ and $\mathcal{R}_{n,y}(\mathsf{Stumps}) \leq C\sqrt{\log p/n_y}$ (conditionally on $n_y > 0$), with $C > 0$ universal.*

Proof of Proposition 3. The first statement is (Bartlett and Mendelson, 2002, Theorem 6, Lemma 4). The second results from the following steps. Given a coordinate $j\in[p]$, use $x^{(j)}$ to denote the j-th coordinate of x. Let Stumps_j denote the set of all functions of the form

$$x\in\mathbb{R}^p \mapsto a\mathbf{1}_{[x^{(j)}\leq\xi]} + b\mathbf{1}_{[x^{(j)}>\xi]}, \text{ with } a,b\in[-1,1],\ \xi\in\mathbb{R}.$$

Each $f\in\mathsf{Stumps}_j$ is a convex combination of the 0 function and functions of the form $\pm 2\mathbf{1}_{[x^{(j)}\leq\xi]}$, $\pm 2\mathbf{1}_{[x^{(j)}>\xi]}$. For each j, each family $\{\mathbf{1}_{[x^{(j)}\leq\xi]}\}\cup\{\mathbf{1}_{[x^{(j)}>\xi]}\}\cup\{0\}$ comprises 0/1-valued functions with VC dimension bounded by an absolute constant. From (Bartlett and Mendelson, 2002, Theorem 6, Lemma 4), their Rademacher complexities are $O(1/\sqrt{n})$, which doesn't change when these functions are multiplied by 2. Moreover, passing to the convex hull does not change the Rademacher complexity, as shown in (Bartlett and Mendelson, 2002, Theorem 12, items 3 and 7). We deduce that $\mathcal{R}_n(\mathsf{Stumps}_j) = O(1/\sqrt{n})$. Now,

$$\mathcal{R}_n(\mathsf{Stumps}) \ -\ \max_{j\in[p]} \mathcal{R}_n(\mathsf{Stumps}_j) \quad\leq\quad \mathbb{E} \max_{j\in[p]} \left[\sup_{h\in\mathsf{Stumps}_j} \frac{1}{n}\sum_{i=1}^n \sigma_i h(X_i) - \mathbb{E}\left(\sup_{h\in\mathsf{Stumps}_j} \frac{1}{n}\sum_{i=1}^n \sigma_i h(X_i) \right) \right].$$

The random variables inside the supremum in the RHS have zero mean. By McDiarmid's inequality McDiarmid (1998), they are also sub-Gaussian with variance proxies $O(1/n)$. By (Vershynin, 2018, Exercise 2.5.10), the expectation of the maximum satisfies:

$$\mathbb{E}\max_{j\in[p]}\left(\sup_{h\in\mathsf{Stumps}_j}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(X_i)-\mathcal{R}_n(\mathsf{Stumps}_j)\right)\leq C\sqrt{\frac{\log p}{n}},\text{ with } C>0 \text{ universal.}$$

This implies $\mathcal{R}_n(\mathsf{Stumps})\leq C\sqrt{\log p/n}$, with a potentially larger (but still universal) $C>0$. The bounds for $\mathcal{R}_{n,y}(\mathsf{Stumps})$ follow similarly once we condition on the number of examples with the two labels. □

A.2 Proofs of the results in the paper

Theorem 4. *Given $\theta>0$, $\delta\in(0,1)$, $n_0,n_1>0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, the following holds with probability at least $1-\delta$: for all score functions $S:\mathbb{R}^p\to[-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$,*

$$\mathrm{AUC}(S)\leq\widehat{\mathrm{AUC}}_\theta(S)+\frac{4}{\theta}\zeta_{\mathrm{AUC}}(\mathcal{H})+\sqrt{\frac{2\log(1/\delta)}{\min\{n_0,n_1\}}},$$

where $\zeta_{\mathrm{AUC}}(\mathcal{H})=\mathcal{R}_{\min\{n_0,n_1\},0}(\mathcal{H})+\mathcal{R}_{\min\{n_0,n_1\},1}(\mathcal{H})$.

Proof of Theorem 4. As we explain below, the proof is a direct application of Lemma 2 to the two distributions $\mathcal{D}_1=\mathcal{D}_Z$ and $\mathcal{D}_0=\mathcal{D}'_Z$ with $\mathcal{G}=\mathrm{conv}(\mathcal{H})$, with $\eta=\theta$. Importantly, the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are equal (Bartlett and Mendelson, 2002, Theorem 12).

The only slightly subtle aspect in our argument, which will also come up in later proofs, is the following. We wish to control the probability of an event E given by "the inequality for $\mathrm{AUC}(S)$ in Theorem 4 holds for all S in the convex hull of $\mathcal{H}$." Now consider what happens when one conditions on specific (non-random) values $n_0=m_0>0$ and $n_1=m_1=n-m_0>0$; that is, $m_0,m_1=n-m_0$ are fixed (non-random) positive integers such that $\Pr(n_0=m_0,n_1=m_1)>0$. Crucially, under this conditioning, the subsamples $\mathbf{X}_1=\{X_i:y_i=1\}$ and $\mathbf{X}_0\{X_i:y_i=0\}$ corresponding to 1- and 0-labelled examples (respectively) are iid with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. Under this conditioning, Lemma 2 gives that E holds with probability $\geq 1-\delta$. This is irrespective of the choice of $m_0,m_1=n-m_0>0$. Therefore, we discover that

$$\begin{aligned}\Pr(E\mid\min\{n_0,n_1\}>0)=&\sum_{m_0=1}^{n-1}\Pr(E\mid n_0=m_0,n_1=n-m_0)\Pr(n_0=m_0,n_1=n-m_0\mid\min\{n_0,n_1\}>0)\\ \geq&1-\delta.\end{aligned}$$

□

Remark 1. *The same reasoning we gave above shows that, for any event E,*

$$\Pr(E\mid\min\{n_0,n_1\}>0)\geq\min\{\Pr(E\mid n_0=m_0,n_1=n-m_0),1\leq m_0\leq n-1\}.$$

Moreover, under the conditioning in the RHS, the subsamples $\mathbf{X}_1=\{X_i:y_i=1\}$ and $\mathbf{X}_0\{X_i:y_i=0\}$ corresponding to 1- and 0-labelled examples (respectively) are iid with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. In later proofs, we will abuse notation slightly and compute $\Pr(E)$ assuming that n_0 and n_1 are fixed positive constants, as all bounds on $\Pr(E\mid n_0=m_0,n_1=n-m_0)$ we obtain are uniform in the choice of $0<m_0<n$.

Theorem 5. *Given $\theta>0$, $\delta\in(0,1)$, $n_0,n_1>0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, the following holds with probability at least $1-\delta$: for all score functions $S:\mathbb{R}^p\to[-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$,*

$$\mathrm{KS}(S)\leq\widehat{\mathrm{KS}}_\theta(S)+\frac{8}{\theta}\zeta_{\mathrm{KS}}(\mathcal{H})+\sqrt{\frac{\log(2/\delta)}{2}}\left(\frac{1}{\sqrt{n_0}}+\frac{1}{\sqrt{n_1}}\right),$$

where $\zeta_{\mathrm{KS}}(\mathcal{H})=\mathcal{R}_{n_0,0}(\mathcal{H})+\mathcal{R}_{n_1,1}(\mathcal{H})+n_0^{-1/2}+n_1^{-1/2}$.

Proof of Theorem 5. We want to prove that, with probability $\geq 1-\delta$, conditionally on $\min\{n_0, n_1\} > 0$, for all S in the convex hull of $\mathcal{H}$,

$$\mathrm{KS}(S) \leq \widehat{\mathrm{KS}}_\theta(S) + \frac{8}{\theta}\zeta_{\mathrm{KS}}(\mathcal{H}) + \sqrt{\frac{\log(2/\delta)}{2}}\left(\frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}}\right), \tag{14}$$

where

$$\zeta_{\mathrm{KS}}(\mathcal{H}) = \mathcal{R}_{n_0,0}(\mathcal{H}) + \mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{1}{\sqrt{n_0}} + \frac{1}{\sqrt{n_1}}.$$

To this end, we apply Lemma 1 from Section A.1.1 to the two subsamples $\mathbf{X}_1$ and $\mathbf{X}_0$, with $\eta = \theta/2$, $\delta/2$ replacing δ, and $\mathcal{G} = \mathrm{conv}(\mathcal{H})$ equal to the convex hull of $\mathcal{H}$. As described in Remark 1 above, we abuse notation slightly and treat n_0, n_1 as fixed (non-random) positive integers in what follows; that is, n_0, n_1 represent specific values of these random variables. Under this (implicit) conditioning, the subsamples $\mathbf{X}_1 = \{X_i : y_i = 1\}$ and $\mathbf{X}_0\{X_i : y_i = 0\}$ corresponding to 1- and 0-labelled examples (respectively) are iid with respective laws $\mathcal{D}_1$ and $\mathcal{D}_0$, and independent from one another. Thus Lemma 1 indeed applies.

To continue, we recall that the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are the same (see (Bartlett and Mendelson, 2002, Theorem 12)). Therefore, Lemma 1 allows us to deduce that, conditionally on specific values of $n_0, n_1 > 0$, with probability at least $1-\delta$, the following two inequalities hold simultaneously for all $S \in \mathrm{conv}(\mathcal{H})$ and $t \in \mathbb{R}$:

$$\begin{aligned}
\Pr_{X\sim\mathcal{D}_1}\{S(X) \leq t\} &\leq \frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\leq t+\frac{\theta}{2}]} + \varepsilon_1, \\
\frac{1}{n_0}\sum_{i\,:\,y_i=0} \mathbf{1}_{[S(X_i)\leq t-\frac{\theta}{2}]} &\leq \Pr_{X\sim\mathcal{D}_0}\{S(X) \leq t\} + \varepsilon_0,
\end{aligned}$$

where, for $y = 0, 1$:

$$\varepsilon_y \coloneqq \frac{8\mathcal{R}_{n_y,y}(\mathcal{G}) + \frac{8}{\sqrt{n_y}}}{\theta} + \sqrt{\frac{\log(2/\delta)}{2n_y}}.$$

Now notice that, when these two inequalities hold, we also have

$$\begin{aligned}
\mathrm{KS}(S) - 1 &= \inf_{t\in\mathbb{R}}\left(\Pr_{X\sim\mathcal{D}_1}\{S(X) \leq t\} - \Pr_{X\sim\mathcal{D}_0}\{S(X) \leq t\}\right) \\
&\leq \inf_{t\in\mathbb{R}}\left(\frac{1}{n_1}\sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\leq t+\frac{\theta}{2}]} - \frac{1}{n_0}\sum_{i\,:\,y_i=0} \mathbf{1}_{[S(X_i)\leq t-\frac{\theta}{2}]}\right) + \varepsilon_0 + \varepsilon_1 \\
&= \widehat{\mathrm{KS}}_\theta(S) - 1 + \varepsilon_0 + \varepsilon_1,
\end{aligned}$$

which inspection reveals to be the same inequality as (14). Therefore, the probability of (14) holding is also at least $1-\delta$ (conditionally on $n_0, n_1 > 0$). □

Theorem 6. *Given $\theta > 0$, $\delta \in (0,1)$, $n_0, n_1 > 0$, and a class of functions $\mathcal{H}$ from $\mathbb{R}^p$ to $[-1,1]$, define*

$$\bar{\eta}_n(\mathcal{H}) \coloneqq \sqrt{4\mathcal{R}_n(\mathcal{H}) + \frac{4}{\sqrt{n}}} + \sqrt{\frac{\log(3/(\delta-\delta^2))}{n}},$$

Assume $\theta > 2\bar{\eta}_n(\mathcal{H})$ and $\Pr(\min\{n_0, n_1\} > 0) \geq 1-\delta$. Then the following holds with probability $\geq 1-\delta$: if $\delta' \coloneqq \delta - \delta^2$, then for all score functions $S : \mathbb{R}^p \to [-1,1]$ obtained as convex combinations of the elements of $\mathcal{H}$, it holds

$$\begin{aligned}
\mathrm{P@k}(S) \leq& \widehat{\mathrm{P@k}}_\theta(S) + \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} \\
&+ \bar{\eta}_n(\mathcal{H}) + \sqrt{2\frac{\log(3/\delta')}{n_1}} + \sqrt{\frac{\log(3/\delta')}{2n}}.
\end{aligned}$$

Proof of Theorem 6. This proof is somewhat more complex than preceding examples. As before, let $\mathcal{G} \coloneqq \text{conv}(\mathcal{H})$ denote the convex hull of $\mathcal{H}$. We will use below that the Rademacher complexities of $\mathcal{G}$ and $\mathcal{H}$ are always equal.

