Ladder and Kaggle

Aaditya Ramdas

Carnegie Mellon University

Thanks, mrtz.org

Blum and Hardt, ICML

Outline

1. *Kaggle and the wacky boosting attack*

2. *The ladder mechanism, and implications for practice*

Kaggle (approximate)

1. Labeled training set released publicly.
   Unlabeled holdout+test set, size N (say 12K), also released (but don't know which point is holdout, which is test).

2. Anyone can submit N labels to the system, multiple times, and it a score on the holdout set of size $n = 0.3N$ is released (score=say, empirical risk on n samples, up to 5-6 digits).

3. There is a public leaderboard, through the period of the competition, with the current best holdout score

4. When the competition ends, the final prize is determined by accuracy on test dataset (size 0.7N)

Kaggle

Dashboard

Public Leaderboard - Heritage Health Prize

This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

#	Δ1w	Team Name * in the money	Score	Entries
1	—	EXL Analytics *	0.443793	555
2	—	POWERDOT	0.447651	671
3	—	Dolphin	0.450403	555
4	↑1	jack3	0.451425	455
5	↓1	Hopkins Biostat	0.451569	444
6	—	Xing Zhao	0.453081	161
7	—	Old Dogs With New Tricks	0.454096	370
8	—	Areté Associates	0.454424	112
9	—	Alice Sasandr	0.454670	376
10	↑9	J.A. Guerrero	0.454728	173

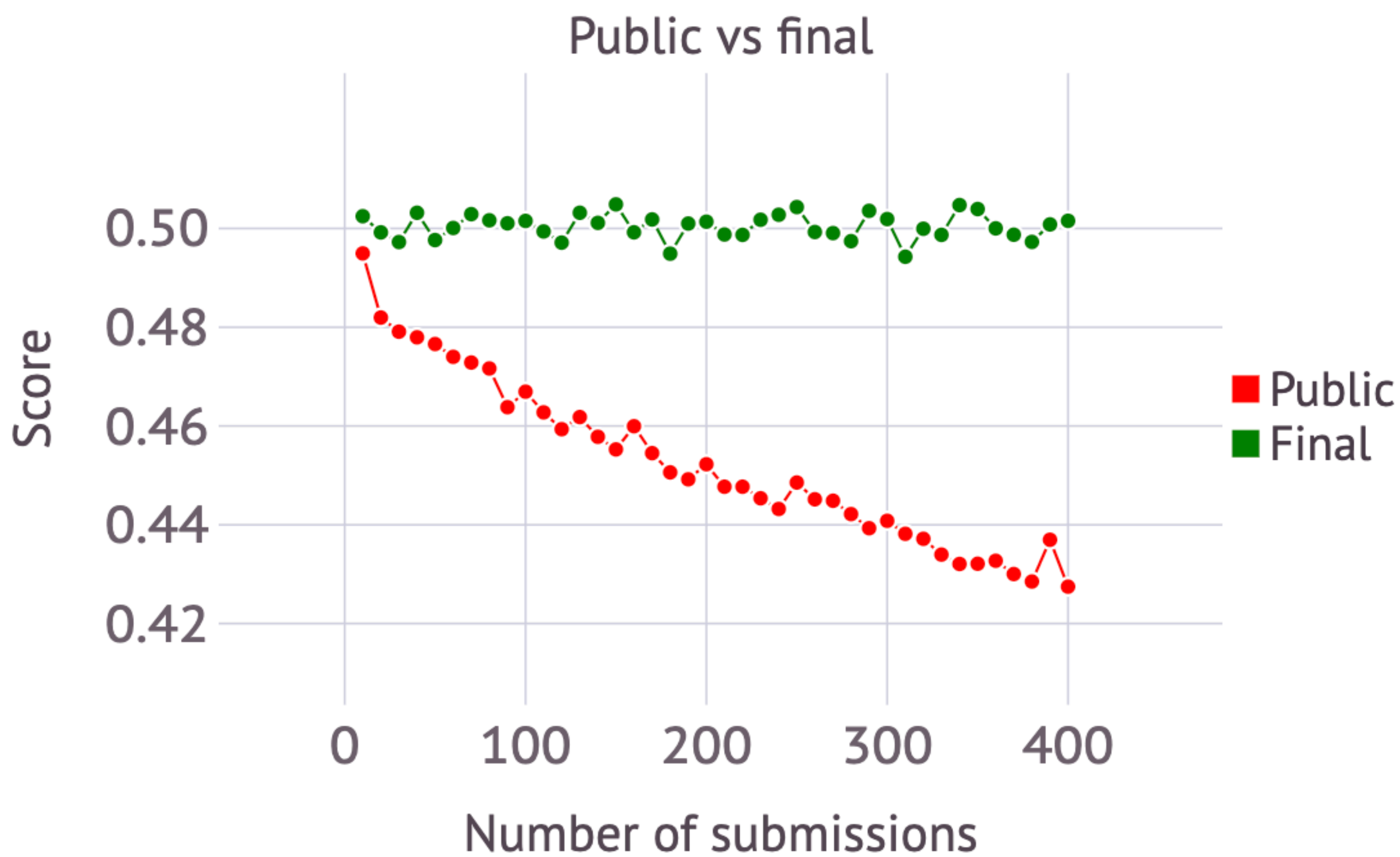
2 year competition, 3 million dollar prize

