COMS 6998-4 Fall 2017 Presenter: Qinyao He
November 29, 2017 Scribe: Geelon So

Leaderboards

1 Introduction

When learning algorithms *overfit* to training data, they fail to generalize onto new data; even though they might have achieved low *empirical risk*, they haven't reduced the *true risk*.

Two techniques commonly used to mitigate overfitting are (1) reducing the capacity of the model, and (2) obtaining more training data. But what can we do if both techniques are unavailable, and as a result, the model must train repeatedly on the same data? One recourse is modifying how the model learns from the training data.

Sometimes, we can change the model itself, implementing *resampling methods* such as crossvalidation or bootstrapping. But sometimes, we can change only the feedback that the model receives, and perhaps different feedback can help prevent overfitting.

Feedback often comes in the form of empirical risk—indeed, if the model is independent of the sample, then the empirical risk is an unbiased estimator of the true risk. And so, minimizing empirical risk should also reduce the true risk. However, empirical risk is no longer unbiased in the *adaptive* setting, where an algorithm may produce classifiers that depend on (and thus, adapt to) previous feedback. Here, if empirical risk forms the feedback, then as time goes on, classifiers become ever more prone to overfitting.

So we turn to determining whether we can provide models with more accurate estimators of the true risk. But of course, as the shrewd reader might ask, 'more accurate' with respect to what? To answer this, we'll define the *leaderboard accuracy*, and from there, we can describe the *Ladder mechanism*, which gives learners that more (leaderboard) accurate feedback.

2 Setting

For motivation, let's concretely place ourselves in the *machine learning competition* situation, say, the Kaggle platform. Here, competitors train different learning models on a public dataset before finally being ranked according to their performance on a private test dataset. While the competition is in progress, Kaggle provides an additional public leaderboard test set that approximates the final leaderboard ranking.

Because competitors are allowed multiple submissions, their models can adapt to the feedback the leaderboard ranking provides; thus, the models may become overfitted to this public leaderboard test set. Both the empirical risk feedback and the estimate of the final leaderboard become biased.

Current methods Kaggle uses to combat overfitting include limiting the number of times a competitor can submit a model and limiting the precision of the returned feedback. Still, we hope for theoretical guarantees, so let's formalize this problem.

2.1 Definitions and Notation

As usual, we have an instance space $\mathcal{X}$ with label set $\mathcal{Y}$. There is an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Then for any classifier $f : \mathcal{X} \to \mathcal{Y}$ and loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, we define the *true loss* as

$$R_{\mathcal{D}}(f) := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\ell(f(x), y)\right].$$

Given a collection of n i.i.d. samples drawn from $\mathcal{D}$, say $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$, we can estimate the true loss by the *empirical loss*,

$$R_S(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

The problem at hand: given a sample S of size n and a collection $f_1, \ldots, f_k$ of k classifiers, can we estimate their respective true losses $R_1, \ldots, R_k$? And can we ensure that with probability greater than $1 - \delta$, for all $t \in [k]$,

$$\left|R_t - R_{\mathcal{D}}(f_t)\right| < \epsilon.$$

Exercise 1. *Let $f_1, \ldots, f_k$ be classifiers fixed independently of a sample S of size n. Assuming that loss is bounded between 0 and 1, use Hoeffding's inequality to prove an upper bound on*

$$\Pr\left[\exists t \in [k] : \left|R_S(f_t) - R_{\mathcal{D}}(f_t)\right| > \epsilon\right].$$

Deduce that we can achieve $o(1)$ error on k classifiers using a sample size $O(\log k)$.

