ExactBoost: Directly Boosting Combinatorial and Non-decomposable Metrics

Daniel Csillag Carolina Piazza Thiago Ramos João Vitor Romano
Roberto Oliveira Paulo Orenstein

Instituto Nacional de Matemática Pura e Aplicada (IMPA)

Introduction

We have data $(X_i, y_i)_{i=1}^n \stackrel{\text{iid}}{\sim} \mathcal{D}$, with $X_i \in \mathbb{R}^p$, and $y_i \in \{0, 1\}$.

We want to learn a good score function $S : \mathbb{R}^p \to [0, 1]$.

Introduction

We have data $(X_i, y_i)_{i=1}^n \overset{\text{iid}}{\sim} \mathcal{D}$, with $X_i \in \mathbb{R}^p$, and $y_i \in \{0, 1\}$.

We want to learn a good score function $S : \mathbb{R}^p \to [0, 1]$.

Our loss functions:

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

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

Combinatorial and Non-Decomposable Loss Functions

Combinatorial and Non-Decomposable Loss Functions

A combinatorial loss function is one that is computed in terms of indicator functions.

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

Combinatorial and Non-Decomposable Loss Functions

A combinatorial loss function is one that is computed in terms of indicator functions.

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

A decomposable loss function is one where

$$\widehat{L}(S, y) = \frac{1}{n} \sum_{i=1}^{n} \widehat{L}(S_i, y_i)$$

Our metrics are <u>non</u>-decomposable.

Empirical Error, Generalization Error and Margin Theory

- Empirical Error: $\widehat{L}(S(X), y)$

Empirical Error, Generalization Error and Margin Theory

- Empirical Error: $\widehat{L}(S(X), y)$

- Generalization Error: $L(S) = \mathbb{E}_{\mathcal{D}}[\widehat{L}(S, y)]$

Empirical Error, Generalization Error and Margin Theory

- Empirical Error: $\widehat{L}(S(X), y)$

- Generalization Error: $L(S) = \mathbb{E}_{\mathcal{D}}[\widehat{L}(S, y)]$

- Margin-adjusted loss: $\widehat{L}_{\theta}(S, y) := \widehat{L}(S - \theta y, y)$

Boosting

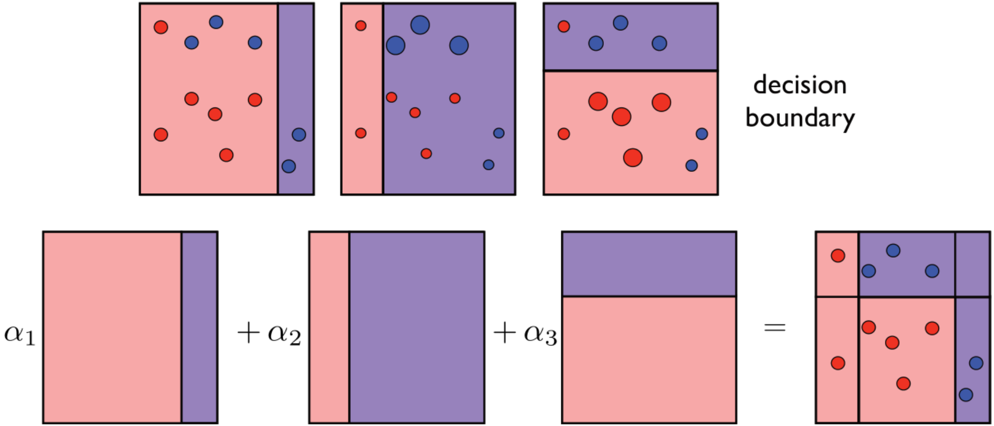


Image taken from Mehryar Mohri, et al. Foundations of Machine Learning, second edition, page 147

