ECE 901: Large-scale Machine Learning and Optimization **Spring 2018**

Lecture 1 — January 25

Lecturer: Dimitris Papailiopoulos *Scribe: Hongyi Wang* × *Harrison Rosenberg*

Note: These lecture notes are still rough, and have only have been mildly proofread.

ML research is multi-disciplinary, combining high-dimensional statistics, algorithms, and optimization.

1.0.1 Some Definitions

The Loss Function, $L(*)$ measures difference between the "correctness" of model predictions and reality. For simplicity, we will assume the loss function always evaluates to a number between zero (0) and one (1).

Training Data:

$$\mathbf{S} = \{\mathbf{z}_1, \dots \mathbf{z}_n\} \tag{1.1}$$

Each Element $\mathbf{z}_i$ in $\mathbf{S}$ represents a tuple containing a set of features AND a label drawn from an unknown distribution $\mathcal{D}$. A machine learning algorithm learns a model from the training data

Empirical Risk Minimization:

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i) \tag{1.2}$$

Algorithms train on training data, they seek to find the model which minimizes the Empirical Risk – (or training error). This problem is not guaranteed to be convex (i.e. neural networks), but can be (Ridge Regression). Theory is somewhat developed for when $\mathbf{S} \overset{iid}{\sim} \mathcal{D}$

Below **true risk** is defined:

$$\mathcal{R}(h_s) = \mathbb{E}_{\mathbf{S}}[L(h_s(\mathbf{x}, \mathbf{y})] \tag{1.3}$$

It almost always impossible to evaluate this quantity, as the distribution $\mathcal{D}$ is unknown, hence ERM is used to generate a classifier

More key terms you should be familiar with from past ML courses: training, validation, test, cross-validation, hold-out set. Please review them.

This lecture focuses on the following questions:

- When is Empirical Risk Minimization a good estimation for true risk (does ERM concentrate about the true risk)?
- How does the choice of the model affect the concentration of the empirical risk?

TL;DR: find a hypothesis $h_{\mathcal{S}} \in \mathcal{H}$ with small True Risk (Equation (1.3))

1.0.2 Generalization

The generalization of a hypothesis h is a function of the following:

•$\mathbf{S}$ • n • $\mathcal{H}$ • $\mathcal{D}$ • Training Algorithm

One way to determine generalization is PAC learning, often associated with **Hoeffding Inequality**. In the interest of avoiding redundancy, it is defined in the linked Wikipedia page. The inequality is rather powerful because you do not need to know much about $\mathbf{x}$. You just need to know if the distribution is sub-Gaussian! That being said, you only get a bound on your estimation error, not approximation error. Which is suboptimal. Another concentration inequality of interest is Bernstein's inequality, which provides a bound on the deviation from the mean.

Hoeffding's Inequality can be used to answer questions such as: "How many samples do I need to guarantee that $\mathbf{S}_n = \mathbb{E}[S_n] \pm \epsilon$ with probability $1 - \delta$?"

For example:

$$\delta = 2e^{-n\epsilon^2} \implies n = \mathcal{O}\left(\frac{(\log(\frac{1}{\delta}))}{\epsilon^2}\right)$$

A set of important assumptions to use Hoeffding's Inequality:

- $h \in \mathcal{H}$ is independent of $\mathbf{S}$.
- $\mathcal{R}_i[h] = L(h(\mathbf{x}_i, y_i))$ ▷ $\mathcal{R}_i$ is true risk of each predictor h trained on the i.i.d. samples of $\mathbf{S}$.
- $\hat{\mathcal{R}}_{\mathbf{S}}[h] = \frac{1}{n}\sum_{i} \mathcal{R}_i[h]$ ▷ The empirical risk of each predictor h as calculated on $\mathbf{S}$

Then, by *Hoeffding Inequality* we get:

$$\mathbb{P}(|\hat{\mathcal{R}}_{\mathbf{S}}[h] - \underbrace{\mathbb{E}[\hat{\mathcal{R}}_{\mathbf{S}}[h]]}_{True\ \ Risk}| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

From Hoeffdings, we see the empirical risk "converges" to true risk $\sim \frac{1}{\sqrt{n}}$

What if $|\mathcal{H}| < \infty$? (it usually is for our purposes. Numerical representations are limited by bits, i.e. 64 bit double)

Example: Say $\mathcal{H}$ consists of all binary linear classifiers, $\text{sign}(\mathbf{w}^\top \mathbf{x} + b) = y,\ \mathbf{w} \in \{0,1\}^d$ ($|\mathcal{H}| = 2^d$). How can we bound the concentration of $\mathcal{H}$?

Union Bound:

$$\mathbb{P}(\cup_i A_i) \leq \sum_i \mathbb{P}\{A_i\}$$

For binary linear classifiers, $n = \mathcal{O}(\frac{d - \log(\delta)}{\epsilon^2})$

These bounds derived from Hoeffding's inequality are oblivious to the algorithm! Only the predictions matter!