For convenience, we define

$$\Gamma \coloneqq \sup_{S \in \mathcal{G}} \left(\frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i) \geq \widehat{t}_\alpha(S)+\theta]} - \Pr_{X \sim \mathcal{D}_1}\{S(X) \geq t_\alpha(S)\} \right), \tag{15}$$

so that we can write, for any $S \in \mathcal{G}$:

$$\begin{aligned} \text{P@k}(S) - \widehat{\text{P@k}}_\theta(S) &\leq \Pr_{(X,y)\sim\mathcal{D}}\{y=1\}\,\Gamma && (16)\\ &\quad + \left(\frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i) \geq \widehat{t}_\alpha(S)+\theta]} \right) \left(\frac{n_1}{n} - \Pr_{(X,y)\sim\mathcal{D}}\{y=1\} \right) && (17)\\ &\leq \Pr_{(X,y)\sim\mathcal{D}}\{y=1\}\,\Gamma + \max\left\{ \left(\frac{n_1}{n} - \Pr_{(X,y)\sim\mathcal{D}}\{y=1\} \right), 0 \right\}. && (18) \end{aligned}$$

If we define an event,

$$C = \left\{ \frac{n_1}{n} \leq \Pr_{(X,y)\sim\mathcal{D}}\{y=1\} + \sqrt{\frac{\log(3/(\delta-\delta^2))}{2n}} \right\}, \tag{19}$$

it is clear that $\Pr(C) \geq 1-\delta/3+\delta^2/3$ due to a simple application of Hoeffding's inequality. Since $\Pr(\min\{n_0, n_1\} > 0) \geq 1-\delta$,

$$\Pr(C \mid \min\{n_0, n_1\} > 0) \geq 1 - \frac{\Pr(C^c)}{\Pr(\min\{n_0, n_1\} > 0)} \geq 1 - \delta/3.$$

Now consider another event D defined as follows: either $\min\{n_0, n_1\} = 0$, or

$$\Gamma \leq \frac{\overline{\eta}_n(\mathcal{H})}{\Pr_{(X,y)\sim\mathcal{D}}\{y=1\}} + \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} + \sqrt{\frac{\log(3/\delta)}{2n_1}}. \tag{20}$$

We see from the above that, if $D \cap C$ holds, then (18) implies that either $\min\{n_0, n_1\} = 0$, or the inequality on $\text{P@k}(S) - \widehat{\text{P@k}}_\theta(S)$ claimed in the statement of the Theorem holds for all $S \in \mathcal{G}$. Therefore, we will be done once we show that $\Pr(D \cap C \mid \min\{n_0, n_1\} > 0) \geq 1-\delta$. In fact, since $\Pr(C \mid \min\{n_0, n_1\} > 0) \geq 1-\delta/3$, it suffices to show $\Pr(D \mid \min\{n_0, n_1\} > 0) \geq 1 - 2\delta/3$. This will be our goal for the remainder of the proof.

To continue, we define a third event which we use to control $t_\alpha(S)$, $\widehat{t}_\alpha(S)$ and related quantities. Define

$$E \coloneqq \left\{ \forall S \in \mathcal{G},\, \forall t \in \mathbb{R} \,:\, \Pr_{X\sim\mathcal{D}}\{S(X) \geq t\} \leq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[S(X_i) \geq t - \bar{\eta}_n(\mathcal{H})]} + \bar{\eta}_n(\mathcal{H}) \right\}. \tag{21}$$

This is the kind of event controlled by Corollary 1, except that we have $S(X_i) \geq t$ and $S(X) \geq t - \theta$ as opposed to "$\leq$" inequalities. However, the corollary still applies if we consider the functions $-S$ as S ranges over $\mathcal{G}$. This is tantamount to applying the corollary to the family of functions $-\mathcal{G} = \{-S \,:\, S \in \mathcal{G}\}$. Since $-\mathcal{G}$ has the same Rademacher complexity as $\mathcal{G}$ and $\mathcal{H}$, we obtain $\Pr(E) \geq 1 - \delta/3 + \delta^2/3$. As noted in the case of C, we obtain that $\Pr(E \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$.

We now claim the following.

Claim 1. *When E holds,*

$$\Pr_{X\sim\mathcal{D}_1}\{S(X) \geq t_\alpha(S)\} \geq \Pr_{X\sim\mathcal{D}_1}\{S(X) \geq \widehat{t}_\alpha(S) + 2\bar{\eta}_n(\mathcal{H})\} - \frac{\bar{\eta}_n(\mathcal{H})}{\Pr_{(X,y)\sim\mathcal{D}}\{y=1\}}. \tag{22}$$

Indeed, the claim is trivial if $t_* := \widehat{t}_\alpha(S) + 2\bar{\eta}_n(\mathcal{H}) \geq t_\alpha(S)$. Otherwise,

$$\begin{aligned}\Pr_{X\sim\mathcal{D}_1}\{S(X) \geq t_*\} - \Pr_{X\sim\mathcal{D}_1}\{S(X) \geq t_\alpha(S)\} &= \frac{\Pr_{(X,y)\sim\mathcal{D}}\{y=1, t_* \leq S(X) < t_\alpha(S)\}}{\Pr_{(X,y)\sim\mathcal{D}}\{y=1\}} \\ &\leq \frac{\Pr_{(X,y)\sim\mathcal{D}}\{t_* \leq S(X) < t_\alpha(S)\}}{\Pr_{(X,y)\sim\mathcal{D}}\{y=1\}}\end{aligned}$$

Since we know from the definition of $t_\alpha(S)$ that $\Pr_{(X,y)\sim\mathcal{D}}\{t_\alpha(S) \leq S(X)\} \geq \alpha$, we will be done if we show $\Pr_{X\sim\mathcal{D}}\{S(X) \geq t_*\} \leq \alpha + \bar{\eta}_n(\mathcal{H})$ whenever D holds. To do this, take $t = t_*$ in the definition of E. Since $t - \bar{\eta}_n(\mathcal{H}) > \widehat{t}_\alpha(S)$, and the latter is a $(1-\alpha)$-quantile for S, under the sample distribution, we obtain that:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[S(X_i) \geq t_* - \bar{\eta}_n(\mathcal{H})]} \leq \alpha,$$

and so, when E holds,

$$\Pr_{X\sim\mathcal{D}}\{S(X) \geq t_*\} \leq \alpha + \bar{\eta}_n(\mathcal{H}).$$

This gives us the claim.

To continue, we go back to the definition of Γ in (15) and notice that, by the Claim, when E holds,

$$\Gamma \leq \frac{\bar{\eta}_n(\mathcal{H})}{\Pr_{(X,y)\sim\mathcal{D}}\{y=1\}} + \Gamma^*,$$

where we define

$$\Gamma^* := \sup_{S\in\mathcal{G}, t\in\mathbb{R}} \left(\frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S(X_i)\geq t]} - \Pr_{X\sim\mathcal{D}_1}\{S(X) \geq t - (\theta - 2\bar{\eta}_n(\mathcal{H}))\} \right).$$

Recall that our goal is to show that the probability $\Pr(D \mid \min\{n_0, n_1\} > 0)$ above is at least $1 - 2\delta/3$. By the above reasoning, we see that $D \supset E \cap F$, where

$$F := \left\{ \Gamma^* \leq \frac{4\mathcal{R}_{n_1,1}(\mathcal{H}) + \frac{4}{\sqrt{n_1}}}{\theta - 2\bar{\eta}_n(\mathcal{H})} + \sqrt{\frac{\log(3/\delta)}{2n_1}} \right\}.$$

Since we know already that $\Pr(E \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$, we will be done once we show that $\Pr(F \mid \min\{n_0, n_1\} > 0) \geq 1 - \delta/3$, which (as seen above) will follow from $\Pr(F) \geq 1 - \delta/3 + \delta^2/3$.

At this last step, we will apply the reasoning in Remark 1 above: that is, we treat n_0 and n_1 as fixed constants and the subsamples $\mathbf{X}_0, \mathbf{X}_1$ as iid and independent. Under this (implicit) conditioning, Γ^* is almost the kind of quantity to which Lemma 1 applies, with $\eta = \theta - 2\bar{\eta}_n(\mathcal{H}) > 0$. The differences one notices is that there is a minus sign in front of η, and there are " $\geq$" signs where "$\leq$" should be. As we observed following (21), one can circumvent this by applying the Lemma to $-\mathcal{G}$. If we do that (with $\delta/3 - \delta^2/3$ replacing δ), we obtain that the event satisfies $\Pr(F) \geq 1 - \delta/3 + \delta^2/3$, as desired. □

Proposition 1. *Let $\widehat{L}$ be either the $\widehat{\mathrm{AUC}}$ or the $\widehat{\mathrm{KS}}$ loss. Consider a subset of indices $I = I_0 \cup I_1 \subset [n]$ chosen independently and uniformly at random with equal number of positive and negative cases, $|I_0| = |I_1| = k$. Let h_R be the optimal stump over the reduced sample $\{(X_j, y_j)\}_{j\in I}$ and score S and h_* the optimal stump over the entire sample $\{(X_i, y_i)\}_{i\in[n]}$. Then,*

$$\mathbb{E}[\widehat{L}(S + h_R)] \leq \widehat{L}(S + h_*) + \frac{e}{k},$$

where the expectation is over the choice of I.

Proof of Proposition 1. The idea of the proof is that any observation from the original sample is close (the precise meaning of this statement will be defined below) to some observation in the subsample with high probability. Moreover, the better is such approximation, the lower is the impact on the minimization of the target loss.

First, consider $\widehat{L} = \widehat{\mathrm{KS}}$. For each $j \in [p]$, let

$$f_{j,\leq}(a,\xi) \coloneqq \sum_{i=1}^{n} \rho_i \, \mathbf{1}_{[a \leq t_i, X_{i,(j)} \leq \xi]};$$
$$f_{j,>}(b,\xi) \coloneqq \sum_{i=1}^{n} \rho_i \, \mathbf{1}_{[b \leq t_i, X_{i,(j)} > \xi]},$$

Note that our objective function $\widehat{\mathrm{KS}}(S+h)$ is one minus the sum of $f_{j,\leq}$ and $f_{j,>}$ where h is a stump with parameters (a,b,j,ξ). Hence, our problem is equivalent to maximizing $f_{j,\leq} + f_{j,>}$. We also use $t_i = \widehat{t}(S) - S(X_i)$. For convenience, we assume the sample has been ordered so that $t_1 \leq t_2 \leq \cdots \leq t_n$.

Now imagine $a = t_i$ is changed to $a' = t_{i'}$ with $i' \leq i$. Notice that:

$$f_{j,\leq}(t_i,\xi) - f_{j,\leq}(t_{i'},\xi) = \sum_{\ell=i'}^{i-1} \rho_i \, \mathbf{1}_{[X_{i,(j)} \leq \xi]} \in \left[-\frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right],$$

where $\mathrm{pos}(i,i')$ and $\mathrm{neg}(i,i')$ count the number of positive and negative examples between t_i and $t_{i'}$, including the largest of the two extreme points (these are well-defined even if $i' > i$). Therefore,

$$\|f_{j,\leq}(t_i,\xi) - f_{j,\leq}(t_{i'},\xi)\| \leq \max\left\{ \frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right\} \tag{23}$$

If we like, we can say that the above implies that $f_{j,\leq}(t_i,\xi)$ is a 1-Lipschitz function of i in the pseudometric:

$$d(i,i') \coloneqq \max\left\{ \frac{\mathrm{pos}(i,i')}{n_1}, \frac{\mathrm{neg}(i,i')}{n_0} \right\}.$$

A similar property holds for the $f_{>,j}$ function.