Our Boosting Framework

$$S = \sum_i \alpha_i h_i(X), \qquad \sum_i \alpha_i = 1$$

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

Our Boosting Framework

$$S = \sum_i \alpha_i h_i(X), \qquad \sum_i \alpha_i = 1$$

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

$$(\alpha_t, h_t) = \arg\min_{\alpha, h} \widehat{L}_\theta \left(\frac{1}{1+\alpha} S_{t-1} + \frac{\alpha}{1+\alpha} h(X), y \right)$$

Our Boosting Framework

$$S = \sum_i \alpha_i h_i(X), \qquad \sum_i \alpha_i = 1$$

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

$$\begin{aligned}(\alpha_t, h_t) &= \arg\min_{\alpha, h} \widehat{L}_\theta \left(\frac{1}{1+\alpha} S_{t-1} + \frac{\alpha}{1+\alpha} h(X), y \right) \\ &= \arg\min_{a,b,\xi,j} \widehat{L}(S_{t-1} + a\mathbf{1}_{[X_{(j)} \le \xi]} + b\mathbf{1}_{[X_{(j)} > \xi]} - \left(1 - \frac{|b-a|}{2}\right) \theta y, y)\end{aligned}$$

Stagewise Minimization Procedure

function $\textsc{ExactBoost}$(data (X, y), initial score S_0)
  $S \leftarrow S_0$
  for $t \in \{1, \ldots, T\}$ **do**
    for $j \in \{1, \ldots, p\}$ **do**
      $(\xi, a, b) \leftarrow \arg\min_{\xi,a,b} \widehat{L}(S + a\mathbf{1}_{[X_{(j)} \leq \xi]} + b\mathbf{1}_{[X_{(j)} > \xi]} - (1 - \frac{|b-a|}{2})\theta y, y)$
      $h_j \leftarrow a\mathbf{1}_{[X_{(j)} \leq \xi]} + b\mathbf{1}_{[X_{(j)} > \xi]}$
    $S' \leftarrow S + \arg\min_{h_j} \widehat{L}(S + h_j, y)$
    if $\widehat{L}(S', y) \leq \widehat{L}(S, y)$ **then**
      $S \leftarrow S'$
  return S

Stagewise Minimization Procedure

function $\textsc{ExactBoost}$(data (X, y), initial score S_0)
  for $e \in \{1, \ldots, E\}$ **do**
    $S_e \leftarrow S_0$
    for $t \in \{1, \ldots, T\}$ **do**
      $X^s, y^s \leftarrow$ subsample X, y
      for $j \in \{1, \ldots, p\}$ **do**
        $(\xi, a, b) \leftarrow \arg\min_{\xi,a,b} \widehat{L}(S_e + a\mathbf{1}_{[X^s_{(j)} \le \xi]} + b\mathbf{1}_{[X^s_{(j)} > \xi]} - (1 - \frac{|b-a|}{2})\theta y^s, y^s)$
        $h_j \leftarrow a\mathbf{1}_{[X^s_{(j)} \le \xi]} + b\mathbf{1}_{[X^s_{(j)} > \xi]}$
      $S'_e \leftarrow S_e + \arg\min_{h_j} \widehat{L}(S_e + h_j, y^s)$
      if $\widehat{L}(S'_e, y) \le \widehat{L}(S_e, y)$ **then**
        $S_e \leftarrow S'_e$
  return $\mathrm{mean}(S_1, \ldots, S_E)$

Generalization Bound for AUC

Theorem

Given $\theta > 0$, $\delta \in (0,1)$, 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

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

Generalization Bound for KS

Theorem

Given $\theta > 0$, $\delta \in (0, 1)$, 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

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

Ensembling

Proposition

Consider the score $S_\star : \mathbb{R}^M \to \mathbb{R}$ *obtained by ExactBoost over the dataset* $(Z_i, y_i)_{i=1}^n$ *with initial score* $S_0 \equiv 0$*. Then:*

