CSC 665: Boosting

Chicheng Zhang

December 18, 2019

1 Boosting

Motivation: spam classification

1. Given: emails in the form of text; Goal: find a good classifier that can tell good emails from spam emails.
2. Observation: there are many "rule of thumb" available: e.g. contains "free offer" / "a million dollar" ⇒ spam
3. However: hard to find a single rule that is accurate
4. Boosting: one systematic way of combining "weak" classification rules to strong classification rules.

Theoretical Formulation:

Definition 1 (weak PAC learning)**.** *$\mathcal{A}$ is a γ-weak PAC learner for hypothesis class $\mathcal{H}$, if for any distribution D realizable by $\mathcal{H}$, any $\epsilon \geq \frac{1}{2} - \gamma$, $\mathcal{A}$ produces a classifier h such that with probability $1 - \delta$,*

$$\mathrm{err}(h, D) \leq \epsilon.$$

$\mathcal{H}$ is called γ-weak PAC learnable if there is a γ-weak PAC learner for $\mathcal{H}$.

Note that the difference between weak PAC learning and the regular notion of PAC learning. In weak PAC learning, we only require that the classifier output by the weak learner has an error slightly better than random guessing (50%), as opposed to arbitrary small ϵ.

A brief history of boosting:

1. [Kearns, 1988] - open question: if $\mathcal{H}$ is a weak PAC learnable, is $\mathcal{H}$ also PAC learnable?
2. [Schapire, 1990]: Affirmative answer to the open question with a new technique now known as "boosting". Proposes the first boosting algoithm (by recursion).
3. [Freund, 1990]: Boost by majority algorithm: combining the output of weak learners by a majority vote
4. [Freund and Schapire, 1997]: AdaBoost, an adaptive and practical boosting algorithm (that does not need to know γ)
5. Since then: many more empirical success stories of boosting, e.g. XGBoost [Chen and Guestrin, 2016] is still dominating many ML competitions (e.g. those in Kaggle) as of now.

Algorithm 1 Adaboost

```
Require: Training examples $(x_i, y_i)_{i=1}^m$, weak learner $\mathcal{B}$.
  Initialize distributions over all training examples $(D_1(i))_{i=1}^m$.
  for $t = 1, 2, \ldots, T$: do
    $h_t \leftarrow \mathcal{B}$ trained on weighted examples $((x_i, y_i), D_t(i))_{i=1}^m$.
    Let $\epsilon_t = \sum_{i=1}^m D_t(i)\mathbf{1}(y_i \neq h_t(x_i))$ be the weighted error of $h_t$ on distribution $D_t$, and $\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$.
    Update weighting on training examples: $D_{t+1}(i) = D_t(i)e^{-\alpha_t y_i h_t(x_i)}/Z_t$ where $Z_t$ is a normalizer that ensures $\sum_{i=1}^m D_{t+1}(i) = 1$.
  end for
  Final classifier $H_T(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$.
```

2 AdaBoost: algorithm and analysis

High-level idea: Maintain a weighting on training examples. Repeatedly call weak learner, and adjust the weightings of training examples so that hard examples get emphasized in subsequent training. See Algorithm 1 for a formal description.

We can show that, if at every round of AdaBoost, $\mathcal{B}$ returns a "useful" classifier, in the sense that ϵ_t is slightly better than 0.5 (by a positive "edge" γ), then AdaBoost will bring the training error down exponentially fast.

Theorem 1. *Suppose for every t, $\epsilon_t \leq \frac{1}{2} - \gamma$. Then* $\text{err}(H_T, S) \leq \exp\{-2T\gamma^2\}$.

Proof. Define exponential loss as $\phi(z) = \exp(-z)$. It can be seen that $\phi(z) \geq \mathbf{1}(z \leq 0)$. Denote by $f_s(x) = \sum_{t=1}^s \alpha_t h_t(x)$. Using the notation, $H_T(x) = \text{sign}(f_T(x))$.