Now let (a_*, b_*, j_*, ξ_*) be the parameters of the optimal h_*. Say $a_* = t_{i_*}$ and $b_* = t_{j_*}$ for indices $i_*, j_* \in [n]$. We consider a modified function $\tilde{h}$ where a_*, b_* are replaced by points $t_{\tilde{i}}, t_{\tilde{j}}$ with $\tilde{i}, \tilde{j} \in I$ chosen to minimize $d(i_*, \tilde{i}) + d(j_*, \tilde{j})$. Notice that:

$$\widehat{\mathrm{KS}}(S + h_R) \leq \widehat{\mathrm{KS}}(S + \tilde{h})$$

because $\tilde{h}$ is feasible for the optimization problem of which h_R achieves the minimum. Therefore,

$$\begin{aligned}
\mathbb{E}[\widehat{\mathrm{KS}}(S+h_R)] &\leq \mathbb{E}[\widehat{\mathrm{KS}}(S+\tilde{h})] \\
&\leq \widehat{\mathrm{KS}}(S+h_*) - \mathbb{E}[\widehat{\mathrm{KS}}(S+h_*) - \widehat{\mathrm{KS}}(S+\tilde{h})] \\
&\leq \widehat{\mathrm{KS}}(S+h_*) + \mathbb{E}[d(i_*,\tilde{i}) + d(j_*,\tilde{j})],
\end{aligned}$$

where the last step uses the Lipschitz property.

To finish, we bound the expected distances in the RHS.

Let $\ell \in \mathbb{R}$. Suppose there are at least $\lfloor \ell n_1 \rfloor$ positive examples to the right of t_{i_*}, denoted $t_{i_1}, \ldots, t_{i_{\lfloor \ell n_1 \rfloor}}$, and at least $\lfloor \ell n_0 \rfloor$ negative examples to the right of t_{i_*}, denoted $t_{j_1}, \ldots, t_{j_{\lfloor \ell n_0 \rfloor}}$. If $t_{i_{\lfloor \ell n_1 \rfloor}} \leq t_{j_{\lfloor \ell n_0 \rfloor}}$, then for any $k \leq \lfloor \ell n_1 \rfloor$ with $i_k \in I_1$, we have $d(i_*, \tilde{i}) \leq \ell$. To see this, note that

$$d(i_*,\tilde{i}) \leq d(i_*,i_k) = \max\left\{ \frac{\mathrm{pos}(i_*,i_k)}{n_1}, \frac{\mathrm{neg}(i_*,i_k)}{n_0} \right\} \leq \max\left\{ \frac{\ell n_1}{n_1}, \frac{\ell n_0}{n_0} \right\}.$$

Then,

$$\begin{aligned}
\Pr\left[d(i_*,\tilde{i}) > \ell\right] &\leq \Pr\left[I_1 \cap \{t_{i_1}, \ldots, t_{i_{\lfloor \ell n_1 \rfloor}}\} = \emptyset\right] \\
&\leq \left(1 - \frac{\lfloor \ell n_1 \rfloor}{n_1}\right)^k \leq \exp\left(-k \frac{\lfloor \ell n_1 \rfloor}{n_1}\right) \\
&\leq \exp\left(-k\left(\frac{\ell n_1}{n_1} - \frac{1}{n_1}\right)\right) = \exp\left(-k\ell\right) \exp\left(k/n_1\right).
\end{aligned}$$

Note that the same reasoning works even if there are less than $\lfloor \ell n_1 \rfloor$ positive examples.

Similarly, if $t_{i_{\lfloor \ell n_1 \rfloor}} > t_{j_{\lfloor \ell n_0 \rfloor}}$ and some $i_k \in I_0$ for $k \leq \lfloor \ell n_0 \rfloor$, $\Pr\left[d(i_*, \tilde{i}) > \ell\right] \leq \exp(-k\ell)\exp(k/n_0)$. Then

$$\mathbb{E}[d(i_*, \tilde{i})] \leq \int_0^\infty \Pr\left[d(i_*, \tilde{i}) > \ell\right] d\ell \leq \max\left\{\mathrm{e}^{k/n_1}, \mathrm{e}^{k/n_0}\right\} \int_0^\infty \mathrm{e}^{-k\ell} d\ell = \frac{\max\left\{\mathrm{e}^{k/n_1}, \mathrm{e}^{k/n_0}\right\}}{k}$$

And if $k \leq \min\{n_1, n_0\}$, we bound

$$\mathbb{E}[d(i_*, \tilde{i})] \leq \frac{\mathrm{e}}{k},$$

and we are done.

Now, let $\widehat{L} = \widehat{\mathrm{AUC}}$. We'll apply the same strategy as above. As with the KS loss, the optimal stump coefficients can be searched on a finite set. In this case, we have $\{t_{ij} : t_{ij} = S(X_i) - S(X_j) \text{ with } i, j \in [n]\}$. For ease of calculation, consider some stump $h(X) = t_{pq}\mathbf{1}_{[X_{(m)} \leq \xi]}$. Then,

$$\begin{aligned}
\widehat{\mathrm{AUC}}(S + h) &= 1 - \frac{1}{n_0 n_1} \sum_{\{i:y_i=1\}} \sum_{\{j:y_j=0\}} \mathbf{1}_{[S(X_i)+h(X_i) > S(X_j)+h(X_j)]} \\
&= 1 - \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} \mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]} \\
&= 1 - \sum_{i=1}^n \sum_{j=1}^n \rho_{ij} \mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]}
\end{aligned}$$

where $\rho_{ij} = \frac{1}{n_0 n_1} \mathbf{1}_{[y_i=1]} \mathbf{1}_{[y_j=0]}$. Note that

$$h(X_i) - h(X_j) = t_{pq}(\mathbf{1}_{[X_{i,(m)} \leq \xi]} - \mathbf{1}_{[X_{j,(m)} \leq \xi]}).$$

If t_{pq} is changed to some $t_{p'q'} \leq t_{pq}$ so that $h'(X) = t_{p'q'}\mathbf{1}_{[X_{(m)} \leq \xi]}$, we have

$$\mathbf{1}_{[t_{ij}+h(X_i)-h(X_j)>0]} - \mathbf{1}_{[t_{ij}+h'(X_i)-h'(X_j)>0]} = \begin{cases} \mathbf{1}_{[t_{ij}+t_{pq}>0]} - \mathbf{1}_{[t_{ij}+t_{p'q'}>0]}, \text{ if } X_{i,(m)} \leq \xi < X_{j,(m)} \\ \mathbf{1}_{[t_{ij}-t_{pq}>0]} - \mathbf{1}_{[t_{ij}-t_{p'q'})>0]}, \text{ if } X_{i,(m)} > \xi \geq X_{j,(m)} \\ 0, \text{ if } X_{i,(m)} \leq \xi, \text{ and } X_{j,(m)} \leq \xi \\ 0, \text{ if } X_{i,(m)} > \xi, \text{ and } X_{j,(m)} > \xi \end{cases}$$

Therefore,

$$\widehat{\mathrm{AUC}}(S + h') - \widehat{\mathrm{AUC}}(S + h) \in \left[-\frac{\#J_-((p,q),(p',q'))}{n_0 n_1}, \frac{\#J_+((p,q),(p',q'))}{n_0 n_1}\right]$$

where

$$\begin{aligned}
J_-((p,q),(p',q')) &= \{(i,j) : y_i = 1,\ y_j = 0,\ -t_{p'q'} > t_{ij} > -t_{pq}\} \\
J_+((p,q),(p',q')) &= \{(i,j) : y_i = 1,\ y_j = 0,\ t_{p'q'} < t_{ij} < t_{pq}\}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\left\|\widehat{\mathrm{AUC}}(S + h) - \widehat{\mathrm{AUC}}(S + h')\right\| &\leq \max\left\{\frac{\#J_-((p,q),(p',q'))}{n_0 n_1}, \frac{\#J_+((p,q),(p',q'))}{n_0 n_1}\right\} \\
&\leq \frac{\#J((p,q),(p',q'))}{n_0 n_1},
\end{aligned}$$

where $J((p,q),(p',q')) = J_-((p,q),(p',q')) \cup J_+((p,q),(p',q'))$. The rest of the proof follows the same strategy used in the $\widehat{\text{KS}}$ loss, replacing the pseudometric d with $\tilde{d}$, where

$$\tilde{d}((p,q),(p',q')) = \frac{\#J((p,q),(p',q'))}{n_0 n_1}.$$

Now let the optimal stump be $h_*(x) = t_{p_*q_*}\mathbf{1}_{[x_{(m_*)} \leq \xi_*]}$ and, again, consider a modified function $\tilde{h}$ where $t_{p_*q_*}$ is replaced by a point $t_{\tilde{p}\tilde{q}}$ with $\tilde{p}, \tilde{q} \in I$ chosen to minimize $\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q}))$. Recall that the optimal stump over the reduced sample, h_R, satisfies

$$\widehat{\text{AUC}}(S + h_R) \leq \widehat{\text{AUC}}(S + \tilde{h})$$

and therefore,

$$\begin{aligned}
\mathbb{E}\Big[\widehat{\text{AUC}}(S+h_R)\Big] &\leq \mathbb{E}\Big[\widehat{\text{AUC}}(S+\tilde{h})\Big] \\
&\leq \widehat{\text{AUC}}(S+h_*) - \mathbb{E}\Big[\widehat{\text{AUC}}(S+h_*) - \widehat{\text{AUC}}(S+\tilde{h})\Big] \\
&\leq \widehat{\text{AUC}}(S+h_*) + \mathbb{E}\Big[\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q}))\Big]
\end{aligned}$$

And finally, we bound the expected distance on the RHS. Let $\ell \in \mathbb{R}$. Suppose there are at least $r = \lfloor \ell n_1 n_0 \rfloor$ pairs $(i,j) \in J((p,q),(p',q'))$ such that $t_{ij} \leq t_{p_*q_*}$, denoted $t_{i_1j_1}, \ldots, t_{i_rj_r}$. Then, $\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) \leq \ell$. To verify this, note that for any pair (p,q) with $t_{pq} \leq t_{i_rj_r}$ such that $p \in I_1, q \in I_0$, we have

$$\begin{aligned}
\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) &\leq \tilde{d}((p_*,q_*),(p,q)) \\
&= \frac{\#J((p_*,q_*),(p,q))}{n_0 n_1} \\
&\leq \frac{\ell n_1 n_0}{n_1 n_0} = \ell
\end{aligned}$$

Moreover, note that $r \leq r_1 r_0$ where r_1 is the number of distinct indices i_s, $s \leq r$, such that $y_{i_s} = 1$ and r_0 is the number of distinct indices j_s, $s \leq r$, such that $y_{j_s} = 0$. Then,

$$\begin{aligned}
\Pr\Big[\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) > \ell\Big] &\leq \Pr\left[I_1 \times I_0 \cap \{(i_1,j_1),\ldots,(i_r,j_r)\} = \emptyset\right] \\
&\leq \left(1 - \frac{r_1}{n_1}\right)^k \left(1 - \frac{r_0}{n_0}\right)^k \\
&\leq \left(1 - \frac{r_1 r_0}{n_1 n_0}\right)^k \\
&\leq \left(1 - \frac{\lfloor \ell n_0 n_1 \rfloor}{n_0 n_1}\right)^k \\
&\leq \exp\left(-k \frac{\lfloor \ell n_0 n_1 \rfloor}{n_0 n_1}\right) \\
&\leq \exp\left(-k\left(\frac{\ell n_0 n_1}{n_0 n_1} - \frac{1}{n_0 n_1}\right)\right) \\
&= \exp(-k\ell)\exp(k/(n_0 n_1))
\end{aligned}$$

where the third inequality follows from the fact that $r_0 \leq n_0$ and $r_1 \leq n_1$. Then,

$$\begin{aligned}
\mathbb{E}\left[\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q}))\right] &\leq \int_0^\infty \Pr\left[\tilde{d}((p_*,q_*),(\tilde{p},\tilde{q})) > \ell\right] d\ell \\
&\leq \mathrm{e}^{k/(n_0 n_1)} \int_0^\infty \mathrm{e}^{-k\ell} d\ell = \frac{\mathrm{e}^{k/(n_0 n_1)}}{k} \leq \frac{\mathrm{e}}{k}.
\end{aligned}$$

□

Proposition 2. *Let $\widehat{L}$ be either the $\widehat{\mathrm{KS}}$ or the $\widehat{\mathrm{P@k}}$ loss. Consider the score $S_* : \mathbb{R}^M \to \mathbb{R}$ obtained by ExactBoost over the dataset $(Z_i, y_i)_{i=1}^n$ with initial score $S_0 \equiv 0$. Then:*

$$\widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \min_{1\leq m\leq M} \widehat{L}_{(X_i,y_i)_{i=1}^n}(S_m),$$

where $\widehat{L}_{(Z_i,y_i)_{i=1}^n}(\cdot)$ and $\widehat{L}_{(X_i,y_i)_{i=1}^n}(\cdot)$ denote the loss over the ensemble and the original data.

