CSE 522: Algorithms and Uncertainty Spring 2017

Lecture 1 — March 27, 2017

Lecturer: Anna R. Karlin

These rough notes follow lectures and notes by Avrim Blum, in some parts, nearly verbatim. However, any errors are mine.

1 PAC Model

We are given a training set $S = \{(x_i, y_i),\ 1 \le i \le m\}$, where each x_i is an instance in some space X, e.g. a feature vector over $\mathbb{R}^d$, and y_i is a binary label (classification). For example, x_i could be a set of features of an email message and y could be a label indicating whether it is spam or not. We assume that

$$y_i = f(x_i) \quad \text{where} \quad f : X \to \{0, 1\}$$

is the correct labeling of the message, i.e., the ground truth. We also assume that each x_i is drawn independently from some distribution $\mathcal{D}$ over the instance space.

Definition 1.1. A **learning algorithm** takes as input a training set S whose elements are sampled i.i.d. from some $\mathcal{D}$ over X and produces as output a hypothesis $h_S \in \mathcal{H}$, where $h : X \to \{0, 1\}$.

Definition 1.2. For a sample $S = \{(x_i, y_i), i = 1 \dots m\}$ and hypothesis h, define the **training error** (also sometimes called the empirical error or empirical risk)

$$\mathrm{err}_S(h) \triangleq \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{h(x_i) \neq y_i}.$$

The typical learning algorithm will find a hypothesis with minimum training error, that is,

$$h_S := \mathrm{argmin}_{h \in \mathcal{H}} \mathrm{err}_S(h).$$

This is called **empirical risk minimization**.

Our goal is to understand how big S needs to be so that the empirical risk minimizer h_S is very likely to satisfy $h_S(x) = f(x)$. Formally,

Definition 1.3. The **generalization error** $\mathrm{err}_D(h)$ of a hypothesis h with respect to a distribution $\mathcal{D}$ is

$$\mathrm{err}_D(h) \triangleq \mathbb{P}_{X \sim D}\left[h(X) \neq f(X)\right].$$

Remark 1.4. Clearly, if we are not careful, e.g., allow $\mathcal{H}$ to contain all possible functions, it's easy to get small training error. But then we are very likely to have large generalization error. This is called **overfitting**.

Definition 1.5. Suppose that $\mathcal{H}$ is **realizable**, that is, $f(\cdot) \in \mathcal{H}$. We say that the hypothesis class $\mathcal{H}$ is **PAC-learnable** if there is a function $m_{\mathcal{H}} : (0, 1)^2 \to \mathbb{N}$ and a learning algorithm such that

for any $\epsilon, \delta > 0$ and $\mathcal{D}$, given a random sample S of size at least $m_{\mathcal{H}}(\epsilon, \delta)$ of correctly labeled data, the algorithm produces a hypothesis $h_S \in \mathcal{H}$ s.t.

$$\mathbb{P}\left[\text{err}_D(h_S) < \epsilon\right] \geq 1 - \delta.$$

In other words, the hypothesis output is **probably** (with probability $1 - \delta$) **approximately correct**(errs with probability at most ϵ on new samples).

In class, we showed that for $\mathcal{H} = \{\text{decision lists over } n \text{ boolean vars}\}$, with

$$m_{\mathcal{H}}(\epsilon, \delta) \approx \frac{1}{\epsilon}\left(n \ln n + \ln\left(\frac{1}{\delta}\right)\right)$$

labeled examples, and in time polynomial in n and $m_{\mathcal{H}}(\epsilon, \delta)$, we are able to find a consistent DL such that with probability at least $1 - \delta$, the generalization error is at most ϵ.

Remark 1.6. Often when people speak about $\mathcal{H}$ being PAC-learnable, they also require that the algorithm for finding a consistent hypothesis runs in time polynomial in $\epsilon^{-1}, \delta^{-1}$, the size of each example (e.g. n for the spam example from class), and the size of the representation of a function in $\mathcal{H}$. In our discussion we aren't going to worry too much about the running time of finding a consistent hypothesis or the best hypothesis in the class. We'll focus on trying to understand the **sample complexity** – how much data is needed to get a certain confidence bound. Often it's the training data that is expensive to get.

