Indian Institute of Technology, Kanpur
Statistical and Algorithmic Learning Theory

Instructors: Purushottam Kar
Authors: Gurpreet Singh
Email: guggu@iitk.ac.in

SCRIBE

3

Parametric Learning in Realizable Settings

Before we attempt to solve a machine learning problem, we must accurately define the model or setting. Hence, we first define a formal model aimed to cover learning tasks.

We only cover parametric learning models *e.g.* Binary Classification, Regression, etc in this chapter and following ones

Definition of a Formal Model and Notations

1. **Instance/Feature Space** $(\mathcal{X})$ — This is the space of the random variable that defines the data distribution. $\mathcal{X}$ is the set of objects that we wish to label. For example, for the problem of classifying an email as spam or not-spam, the set of emails defines the set $\mathcal{X}$

2. **Output Space** $(\mathcal{Y})$ — This defines the space of the random variable that is the label or value mapped to each data point. For example, in the problem of email classification, the output space is $\{0, 1\}$ where 1 is for spam and 0 for not-spam. There can be different Output Spaces depending on the problem.

   - Labeling: $\{-1, 1\}, \mathbb{R}^p$
   - Permutation: S_n
   - Hierarchy / Tree
   - Alternate Representation: $\mathbb{R}^D \to \mathbb{R}^d$ (*e.g.* Dimensionality Reduction)

3. **Distribution** $(\mathcal{D})$ — This is also known as the data generation distribution. $\mathcal{D}$ defines the probability distribution of the data points (and sometimes also labels, more in agnostic setting). For the realizable setting, this distribution is used to sample data points say $\{x^1, x^2 \ldots x^n\}$ and a *prediction rule* $f : \mathcal{X} \to \mathcal{Y}$ is used to map the data points to their respective labels, as per defined by the prediction rule *i.e.* $y^i = f(x^i)$

4. **Training Sample** (S) — This is a tuple or sequence of n data points and their mapped label/parameter sampled from the *distribution* or are randomly obtained. $S = \big((x^1, y^1), (x^2, y^2) \ldots (x^n, y^n$ $(\mathcal{X} \times \mathcal{Y})^n$. This is the part of data that is generally visible to us, and is used for learning prediction models.

5. **Loss Function** (l) — Loss function defines the similarity / dissimilarity in the estimated label value and the actual / observed label value. We say $l : (f(\mathcal{X}), \mathcal{Y}) \to \mathbb{R}$. There are different types of loss functions, for example the squared loss function l^{sq} and the $0-1$ loss function l^{0-1}.

6. **Algorithm** (A) — We say that the algorithm maps the generated / sampled training sample to a function belonging in the hypothesis space *i.e.*

$$A : S \mapsto \hat{f}_s \in \mathcal{H}$$

$$A : \bigcup_{n=1}^{\infty} (\mathcal{X}, \mathcal{Y})^n \mapsto \mathcal{H}$$

7. **l-risk** $(\mathrm{er}^l_D\left[\, f\,\right])$ — This is defined as the expected loss/error that is obtained for a function f given a loss function $l : (f(\mathcal{X}), \mathcal{Y}) \to \mathbb{R}$.

$$\mathrm{er}^l_D\left[\, f\,\right] \quad \triangleq \quad \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\, l(f(x), y)\,\right]$$

8. **emperical risk** $(\mathrm{er}^l_D\left[\, f\,\right])$ — This is defined as the weighted loss/error on the training sample that is obtained for a function f given a loss function $l : (f(\mathcal{X}), \mathcal{Y}) \to \mathbb{R}$

$$\mathrm{er}^l_S\left[\, f\,\right] \quad \triangleq \quad \frac{1}{n}\sum_{i=1}^{n} l(f(x^i), y^i)$$

Exercise 3.1. Show that $\mathrm{er}^l_D\left[\, f\,\right] = \mathbb{E}\left[\mathrm{er}^l_S\left[\, f\,\right]\right]$

Exercise 3.2. Show that $\mathrm{er}^{l^{0-1}}_D\left[\, f\,\right] = \mathrm{er}^{0-1}_S\left[\, f\,\right]$ where

$$l^{0-1}(\hat{y}, y) \quad = \quad 1 - \mathbb{I}\left[\, y = \hat{y}\,\right]$$

Exercise 3.3. Find out $\mathrm{er}^{l^{0-1}_\alpha}_D\left[\, f\,\right]$ in terms of the empirical risk where

$$l^{0-1}_\alpha \quad = \quad \begin{cases} \alpha & \hat{y} \neq y, y = 1 \\ 1-\alpha & \hat{y} \neq y, y = 0 \\ 0 & \hat{y} = y \end{cases}$$

Empirical Risk Minimization

We need to have a learning algorithm to reduce the l-risk / true error on the distribution. However, as discussed earlier, we can only see the training sample S, and hence we have no direct way to minimize error on the true distribution (given some loss function) *i.e.* $\mathrm{er}^l_D\left[\, l\,\right]$.

To tackle this, we try to minimize the empirical error or empirical risk. The intuition behind this is the assumption / expectation that the training sample is a snapshot of the true data distribution and it explains the universe well.

Hence, our learning algorithm gives us a prediction rule $\hat{f} \in \mathcal{H}$ which is the estimate of the true f^* where

$$f^* \quad = \quad \arg\min_{f} \; \mathrm{er}^l_D\left[\, f\,\right]$$

$$f^* \quad \approx \quad f_{ERM} \quad = \quad \arg\min_{f\in\mathcal{H}} \; \mathrm{er}^l_S\left[\, f\,\right]$$

Although we have a nice prediction rule which allows us to learn our sample training, and is expected to be a good predictor for the complete data / universe as well. However, the ERM predictor may highly overfit the sample training data.