Proof of Proposition 2. For any loss $\widehat{L}$, ExactBoost obtains a sequence of score functions with decreasing values of $\widehat{L}$. Therefore, the loss of S_* is upper bounded by that of $S_{*,1}$, the stump function obtained in the first round of ExactBoost.

Now take any $1 \leq m \leq M$ and $t \in \mathbb{R}$ and consider the stump function $h_{m,t} : \mathbb{R}^M \to \mathbb{R}$, defined via $h_{m,t}(z) = \mathbf{1}_{[z^{(m)}\geq t]}$, where $z^{(m)}$ denotes the mth entry of z. Since $S_{*,1}$ has the smallest loss over training data of all stumps, for all $t \in \mathbb{R}$ and $1 \leq m \leq M$, it holds that:

$$\widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_{*,1}) \leq \widehat{L}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}). \tag{24}$$

The remainder of the proof consists of applying (24) judiciously. First, consider the $\widehat{\mathrm{KS}}$ loss. To estimate the $\widehat{\mathrm{KS}}$ loss for $h_{m,t}$, let n_0, n_1 denote the numbers of 0- and 1-labelled examples in (X_i, y_i). Then

$$\widehat{\mathrm{KS}}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}) = \inf_{s\in\mathbb{R}} \left(\frac{1}{n_1} \sum_{i:y_i=1} \mathbf{1}_{[h_{m,t}(Z_i)\leq s]} + \frac{1}{n_0} \sum_{i:y_i=0} \mathbf{1}_{[h_{m,t}(Z_i)>s]} \right).$$

In particular, taking the specific value $s = 0$ instead of the infimum in the right-hand side gives an upper bound for the $\widehat{\mathrm{KS}}$ losses of S_*, $S_{*,1}$ and $h_{m,t}$. Since $\mathbf{1}_{[h_{m,t}(Z_i)\leq 0]} = \mathbf{1}_{[S_m(X_i)\leq t]}$ and $\mathbf{1}_{[h_{m,t}(Z_i)>0]} = \mathbf{1}_{[S_m(X_i)>t]}$, from (24) it follows that, for all $t \in \mathbb{R}$ and $1 \leq m \leq M$,

$$\widehat{\mathrm{KS}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \frac{1}{n_1} \sum_{i:y_i=1} \mathbf{1}_{[S_m(X_i)\leq t]} + \frac{1}{n_0} \sum_{i:y_i=0} \mathbf{1}_{[S_m(X_i)>t]}.$$

Minimizing the right-hand side over t for a given m shows that

$$\widehat{\mathrm{KS}}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}) \leq \widehat{\mathrm{KS}}_{(X_i,y_i)_{i=1}^n}(S_m),$$

and taking the minimum over m finishes the proof in the case $\widehat{L} = \widehat{\mathrm{KS}}$.

Now consider the metric $\widehat{\mathrm{P@k}}$. For each $1 \leq m \leq M$, let $\widehat{t}_\alpha(S_m)$ denote the $(1-\alpha)$-quantile of the score S_m on the dataset $(X_i, y_i)_{i=1}^n$. Apply (24) to each m and to values $t < \widehat{t}_\alpha(S_m)$. To compute $\widehat{\mathrm{P@k}}_{(Z_i,y_i)}(h_{m,t})$, note that, for $0 \leq s < 1$, $h_{m,t}(Z_i) \leq s$ if and only if $Z_i^{(m)} = S_m(X_i) < t$. Since t is smaller than the $(1-\alpha)$-quantile, for all $0 \leq s < 1$:

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{[h_{m,t}(Z_i)\leq s]} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{[S_m(X_i)\leq t]} < 1 - \alpha.$$

Since $h_{m,t}$ takes binary values, the $(1-\alpha)$-quantile of the vector $(h_{m,t}(Z_i))_{i=1}^n$ is 1, and from (24) it follows that for any $1 \leq m \leq M$ and $t < \widehat{t}_\alpha(S_m)$,

$$\widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(h_{m,t}) = 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[h_{m,t}(Z_i)\geq 1]} = 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S_m(X_i)\geq t]}.$$

When $t \nearrow \widehat{t}_\alpha(S_m)$, it holds that $\mathbf{1}_{[S_m(X_i)\geq t]} \to \mathbf{1}_{[S_m(X_i)\geq \widehat{t}_\alpha(S_m)]}$, so, for all $1 \leq m \leq M$,

$$\widehat{\mathrm{P@k}}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq 1 - \frac{1}{n_1} \sum_{i\,:\,y_i=1} \mathbf{1}_{[S_m(X_i)\geq \widehat{t}_\alpha(S_m)]} = \widehat{\mathrm{P@k}}_{(X_i,y_i)_{i=1}^n}(S_m).$$

Minimizing the right-hand side over m finishes the proof. □

B Pseudocodes for evaluating metrics

Algorithm 3 AUC calculation. Complexity: $O(n \log n)$

function AUC(labels $\mathbf{y}$, scores S)
  $n^{(0)} \leftarrow \text{count}(\{S_i : i = 1, \ldots, n; y_i = 0\})$
  $n^{(1)} \leftarrow \text{count}(\{S_i : i = 1, \ldots, n; y_i = 1\})$
  $S^{(0)} \leftarrow \text{sort}(\{(S_i, y_i) : i = 1, \ldots, n; y_i = 0\})$
  $S^{(1)} \leftarrow \text{sort}(\{(S_i, y_i) : i = 1, \ldots, n; y_i = 1\})$

  $v \leftarrow 0, m \leftarrow 0$
  $i \leftarrow 0, j \leftarrow 0$
  while $i \leq n^{(1)} \wedge j \leq n^0$ **do**
    if $S_i^{(1)} < S_j^{(0)}$ **then**
      $v \leftarrow v + m$
      $i \leftarrow i + 1$
    else
      $m \leftarrow m + 1$
      $j \leftarrow j + 1$
    end if
  end while
  $v \leftarrow v + n^{(0)}(n^{(1)} - i)$
  return v
end function

Algorithm 4 KS threshold calculation. Complexity: $O(n \log n)$

function KSTHRESHOLD(labels $\mathbf{y}$, scores S)
  $n^{(0)} \leftarrow \text{count}(\{S_i : i = 1, \dots, n; y_i = 0\})$
  $n^{(1)} \leftarrow \text{count}(\{S_i : i = 1, \dots, n; y_i = 1\})$
  $S^{(0)} \leftarrow \text{sort}(\{(S_i, y_i) : i = 1, \dots, n; y_i = 0\})$
  $S^{(1)} \leftarrow \text{sort}(\{(S_i, y_i) : i = 1, \dots, n; y_i = 1\})$

  $v \leftarrow 0, v_\star \leftarrow 0, t_\star \leftarrow 0$
  $i \leftarrow 0, j \leftarrow 0$
  while $i \leq n^{(1)} \wedge j \leq n^{(0)}$ **do**
    if $S_i^{(1)} < S_j^{(0)}$ **then**
      $t \leftarrow S_i^{(1)}$
      $i \leftarrow i + 1$
      $v \leftarrow v - 1/n^{(1)}$
    else
      $t \leftarrow S_j^{(0)}$
      $j \leftarrow j + 1$
      $v \leftarrow v + 1/n^{(0)}$
    end if
    if $v > v_\star$ **then**
      $v_\star \leftarrow v$
      $t_\star \leftarrow t$
    end if
  end while
  return $t_\star$
end function

Algorithm 5 KS calculation given threshold. Complexity: $O(n)$

function KS(labels $\mathbf{y}$, scores S, threshold t)
  $n^{(0)} \leftarrow \text{count}(\{S_i : i = 1, \dots, n; y_i = 0\})$
  $n^{(1)} \leftarrow \text{count}(\{S_i : i = 1, \dots, n; y_i = 1\})$

  $v^{(0)} \leftarrow 0, v^{(1)} \leftarrow 0$
  for $i = 0, \dots, n$ **do**
    if $y_i = 0$ **then**
      $v^{(0)} \leftarrow v^{(0)} + \mathbf{1}_{[S_i \leq t]}$
    else
      $v^{(1)} \leftarrow v^{(1)} + \mathbf{1}_{[S_i \leq t]}$
    end if
  end for

  return $v^{(0)}/n^{(0)} - v^{(1)}/n^{(1)}$
end function

Algorithm 6 P@k calculation. Complexity: $O(n \log n)$

```
function P@k(labels y, scores S, k)
    n^(0) ← count({S_i : i = 1,...,n; y_i = 0})
    n^(1) ← count({S_i : i = 1,...,n; y_i = 1})
    S^(0) ← sort({(S_i, y_i) : i = 1,...,n; y_i = 0})
    S^(1) ← sort({(S_i, y_i) : i = 1,...,n; y_i = 1})

    v ← 0
    i ← n^(1), j ← n^(0)
    while 1 ≤ i ∧ 0 ≤ j ∧ k > 0 do
        if S_i^(1) > S_j^(0) then
            v ← v + 1
            i ← i − 1
        else
            j ← j − 1
        end if
        k ← k − 1
    end while
    v ← v + min{k, i}
    return v
end function
```

C Further details on Algorithm 2

Let $S'(A, B, \Xi) = \mathbf{S} + A\mathbf{1}_{[\mathbf{X}_{(j)} \leq \Xi]} + B\mathbf{1}_{[\mathbf{X}_{(j)} > \Xi]} - (1 + |B - A|/2)\theta\mathbf{y}$, and let $\odot$ be elementwise multiplication.

To solve (4), assign intervals A, B, Ξ as the search domain [lines 1-2]. Then, c times [line 3], halve it as follows: for each possible way to halve A, B, Ξ [line 5] (by the bissections $(\Xi^{(b)}, A^{(b)}, B^{(b)}) \in b(A) \times b(B) \times b(\Xi)$ where $b([a, b]) = \{[a, (a + b)/2], [(a + b)/2, b]\}$), compute the IA lower bound in the subinterval, $\underline{\widehat{L}(S'(A^{(b)}, B^{(b)}, \Xi^{(b)}), \mathbf{y})}$ [lines 6-9], using $\underline{\widehat{L}(S'(A^{(b)}, B^{(b)}, \Xi^{(b)}), \mathbf{y})} = \widehat{L}(\mathbf{y} \odot \overline{S'(A^{(b)}, B^{(b)}, \Xi^{(b)})} + (\mathbf{1} - \mathbf{y}) \odot \underline{S'(A^{(b)}, B^{(b)}, \Xi^{(b)})}, \mathbf{y})$, i.e., evaluating the IA lower bound for the losses is equivalent to evaluating the loss on the upper bound produced for the score when $y = 1$ and on the lower bound for the score when $y = 0$ (this follows from standard definitions of IA operations on our losses). Then, pick as the new search domain the subdomain with lowest IA lower bound [lines 4,10]. Since each step halves the search domain and gives an extra bit of numerical precision, c is fixed as the precision of the floating-point type. After c iterations, we have small intervals A, B, Ξ containing the (greedy) minimum. To produce values $\xi_\star, a_\star, b_\star$, select the corner of the cube $A \times B \times \Xi$ with smallest loss [lines 13-15].

Note the algorithm's complexity is $O((c+1)f(n)2^k)$, where c is fixed as bits of float precision, $f(n) = O(n \log n)$ is the cost of evaluating $\widehat{L}$, and $k = 3$ parameters, totalling $O(n \log n)$; as it relies on fast operations, the algorithm is very quick (see Table 2).