Using this relationship, we can upper bound the training error of H_t using its empirical exponential loss:

$$\begin{aligned}\text{err}(H_T, S) &= \frac{1}{m}\sum_{i=1}^m \mathbf{1}(H_T(x_i) \neq y_i) \\ &= \frac{1}{m}\sum_{i=1}^m \mathbf{1}(y_i \cdot f_T(x) \leq 0) \\ &\leq \frac{1}{m}\sum_{i=1}^m \exp\{-y_i f_T(x_i)\}\end{aligned}$$

What do we know about the expoential loss for the i-th example, $\exp\{-y_i f_T(x_i)\}$? In fact it is propotional to $D_{T+1}(i)$. To see this, let us unwrap $D_{T+1}(i)$:

$$\begin{aligned}D_{T+1}(i) &= \frac{D_T(i)e^{-\alpha_T y_i h_T(x_i)}}{Z_T} \\ &= \frac{D_{T-1}(i)e^{-(\alpha_{T-1} y_i h_{T-1}(x_i) + \alpha_T y_i h_T(x_i))}}{Z_{T-1}Z_T} \\ &= \ldots \\ &= \frac{\frac{1}{m}e^{-\sum_{t=1}^T \alpha_t y_i h_t(x_i)}}{\prod_{t=1}^T Z_t} \\ &= \frac{\frac{1}{m}\sum_{i=1}^m \exp\{-y_i f_T(x_i)\}}{\prod_{t=1}^T Z_t}\end{aligned}$$

As $D_{T+1}(i)$ is a distribution over training examples, $\sum_{i=1}^m D_{T+1}(i) = 1$. This implies that the exponential loss, $\frac{1}{m}\sum_{i=1}^m \exp\{-y_i f_T(x_i)\}$, equals $\prod_{t=1}^T Z_t$, the product of the normalization factors at all rounds.

What can we say about each Z_t? Note that

$$\begin{aligned} Z_t &= \sum_{i=1}^m D_t(i) e^{-\alpha_t y_i h_t(x_i)} \\ &= \sum_{i: y_i = h_t(x_i)} D_t(i) e^{-\alpha_t} + \sum_{i: y_i \neq h_t(x_i)} D_t(i) e^{\alpha_t} \\ &= (1-\epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \\ &= (1-\epsilon_t) \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \\ &= 2\sqrt{\epsilon_t(1-\epsilon_t)} \leq \sqrt{1-4\gamma^2} \leq \exp\left\{-2\gamma^2\right\}. \end{aligned}$$

Therefore, $\prod_{t=1}^T Z_t$ is at most $\exp\left\{-2T\gamma^2\right\}$, which concludes that the training error of H_T is at most $\exp\left\{-2T\gamma^2\right\}$.

□

3 Margin bound of Boosting

An intriguing feature of AdaBoost is that, it is "immune" to overfitting. When the number of iterations T increases, one should expect the returned classifier to be more complex - specifically, if at each round, weak learner chooses classifier h_t from some base hypothesis class $\mathcal{H}$, then $H_T(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$ can be seen as coming from the hypothesis class of weighted majority vote over the base classifiers: $\mathcal{H}_T = \left\{\sum_{t=1}^T \alpha_t h_t(x) : \forall t, \alpha_t \in \mathbb{R}, h_t \in \mathcal{H}\right\}$. As T grows, it can be shown that the VC dimension of $\mathcal{H}_T$ also grows (specifically, in the order of $T\,\text{VC}(\mathcal{H})$). By a straightforward application of VC theory, we have that with high probability,

$$\text{err}(H_T, D) \leq \text{err}(H_T, S) + O(\sqrt{\frac{T\,\text{VC}(\mathcal{H})}{m}}).$$

Therefore, according to VC theory, AdaBoost is expected to overfit, as the generalization bound on the right hand side is growing.

However, it is noted in many datasets that as T grows, the generalization error of the classifier output by AdaBoost keeps decreasing, even if training error already reaches zero! What is going on in AdaBoost? To explain the discrepancy between the theory and the experiments, works have shown that similar to SVM, AdaBoost also implicitly performs margin maximization [Schapire et al., 1998]. Moreover, similar to linear classifiers, there is a theory of margin-based generalization error bounds for voting classifiers [Schapire et al., 1998, Breiman et al., 1998, Koltchinskii et al., 2002, Wang et al., 2011].

