**STAT 425: Introduction to Nonparametric Statistics** **Winter 2018**

# Lecture 16: Learning Theory: Empirical Risk Minimization

*Instructor: Yen-Chi Chen*

## 16.1 Introduction

Recall that in (binary) classification problem, we observe

$$(X_1, Y_1), \cdots, (X_n, Y_n)$$

that are IID from some distribution $P$ and each $Y_i$ is a binary class label, i.e. $Y = 0$ or 1 and each $X_i \in \mathcal{X} \subset \mathbb{R}^d$ contains $d$ variables/features/covariates.

A classifier $c : \mathcal{X} \mapsto \{0, 1\}$ is a function such that the input is the feature of a new observation (without label) and then the output is a predicted class label.

To measure the quality of our classifier, we use a loss function $L(c(x), y)$ and a common loss function is the $0-1$ loss, which is $L(c(x), y) = I(c(x) \neq y)$. Note that the MLE method for the logistic regression uses another loss function. The expected loss is the risk function

$$R(c) = \mathbb{E}(L(c(X), Y)),$$

where $(X, Y) \sim P$. $R(c)$ is the expected loss for the future prediction.

Let $\mathcal{C}$ be a collection of classifiers, we want to find the one $c^* \in \mathcal{C}$ such that the risk function is minimized, i.e.,

$$c^* = \mathsf{argmin}_{c \in \mathcal{C}} R(c).$$

This classifier is called the Bayes classifier and it it is the one with the best predictive performance for the future data.

Because $R(c)$ is an unknown quantity, so a sample analogue is

$$\widehat{R}_n(c) = \frac{1}{n} \sum_{i=1}^{n} L(c(X_i), Y_i).$$

$\widehat{R}_n(c)$ is called the **empirical risk**. By the law of large number, $\widehat{R}_n(c) - R(c) = o_P(1)$ for any given $c$.

The **empirical risk minimization (ERM)** is to find the classifier $c^*$ by minimizing the empirical risk

$$\widehat{c} = \mathsf{argmin}_{c \in \mathcal{C}} \widehat{R}_n(c). \tag{16.1}$$

As we have seen in the past few lectures, the ERM may not work because we not only need $\widehat{R}_n(c) - R(c) \approx 0$ for a given $c$ but also $\widehat{R}_n(c) - R(c) \approx 0$ *uniformly for all* $c \in \mathcal{C}$. Namely, we need

$$\sup_{c \in \mathcal{C}} |\widehat{R}_n(c) - R(c)| = o_P(1). \tag{16.2}$$

When $\mathcal{C}$ contains only finite number of classifiers, say $N$ classifiers, then Hoeffding's inequality shows that

$$\sup_{c \in \mathcal{C}} |\widehat{R}_n(c) - R(c)| = O_P\left(\sqrt{\frac{\log N}{n}}\right),$$

so as long as $N$ is not too large compared to the sample size $n$, we eventually have (16.2).

Even when $\mathcal{C}$ contains infinite number of classifiers, as long as its VC dimension is not too large, the VC theory tells us that

$$\sup_{c\in\mathcal{C}} |\widehat{R}_n(c) - R(c)| = O_P\left(\sqrt{\nu \frac{\log n}{n}}\right),$$

where $\nu$ is the VC dimension of $\mathcal{C}$ (which is often a fixed number).

## 16.2 Excess Risk

Recall that the **excess risk** of a classifier $c$ is defined as

$$\mathcal{E}(c) = R(c) - R(c^*) = R(c) - \min_{c\in\mathcal{C}} R(c).$$

Namely, the excess risk describes the amount of decreased performance of a classifier $c$ compared to the Bayes classifier $c^*$.

If a classifier $\widehat{c}_n$ is from the ERM (equation (16.1)), how will its excess risk be like? It turns out that we can control the excess risk using the uniform error bound

$$\epsilon_n = \sup_{c\in\mathcal{C}} |\widehat{R}_n(c) - R(c)|.$$

Because $\widehat{c}$ is the minimizer of $\widehat{R}_n(c)$,

$$\widehat{R}_n(\widehat{c}) \le \widehat{R}_n(c)$$

for any classifier $c \in \mathcal{C}$, including $c^*$. Thus,

$$\widehat{R}_n(\widehat{c}) \le \widehat{R}_n(c^*).$$

Because $\epsilon_n$ is the uniform bound, which implies

$$|\widehat{R}_n(\widehat{c}) - R(\widehat{c})| \le \epsilon_n, \quad |\widehat{R}_n(c^*) - R(c^*)| \le \epsilon_n.$$

This further implies

$$\begin{aligned} R(\widehat{c}) &\le \widehat{R}_n(\widehat{c}) + \epsilon_n \\ &\le \underbrace{\widehat{R}_n(c^*)}_{\le R(c^*)+\epsilon_n} + \epsilon_n \\ &\le R(c^*) + 2\epsilon_n \end{aligned}$$

and

$$\mathcal{E}(\widehat{c}) = R(\widehat{c}) - R(c^*) \le 2\epsilon_n = 2 \times \sup_{c\in\mathcal{C}} |\widehat{R}_n(c) - R(c)|.$$

Namely, the excess risk of a classifier from ERM is no more than 2 times the uniform error.