D Datasets characteristics and sources

Dataset	Observations	Features	Positives
a1a	1605	119	24.6%
australian	690	14	44.5%
banknote	1372	4	44.5%
breast-cancer	683	10	35.0%
cod-rna	59535	8	33.3%
colon-cancer	62	2000	35.5%
covtype	581012	54	48.8%
cskaggle	307511	97	8.1%
diabetes	768	8	34.9%
fourclass	862	2	35.6%
german	1000	20	70.0%
gisette	6000	5000	50.0%
gmsc	150000	10	6.7%
heart	303	21	45.9%
housing	506	13	6.9%
ijcnn1	49990	22	9.7%
ionosphere	351	34	64.1%
liver-disorders	145	5	37.9%
madelon	2000	500	50.0%
mammography	11183	6	2.3%
mq2008	15211	46	19.3%
oil-spill	937	49	4.4%
phishing	11055	68	55.7%
phoneme	5404	5	29.3%
skin-nonskin	245057	3	79.2%
sonar	208	60	46.6%
splice	1000	60	48.3%
svmguide1	3089	4	35.3%
svmguide3	1243	22	23.8%
taiwan	30000	24	22.1%
w1a	2477	300	2.9%

Table 5: Datasets characteristics

Many datasets above were retrieved from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary`. Exceptions are:

- `australian`:
  `http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/australian/`
- `banknote`:
  `https://archive.ics.uci.edu/ml/machine-learning-databases/00267`
- `cskaggle`:
  `https://www.kaggle.com/c/home-credit-default-risk/data`
- `diabetes`:
  `https://github.com/jbrownlee/Datasets/`
- `german`:
  `https://online.stat.psu.edu/stat508/resource/analysis/gcd`
- `gmsc`:
  `https://www.kaggle.com/c/GiveMeSomeCredit/data`
- `heart`:
  `https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease`

- housing:
  https://archive.ics.uci.edu/ml/machine-learning-databases/housing
- ionosphere:
  https://github.com/jbrownlee/Datasets/
- mammography:
  https://github.com/jbrownlee/Datasets/
- mq2008:
  https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/
- oil-spill:
  https://github.com/jbrownlee/Datasets/
- phoneme:
  https://github.com/jbrownlee/Datasets/
- sonar:
  https://github.com/jbrownlee/Datasets/
- taiwan:
  https://archive.ics.uci.edu/ml/machine-learning-databases/00350

Pre-processing mainly involved converting categorical variables to binary dummies and scaling labels to $\{0, 1\}$. Scripts are included in `src/data/` to both download and process all the datasets above.

Some datasets are made available as a single file containing all observations, while some datasets are split into more than one file. To ease processing and to simplify the data acquisition pipeline, only a single sample from datasets split across several files was considered.

E Running time measurements

Dataset	E.B. (AUC)	RankBoost	E.B. (KS)	DMKS	E.B. (P@k)	TopPush
a1a	2.28s	127.52s	2.45s	251.95s	1.67s	1.37s
german	0.53s	12.80s	0.67s	0.86s	0.53s	1.54s
gisette	336.83s	OOT	355.37s	19786.38s	281.20s	6.39s
gmsc	40.55s	OOT	40.84s	934.80s	21.95s	1.73s
heart	0.32s	1.05s	0.29s	5.46s	0.28s	1.46s
ionosphere	0.71s	2.83s	0.65s	2.26s	0.52s	1.39s
liver-disorders	0.12s	0.23s	0.12s	0.79s	0.11s	1.35s
oil-spill	0.90s	5.32s	0.84s	6.68s	0.70s	1.54s
splice	1.17s	58.46s	1.31s	1.57s	1.14s	1.37s
svmguide1	0.42s	91.94s	0.47s	0.89s	0.33s	1.40s

F Hyperparameters

All surrogate-based models were trained with fixed hyperparameters, set to be the default values provided by their corresponding packages — Scikit-learn (version 0.22.1) for AdaBoost, kNN, Logistic and Random Forest; XGBoost (version 1.0.2) for gradient boosting and TensorFlow (version 2.2.0) for the neural network.

AdaBoost The base estimator was set to be a decision tree with depth of 1; the number of estimators was set to 50; learning rate set to 1 and the default algorithm is SAMME.R.

kNN The model was trained with 5 neighbors and uniform weights; the distance metric used for the tree was the Minkowski metric with power 2; the package decides which algorithm to use automatically, with leaf size 30 passed to BallTree or KDTree.

Logistic Regression The penalty norm used was L2; the model was solved in primal formulation; tolerance is set to 0.0001; the (inverse) of regularization strength is set to 1; the model fits an intercept constant and uses no class weights; the solver was set to `lbfgs` with 100 maximum iterations.

Neural Network The neural network had four fully connected layers, the first three with relu activation functions and the last with a sigmoid activation function. The number of output units were 26, 12, 12 and 1, respectively. The model was trained with the Adam optimizer for the binary cross-entropy loss; the number of epochs was set to 30 and batch size to 4.

Random Forest The model was trained with 100 estimator using the Gini criterion; in the default settings, nodes are expanded until leaves are pure or until all leaves have less than 2 samples; the minimum number of samples in each leaf is 1; the trees have an unlimited number of nodes; the maximum number of features is set to $\sqrt{p}$, where p is the number of features; the trees are built using bootstrapped samples but out-of-bag samples are not used to estimate the score; both classes have the same weight.

XGBoost The model used a tree booster with learning rate set to 0.3; trees had maximum depth of 6; the minimum loss required to make a partition in a leaf was set to 0; no subsampling was used in each boosting iteration; the L2 regularization term on weights was set to 1 and the L1 regularization term set to 0; the tree construction algorithm was automatically chosen by the package.

As for the exact benchmarks, the implementations of RankBoost and TopPush used were publicly available.

RankBoost The number of rounds was set to 100.

TopPush The regularization parameter λ was set to 0.001.

DMKS The model uses the normalized coefficients obtained in the logistic regression as a starting point to optimize the KS loss. Weights were also all set to 1.

G Evaluation results: AUC, KS and P@k

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	RankBoost
a1a	**0.11 ± 0.03**	0.13 ± 0.04	0.18 ± 0.03	**0.11 ± 0.03**	0.15 ± 0.03	0.13 ± 0.03	0.13 ± 0.03	0.13 ± 0.04
australian	**0.06 ± 0.02**	0.09 ± 0.01	0.27 ± 0.01	0.12 ± 0.03	0.18 ± 0.03	0.07 ± 0.01	0.07 ± 0.02	0.10 ± 0.01
banknote	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
breast-cancer	**0.01 ± 0.01**	**0.01 ± 0.01**	0.43 ± 0.04	0.43 ± 0.03	0.38 ± 0.11	**0.01 ± 0.00**	**0.01 ± 0.01**	**0.01 ± 0.01**
cod-rna	0.14 ± 0.01	0.02 ± 0.00	0.06 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	**0.01 ± 0.00**	**0.01 ± 0.00**	OOT[1]
colon-cancer	0.24 ± 0.09	0.14 ± 0.10	0.13 ± 0.07	0.17 ± 0.16	0.14 ± 0.12	**0.11 ± 0.10**	0.15 ± 0.16	0.13 ± 0.14
covtype	0.21 ± 0.00	0.16 ± 0.00	**0.00 ± 0.00**	0.34 ± 0.00	0.29 ± 0.17	0.01 ± 0.00	0.05 ± 0.00	OOM[2]
cskaggle	0.30 ± 0.00	**0.26 ± 0.00**	0.45 ± 0.00	0.37 ± 0.00	0.50 ± 0.00	0.29 ± 0.00	**0.26 ± 0.00**	OOM
diabetes	0.18 ± 0.01	0.20 ± 0.03	0.25 ± 0.04	**0.17 ± 0.02**	0.28 ± 0.03	0.18 ± 0.02	0.21 ± 0.02	0.20 ± 0.03
fourclass	0.11 ± 0.04	0.04 ± 0.01	**0.00 ± 0.00**	0.17 ± 0.03	0.17 ± 0.03	**0.00 ± 0.00**	**0.00 ± 0.00**	0.04 ± 0.01
german	0.23 ± 0.02	0.24 ± 0.01	0.42 ± 0.03	0.23 ± 0.02	0.39 ± 0.11	**0.22 ± 0.03**	0.23 ± 0.03	0.24 ± 0.01
gisette	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	**0.00 ± 0.00**	0.01 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	OOT
gmsc	0.21 ± 0.01	**0.14 ± 0.00**	0.43 ± 0.00	0.32 ± 0.01	0.42 ± 0.05	0.16 ± 0.00	**0.14 ± 0.00**	OOT
heart	**0.09 ± 0.03**	0.16 ± 0.07	0.30 ± 0.03	**0.09 ± 0.02**	0.10 ± 0.02	0.11 ± 0.02	0.12 ± 0.03	0.13 ± 0.03
housing	0.14 ± 0.04	0.25 ± 0.08	0.23 ± 0.05	0.19 ± 0.03	0.31 ± 0.05	**0.10 ± 0.03**	0.11 ± 0.04	0.24 ± 0.07
ijcnn1	0.10 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.07 ± 0.00	**0.00 ± 0.00**	0.01 ± 0.00	**0.00 ± 0.00**	OOT
ionosphere	0.04 ± 0.02	0.05 ± 0.02	0.09 ± 0.05	0.09 ± 0.04	0.04 ± 0.02	**0.02 ± 0.02**	0.03 ± 0.01	0.04 ± 0.02
liver-disorders	**0.22 ± 0.06**	0.33 ± 0.07	0.29 ± 0.13	0.24 ± 0.08	0.34 ± 0.08	0.24 ± 0.13	0.27 ± 0.17	0.32 ± 0.08
madelon	0.35 ± 0.02	0.36 ± 0.02	0.23 ± 0.03	0.44 ± 0.03	0.50 ± 0.00	0.24 ± 0.02	**0.14 ± 0.01**	0.36 ± 0.02
mammography	0.07 ± 0.02	0.07 ± 0.02	0.11 ± 0.01	0.08 ± 0.02	**0.05 ± 0.01**	0.06 ± 0.02	0.06 ± 0.02	0.07 ± 0.02
mq2008	0.21 ± 0.01	0.18 ± 0.00	0.29 ± 0.01	0.22 ± 0.01	0.22 ± 0.01	**0.13 ± 0.01**	**0.13 ± 0.01**	0.17 ± 0.01
oil-spill	0.09 ± 0.09	0.14 ± 0.10	0.32 ± 0.11	0.29 ± 0.10	0.50 ± 0.09	0.09 ± 0.06	**0.07 ± 0.06**	0.09 ± 0.08
phishing	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	0.01 ± 0.00
phoneme	0.17 ± 0.02	0.11 ± 0.01	0.07 ± 0.01	0.19 ± 0.02	0.09 ± 0.01	**0.04 ± 0.01**	0.05 ± 0.01	0.11 ± 0.01
skin-nonskin	0.01 ± 0.00	0.02 ± 0.00	**0.00 ± 0.00**	0.05 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	OOM
sonar	0.08 ± 0.03	0.12 ± 0.03	0.11 ± 0.04	0.14 ± 0.04	0.09 ± 0.04	**0.05 ± 0.02**	**0.05 ± 0.03**	0.10 ± 0.02
splice	0.04 ± 0.01	0.03 ± 0.00	0.18 ± 0.02	0.12 ± 0.03	0.09 ± 0.02	**0.01 ± 0.00**	**0.01 ± 0.00**	0.02 ± 0.00
svmguide1	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**
svmguide3	0.19 ± 0.02	0.16 ± 0.02	0.24 ± 0.04	0.24 ± 0.03	0.18 ± 0.03	0.13 ± 0.02	**0.12 ± 0.02**	0.14 ± 0.02
taiwan	0.27 ± 0.00	**0.23 ± 0.01**	0.40 ± 0.00	0.36 ± 0.01	0.50 ± 0.00	**0.23 ± 0.00**	**0.23 ± 0.00**	**0.23 ± 0.01**
w1a	0.12 ± 0.05	0.15 ± 0.05	0.24 ± 0.04	0.08 ± 0.02	0.15 ± 0.04	**0.07 ± 0.03**	0.13 ± 0.07	0.12 ± 0.04