Theorem 2. *Suppose $\mathcal{H}$ is finite. Define $\text{C}(\mathcal{H}) := \left\{\sum_{h\in\mathcal{H}} \alpha_h h(x) : \sum_{h\in\mathcal{H}} |\alpha_h| \leq B_1\right\}$ be the set of voting classifiers over $\mathcal{H}$. Fix margin value $\gamma \in (0,1]$. Then, with probability $1-\delta$ over the draw of m training examples S, for all predictors f on $\text{C}(\mathcal{H})$,*

$$\mathbb{P}_D(yf(x) \leq 0) \leq \mathbb{P}_S(yf(x) \leq \gamma) + O\left(\frac{B_1}{\gamma}\sqrt{\frac{\ln\frac{|\mathcal{H}|}{\delta}}{m}}\right).$$

Remark. Note the similarity between this bound and the margin bound of linear classification we discussed in the analysis of SVM. In fact this bound can also be viewed as a statement of linear classification: suppose each x is represented by a $d = |\mathcal{H}|$-dimensonal vector $\phi(x) = (h(x))_{\mathcal{H}}$, we can alternatively view the theorem statement as: with probability $1-\delta$: for all w such that $\sum_{i=1}^d |w_i| \leq B_1$,

$$\mathbb{P}_D(y \left\langle w, \phi(x) \right\rangle \leq 0) \leq \mathbb{P}_S(y \left\langle w, \phi(x) \right\rangle \leq \gamma) + O\left(\frac{B_1}{\gamma} \sqrt{\frac{\ln \frac{d}{\delta}}{m}} \right).$$

In light of the above connection, it is not hard to show a slight generalization of Theorem 2 in the context of linear classification, allowing each example to have bounded ℓ_∞ norms.

Theorem 3. *Suppose labeled data distribution D is support on $\left\{x \in \mathbb{R}^d : \|x\|_\infty \leq R_\infty\right\} \times \{\pm 1\}$. Fix margin value $\gamma \in (0,1]$. Then, with probability $1-\delta$ over the draw of m training examples S, for all predictors w such that $\|w\|_1 \leq B_1$,*

$$\mathbb{P}_D(y \left\langle w, x \right\rangle \leq 0) \leq \mathbb{P}_S(y \left\langle w, x \right\rangle \leq \gamma) + O\left(\frac{B_1 R_\infty}{\gamma} \sqrt{\frac{\ln \frac{d}{\delta}}{m}} \right).$$

We usually refer to the Theorem 3 as a ℓ_1(predictor norm)-ℓ_∞ (example norm) margin bound, which is in contrast with the ℓ_2-ℓ_2 margin bound for SVMs (Theorem 2 of the note on SVM). Let us make a detailed comparison here. For notational simplicity, denote by $\mathrm{B}_p(v_0, r) = \left\{v : \|v - v_0\|_p \leq r\right\}$ as the ℓ_p ball centered at v_0 with radius r.

Bound type	Constraint on x	Constraint on w	Generalization error bound
ℓ_1-ℓ_∞	$\mathrm{B}_\infty(0, R_\infty)$	$\mathrm{B}_1(0, B_1)$	$\tilde{O}(\frac{B_1 R_\infty}{\gamma}\sqrt{\frac{1}{m}})$
ℓ_2-ℓ_2	$\mathrm{B}_2(0, R_2)$	$\mathrm{B}_2(0, B_2)$	$\tilde{O}(\frac{B_2 R_2}{\gamma}\sqrt{\frac{1}{m}})$

Ignoring logarithmic factors, both bounds have the common term of $\frac{1}{\gamma}\sqrt{\frac{1}{m}}$. So the key difference lies in the factors $B_2 R_2$ and $B_1 R_\infty$, respectively. Let us consider two settings:

1. Suppose all examples (x, y) are such that $x \in \mathrm{B}_\infty(0, R_\infty)$, and the predictor w of interest satisfies $w \in \mathrm{B}_1(0, B_1)$. We already know that ℓ_1-ℓ_∞ generalization error bound has a factor of $B_1 R_\infty$.

   It is also possible to apply ℓ_2-ℓ_2 generalization bound here. Specifically, the smallest ℓ_2 ball that contains $\mathrm{B}_\infty(0, R_\infty)$ is $\mathrm{B}_2(0, \sqrt{d} R_\infty)$ [1], and the smallest ℓ_2 ball that contains $\mathrm{B}_1(0, B_1)$ is $\mathrm{B}_2(0, B_1)$. This implies that the ℓ_1-ℓ_∞ generalization bound is a factor of $\sqrt{d}$ worse than the ℓ_2-ℓ_2 bound, when applied to this setting.