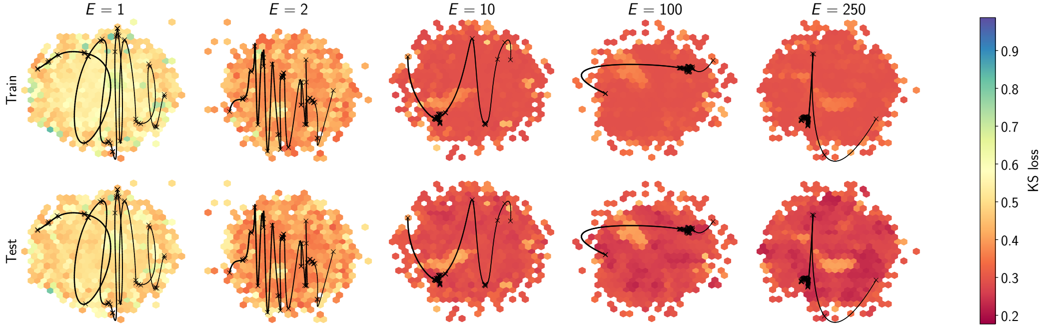
$$\widehat{L}_{(Z_i,y_i)_{i=1}^n}(S_*) \leq \min_{1 \leq m \leq M} \widehat{L}_{(X_i,y_i)_{i=1}^n}(S_m),$$

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.*

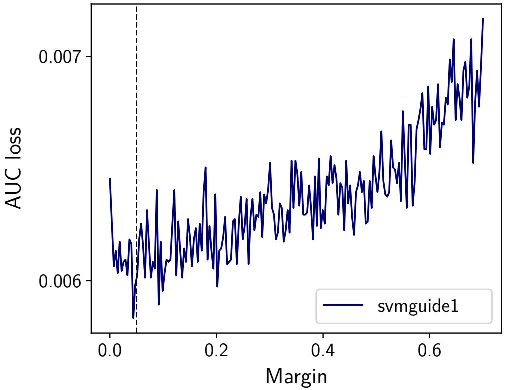
Datasets

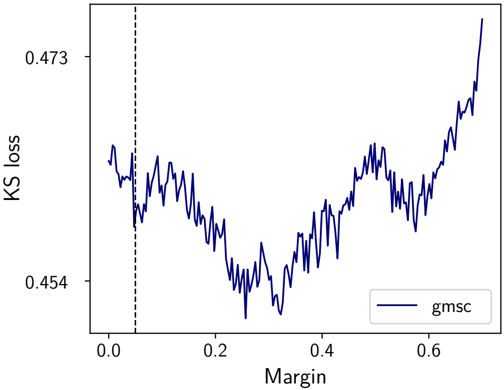
Dataset	Observations	Features	Positives
a1a	1605	119	24.61%
german	1000	20	70.0%
gisette	6000	5000	50.0%
gmsc	150000	10	6.68%
heart	303	21	45.87%
ionosphere	351	34	64.1%
liver-disorders	145	5	37.93%
oil-spill	937	49	4.38%
splice	1000	60	48.3%
svmguide1	3089	4	35.25%