So when the uniform error $\epsilon_n$ converges to 0, the excess risk converges to 0, implying that the classifier $\widehat{c}$ is as good as the Bayes classifier.

Moreover, the distribution of $\epsilon_n$ can be used to construct a confidence interval for the Bayes risk $R(c^*) = \min_{c\in\mathcal{C}} R(c)$. Let $t_{1-\alpha}$ be the $1-\alpha$ upper quantile of $\epsilon_n$, i.e,

$$P(\epsilon_n \le t_{1-\alpha}) = 1 - \alpha.$$

Recall that to construct a CI, we often need a lower bound and an upper bound.

Because

$$\widehat{R}_n(\widehat{c}) \le \widehat{R}_n(c^*) \le R(c^*) + \epsilon_n,$$

a lower bound of $R(c^*)$ can be obtained by

$$\widehat{R}_n(\widehat{c}) - \epsilon_n \le R(c^*).$$

For the upper bound,

$$R(c^*) \le R(\widehat{c}) \le \widehat{R}_n(\widehat{c}) + \epsilon_n.$$

Thus, we conclude that

$$\widehat{R}_n(\widehat{c}) - \epsilon_n \le R(c^*) \le \widehat{R}_n(\widehat{c}) + \epsilon_n.$$

Namely,

$$|R(c^*) - \widehat{R}_n(\widehat{c})| \le \epsilon_n.$$

Thus, a $1-\alpha$ CI of the Bayes risk $R(c^*)$ is

$$\left[\widehat{R}_n(\widehat{c}) - t_{1-\alpha}, \widehat{R}_n(\widehat{c}) + t_{1-\alpha}\right].$$

## 16.3 Data Splitting

After finding a classifier $\widehat{c}$ from the ERM, we may be interested in the actual performance of this estimator $R(\widehat{c})$. The empirical $\widehat{R}_n(\widehat{c})$ may be misleading because we choose $\widehat{c}$ by minimizing $\widehat{R}_n(\cdot)$ so in most cases, we will underestimate the actual risk $R(\widehat{c})$. Although the uniform bound $\epsilon_n$ implies that $R(\widehat{c})$ cannot be larger than $\widehat{R}_n(\widehat{c}) + \epsilon_n$, the upper bound $\widehat{R}_n(\widehat{c}) + \epsilon_n$ may still be quiet large so that it is not useful in practice.

Is there any way we can get a better estimate of $R(\widehat{c})$?

In some special case, the answer is yes. Now assume that after training our classifier $\widehat{c}$, somehow we have a set of new data that are IID from the same population and are independent of the original data. Let the new data be

$$(X_1^*, Y_1^*), \cdots, (X_m^*, Y_m^*).$$

Then we can use the new data to evaluate the performance of our classifier, i.e.

$$\widehat{R}_m^*(\widehat{c}) = \frac{1}{m}\sum_{j=1}^{m} L(\widehat{c}(X_j^*), Y_j^*).$$

Because now the classifier $\widehat{c}$ is fixed and the new data is independent of the classifier $\widehat{c}$,

$$\widehat{R}_m^*(\widehat{c}) - R(\widehat{c}) = O_P\left(\sqrt{\frac{1}{m}}\right)$$

and $\mathbb{E}(\widehat{R}_m^*(\widehat{c})|\widehat{c}) = R(\widehat{c})$, implying that $\widehat{R}_m^*(\widehat{c})$ is an unbiased estimator of $R(\widehat{c})$.

Thus, as long as we have a set of new data, the empirical risk on the new dataset is an unbaised estimator of the actual risk.

However, in reality we may not have a new dataset, what can we do? A simple approach is to create the new dataset by splitting the original data set. Namely, we will random split

$$\mathcal{D} = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$$

into $\mathcal{D}_{Tr}$ and $\mathcal{D}_{Val}$ where

$$\mathcal{D} = \mathcal{D}_{Tr} \cup \mathcal{D}_{Val}, \quad \mathcal{D}_{Tr} \cap \mathcal{D}_{Val} = \emptyset.$$

Namely, some observations are in $\mathcal{D}_{Tr}$ while some are in $\mathcal{D}_{Val}$ and each observation is either in $\mathcal{D}_{Tr}$ or $\mathcal{D}_{Val}$. We will call $\mathcal{D}_{Tr}$ the *training set* and $\mathcal{D}_{Val}$ the *validation set*.

We then apply ERM of $\mathcal{D}_{Tr}$ to find the classifier $\widehat{c}$ and then evaluate its performance on $\mathcal{D}_{Val}$. Namely, we are treating the training set as our original data and the validation set as the new data. Indeed, both datasets are independent of each other and they are from the same population so the analysis above can be applied in this case. This technique is called *data splitting* and is a common way when we have a multi-stage estimation procedure (here we have two stages, the first stage is finding the classifier $\widehat{c}$ and the second stage is estimating the risk $R(\widehat{c})$).

By using the data splitting, we obtain a good classifier with a good estimate about its performance.