Solution. Since the classifiers are fixed independently of S, the empirical loss is an unbiased estimator of the true loss. Thus, we may apply Hoeffding's inequality, which states that

$$\Pr\left[\left|R_S(f_t) - R_{\mathcal{D}}(f_t)\right| \geq \epsilon\right] \leq 2\exp\left(-2\epsilon^2 n\right).$$

Then, union bound over k empirical losses gives us an upper bound $2k\exp(-2\epsilon^2 n)$. Thus, we don't need more than $O(\log k)$ samples to achieve constant probability of success. □

Now, we consider the *adaptive setting*, in which the classifiers are no longer independent of the sample S. In particular, the learning algorithm $\mathcal{A}$ generates new classifiers with respect to previous feedback—let R_t the loss estimator for f_t. Then, for all $t \in [k]$,

$$f_t = \mathcal{A}(f_1, R_1, \ldots, f_{t-1}, R_{t-1}).$$

Here, as [1] writes, "no computationally efficient estimator can achieve error $o(1)$ on more than $n^{2+o(1)}$ adaptively chosen functions" (assuming one-way functions exist).[2, 3]

That is, in the adaptive setting, to achieve $o(1)$ error on k classifiers using a computationally efficient estimator, we'll essentially need at least $\Omega(\sqrt{k})$ samples, which is exponentially worse than the upper bound in the nonadaptive setting of $O(\log k)$.

This suggests that within the adaptive setting, achieving bounds with respect to our usual notion of accuracy, $\left|R_t - R_{\mathcal{D}}(f_t)\right| < \epsilon$, is too hard. Let us define a weaker notion of accuracy with respect to leaderboards. Let R_t estimate the error of the best classifier so far:

$$R_t^{\text{lb}} := \min_{1 \le i \le t} R_{\mathcal{D}}(f_i).$$

Then, we can think of *leaderboard accuracy* as the error of our estimator $\left|R_t^{\text{lb}} - R_t\right|$. And so, if we want to bound $\left|R_t^{\text{lb}} - R_t\right|$ by ϵ for the sequence of estimates, we want

$$\left\| (R_t^{\text{lb}})_{t=1}^k - (R_t)_{t=1}^k \right\|_\infty < \epsilon.$$

Let's define the left-hand side term as the *leaderboard error*. Explicitly,

Definition 2. *Given an adaptively chosen sequence of classifiers $f_1, \ldots, f_k$, the* leaderboard error *of estimates $R_1, \ldots, R_k$ is*

$$\text{lberr}(R_1, \ldots, R_k) := \max_{1 \le t \le k} \left| \min_{1 \le i \le t} R_{\mathcal{D}}(f_i) - R_t \right|.$$

3 The Ladder Mechanism

Here, we'll give an example of a leaderboard that achieves low leaderboard error. In the algorithm below, we'll use the notation $[x]_\eta$ to denote x rounded to the nearest integer multiple of η.

Algorithm 1 Ladder Mechanism

input Data set S, step size parameter $\eta > 0$

Assign initial estimate $R_0 \leftarrow \infty$
for each round $t \leftarrow 1, 2, \ldots$ **do**
  Receive classifier $f_t : \mathcal{X} \to \mathcal{Y}$
  if $R_S(f_t) < R_{t-1} - \eta$ **then**
    Assign $R_t \leftarrow \left[R_S(f_t)\right]_\eta$
  else
    Assign $R_t \leftarrow R_{t-1}$
  end if
  return R_t
end for

Theorem 3. *For any sequence of adaptively chosen classifiers $f_1, \ldots, f_k$, the Ladder Mechanism achieves with high probability*

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

This theorem then implies that in the adaptive setting, we can achieve $o(1)$ leaderboard error provided that the number of classifiers is less than $O\left(\frac{1}{n}\exp(n)\right)$.

To prove this, we'll show something a bit more general:

Lemma 4. *Given the previous conditions,*

$$\Pr\left[\left|\min_{1\le i\le t} R_{\mathcal{D}}(f_i) - R_t\right| > \epsilon + \eta\right] \le \exp\left(-2\epsilon^2 n + (1/\eta + 2)\log(4t/\eta) + 1\right)$$

.

Letting $\eta = O(n^{-1/3}\log^{1/3}(kn))$ immediately obtains Theorem 3.

Proof of lemma. First, notice that the ladder mechanism truncates the feedback to the nearest integer multiple of η. So, the returned value for R_t can take on at most $\left\lceil \frac{1}{\eta} \right\rceil$ different values.