ExactBoost Hyperparameters — Run Averaging



ExactBoost Hyperparameters — Margin





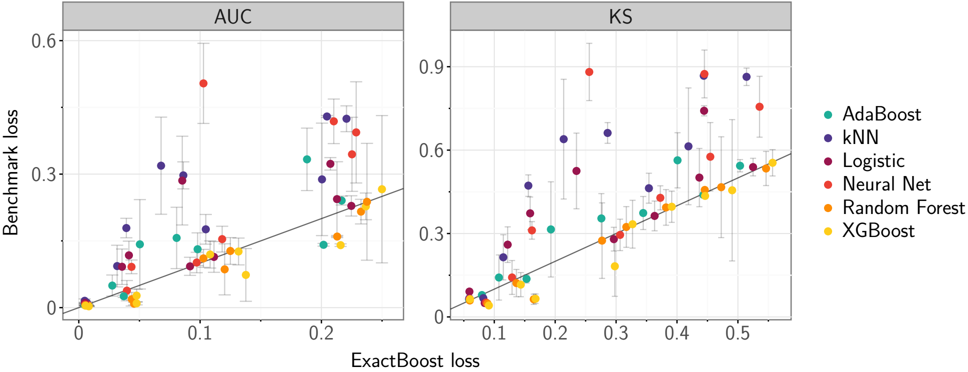
Experimental Benchmarks

- Surrogate benchmarks:
  - AdaBoost;
  - *k*-nearest neighbors;
  - Logistic Regression;
  - Random Forest;
  - XGBoost (Gradient Boosting);
  - Neural Network (4-layer fully connected).
- Exact benchmarks:
  - RankBoost (optimizes AUC);
  - DMKS (optimizes KS);

ExactBoost as an Estimator vs. Exact Benchmarks

	AUC		KS	
Dataset	ExactBoost	RankBoost	ExactBoost	DMKS
a1a	$\mathbf{0.11 \pm 0.0}$	0.13 ± 0.0	$\mathbf{0.37 \pm 0.0}$	$\mathbf{0.37 \pm 0.0}$
german	$\mathbf{0.23 \pm 0.0}$	0.24 ± 0.0	$\mathbf{0.53 \pm 0.0}$	0.55 ± 0.0
gisette	$\mathbf{0.01 \pm 0.0}$	OOT	0.09 ± 0.0	$\mathbf{0.06 \pm 0.0}$
gmsc	$\mathbf{0.21 \pm 0.0}$	OOT	$\mathbf{0.44 \pm 0.0}$	0.45 ± 0.0
heart	$\mathbf{0.09 \pm 0.0}$	0.13 ± 0.0	0.30 ± 0.0	$\mathbf{0.28 \pm 0.0}$
iono.	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	$\mathbf{0.13 \pm 0.0}$	0.28 ± 0.0
liver	$\mathbf{0.22 \pm 0.1}$	0.32 ± 0.1	$\mathbf{0.45 \pm 0.1}$	0.50 ± 0.1
oil-spill	$\mathbf{0.09 \pm 0.1}$	$\mathbf{0.09 \pm 0.1}$	$\mathbf{0.25 \pm 0.1}$	0.45 ± 0.1
splice	0.04 ± 0.0	$\mathbf{0.02 \pm 0.0}$	$\mathbf{0.16 \pm 0.0}$	0.36 ± 0.0
svmguide1	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.06 \pm 0.0}$	0.09 ± 0.0

ExactBoost as an Estimator vs. Surrogate Benchmarks



ExactBoost as an Ensembler

AUC

Dataset	ExactBoost	AdaBoost	Logistic	Neural Net	Rand. For.	XGBoost	Exact Bench.
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

ExactBoost as an Ensembler

KS

Dataset	ExactBoost	AdaBoost	Logistic	Neural Net	Rand. For.	XGBoost	Exact Bench.
a1a	**0.37 ± 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	**0.50 ± 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	**0.04 ± 0.0**	**0.04 ± 0.0**	0.07 ± 0.0	0.07 ± 0.0	0.06 ± 0.0	**0.04 ± 0.0**	0.10 ± 0.0
gmsc	**0.43 ± 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	**0.34 ± 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.	**0.13 ± 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	**0.53 ± 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	**0.33 ± 0.2**	**0.33 ± 0.2**	0.47 ± 0.2	0.89 ± 0.1	0.76 ± 0.2	0.69 ± 0.3	0.63 ± 0.3
splice	**0.06 ± 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	**0.06 ± 0.0**	0.08 ± 0.0	**0.06 ± 0.0**	**0.06 ± 0.0**	0.07 ± 0.0	0.07 ± 0.0	**0.06 ± 0.0**

Conclusion

- ExactBoost is a competitive estimator and an even better ensembler;
- There is value to be gained in working with the intended loss function;
- Novel theoretical results bound the generalization error for AUC and KS;
- Paper and source code to be released.

Thank you!