**ECE 901: Large-scale Machine Learning and Optimization** **Fall 2016**

Lecture 3 — 09/13

*Lecturer: Dimitris Papailiopoulos* *Scribe: Ananth Sridhar, Ashwin Varadarajan*

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

## 3.1 Review

- The Empirical Risk concentrates for **finite hypothesis classes**
  - *Floating Point Assumption:*

$$n_{\text{samples}} = O\left(\frac{n_{\text{parameters}}}{\epsilon^2}\right) \tag{3.1}$$

- Empirical risk and true risk (concentration). If, $h$ is a predictor,

$$\hat{R}[h] \leq R[h] + \epsilon \tag{3.2}$$

- Earlier, we assumed that the cardinality of the hypothesis class H is finite. Can we handle infinite hypothesis classes? Yes, using the VC-dimension (section 3.2).

## 3.2 VC dimension

- From Vapnik-Chervonenkis Theory [1971]
- The VC-dimension expresses the predictive power of a classifier
- In the general case, finding the VC-dimension of a hypothesis class is not trivial, and we only know the VC-dimensions for a handful of hypothesis classes. However, there are a few special cases of hypothesis classes where the VC-dimension is easy to calculate (see example). Also, we can always bound the VC-dimension if the Hypothesis class is a finite set, as in (eqn 3.3).

If $\mathcal{H}$ is the hypothesis class,

$$\text{VC}_{\text{dimension}}[\mathcal{H}] \leq \log|\mathcal{H}| \tag{3.3}$$

The above inequality implies that the VC dimension is a stronger notion of complexity than simply the number of predictors.

**Example:** if $h$ is a predictor, and $\mathcal{H}$ is the hypothesis class

$$\text{If,} \quad h(x_i) = y_i \quad \forall \quad i \in \{1 \dots n\}, \quad \text{for any } n$$
$$\text{VC}_{\text{dimension}}[\mathcal{H}] = \infty$$

$$\text{If,} \quad \mathcal{H} = \{\text{All Hyperplanes in } d \text{ dimensions}\}$$
$$\text{VC}_{\text{dimension}}[\mathcal{H}] = d + 1$$

## 3.3 Concentration of Empirical Risk

The following holds with probability $1 - \delta$, for finite and infinite hypothesis classes.

$$\sup_{h \in \mathcal{H}} \left| \hat{R}_n[h] - R[h] \right| \leq O\left( \sqrt{\frac{\text{VC}_{\text{dimension}}[\mathcal{H}] \log\left(\frac{n}{\text{VC}_{\text{dimension}}[\mathcal{H}]}\right) + \log \frac{1}{\delta}}{n}} \right) \tag{3.4}$$

### 3.3.1 Concentration of Empirical Risk for ERM

$$\hat{h}^* \in \arg\min_{h \in \mathcal{H}} \quad \hat{R}_n[h] \qquad \text{(Empirical Risk Minimizer)} \tag{3.5}$$
$$h^* \in \arg\min_{h \in \mathcal{H}} \quad R[h] \qquad \text{(True Risk Minimizer)} \tag{3.6}$$
$$R[\hat{h}^*] - R[h^*] = \quad ? \qquad \text{(How close are we?)} \tag{3.7}$$

For the purposes of the lectures, we will assume that the set containment is an equality, that is, minimizers are unique.

**Theorem 3.1.** *The empirical risk of the empirical risk minimizer concentrates around the true risk of the true risk minimizer*

**Proof:**

$$R[\hat{h}^*] = \begin{cases} R[\hat{h}^*] + (R[h^*] - R[h^*]) \\ \qquad + \left(\hat{R}_n[h^*] - \hat{R}_n[h^*]\right) \\ \qquad + \left(\hat{R}_n[\hat{h}^*] - \hat{R}_n[\hat{h}^*]\right) \end{cases} \tag{3.8}$$

$$R[\hat{h}^*] - R[h^*] = \begin{cases} \left(\hat{R}_n[h^*] - R[h^*]\right) & \text{concentrates: } \leq \epsilon \\ + \left(R[\hat{h}^*] - \hat{R}_n[\hat{h}^*]\right) & \text{concentrates: } \leq \epsilon \\ + \left(\hat{R}_n[\hat{h}^*] - \hat{R}_n[h^*]\right) & \text{concentrates: } \leq 0 \end{cases} \tag{3.9}$$

$$R[\hat{h}^*] - R[h^*] \leq 2\epsilon \tag{3.10}$$

□

## 3.4 Examples of learning problems

The following are some typical classes of problems in Machine Learning:

### 3.4.1 Linear Regression

Fitting a line or hyperplane to data points so as to minimize the sum of distance of predicted labels to the observed ones. Involves minimizing the norm:

$$\min \quad \frac{1}{n}\sum_{i=1}^{n}(x_i^T w - y_i)^2 \tag{3.11}$$

### 3.4.2 Binary Classification

Assigning binary labels to data, hence the variable $y$ is discrete and $y|x$ is a Bernoulli distribution. Involves solving the logistic regression:

$$\min \quad \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i x_i^T w)) \tag{3.12}$$

### 3.4.3 Support Vector Machines

Involves minimizing the maximum margin from a classifier, with a regularization term.

$$\min \quad \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i x_i^T w) + \lambda\|w\| \tag{3.13}$$

### 3.4.4 Feed Forward Neural Network

Each layer involves a linear sum of weighted features, passed through an activation function $\sigma$, which is usually a step function or a smooth version of it, such as $\sigma(x) = -1/(1+exp(-x))$. This can be repeated over a number of hidden layers between the input and the output layers (deep networks).

## 3.5 Useful properties for loss functions

In general, the problem $\min_x f(x)$ is hard to solve as is, and hence we use additional properties of the loss function $f$ to speed up computation.

### 3.5.1 Convex Functions

Any local minimum of the function is also a global minimum.

$$\begin{gathered} f(\alpha\vec{x} + (1-\alpha)\vec{y}) \leq \alpha f(\vec{x}) + (1-\alpha) f(\vec{y}) \\ \forall \alpha \in [0,1], \quad \vec{x}, \vec{y} \end{gathered} \tag{3.14}$$

### 3.5.2 L-Lipschitz functions

The function does not change too fast.

$$|f(\vec{x}) - f(\vec{y})| \leq L||\vec{x} - \vec{y}|| \tag{3.15}$$

### 3.5.3 $\beta$-smooth functions

The function has a Lipschitz gradient.

$$||\nabla f(\vec{x}) - \nabla f(\vec{y})|| \leq \beta||\vec{x} - \vec{y}|| \tag{3.16}$$

### 3.5.4 $\lambda$-strongly convex functions

The function has a unique global minimum.

$$\begin{gathered} f(a\vec{x} + (1-a)\vec{y}) \leq a f(\vec{x}) + (1-a) f(\vec{y}) - \frac{\lambda}{2} a(1-a)||\vec{x} - \vec{y}||^2 \\ \forall a \in [0,1], \quad \vec{x}, \vec{y} \end{gathered} \tag{3.17}$$

In other words, $f$ is $\lambda$-strongly convex **iff**

$$f(\vec{x}) - \lambda||\vec{x}||^2 \quad \text{is convex} \tag{3.18}$$

This property is especially nice, since we can achieve this for ERM by regularization. Strong convexity gives improved algorithmic convergence rates (the algorithms make use of the regularization for faster convergence). Note: All convex functions are 0-strongly convex (trivial case).

## 3.6 Additional Reading

- A nice reference on the VC-dimension (explained with an example): Link
- An additional resource for insights into strong convexity and some settings where it is utilized: Link