ECE901 Spring 2004 Statistical Regularization and Learning Theory **Lecture: 6**

Probably Approximately Correct (PAC) Learning

Lecturer: Rob Nowak *Scribe: Badri Narayan*

1 Introduction

1.1 Overview of the Learning Problem

The fundamental problem in learning from data is proper Model Selection. As we have seen in the previous lectures, a model that is too complex could overfit the training data (causing an estimation error) and a model that is too simple could be a bad approximation of the function that we are trying to estimate (causing an approximation error). The estimation error arises because of the fact that we do not know the true joint distribution of data in the input and output space, and therefore we minimize the empirical risk (which is a random number depending on the data) and estimate the average risk again from the limited number of training samples we have. The approximation error measures how well the functions in the chosen model space can approximate the true dependence of the output space on the input space, and in general improves as the "size" of our model space increases.

1.2 Lecture Outline

In the preceding lectures, we looked at some proposed solutions to deal with the overfitting problem and investigated in detail the application of Method of Sieves which suggests a way of making the complexity of the model space depend on the number of data samples. We refine our idea further and propose that we may do better if we allowed the complexity of the model space to adapt to the distribution of the training data, rather than just the number of samples. The premise is that such a choice of model space could reduce the probability of overfitting as we only choose a model that is no more complex than what the data can reliably tell us.

Based on these ideas, the second part of the lecture introduces a learning model called "Probably Approximately Correct" learning, where we derive bounds on estimation error for a choice of model space and given training data.

2 Recap: Method of Sieves

The method of Sieves underpinned our approaches in the denoising problem and in the histogram classification problem. Recall that the basic idea is to define a sequence of model spaces $\mathcal{F}_1$, $\mathcal{F}_2$, ... of increasing complexity, and then given the training data $\{X_i, Y_i\}_{i=1}^n$ select a model according to

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}_n} \hat{R}_n(f)$$

The choice of the model space $\mathcal{F}_n$ (and hence the model complexity and structure) is determined completely by the sample size n, and does not depend on the values of the training data. This is a major limitation of the sieve method. In a nutshell, the method of sieves tells us to average the data in a certain way (e.g., over a partition $\mathcal{X}$) based on the sample size, independent on the sample values.

Learning basically comprises of two things:

1. Averaging data to reduce variability

2. Deciding *where (or how)* to average to reduce bias

Sieves basically force us to deal with (2) *a priori* (before we analyze the tranining data). This will lead to suboptimal classifiers and estimators, in general. Indeed (2) is the really interesting and fundamental aspect of learning; once we learn to deal with (2) we have solved the learing problem. There are at least two possibilities for breaking the rigidity of the method of sieves, as we shall see in the following section.

3 Data dependent Model Selection

1. Structural Risk Minimization
2. Complexity Regularization

3.1 Structural Risk Minimization

The basic idea is to select $\mathcal{F}_n$ based on the training data themselves. Let $\mathcal{F}_1$, $\mathcal{F}_2$, ... be a sequence of model spaces of increasing complexities with

$$\lim_{k\to\infty} \inf_{f\in\mathcal{F}_k} R(f) = R^*$$

Let

$$\hat{f}_{n,k} = \arg\min_{f\in\mathcal{F}_k} \hat{R}_n(f)$$

, a function from $\mathcal{F}_k$ that minimizes the empirical risk. This gives us a sequence of selected models $\hat{f}_{n,1}, \hat{f}_{n,2}, \ldots$ Also associate with each set $\mathcal{F}_k$ a value $C_{n,k} > 0$ that measures the complexity or "size" of the set $\mathcal{F}_k$. Typically, $C_{n,k}$ is monotonically increasing with k (since the sets are of increasing complexity) and decreasing with n (since we become more confident with more training data). A typical example could be the use of VC dimension to characterize the complexity of the collection of model spaces *i.e.,*$C_{n,k}$ is derived from a bound on the estimation error.