2. Suppose all examples (x, y) are such that $x \in \mathrm{B}_2(0, R_2)$, and the predictor w of interest satisfies $w \in \mathrm{B}_2(0, B_2)$. We already know that ℓ_2-ℓ_2 generalization error bound has a factor of $B_2 R_2$.

   It is also possible to apply ℓ_1-ℓ_∞ generalization bound here. Specifically, the smallest ℓ_∞ ball that contains $\mathrm{B}_2(0, R_2)$ is $\mathrm{B}_\infty(0, R_2)$, and the smallest ℓ_1 ball that contains $\mathrm{B}_2(0, R_1)$ is $\mathrm{B}_1(0, \sqrt{d} R_1)$. This implies that ℓ_2-ℓ_2 generalization bound is a factor of $\sqrt{d}$ worse than the ℓ_1-ℓ_∞ bound, when applied to this setting.

Proof of Theorem 2. The proof uses the same line of reasoning as the ℓ_2-ℓ_2 margin bound. We can show that (with details left to the reader) with probability $1-\delta$, for all f in $\mathrm{C}(\mathcal{H})$,

$$\mathbb{P}_D(yf(x) \leq 0) \leq \mathbb{P}_S(yf(x) \leq \gamma) + \sqrt{\frac{\ln \frac{2}{\delta}}{m}} + \frac{2}{\gamma} \mathbb{E}\, \mathrm{Rad}_S(\mathcal{F}). \tag{1}$$

[1] To see this, draw a picture of a ℓ_∞ ball in 2d, and try to grow a ℓ_2 ball that is just enough to encapsulate it.

Here $\mathcal{F}$ is the class of margin functions, each induced by one weighting over the base hypothesis class $\mathcal{H}$:

$$\mathcal{F} = \left\{ m_\alpha : \|\alpha\|_1 \le B_1 \right\},$$

where

$$m_\alpha(x, y) = y \sum_{h \in \mathcal{H}} \alpha_h h(x).$$

We now bound the empirical Rademacher complexity $\mathrm{Rad}_S(\mathcal{F})$ for dataset S differently:

$$\begin{aligned}
\mathrm{Rad}_S(\mathcal{F}) &= \frac{1}{m} \mathbb{E}_\sigma \sup_{\alpha : \|\alpha\|_1 \le B_1} \sum_{i=1}^m \sigma_i y_i \left(\sum_{h \in \mathcal{H}} \alpha_h h(x_i) \right) \\
&= \frac{1}{m} \mathbb{E}_\sigma \sup_{\alpha : \|\alpha\|_1 \le B_1} \sum_{i=1}^m \sigma_i \left(\sum_{h \in \mathcal{H}} \alpha_h h(x_i) \right) \\
&= \frac{1}{m} \mathbb{E}_\sigma \sup_{\alpha : \|\alpha\|_1 \le B_1} \sum_{h \in \mathcal{H}} \alpha_h \sum_{i=1}^m \sigma_i h(x_i) \\
&= \frac{B_1}{m} \mathbb{E}_\sigma \sup_{h \in \mathcal{H}} \left| \sum_{i=1}^m \sigma_i h(x_i) \right| .
\end{aligned}$$

But we have seen this before! This is the Rademacher complexity (with absolute value sign) of class $\mathcal{H}$. As seen before, by Massart's Lemma, the above is at most

$$\frac{1}{m} \sqrt{m \ln\bigl(2|\mathcal{H}|\bigr)} = \sqrt{\frac{\ln\bigl(2|\mathcal{H}|\bigr)}{m}}.$$

The proof is concluded by combining the above fact with Equation (1), along with simple algebra. □

References

Leo Breiman et al. Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849, 1998.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

Y Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, 1990.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Michael Kearns. Thoughts on hypothesis boosting. 1988.

Vladimir Koltchinskii, Dmitry Panchenko, et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

Liwei Wang, Masashi Sugiyama, Zhaoxiang Jing, Cheng Yang, Zhi-Hua Zhou, and Jufu Feng. A refined margin analysis for boosting algorithms via equilibrium margin. *Journal of Machine Learning Research*, 12(Jun):1835–1863, 2011.