(Leaderboard standings do NOT affect who gets prize)

Wacky boosting v1

Algorithm (Wacky Boosting):

1. Choose $y_1, \dots, y_k \in \{0,1\}^N$ uniformly at random.
2. Let $I = \{i \in [k] \colon s_H(y_i) < 0.5\}$.
3. Output $\hat{y} = \mathbf{majority}\{y_i \colon i \in I\}$, where the majority is component-wise.



In this plot, $n = 4000$ and all numbers are averaged over 5 independent repetitions.

Wacky boosting v1

Algorithm (Wacky Boosting):

1. Choose $y_1, \ldots, y_k \in \{0,1\}^N$ uniformly at random.
2. Let $I = \{i \in [k] \colon s_H(y_i) < 0.5\}$.
3. Output $\hat{y} = \mathbf{majority}\{y_i \colon i \in I\}$, where the majority is component-wise.

$$s_H(y_i) \approx 1/2 \pm O(1/\sqrt{n})$$

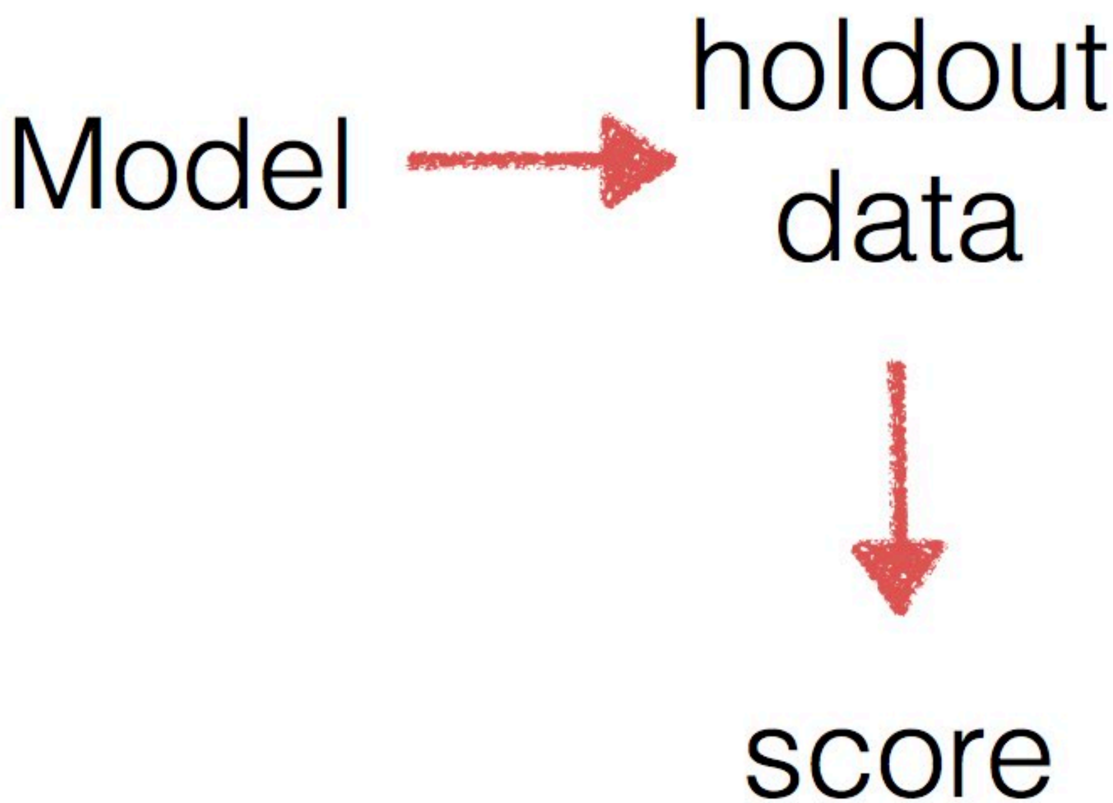
gets boosted to

$$s_H(\hat{y}) \approx 1/2 - O(\sqrt{k/n})$$

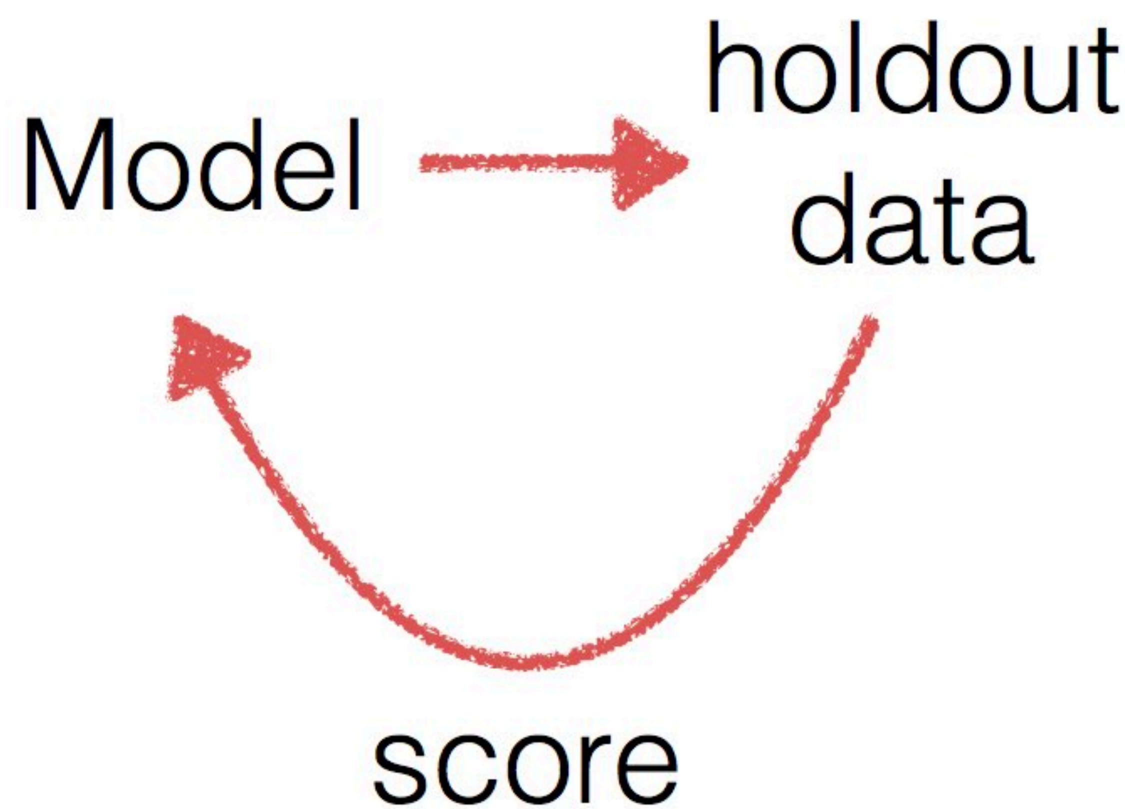
This is simply majority vote amongst the ones with a small edge.

Can generalize beyond 0/1 loss, to [0,1]-bounded loss.
Wacky boosting works with rounded or $1/\sqrt{n}$-approx scores.