Table 6: Evaluation of estimators with AUC as metric

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	RankBoost
a1a	**0.13 ± 0.04**	0.17 ± 0.05	0.18 ± 0.06	0.14 ± 0.05	0.15 ± 0.05	0.27 ± 0.07	0.28 ± 0.06	0.16 ± 0.05
australian	**0.07 ± 0.02**	0.11 ± 0.03	0.33 ± 0.04	0.12 ± 0.03	0.21 ± 0.04	0.14 ± 0.03	0.15 ± 0.03	0.11 ± 0.03
banknote	**0.00 ± 0.00**	0.01 ± 0.01	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	0.01 ± 0.01	**0.00 ± 0.00**
breast-cancer	**0.01 ± 0.00**	0.02 ± 0.02	0.44 ± 0.04	0.26 ± 0.20	0.49 ± 0.06	0.03 ± 0.02	0.04 ± 0.01	0.02 ± 0.01
cod-rna	**0.01 ± 0.00**	**0.01 ± 0.00**	0.04 ± 0.00	**0.01 ± 0.00**	**0.01 ± 0.00**	0.03 ± 0.00	0.04 ± 0.00	**0.01 ± 0.00**
colon-cancer	0.48 ± 0.21	0.42 ± 0.07	0.29 ± 0.09	**0.26 ± 0.17**	0.31 ± 0.12	0.43 ± 0.04	0.34 ± 0.12	**0.26 ± 0.18**
covtype	**0.00 ± 0.00**	**0.00 ± 0.00**	0.02 ± 0.00	0.16 ± 0.08	**0.00 ± 0.00**	0.02 ± 0.00	0.02 ± 0.00	OOM
cskaggle	0.27 ± 0.00	**0.26 ± 0.00**	0.46 ± 0.00	0.37 ± 0.01	0.50 ± 0.00	0.50 ± 0.00	0.48 ± 0.00	OOM
diabetes	**0.19 ± 0.03**	0.25 ± 0.06	0.28 ± 0.04	0.21 ± 0.02	0.37 ± 0.05	0.30 ± 0.04	0.32 ± 0.05	0.24 ± 0.05
fourclass	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	0.03 ± 0.02	0.01 ± 0.01	**0.00 ± 0.00**	**0.00 ± 0.00**
german	**0.23 ± 0.03**	0.32 ± 0.05	0.45 ± 0.06	0.24 ± 0.04	0.50 ± 0.07	0.33 ± 0.02	0.35 ± 0.03	0.30 ± 0.06
gisette	**0.00 ± 0.00**	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
gmsc	0.15 ± 0.01	**0.14 ± 0.00**	0.45 ± 0.00	0.31 ± 0.01	0.46 ± 0.01	0.42 ± 0.02	0.41 ± 0.01	0.15 ± 0.00
heart	**0.12 ± 0.03**	0.18 ± 0.06	0.34 ± 0.06	**0.12 ± 0.03**	0.23 ± 0.07	0.19 ± 0.04	0.23 ± 0.06	0.15 ± 0.05
housing	**0.15 ± 0.04**	0.25 ± 0.15	0.42 ± 0.07	0.29 ± 0.09	0.37 ± 0.09	0.46 ± 0.06	0.44 ± 0.06	0.24 ± 0.10
ijcnn1	**0.00 ± 0.00**	**0.00 ± 0.00**	0.02 ± 0.00	**0.00 ± 0.00**	0.01 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	**0.00 ± 0.00**
ionosphere	**0.04 ± 0.03**	0.05 ± 0.03	0.12 ± 0.04	0.07 ± 0.04	0.07 ± 0.05	0.07 ± 0.03	0.09 ± 0.03	0.05 ± 0.04
liver-disorders	**0.30 ± 0.11**	0.34 ± 0.10	0.37 ± 0.11	0.34 ± 0.08	0.34 ± 0.08	0.38 ± 0.04	0.38 ± 0.02	0.38 ± 0.10
madelon	**0.16 ± 0.02**	0.24 ± 0.03	0.28 ± 0.01	0.46 ± 0.03	0.50 ± 0.01	0.29 ± 0.02	0.23 ± 0.01	0.22 ± 0.02
mammography	**0.05 ± 0.02**	0.07 ± 0.02	0.13 ± 0.02	0.06 ± 0.02	0.08 ± 0.03	0.19 ± 0.02	0.19 ± 0.03	0.07 ± 0.03
mq2008	**0.13 ± 0.01**	0.15 ± 0.01	0.34 ± 0.01	0.22 ± 0.01	0.28 ± 0.00	0.29 ± 0.01	0.29 ± 0.01	0.15 ± 0.01
oil-spill	**0.17 ± 0.12**	0.19 ± 0.11	0.41 ± 0.08	0.29 ± 0.19	0.46 ± 0.09	0.38 ± 0.09	0.35 ± 0.17	0.19 ± 0.13
phishing	**0.00 ± 0.00**	0.01 ± 0.00	0.01 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	0.03 ± 0.00	0.03 ± 0.01	0.01 ± 0.00
phoneme	**0.04 ± 0.01**	0.05 ± 0.01	0.08 ± 0.01	**0.04 ± 0.01**	**0.04 ± 0.01**	0.12 ± 0.02	0.13 ± 0.01	0.05 ± 0.01
skin-nonskin	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	OOM
sonar	0.09 ± 0.03	0.19 ± 0.09	0.09 ± 0.03	**0.06 ± 0.02**	0.08 ± 0.02	0.17 ± 0.03	0.18 ± 0.03	0.18 ± 0.11
splice	**0.01 ± 0.00**	**0.01 ± 0.01**	0.16 ± 0.03	0.08 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.04 ± 0.02	0.02 ± 0.01
svmguide1	**0.00 ± 0.00**	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.04 ± 0.00	0.01 ± 0.00
svmguide3	0.15 ± 0.02	0.21 ± 0.04	0.21 ± 0.04	**0.14 ± 0.03**	0.16 ± 0.03	0.27 ± 0.03	0.28 ± 0.04	0.20 ± 0.02
taiwan	**0.23 ± 0.01**	0.24 ± 0.01	0.41 ± 0.01	0.36 ± 0.01	0.50 ± 0.00	0.34 ± 0.01	0.35 ± 0.01	0.24 ± 0.00
w1a	**0.11 ± 0.03**	0.15 ± 0.03	0.27 ± 0.06	**0.11 ± 0.04**	0.20 ± 0.05	0.37 ± 0.06	0.27 ± 0.05	0.13 ± 0.06

Table 7: Evaluation of ensemblers with AUC as metric

[1]OOT (out-of-time): the budget time of 5 days was exceeded.
[2]OOM (out-of-memory): 810GB of RAM were exceeded.

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	DMKS
a1a	0.37 ± 0.05	0.37 ± 0.06	0.46 ± 0.05	$\mathbf{0.36 \pm 0.05}$	0.43 ± 0.04	0.39 ± 0.06	0.40 ± 0.06	0.37 ± 0.05
australian	0.24 ± 0.04	0.29 ± 0.03	0.62 ± 0.03	0.33 ± 0.07	0.42 ± 0.07	0.25 ± 0.04	$\mathbf{0.23 \pm 0.04}$	0.27 ± 0.06
banknote	0.06 ± 0.02	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01
breast-cancer	$\mathbf{0.03 \pm 0.01}$	0.06 ± 0.02	0.86 ± 0.04	0.83 ± 0.03	0.69 ± 0.23	$\mathbf{0.03 \pm 0.02}$	0.04 ± 0.03	0.40 ± 0.04
cod-rna	0.51 ± 0.01	0.12 ± 0.01	0.20 ± 0.01	0.12 ± 0.00	0.12 ± 0.00	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	0.12 ± 0.00
colon-cancer	0.39 ± 0.22	0.28 ± 0.20	0.24 ± 0.13	0.23 ± 0.22	0.24 ± 0.22	$\mathbf{0.22 \pm 0.21}$	$\mathbf{0.22 \pm 0.22}$	0.46 ± 0.26
covtype	0.52 ± 0.00	0.47 ± 0.00	$\mathbf{0.05 \pm 0.00}$	0.75 ± 0.01	0.66 ± 0.28	0.07 ± 0.00	0.25 ± 0.00	0.50 ± 0.01
cskaggle	0.68 ± 0.00	$\mathbf{0.64 \pm 0.01}$	0.90 ± 0.01	0.80 ± 0.01	1.00 ± 0.00	0.69 ± 0.01	$\mathbf{0.64 \pm 0.00}$	0.73 ± 0.01
diabetes	$\mathbf{0.46 \pm 0.03}$	0.50 ± 0.05	0.64 ± 0.05	$\mathbf{0.46 \pm 0.04}$	0.62 ± 0.07	0.47 ± 0.03	0.54 ± 0.04	$\mathbf{0.46 \pm 0.05}$
fourclass	0.30 ± 0.07	0.11 ± 0.02	$\mathbf{0.00 \pm 0.00}$	0.50 ± 0.04	0.47 ± 0.07	$\mathbf{0.00 \pm 0.01}$	0.01 ± 0.01	0.49 ± 0.04
german	$\mathbf{0.53 \pm 0.05}$	0.54 ± 0.02	0.86 ± 0.03	0.54 ± 0.03	0.76 ± 0.11	$\mathbf{0.53 \pm 0.06}$	0.55 ± 0.05	0.55 ± 0.05
gisette	0.09 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	$\mathbf{0.04 \pm 0.00}$	0.06 ± 0.01
gmsc	$\mathbf{0.44 \pm 0.00}$	$\mathbf{0.44 \pm 0.01}$	0.87 ± 0.00	0.74 ± 0.02	0.87 ± 0.09	0.46 ± 0.00	$\mathbf{0.44 \pm 0.01}$	0.45 ± 0.01
heart	0.30 ± 0.05	0.35 ± 0.09	0.66 ± 0.04	$\mathbf{0.28 \pm 0.04}$	0.30 ± 0.06	0.32 ± 0.07	0.33 ± 0.09	$\mathbf{0.28 \pm 0.04}$
housing	0.25 ± 0.03	0.48 ± 0.15	0.50 ± 0.12	0.37 ± 0.06	0.56 ± 0.04	$\mathbf{0.20 \pm 0.08}$	0.27 ± 0.10	0.36 ± 0.10
ijcnn1	0.30 ± 0.01	0.21 ± 0.01	0.10 ± 0.00	0.27 ± 0.02	$\mathbf{0.04 \pm 0.00}$	0.07 ± 0.01	0.05 ± 0.00	0.25 ± 0.01
ionosphere	0.13 ± 0.04	0.14 ± 0.08	0.21 ± 0.08	0.26 ± 0.06	0.14 ± 0.06	$\mathbf{0.12 \pm 0.05}$	$\mathbf{0.12 \pm 0.04}$	0.28 ± 0.04
liver-disorders	$\mathbf{0.45 \pm 0.09}$	0.56 ± 0.10	0.61 ± 0.21	0.50 ± 0.10	0.58 ± 0.12	0.47 ± 0.18	0.46 ± 0.25	0.50 ± 0.10
madelon	0.71 ± 0.03	0.77 ± 0.04	0.59 ± 0.06	0.87 ± 0.05	1.00 ± 0.00	0.60 ± 0.05	$\mathbf{0.41 \pm 0.03}$	0.87 ± 0.05
mammography	0.20 ± 0.04	0.21 ± 0.04	0.23 ± 0.02	0.20 ± 0.04	$\mathbf{0.15 \pm 0.02}$	0.18 ± 0.01	0.18 ± 0.02	0.19 ± 0.04
mq2008	0.57 ± 0.01	0.50 ± 0.01	0.68 ± 0.02	0.58 ± 0.02	0.56 ± 0.03	0.41 ± 0.01	$\mathbf{0.40 \pm 0.02}$	0.58 ± 0.01
oil-spill	0.25 ± 0.10	0.31 ± 0.17	0.64 ± 0.22	0.53 ± 0.14	0.88 ± 0.10	0.27 ± 0.14	$\mathbf{0.18 \pm 0.11}$	0.45 ± 0.14
phishing	0.14 ± 0.01	0.12 ± 0.01	0.09 ± 0.01	0.12 ± 0.01	0.07 ± 0.01	$\mathbf{0.05 \pm 0.01}$	0.06 ± 0.00	0.11 ± 0.01
phoneme	0.45 ± 0.03	0.35 ± 0.02	0.27 ± 0.02	0.48 ± 0.02	0.32 ± 0.03	$\mathbf{0.19 \pm 0.02}$	0.22 ± 0.02	0.45 ± 0.02
skin-nonskin	0.04 ± 0.00	0.06 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.10 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.09 ± 0.00
sonar	0.32 ± 0.04	0.29 ± 0.07	0.37 ± 0.09	0.32 ± 0.06	0.23 ± 0.10	0.22 ± 0.04	$\mathbf{0.15 \pm 0.07}$	0.33 ± 0.06
splice	0.16 ± 0.01	0.14 ± 0.01	0.47 ± 0.04	0.37 ± 0.06	0.31 ± 0.03	$\mathbf{0.06 \pm 0.02}$	0.07 ± 0.02	0.36 ± 0.05
svmguide1	$\mathbf{0.06 \pm 0.00}$	$\mathbf{0.06 \pm 0.01}$	0.07 ± 0.01	0.09 ± 0.01	0.07 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.01}$	0.09 ± 0.01
svmguide3	0.50 ± 0.03	0.43 ± 0.04	0.60 ± 0.06	0.53 ± 0.05	0.48 ± 0.05	0.41 ± 0.05	$\mathbf{0.36 \pm 0.05}$	0.53 ± 0.06
taiwan	0.59 ± 0.01	$\mathbf{0.58 \pm 0.01}$	0.86 ± 0.01	0.78 ± 0.01	1.00 ± 0.00	0.59 ± 0.01	0.59 ± 0.00	0.62 ± 0.01
w1a	0.30 ± 0.11	0.39 ± 0.09	0.48 ± 0.07	0.26 ± 0.07	0.41 ± 0.08	$\mathbf{0.24 \pm 0.08}$	0.36 ± 0.13	0.59 ± 0.09