This lets us view an adaptive algorithm $\mathcal{A}$ as just a $\left\lceil \frac{1}{\eta} \right\rceil$-ary decision tree: at each time step, depending on the feedback the learner receives, the learner then presents the next classifier for testing. Recalling

$$f_t = \mathcal{A}(f_1, R_1, \ldots, f_{t-1}, R_{t-1}),$$

the sequence $(R_1, \ldots, R_{t-1})$ corresponds to the path through the decision tree that leads to the node containing f_t. But in fact, the tree $\mathcal{T}$ representing $\mathcal{A}$ is smaller than the full decision tree because the ladder mechanism will never return a R_t that is greater than R_{t-1}.

This means that the total number of possible classifiers that the algorithm can possibly produce is finite, bounded by the number of nodes in the tree $\mathcal{T}$. To this end, we have:

Claim: an upper bound on the number of nodes in $\mathcal{T}$ is $2^{(1/\eta+2)\lg(4k/\eta)}$.

For now, let's assume the claim. And let's call F the collection of classifiers, so $|F| \le |\mathcal{T}|$. Then, we can bound the error of their empirical risks in the same way as in Exercise 1:

$$\Pr\left[\exists f \in F : \left|R_{\mathcal{D}}(f) - R_S(f)\right| > \epsilon\right] \le 2|F|\exp(-2\epsilon^2 n).$$

Substituting the upper bound in and using the fact that $2^x \le e^x$ when x is nonnegative, we obtain a right-hand side that matches that of Theorem 3. And so, with high probability, all estimates $R_S(f)$ are within ϵ of $R_{\mathcal{D}}(f)$.

If we consider the classifier f_{i^*} with the lowest true risk,

$$i^* = \arg\min_i R_{\mathcal{D}}(f_i)$$

it follows that with high probability,

$$\left|R_{\mathcal{D}}(f_{i^*}) - \min_{1\le i\le t} R_S(f_i)\right| \le \left|R_{\mathcal{D}}(f_{i^*}) - R_S(f_{i^*})\right| \le \epsilon.$$

In other words, with probability less than $2|F|\exp(-2\epsilon^2 n)$,

$$\left|\min_{1\le i\le t} R_{\mathcal{D}}(f_i) - \min_{1\le i\le t} R_S(f_i)\right| > \epsilon.$$

Now, as R_t truncates $\min_{1\le i\le t} R_S(f_i)$ to the nearest η, we obtain:

$$\Pr\left[\left|\min_{1\le i\le t} R_{\mathcal{D}}(f_i) - R_t\right| > \epsilon + \eta\right] \le 2|F|\exp(-2\epsilon^2 n),$$

as desired. All that's left to prove is the above claim. □

Exercise 5. *Prove the above claim. Specifically, note that each node at depth t is uniquely determined by the sequence of feedback $(R_1, \ldots, R_{t-1})$, where each of the R_i's can take on at most $\left\lceil \frac{1}{\eta} \right\rceil \le \frac{2}{\eta}$ values. Furthermore, the sequence (R_i) is nonincreasing. How many nodes are there? (Hint: n objects can be encoded using no less than $\lg n$ bits).*

Solution. Let's construct an encoding scheme for the tree. We have k classifiers, so all nodes may be specified by a sequence $(R_1, \ldots, R_k)$ of at most length k. Thus, naively, this encoding scheme requires at most $k \cdot \lg \frac{2}{\eta}$ bits.

However, many bits are wasted because the sequence is monotonic decreasing:

$$R_1 \ge R_2 \ge \cdots \ge R_k.$$

And so, we could just specify the indices i where the sequence is *strictly* monotonic decreasing. That is, $R_i > R_j$. It takes $\lceil \lg k \rceil \le \lg 2k$ bits to encode the index i, and as before, $\lg \frac{2}{\eta}$ bits the corresponding value R_i. There are at most $\frac{1}{\eta}$ such pairs we need to encode, in addition to two more for the head and tail of the path. Thus, the number of required bits B to encode a node is at most

$$B = \left(\frac{1}{\eta} + 2\right)\left(\lg 2k + \lg \frac{2}{\eta}\right).$$

And so, the number of nodes in $\mathcal{T}$ is at most $2^B = 2^{(1/\eta+2)\lg(4k/\eta)}$. □

4 A Parameter-Free Ladder Mechanism

In the Ladder Mechanism algorithm, it is not clear what the choice of η should be. A larger η means that the feedback will be less sensitive to improvements, while a smaller η might pick up on less statistically significant improvements.