Static
data analysis



Interactive
data analysis



Outline

1. *Kaggle and the wacky boosting attack*

2. *The ladder mechanism, and implications for practice*

Leaderboard error

Given a sequence $f_1, \ldots, f_k$ of classifiers,
a finite sample S of size n,
compute "empirical risks" $R_1, \ldots, R_k$ such that

$$\mathbb{P}\mathrm{r}\left\{\exists t \in [k]:\ |R_t - R_{\mathcal{D}}(f_t)| > \varepsilon\right\} \leqslant \delta\,.$$

If $f_1, \ldots, f_k$ are independent, apply Hoeffding's inequality:

$$\mathbb{P}\mathrm{r}\left\{\exists t \in [k]:\ |R_S(f_t) - R_{\mathcal{D}}(f_t)| > \varepsilon\right\} \leqslant 2k\exp(-2\varepsilon^2 n)\,.$$

Does not work when $f_t = \mathcal{A}(f_1, R_1, \ldots, f_{t-1}, R_{t-1})$.

$$\mathrm{lberr}(R_1, \ldots, R_k) \stackrel{\mathrm{def}}{=} \max_{1 \leqslant t \leqslant k} \left| \min_{1 \leqslant i \leqslant t} R_{\mathcal{D}}(f_i) - R_t \right|$$

The Ladder algorithm

Input: Data set S, step size $\eta > 0$
Algorithm:

- Assign initial estimate $R_0 \leftarrow \infty$.
- **For each** round $t \leftarrow 1, 2 \ldots$:
  1. Receive function $f_t : X \to Y$
  2. **If** $R_S(f_t) < R_{t-1} - \eta$, assign $R_t \leftarrow [R_S(f_t)]_\eta$. **Else** assign $R_t \leftarrow R_{t-1}$.
  3. **Output** R_t

$[x]_\eta$ is a rounding of x to the nearest multiple of η.

Theorem 3.1. *For any sequence of adaptively chosen classifiers $f_1, \ldots, f_k$, the Ladder Mechanism satisfies for all $t \leqslant k$ and $\varepsilon > 0$,*

$$\mathbb{P}\mathrm{r}\left\{\left|\min_{1\leqslant i\leqslant t} R_{\mathcal{D}}(f_i) - R_t\right| > \varepsilon + \eta\right\} \leqslant \exp\left(-2\varepsilon^2 n + (1/\eta + 2)\log(4t/\eta) + 1\right). \qquad (4)$$

In particular, for some $\eta = O(n^{-1/3}\log^{1/3}(kn))$, the Ladder Mechanism achieves with high probability,

$$\mathrm{lberr}(R_1, \ldots, R_k) \leqslant O\left(\frac{\log^{1/3}(kn)}{n^{1/3}}\right).$$

Lower bound

Theorem 3.3. *There are classifiers $f_1, \ldots f_k$ and a bounded loss function for which we have the minimax lower bound*

$$\inf_R \sup_{\mathcal{D}} \mathbb{E}[\text{lberr}(R(x_1, \ldots, x_n))] \geqslant \Omega\left(\sqrt{\frac{\log k}{n}}\right).$$

Here the infimum is taken over all estimators $R \colon X^n \to [0,1]^k$ that take n samples from a distribution $\mathcal{D}$ and produce k estimates $R_1, \ldots, R_k = \widehat{\theta}(x_1, \ldots, x_n)$. The expectation is taken over n samples from $\mathcal{D}$.

"Parameter free ladder"

Input: Data set $S = \{(x_1, y_1), \dots (x_n \dots, y_n)\}$ of size n
Algorithm:

- Assign initial estimate $R_0 \leftarrow \infty$, and loss vector $\ell_0 = (0)_{i=1}^n$.
- **For each** round $t \leftarrow 1, 2 \dots, k$:
  1. Receive function $f_t : X \rightarrow Y$.
  2. Compute loss vector $l_t \leftarrow (\ell(f_t(x_i), y_i))_{i=1}^n$
  3. Compute the sample standard deviation $s \leftarrow \mathrm{std}(l_t - l_{t-1})$.
  4. **If** $R_S(f_t) < R_{t-1} - s/\sqrt{n}$
     (a) $R_t \leftarrow [R_S(f_t)]_{1/n}$.
  5. **Else** assign $R_t \leftarrow R_{t-1}$ and $l_t \leftarrow l_{t-1}$.
  6. **Output** R_t

A Kaggle reference mechanism

As we did for the Ladder Mechanism we describe the algorithm as if the analyst was submitting classifiers $f : X \rightarrow Y$. In reality the analyst only submits a list of labels. It is easy to see that such a list of labels is sufficient to compute the empirical loss which is all the algorithm needs to do. The input set S in the description of our algorithm corresponds to the set of data points (and corresponding labels) that Kaggle uses for the public leaderboard.