Table 8: Evaluation of estimators with KS as metric

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	DMKS
a1a	$\mathbf{0.37 \pm 0.07}$	0.44 ± 0.09	0.48 ± 0.09	0.40 ± 0.09	0.41 ± 0.09	0.54 ± 0.13	0.57 ± 0.12	0.49 ± 0.08
australian	$\mathbf{0.22 \pm 0.03}$	0.28 ± 0.05	0.70 ± 0.07	0.34 ± 0.04	0.48 ± 0.06	0.29 ± 0.05	0.30 ± 0.07	0.30 ± 0.05
banknote	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$
breast-cancer	$\mathbf{0.03 \pm 0.03}$	0.06 ± 0.05	0.86 ± 0.06	0.52 ± 0.37	0.94 ± 0.12	0.07 ± 0.04	0.08 ± 0.02	0.43 ± 0.36
cod-rna	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	0.16 ± 0.01	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$	$\mathbf{0.07 \pm 0.00}$
colon-cancer	0.78 ± 0.23	0.83 ± 0.14	0.62 ± 0.16	$\mathbf{0.41 \pm 0.25}$	0.48 ± 0.15	0.85 ± 0.08	0.69 ± 0.24	0.65 ± 0.28
covtype	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$	0.15 ± 0.00	0.45 ± 0.16	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$	0.05 ± 0.00	0.05 ± 0.00
cskaggle	$\mathbf{0.64 \pm 0.00}$	$\mathbf{0.64 \pm 0.01}$	0.92 ± 0.01	0.80 ± 0.01	1.00 ± 0.00	0.99 ± 0.00	0.97 ± 0.01	0.70 ± 0.02
diabetes	$\mathbf{0.47 \pm 0.05}$	0.56 ± 0.08	0.65 ± 0.08	0.52 ± 0.03	0.72 ± 0.07	0.61 ± 0.08	0.63 ± 0.09	0.51 ± 0.02
fourclass	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.03 ± 0.01	$\mathbf{0.00 \pm 0.00}$	0.12 ± 0.09	0.01 ± 0.01	$\mathbf{0.00 \pm 0.01}$	$\mathbf{0.00 \pm 0.01}$
german	$\mathbf{0.50 \pm 0.06}$	0.68 ± 0.07	0.90 ± 0.07	0.53 ± 0.07	0.89 ± 0.08	0.66 ± 0.04	0.69 ± 0.06	0.53 ± 0.06
gisette	$\mathbf{0.04 \pm 0.01}$	$\mathbf{0.04 \pm 0.01}$	0.11 ± 0.01	0.07 ± 0.01	0.07 ± 0.00	0.06 ± 0.01	$\mathbf{0.04 \pm 0.01}$	0.10 ± 0.02
gmsc	$\mathbf{0.43 \pm 0.01}$	0.44 ± 0.01	0.90 ± 0.00	0.73 ± 0.02	0.95 ± 0.01	0.85 ± 0.04	0.83 ± 0.02	0.46 ± 0.00
heart	$\mathbf{0.34 \pm 0.06}$	0.38 ± 0.07	0.70 ± 0.10	0.37 ± 0.07	0.52 ± 0.13	0.38 ± 0.07	0.46 ± 0.12	0.40 ± 0.05
housing	$\mathbf{0.30 \pm 0.04}$	0.42 ± 0.22	0.82 ± 0.12	0.54 ± 0.13	0.59 ± 0.11	0.91 ± 0.11	0.88 ± 0.11	0.55 ± 0.13
ijcnn1	$\mathbf{0.04 \pm 0.00}$	0.05 ± 0.00	0.05 ± 0.00	$\mathbf{0.04 \pm 0.01}$	0.05 ± 0.01	0.06 ± 0.01	0.07 ± 0.01	$\mathbf{0.04 \pm 0.01}$
ionosphere	$\mathbf{0.13 \pm 0.07}$	0.18 ± 0.07	0.27 ± 0.10	0.18 ± 0.07	0.17 ± 0.07	0.15 ± 0.06	0.19 ± 0.06	0.27 ± 0.11
liver-disorders	$\mathbf{0.53 \pm 0.12}$	0.60 ± 0.17	0.72 ± 0.12	0.59 ± 0.17	0.61 ± 0.09	0.76 ± 0.07	0.76 ± 0.04	0.60 ± 0.17
madelon	$\mathbf{0.43 \pm 0.03}$	0.55 ± 0.05	0.66 ± 0.02	0.91 ± 0.03	0.98 ± 0.04	0.58 ± 0.04	0.46 ± 0.02	0.95 ± 0.05
mammography	$\mathbf{0.16 \pm 0.02}$	0.19 ± 0.02	0.27 ± 0.03	0.19 ± 0.04	0.20 ± 0.03	0.39 ± 0.05	0.38 ± 0.06	0.21 ± 0.04
mq2008	$\mathbf{0.40 \pm 0.01}$	0.45 ± 0.01	0.74 ± 0.02	0.59 ± 0.02	0.61 ± 0.01	0.58 ± 0.02	0.58 ± 0.02	0.47 ± 0.02
oil-spill	$\mathbf{0.33 \pm 0.18}$	$\mathbf{0.33 \pm 0.17}$	0.81 ± 0.11	0.47 ± 0.23	0.89 ± 0.14	0.76 ± 0.17	0.69 ± 0.33	0.63 ± 0.29
phishing	$\mathbf{0.05 \pm 0.01}$	0.06 ± 0.01	0.06 ± 0.01	$\mathbf{0.05 \pm 0.01}$	0.06 ± 0.01	0.06 ± 0.01	0.07 ± 0.01	$\mathbf{0.05 \pm 0.01}$
phoneme	$\mathbf{0.20 \pm 0.02}$	0.23 ± 0.03	0.26 ± 0.03	$\mathbf{0.20 \pm 0.02}$	$\mathbf{0.20 \pm 0.02}$	0.25 ± 0.03	0.27 ± 0.03	$\mathbf{0.20 \pm 0.02}$
skin-nonskin	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
sonar	0.27 ± 0.05	0.35 ± 0.11	0.29 ± 0.03	$\mathbf{0.20 \pm 0.03}$	$\mathbf{0.20 \pm 0.06}$	0.34 ± 0.05	0.37 ± 0.05	0.22 ± 0.02
splice	$\mathbf{0.06 \pm 0.02}$	0.09 ± 0.02	0.46 ± 0.06	0.28 ± 0.03	0.21 ± 0.04	0.09 ± 0.02	0.09 ± 0.03	0.28 ± 0.03
svmguide1	$\mathbf{0.06 \pm 0.00}$	0.08 ± 0.02	0.08 ± 0.01	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.06 \pm 0.00}$	0.07 ± 0.00	0.07 ± 0.01	$\mathbf{0.06 \pm 0.01}$
svmguide3	0.41 ± 0.02	0.52 ± 0.07	0.52 ± 0.09	$\mathbf{0.36 \pm 0.05}$	0.40 ± 0.03	0.55 ± 0.05	0.56 ± 0.07	0.38 ± 0.04
taiwan	$\mathbf{0.57 \pm 0.01}$	0.60 ± 0.02	0.86 ± 0.01	0.78 ± 0.01	1.00 ± 0.00	0.68 ± 0.01	0.71 ± 0.01	0.63 ± 0.01
w1a	$\mathbf{0.30 \pm 0.07}$	0.38 ± 0.04	0.54 ± 0.12	$\mathbf{0.30 \pm 0.09}$	0.45 ± 0.08	0.74 ± 0.12	0.54 ± 0.09	0.75 ± 0.10

Table 9: Evaluation of ensemblers with KS as metric

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	TopPush
a1a	0.26 ± 0.11	0.23 ± 0.09	0.30 ± 0.09	0.24 ± 0.10	0.28 ± 0.13	$\mathbf{0.22 \pm 0.09}$	0.24 ± 0.13	0.29 ± 0.10
australian	$\mathbf{0.04 \pm 0.03}$	0.06 ± 0.04	0.23 ± 0.07	0.08 ± 0.05	0.14 ± 0.07	$\mathbf{0.04 \pm 0.05}$	0.05 ± 0.03	0.19 ± 0.10
banknote	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.01}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
breast-cancer	$\mathbf{0.01 \pm 0.02}$	$\mathbf{0.01 \pm 0.02}$	0.58 ± 0.07	0.60 ± 0.11	0.55 ± 0.10	$\mathbf{0.01 \pm 0.02}$	$\mathbf{0.01 \pm 0.02}$	0.59 ± 0.10
cod-rna	0.28 ± 0.01	0.05 ± 0.00	0.06 ± 0.00	0.05 ± 0.00	0.05 ± 0.01	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.01 \pm 0.00}$	0.79 ± 0.31
colon-cancer	$\mathbf{0.20 \pm 0.16}$	$\mathbf{0.20 \pm 0.16}$	0.27 ± 0.13	0.27 ± 0.25	0.27 ± 0.25	$\mathbf{0.20 \pm 0.27}$	0.27 ± 0.25	$\mathbf{0.20 \pm 0.16}$
covtype	0.22 ± 0.00	0.16 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.38 ± 0.00	0.31 ± 0.17	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.00	0.49 ± 0.05
cskaggle	0.75 ± 0.01	0.68 ± 0.01	0.87 ± 0.01	0.84 ± 0.01	0.92 ± 0.00	0.70 ± 0.01	$\mathbf{0.67 \pm 0.01}$	0.92 ± 0.01
diabetes	0.24 ± 0.09	0.28 ± 0.06	0.24 ± 0.11	$\mathbf{0.20 \pm 0.06}$	0.36 ± 0.07	0.23 ± 0.07	0.25 ± 0.05	0.52 ± 0.27
fourclass	0.12 ± 0.04	0.09 ± 0.06	$\mathbf{0.00 \pm 0.00}$	0.11 ± 0.03	0.10 ± 0.07	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	0.35 ± 0.32
german	0.11 ± 0.02	0.12 ± 0.03	0.23 ± 0.02	0.12 ± 0.03	0.23 ± 0.10	$\mathbf{0.09 \pm 0.02}$	0.11 ± 0.03	0.26 ± 0.06
gisette	0.02 ± 0.01	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00
gmsc	0.52 ± 0.01	$\mathbf{0.48 \pm 0.01}$	0.83 ± 0.01	0.76 ± 0.06	0.83 ± 0.08	0.50 ± 0.01	$\mathbf{0.48 \pm 0.01}$	0.96 ± 0.00
heart	$\mathbf{0.04 \pm 0.06}$	0.13 ± 0.05	0.24 ± 0.10	$\mathbf{0.04 \pm 0.06}$	0.07 ± 0.08	$\mathbf{0.04 \pm 0.06}$	0.07 ± 0.05	0.13 ± 0.07
housing	0.80 ± 0.19	0.80 ± 0.19	0.70 ± 0.19	0.90 ± 0.20	0.95 ± 0.10	0.70 ± 0.10	$\mathbf{0.65 \pm 0.12}$	0.95 ± 0.10
ijcnn1	0.24 ± 0.02	0.20 ± 0.01	0.01 ± 0.00	0.29 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.80 ± 0.03
ionosphere	0.03 ± 0.02	0.03 ± 0.03	0.11 ± 0.07	0.07 ± 0.07	0.02 ± 0.02	$\mathbf{0.01 \pm 0.02}$	0.02 ± 0.02	0.15 ± 0.07
liver-disorders	0.23 ± 0.13	0.33 ± 0.11	0.43 ± 0.13	0.30 ± 0.19	0.33 ± 0.24	$\mathbf{0.20 \pm 0.19}$	0.23 ± 0.17	0.47 ± 0.19
madelon	0.33 ± 0.01	0.33 ± 0.06	0.16 ± 0.05	0.42 ± 0.07	0.49 ± 0.06	0.17 ± 0.03	$\mathbf{0.10 \pm 0.01}$	0.44 ± 0.05
mammography	0.21 ± 0.07	0.17 ± 0.04	0.12 ± 0.03	0.25 ± 0.07	0.10 ± 0.07	$\mathbf{0.07 \pm 0.04}$	0.08 ± 0.06	0.53 ± 0.21
mq2008	0.43 ± 0.02	0.39 ± 0.01	0.51 ± 0.01	0.46 ± 0.03	0.47 ± 0.07	$\mathbf{0.24 \pm 0.01}$	0.25 ± 0.01	0.83 ± 0.06
oil-spill	0.52 ± 0.27	0.52 ± 0.24	0.80 ± 0.18	0.72 ± 0.16	0.92 ± 0.16	$\mathbf{0.44 \pm 0.27}$	0.48 ± 0.24	0.96 ± 0.08
phishing	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
phoneme	0.33 ± 0.03	0.19 ± 0.02	0.06 ± 0.01	0.43 ± 0.03	0.13 ± 0.02	$\mathbf{0.03 \pm 0.01}$	0.06 ± 0.02	0.73 ± 0.15
skin-nonskin	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
sonar	0.06 ± 0.08	0.10 ± 0.09	0.10 ± 0.09	0.06 ± 0.05	$\mathbf{0.02 \pm 0.04}$	0.04 ± 0.05	0.04 ± 0.08	0.04 ± 0.05
splice	0.03 ± 0.02	0.02 ± 0.02	0.19 ± 0.04	0.11 ± 0.06	0.07 ± 0.05	$\mathbf{0.00 \pm 0.01}$	0.01 ± 0.01	0.12 ± 0.05
svmguide1	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.01}$	0.02 ± 0.01	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
svmguide3	0.27 ± 0.08	0.19 ± 0.06	0.28 ± 0.06	0.30 ± 0.09	0.23 ± 0.08	$\mathbf{0.17 \pm 0.07}$	$\mathbf{0.17 \pm 0.07}$	0.54 ± 0.16
taiwan	0.39 ± 0.01	$\mathbf{0.33 \pm 0.02}$	0.65 ± 0.01	0.65 ± 0.02	0.79 ± 0.01	$\mathbf{0.33 \pm 0.01}$	0.34 ± 0.01	0.64 ± 0.06
w1a	0.25 ± 0.11	0.40 ± 0.20	0.42 ± 0.06	0.32 ± 0.17	0.32 ± 0.17	0.25 ± 0.14	0.38 ± 0.14	$\mathbf{0.18 \pm 0.10}$