What follows is the most basic sample complexity bound.

Theorem 1.7. *Let* $|S| = m$. *If* $m \geq \frac{1}{\epsilon}\ln\left(\frac{|\mathcal{H}|}{\delta}\right)$, *then with probability at least* $1 - \delta$, *all* $h \in \mathcal{H}$ *with* $\text{err}_S(h) = 0$ *have* $\text{err}_D(h) < \epsilon$.

Proof. Suppose that $h \in \mathcal{H}$ and $\text{err}_D(h) \geq \epsilon$. Then

$$\mathbb{P}\left[\text{err}_S(h) = 0\right] \leq (1 - \epsilon)^m.$$

Therefore, using a union bound,

$$\mathbb{P}\left[\exists h \in H \text{ s.t. } \text{err}_D(h) \geq \epsilon \text{ and } \text{err}_S(h) = 0\right] \leq |\mathcal{H}|(1 - \epsilon)^m \leq |\mathcal{H}|e^{-\epsilon m}.$$

Solving for $|\mathcal{H}|e^{-\epsilon m} \leq \delta$ gives the bound. □

1.1 Beyond the realizable setting: uniform convergence

The last result only bounds the chance that a bad hypothesis looks perfect on the data. What if there is no perfect $h \in \mathcal{H}$? This could happen for 2 reasons: First, the correct labeling may not be in the set $\mathcal{H}$. Second, it may not even be possible to label correctly given the features. For example, suppose we are trying to construct a classifier for determining if a particular person has a particular disease based on various medical indicators like blood pressure, temperature, etc. Then it is unlikely that these indicators are sufficient to uniquely determine if the person has the disease or not.

This motivates the notion of "uniform convergence". Here we try to determine how many samples are needed to guarantee that all $h \in \mathcal{H}$ satisfy $|\mathrm{err}_D(h) - \mathrm{err}_S(h)| \leq \epsilon$ with high probability? If we can show that such a statement holds, then if we are so lucky as to find a hypothesis with small training error, we can be confident that it has low generalization error as well. This motivates optimizing over S, even if we can't find a perfect function. To prove bounds like this, use tail inequalities (see §3).

Theorem 1.8. *If $|S| \geq \frac{1}{2\epsilon^2} \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)$, then with probability at least $1-\delta$, all $h \in \mathcal{H}$ have $|\mathrm{err}_D(h) - \mathrm{err}_S(h)| < \epsilon$.*

Proof. Fix $h \in \mathcal{H}$ and suppose that $\mathrm{err}_D(h) = p$. Then by Hoeffding (Theorem 3.1), we have

$$\mathbb{P}\left[|\mathrm{err}_S(h) - \mathrm{err}_D(h)| \geq \epsilon\right] \leq 2e^{-2|S|\epsilon^2}.$$

Set to δ and solve. □

Remark 1.9. This is worse than previous bound ϵ^{-1} has become ϵ^{-2} because we are asking for something stronger.

1.2 Occam's Razor

Occam's razor is the notion, stated by William of Occam around 1320, that one should prefer simpler explanations over more complicated ones. The theorems we just saw give a mathematical sense in which this is true.