3.2 Complexity Regularization

In this case, to every $f \in \mathcal{F}$ assign a complexity $C_n(f)$ (e.g., code length) and select

$$\hat{f}_n = \arg\min_{f\in\mathcal{F}} \{\hat{R}_n(f) + C_n(f)\}$$

Complexity Regularization and Structural Risk Minimization (SRM) are very similar (differing only in how one measures complexity) and can be equivalent in certain instances. The key point of SRM and complexity requlrarization techniques is that the complexity and structure of the model is not fixed prior to examining the data, the data aid in the selection of the best complexity. In fact, the key difference compared to the method of Sieves is that these methods can allow the data to play an integral role in deciding where and how to average the data.

4 Probably Approximately Correct (PAC) learning

In this section, we look at the development of the theory of the PAC learning model which relates the sample size of the model space and the complexity of the class k required to bound the estimation error to be with a certain accuracy ε, and with a certain confidence $1 - \delta$

4.1 Approximation and Estimation Errors

In order to develop complexity regularization schemes we will need to revisit the estimation error / approximation error trade-off. Let $\hat{f}_n = \arg\min_{f\in\mathcal{F}} \hat{R}_n(f)$ for some space of models $\mathcal{F}$.

$$E[R(\hat{f}_n)] - R^* = \underbrace{\inf_{f\in\mathcal{F}} R(f) - R^*}_{\text{approximation error}} + \underbrace{E[R(\hat{f}_n)] - \inf_{f\in\mathcal{F}} R(f)}_{\text{Estimation Error}}$$

The approximation error depends on how close f^* is close to $\mathcal{F}$, and without making assumptions, this is unknown. The estimation error is quantifiable, and depends on the complexity of the size of $\mathcal{F}$. The error decomposition is illustrated in Figure 1. The estimation error quantifies how much we can "trust" the empirical risk minimization process to select a model close to the best in a given class.

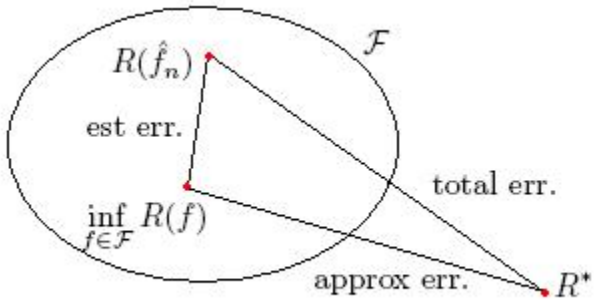

Figure 1: Relationship between the errors

In SRM, we have a sequence of classes $\mathcal{F}_1$, $\mathcal{F}_2$, ... of increasing complexity. We bound the estimation error

$$E[R(\hat{f}_n)] - \inf_{f\in\mathcal{F}} R(f) \leq C(n, \mathcal{F}_k)$$

in each class, and then use $C(n, \mathcal{F}_k), k = 1, 2, \cdots$ to help us avoid overfitting to the training model. That is we select a model from a class $\mathcal{F}_k$ that is not too simple and not too complex; one that balances the empirical risk minimization and the estimation error boubd in an appropriate way.

4.2 The PAC Learning Model (Valiant '84)

The estimation error will be small if $R(\hat{f}_n)$ is close to $\inf_{f\in\mathcal{C}} R(f)$. PAC learning expresses this as follows. We want $\hat{f}_n$ to be the "probably approximately correct" model from $\mathcal{C}$. Finally $\hat{f}_n$ is PAC if for some $\varepsilon > 0, \delta > 0$,

$$P\left(R(\hat{f}_n) - \inf_{f\in\mathcal{C}} R(f) > \varepsilon\right) < \delta$$

This says that the difference between $R(\hat{f}_n)$ and $\inf_{f\in\mathcal{C}} R(f)$ is greater than ε with probability less than δ. Using a PAC bound of this form, one can easily bound the estimation error $E[R(\hat{f}_n)] - \inf_{f\in\mathcal{C}} R(f)$ as we will see.