Table 10: Evaluation of estimators with P@k as metric

Dataset	ExactBoost	AdaBoost	kNN	Logistic	Neural Net	Rand. For.	XGBoost	TopPush
a1a	$\mathbf{0.22 \pm 0.12}$	0.34 ± 0.09	0.28 ± 0.14	0.28 ± 0.14	0.32 ± 0.12	0.34 ± 0.15	0.40 ± 0.10	0.29 ± 0.09
australian	$\mathbf{0.05 \pm 0.03}$	0.10 ± 0.04	0.37 ± 0.11	0.07 ± 0.06	0.17 ± 0.08	0.14 ± 0.08	0.14 ± 0.07	0.06 ± 0.03
banknote	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.02	$\mathbf{0.00 \pm 0.00}$
breast-cancer	0.02 ± 0.03	$\mathbf{0.01 \pm 0.02}$	0.58 ± 0.05	0.32 ± 0.23	0.63 ± 0.09	0.07 ± 0.04	0.04 ± 0.03	0.60 ± 0.14
cod-rna	0.04 ± 0.00	$\mathbf{0.01 \pm 0.00}$	0.07 ± 0.01	$\mathbf{0.01 \pm 0.00}$	$\mathbf{0.01 \pm 0.00}$	0.07 ± 0.01	0.07 ± 0.01	0.16 ± 0.19
colon-cancer	0.60 ± 0.33	$\mathbf{0.40 \pm 0.33}$	$\mathbf{0.40 \pm 0.25}$	$\mathbf{0.40 \pm 0.25}$	$\mathbf{0.40 \pm 0.13}$	0.47 ± 0.27	$\mathbf{0.40 \pm 0.13}$	$\mathbf{0.40 \pm 0.13}$
covtype	0.02 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	0.17 ± 0.08	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.00	0.02 ± 0.00	0.27 ± 0.13
cskaggle	0.69 ± 0.01	$\mathbf{0.68 \pm 0.01}$	0.87 ± 0.01	0.84 ± 0.00	0.92 ± 0.00	0.90 ± 0.01	0.86 ± 0.01	0.92 ± 0.02
diabetes	$\mathbf{0.21 \pm 0.08}$	0.36 ± 0.15	0.39 ± 0.06	0.25 ± 0.10	0.47 ± 0.04	0.34 ± 0.13	0.36 ± 0.09	0.28 ± 0.11
fourclass	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.04	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
german	$\mathbf{0.13 \pm 0.04}$	0.16 ± 0.07	0.26 ± 0.04	$\mathbf{0.13 \pm 0.04}$	0.33 ± 0.05	0.20 ± 0.05	0.21 ± 0.06	0.18 ± 0.03
gisette	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01
gmsc	0.51 ± 0.02	$\mathbf{0.48 \pm 0.01}$	0.85 ± 0.01	0.74 ± 0.06	0.88 ± 0.03	0.65 ± 0.07	0.62 ± 0.04	0.96 ± 0.01
heart	0.07 ± 0.08	0.19 ± 0.09	0.33 ± 0.11	$\mathbf{0.06 \pm 0.05}$	0.19 ± 0.07	0.23 ± 0.08	0.29 ± 0.10	0.14 ± 0.15
housing	$\mathbf{0.65 \pm 0.25}$	0.70 ± 0.10	0.85 ± 0.12	0.75 ± 0.16	0.85 ± 0.20	0.75 ± 0.22	0.75 ± 0.22	$\mathbf{0.65 \pm 0.20}$
ijcnn1	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	0.05 ± 0.01	0.06 ± 0.01	$\mathbf{0.00 \pm 0.00}$
ionosphere	$\mathbf{0.03 \pm 0.03}$	0.04 ± 0.09	0.13 ± 0.04	0.05 ± 0.05	0.06 ± 0.06	0.09 ± 0.07	0.10 ± 0.06	0.10 ± 0.06
liver-disorders	$\mathbf{0.27 \pm 0.23}$	0.33 ± 0.18	0.40 ± 0.23	0.33 ± 0.21	0.40 ± 0.27	0.40 ± 0.25	0.33 ± 0.21	0.30 ± 0.24
madelon	$\mathbf{0.14 \pm 0.05}$	0.22 ± 0.03	0.24 ± 0.06	0.45 ± 0.04	0.48 ± 0.05	0.30 ± 0.06	0.26 ± 0.03	0.46 ± 0.04
mammography	$\mathbf{0.09 \pm 0.04}$	0.13 ± 0.08	0.14 ± 0.04	0.13 ± 0.06	0.12 ± 0.04	0.17 ± 0.04	0.25 ± 0.06	0.23 ± 0.13
mq2008	0.31 ± 0.01	$\mathbf{0.29 \pm 0.01}$	0.60 ± 0.02	0.47 ± 0.03	0.66 ± 0.03	0.30 ± 0.04	0.38 ± 0.02	0.85 ± 0.07
oil-spill	$\mathbf{0.44 \pm 0.23}$	0.72 ± 0.20	0.88 ± 0.16	0.84 ± 0.15	0.92 ± 0.10	0.72 ± 0.16	0.68 ± 0.27	0.68 ± 0.24
phishing	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.03 ± 0.01	0.03 ± 0.01	$\mathbf{0.00 \pm 0.00}$
phoneme	0.07 ± 0.02	0.05 ± 0.02	0.07 ± 0.02	$\mathbf{0.04 \pm 0.01}$	0.06 ± 0.01	0.16 ± 0.04	0.17 ± 0.03	0.05 ± 0.02
skin-nonskin	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
sonar	$\mathbf{0.02 \pm 0.04}$	0.24 ± 0.21	0.06 ± 0.08	$\mathbf{0.02 \pm 0.04}$	0.04 ± 0.05	0.16 ± 0.10	0.18 ± 0.04	$\mathbf{0.02 \pm 0.04}$
splice	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$	0.16 ± 0.01	0.04 ± 0.02	0.04 ± 0.01	0.05 ± 0.03	0.05 ± 0.02	0.05 ± 0.03
svmguide1	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.00	0.01 ± 0.01	$\mathbf{0.00 \pm 0.00}$	0.01 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	$\mathbf{0.00 \pm 0.00}$
svmguide3	0.21 ± 0.06	0.25 ± 0.12	0.24 ± 0.09	$\mathbf{0.17 \pm 0.07}$	$\mathbf{0.17 \pm 0.06}$	0.25 ± 0.07	0.35 ± 0.07	0.22 ± 0.07
taiwan	0.37 ± 0.00	$\mathbf{0.35 \pm 0.02}$	0.65 ± 0.01	0.65 ± 0.02	0.79 ± 0.01	0.36 ± 0.02	0.41 ± 0.01	0.65 ± 0.05
w1a	$\mathbf{0.25 \pm 0.14}$	0.32 ± 0.13	0.70 ± 0.10	0.35 ± 0.20	0.38 ± 0.08	0.57 ± 0.20	0.32 ± 0.15	0.30 ± 0.17

Table 11: Evaluation of ensemblers with P@k as metric

H Visualizing ExactBoost trajectories

This section details how Figure 1 in the paper was generated.

To visualize ExactBoost's trajectories in 2D, we start by training several ExactBoost models, varying the hyperparameters used, and each model is represented as a vector in $\mathbb{R}^{4ET}$, with E being the number of runs averaged and T the number of rounds. We then run denseMAP McInnes and Healy (2018) on these vectors in order to reduce their dimensions from $\mathbb{R}^{4ET}$ to $\mathbb{R}^2$. Note that this usage of UMAP ensures that, with high probability, similar trajectories (and similar trained ExactBoosts) are plotted close to each other. Finally, we plot these points using hexbins, colored with the value of the losses of their corresponding model.

In order to transform a trained model into a vector, we start with a vector $v = 0 \in \mathbb{R}^{4ET}$. If we want to represent a model which has $m < T$ rounds, we act as if it was trained for T rounds, but with all stumps after round m being null stumps (i.e., $j = \xi = a = b = 0$). Then, for each round t and estimator l, we set the positions $v_{4Tl+4t+0} = j_{t,l}$, $v_{4Tl+4t+1} = \xi_{t,l}$, $v_{4Tl+4t+2} = a_{t,l}$ and $v_{4Tl+4t+3} = b_{t,l}$.

Note that generating these landscapes based on trained models gives us a visualization of places where ExactBoost actually navigates around, omitting points where it is not likely to visit. In order to keep this visualization reasonably fair, the models are trained both using sets of hyperparameters for which it performs well and others which can intentionally throw it off-course. This way, we maintain our bias towards points that ExactBoost can visit, while also illustrating bad minima.

For more details, see the scripts in `src/eval/trajectory_plots/` to generate, evaluate, project and plot landscapes and trajectories.

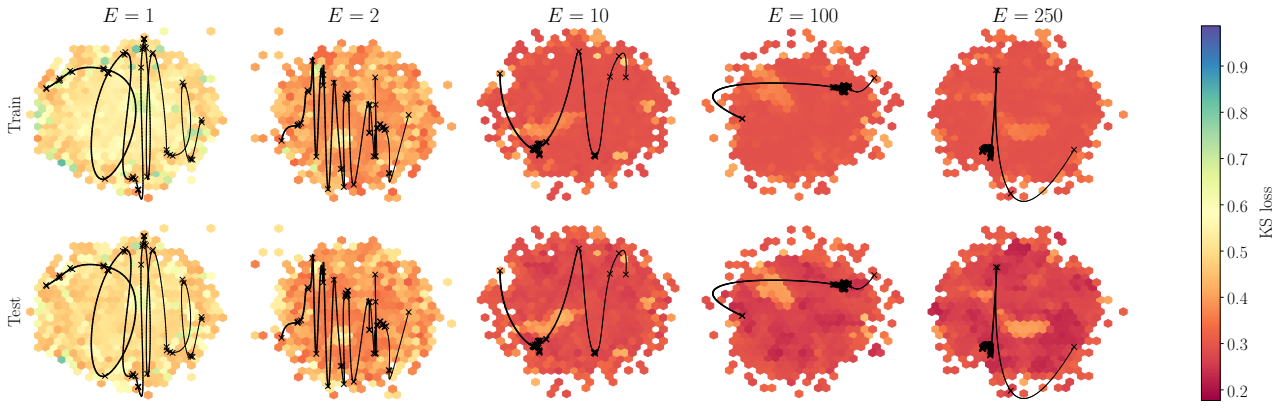

Figure 4: KS loss landscape visualizations via UMAP highlighting ExactBoost's optimization trajectories, which go from left to right. More averaged runs E lead to better train and test losses.