One way we could say that explanations of a certain type are simple is by saying that any explanation in this class can be described with few bits. Since there are at most 2^b rules that can be described with fewer than b bits, we can say

Theorem 1.10. *(Occam's Razor) Fix a description language for rules and consider a training set S from distribution $\mathcal{D}$. Then with probability $1-\delta$, any rule h consistent with S that can be described in this language using fewer than b bits will have $\mathrm{err}_D(h) \leq \epsilon$ for $|S| = \epsilon^{-1}(b\ln(2/\delta))$. Alternatively, with probability at least $1-\delta$, all rules that can be described with less than b bits have*

$$\mathrm{err}_D(h) \leq \frac{b\ln 2 + \ln(\delta^{-1})}{|S|}.$$

1.3 Notes about PAC framework

- This is a distribution-free (prior-independent) model: no assumptions are made about the distribution from which the training examples are drawn.
- The distribution independence is "mitigated" by the fact that the training examples and the examples used to define generalization error are drawn from the same distribution.
- The PAC framework deals with learnability of a hypothesis class and not a particular hypothesis. The algorithm knows the class is is trying to optimize over, but not the particular hypothesis we are looking for.

2 Improving the measure of complexity for a hypothesis class

So far, we used $\ln(|\mathcal{H}|)$ as a measure of complexity for a hypothesis class (i.e., bound on sample complexity). But this is useless when $|\mathcal{H}|$ is infinite, and even when it is finite, there are tighter measures. Again, our question is: how big does S have to be so that whp, $\text{err}_S(h) = 0$ implies that $\text{err}_D(h) \leq \epsilon$.

Example 2.1. Axis-aligned rectangles in $\mathbb{R}^2$:

$$\mathcal{H} = \{R(x, y) = \mathbb{1}_{x_1 \leq x \leq x_2, y_1 \leq y \leq y_2} \text{ for some } x_1, x_2, y_1 y_2 \in \mathbb{R}\}.$$

Suppose that we are in the realizable case, i.e., there is a true rectangle R^* that precisely defines the positive examples. Say we see a sample of size m. How should our learning algorithm work? One possibility is to find the smallest bounding rectangle for the positive examples. Let's call this rectangle R. What is the generalization error after seeing m samples? Clearly, there are no no false positives. But there could be false negatives. Let's analyze the probability that $\text{err}_D(R) > \epsilon$. This is precisely the probability of getting an example in $R^* \setminus R$.

First, we observe that if the probability of R^* under $\mathcal{D}$ is less than ϵ then $\text{err}_D(R) < \epsilon$. Next consider four subrectangles of R^*, the minimal rightmost, leftmost, topmost and bottom-most rectangles inside R^*, for which the probability of a sample under $\mathcal{D}$ is at least $\epsilon/4$. If S contains an example within each of these rectangles, then $\mathbb{P}[R^* \setminus R] \leq \epsilon$. Bad case is if it doesn't. For each subrectangle, probability of no samples inside there is at most $(1 - \epsilon/4)^m$. So the probability there is a subrectangle with no sample inside is at most $4(1 - \epsilon/4)^m$. Setting this to be at most δ and solving for m shows that

3 Tail Bounds

Theorem 3.1 (Hoeffding bounds)**.** *Let $X \sim Bin(m, p)$, $\epsilon \in [0, 1]$. Then*

$$\mathbb{P}\left[\frac{X}{m} > p + \epsilon\right] \leq e^{-2m\epsilon^2} \quad \text{and} \quad \mathbb{P}\left[\frac{X}{m} < p - \epsilon\right] \leq e^{-2m\epsilon^2}.$$

Theorem 3.2 (Chernoff bounds)**.** *Let $X \sim Bin(m, p)$, $\alpha \in [0, 1]$. Then*

$$\mathbb{P}\left[\frac{X}{m} > p(1 + \alpha)\right] \leq e^{-mp\alpha^2/3} \quad \text{and} \quad \mathbb{P}\left[\frac{X}{m} < p(1 - \alpha)\right] \leq e^{-mp\alpha^2/2}.$$

4 Notes

PAC learning was introduced by Valiant [4] in 1984. For detailed expositions of this material, see Kearns and Vazirani [1] (chapters 1 and 2) Shalev-Schwartz and Ben-David [3] (chapters 2-4) and Mohri, Rostamizadeh and Talwalkar [2] (chapter 2).

References

[1] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory.* MIT press, 1994.

[2] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[3] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[4] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.