Sometimes, especially in computer science jargon, PAC bounds are stated as, "with probability of at least $1-\delta$, $|R(\hat{f}_n) - \inf_{f\in\mathcal{F}} R(f)| \leq \varepsilon$"

To introduce PAC bounds, let us consider a simple case. Let $\mathcal{F}$consist of a finite number of models, and let —$\mathcal{F}$— denote that number. Furthermore, assume that $\min_{f\in\mathcal{F}} R(f) = 0$.

Example 1 *$\mathcal{F}$= set of all histogram classifiers with M bins* $\implies |\mathcal{F}| = 2^M$

$$\min_{f\in\mathcal{F}} R(f) = 0 \implies \exists \text{ a classifier in } \mathcal{F} \text{ that has a zero probability of error}$$

Theorem 1 *Assume $\mathcal{F} < \infty$ and $\min_{f\in\mathcal{F}} R(f) = 0$, where $R(f) = P(f(X) \neq Y)$. Then for every n and $\varepsilon > 0$,*

$$P(R(\hat{f}_n) > \varepsilon) \leq |\mathcal{F}|e^{-n\varepsilon} \equiv \delta$$

Proof: Since $\min_{f\in\mathcal{F}} R(f) = 0$, it follows that $\hat{R}_n(f_n) = 0$. In fact, there may be several $f \in \mathcal{F}$ such that $\hat{R}_n(f) = 0$. Let $\mathcal{G} = \{f : \hat{R}_n(f) = 0\}$.

$$\begin{aligned}
P(R(\hat{f}_n) > \varepsilon) &\leq P\left(\bigcup_{f\in\mathcal{G}} \{R(f) > \varepsilon\}\right) \\
&= P\left(\bigcup_{f\in\mathcal{F}} \{R(f) > \varepsilon,\ \hat{R}_n(f) = 0\}\right) \\
&= P\left(\bigcup_{f\in\mathcal{F}:R(f)>\varepsilon} \{\hat{R}_n(f) = 0\}\right) \\
&\leq \sum_{f\in\mathcal{F}:R(f)>\varepsilon} P(\hat{R}_n(f) = 0) \\
&\leq |\mathcal{F}|.(1-\varepsilon)^n
\end{aligned}$$

(since the probability that none of the training samples fall into the set $\{(X, Y) : f(X) \neq Y\}$ is less than $(1 - \varepsilon)^n$.

i.e.,

$$\begin{aligned}
R(f) > \varepsilon &\implies P(f(X) \neq Y) > \varepsilon \\
&\implies P(\{X, Y\} : f(X) \neq Y) > \varepsilon \\
&\implies P(\{X, Y\} : f(X) = Y) \leq 1 - \varepsilon
\end{aligned}$$

Finally apply the inequality $1 - x \leq e^{-x}$

Note that for n sufficiently large,

$$\delta = |\mathcal{F}|e^{-n\varepsilon}$$

is arbitrarily small. To achieve a (ε, δ)-PAC bound for a desired $\varepsilon > 0, \delta > 0$ we require at least

$$n = \frac{\log|\mathcal{F}| - \log\delta}{\varepsilon}$$

training examples. ■

Corollary 1 *Assume that $|\mathcal{F}| < \infty$ and $\min_{f\in\mathcal{F}} R(f) = 0$. Then for every n and $\varepsilon > 0$,*

$$E[R(\hat{f}_n)] \leq \frac{1 + \log|\mathcal{F}|}{n}$$

Proof: Recall that for any non-negative random variable Z with finite mean, $E[Z] = \int_0^\infty P(Z > t)dt$. This follows from an application of integration by parts.

$$\begin{aligned}
E[R(\hat{f}_n)] &= \int_0^\infty P(R(\hat{f}_n) > t)dt \\
&= \int_0^u \underbrace{P(R(\hat{f}_n) > t)}_{\leq 1} dt + \int_u^\infty P(R(\hat{f}_n) > t)dt, \quad \text{for any } u > 0 \\
&\leq u + |\mathcal{F}| \int_u^\infty e^{-nt} dt \\
&= u + \frac{|\mathcal{F}|}{n} e^{-nu}
\end{aligned}$$

Minimizing with respect to u produces the smallest upper bound with $u = \frac{\log |\mathcal{F}|}{n}$ ■