If the goal is to report back whenever a classifier has significantly improved over the previous classifier, then it makes sense to increase sensitivity over time, so that as classifiers become more accurate, the value of η decreases as well. The following algorithm implements this heuristic:

Algorithm 2 Parameter-Free Ladder Mechanism

input Data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of size n

Assign initial estimate $R_0 \leftarrow \infty$ and loss vector $l_0 = (0)_{i=1}^n$
for each round $t \leftarrow 1, 2, \ldots$ **do**
  Receive classifier $f_t : \mathcal{X} \to \mathcal{Y}$
  Compute loss vector $l_t \leftarrow (\ell(f_t(x_i), y_i))_{i=1}^n$
  Compute the sample standard deviation $s \leftarrow \mathrm{SD}(l_t - l_{t-1})$
  if $R_S(f_t) < R_{t-1} - s/\sqrt{n}$ **then**
    Assign $R_t \leftarrow \left[R_S(f_t)\right]_{1/n}$
  else
    Assign $R_t \leftarrow R_{t-1}$ and $l_t \leftarrow l_{t-1}$
  end if
  return R_t
end for

5 Boosting Attack

We will now construct a method to overfit to the sample data by incorporating the feedback given by the leaderboard. In particular, let's assume we're in the a binary classification problem with the 0/1 loss. Suppose a learner submits a classification $u \in \{0,1\}^n$ on n objects, while y is the true classification. Then, the loss on the submission is the average loss:

$$\ell(u, y) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{u_i \neq y_i\}.$$

In this scenario, consider the following attack:

1. Pick $u_1, \ldots, u_k \in \{0,1\}^n$ uniformly at random.
2. Observe loss estimates $l_1, \ldots, l_k \in [0,1]$.
3. Let $I = \{i : l_i \leq 1/2\}$ be the collection of indices corresponding to classifications u_i that performed better than random.
4. Output $u^* = \mathrm{maj}(\{u_k : i \in I\})$ the majority vote over these classifiers.

Theorem 6. *Assume that* $\left|l_i - \ell(u_i, y)\right| \leq n^{-1/2}$ *for all* $i \in [k]$. *Then, the boosting attack finds a vector* $u^* \in \{0,1\}^n$ *so that with probability 2/3,*

$$\ell(u^*, y_i) \leq \frac{1}{2} - \Omega\left(\sqrt{\frac{k}{n}}\right).$$

So, if y is chosen uniformly at random, then the leaderboard error is

$$\mathrm{lberr}(R_1, \ldots, R_k) \geq \Omega\left(\sqrt{\frac{k}{n}}\right).$$

6 Discussion

One interesting application of the leaderboard mechanism might be to adapt it into a resampling technique that a learner uses. One problem we would run into, however, is that the feedback given by the leaderboard mechanism reports the empirical risk of the best classifier so far. But, there isn't a way to immediately determine which classifier achieved that best performance. Naturally, one way to try determining this is by using a private dataset, as done in a machine learning competition.

Another interesting point to consider is how the mechanism truncated the feedback. This is very natural in some sense—the sample data only contains a fraction of the information of the whole dataset. Reminiscent of how significant figures are propagated in other sciences, the truncation here reflects that the sample data, as a 'measurement' of the whole data, has limited precision. Perhaps this question could be pursued more information-theoretically.

Bibliographic notes

This lecture followed [1].

References

[1] A. Blum, M. Hardt. The ladder: a reliable leaderboard for machine learning competitions. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 1006–1014, 2015. 54–463. IEEE, 2014.

[2] M. Hardt, J. Ullman. Preventing false discovery in interactive data analysis is hard. In *Proc. 55th Foundations of Computer Science (FOCS)*, pages 4

[3] T. Steinke, J. Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Conference of Learning Theory*, pages 1588–1628, 2015.