Input: Data set S, rounding parameter $\alpha > 0$ (typically 0.00001)
Algorithm:

- **For each** round $t \leftarrow 1, 2 \dots, k$:
  1. Receive function $f_t : X \rightarrow Y$
  2. **Output** $[R_S(f_t)]_\alpha$.

How does it do?

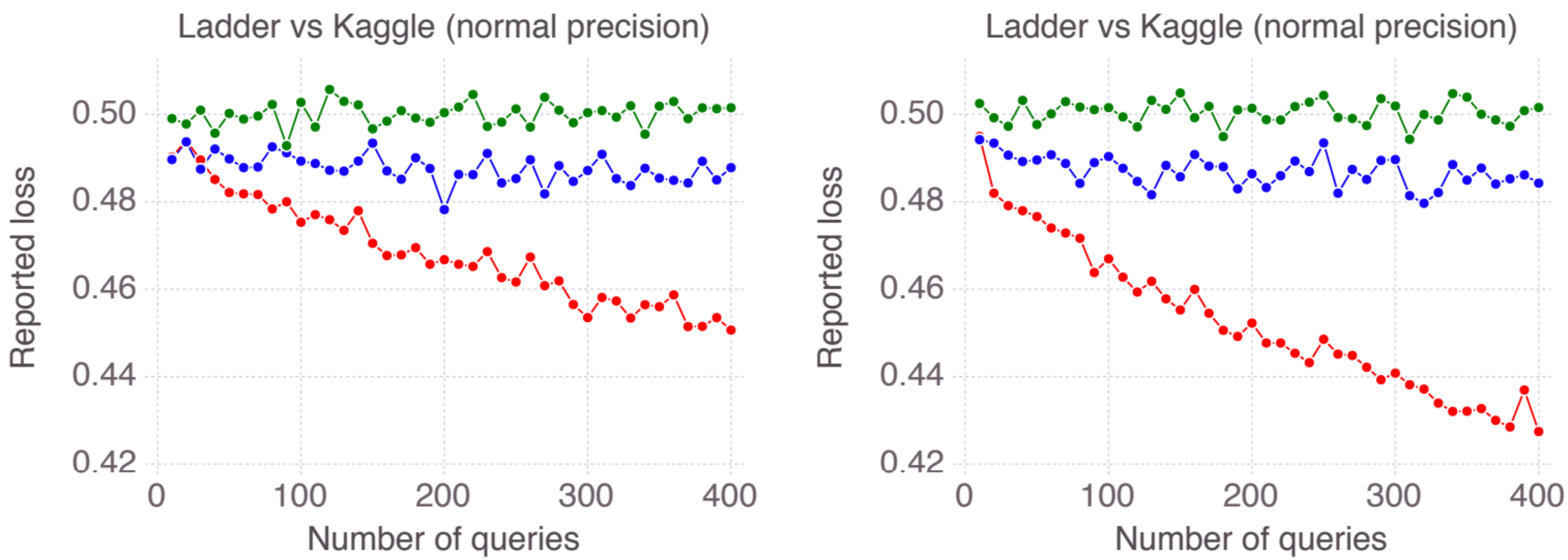


Figure 3: Performance of the parameter free Ladder Mechanism compared with the Kaggle Mechanism. Top **green** line: Independent test set. Middle **blue** line: Ladder. Bottom **red** line: Kaggle. **Left**: Kaggle with large rounding parameter $1/\sqrt{n} \approx 0.0158$. **Right:** Kaggle with normal rounding parameter 0.00001. All numbers are averaged over 5 independent repetitions of the experiment. Number of labels used is $n = 4000$.

Implications for practice?

1. Important to think about double-dipping into the "holdout" set (holdout score and test score could differ)

2. Perhaps what practitioners do is ladder-like anyway? (if their newest "tweak" was not considerably better, discard)

3. Interactive or adaptive data analysis is an interesting area (check out Boyan Duan's thesis this summer)

4. Many followup works to Ladder (reusable holdout, etc)

5. Connected to differential privacy, selective inference, post-selection inference, conditional inference, etc. (check out WADAPT workshop, NeurIPS 2015/16)

Outline

1. *Kaggle and the wacky boosting attack*

2. *The ladder mechanism, and implications for practice*