Overfitting of Training Sample

To understand what overfitting is, let us take a simple binary classification example.

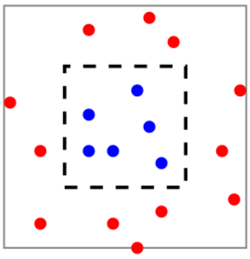

Figure 1: Randomly sampled data points with red / blue labels. Figure taken from **?**

Suppose we have some data points (universe) sampled randomly within a box, as shown in Figure 1. Also, for some S, the sampled data points for the training sample are the points within the gray box.

Therefore, we can write a prediction rule which always labels the data point as 1. Since this gives training error as 0 *i.e.* $\mathrm{er}^l_S\,[\,f\,] = 0$. However, the same prediction rule gives high error when run on the complete data.

Therefore, a predictor that performs "too well" on a training sample, whereas gives a high error on the complete data is said to *overfit* the data. We can also write the formal definition of overfitting.

Definition 3.1 (Overfitting (**?**)). The production of an analysis which corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably.

Empirical Risk Minimization with Inductive Bias

We know that the ERM method might lead to overfitting. In order to avoid that, we need to understand the reason behind it.

The overfitting problem is caused by the learner fitting a function to accurately predict the labels for all the data points in the training sample. Hence, rather than predicting the general pattern in the data.

A simple solution to this is to restrict the search space of the prediction function. Hence, instead of minimizing the empirical risk over all functions, we will look at only a set of possible functions. This set is called the *hypothesis class* or the *model class* (or *hypothesis space*). We can formally define this as below.

Definition 3.2 (Hypothesis/Model Space). The hypothesis class (represented as $\mathcal{H}$) defines the set of all functions $\{f : \mathcal{X} \to \mathcal{Y}\}$ belonging to a certain function class, for example for a regression case, the hypothesis space can be all the linear functions passing through $\mathbf{0}$. Since this is the set of different mappings (prediction rules) of all data points, we can say $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$.

Therefore, we now redefine the ERM model to learn a predictor belonging to the hypothesis class only.

$$\hat{f} \quad = \quad \operatorname*{arg\,min}_{f \in \mathcal{H}} \; \mathrm{er}^l_D\,[\,f\,]$$

Such restrictions, for exampling, fixing the hypothesis space is called the *inductive bias.*

Avoiding Overfitting with Finite Hypothesis Spaces

We claimed in the previous section that we can avoid overfitting if we use Hypothesis space. However, overfitting is highly probably avoided in case of finite hypothesis spaces. More formally, if the hypothesis space $\mathcal{H}$ is a finite class of size m, where m is *sufficiently large*, then the risk of the predictor function is bounded with a high probability.

Note. We will prove this for the realizable setting, however this also holds for the agnostic setting as well, but only the bounds will change by a constant term, as we will see later.

We still need to define when is m "sufficiently large", however, first we need to take some generic assumptions for our model.

1. **Indepent and Identically Distributed Assumption.** This assumption states that the data points in the training sample are independent and identically distributed according to the distribution $\mathcal{D}$. Therfore, S is essentially a set of n independent data points all sampled from $\mathcal{D}$. We represent this as $S \overset{iid}{\sim} \mathcal{D}^n$.
2. **The Realizability Assumption.** The realizability assumption states that there exists $f^* \in \mathcal{H}$ such that $\mathrm{er}_D^l\left[\, f^* \,\right] = 0$. This assumption says that the with probability 1 over any random sample, S, $\mathrm{er}_S^l\left[\, f^* \,\right]$

Since S is sampled using the data distribution, $\mathcal{D}$, we can say that S and therefore $\mathrm{er}_S^l\left[\, f \,\right]$ are random variables for some $f \in \mathcal{H}$. We say an algorithm for a learning model is very less likely to overfit if the l-risk for all functions learned by the algorithm for any sample is bounded by ϵ with probability more than $1 - \delta$ where ϵ is the accuracy parameter and δ is the confidence parameter.

Therefore, we need to bound $\mathbb{P}\left[\mathrm{er}_D^l\left[\, f_S \,\right] > \epsilon\right]$. Since this is just dependent on S, we can write this in an alternate way

$$\mathbb{P}\left[\mathrm{er}_D^l\left[\, f_S \,\right] > \epsilon\right] \quad = \quad \mathcal{D}^m\left(S \,\middle|\, \mathrm{er}_D^l\left[\, f_S \,\right] > \epsilon\right)$$

Hence, we want to bound $\mathcal{D}^m\left(S \,\middle|\, \mathrm{er}_D^l\left[\, f_S \,\right] > \epsilon\right)$. We can define a few more terms, which will make the proof easier.

$$\begin{aligned} \mathcal{H}_B &= \left\{f \in \mathcal{H} \,\middle|\, \mathrm{er}_D^l\left[\, f \,\right] > \epsilon\right\} \\ M &= \left\{S \,\middle|\, \exists\, f \in \mathcal{H}, \mathrm{er}_S^l\left[\, f \,\right] = 0\right\} \end{aligned}$$

where $\mathcal{H}_B$ defines the set of (bad) functions which give l-risk more than ϵ and M as the set of (misleading) samples for which there is at least one function which gives 0 empirical risk. We say that we are misleaded only when our sample gives 0 empirical risk, but high l-risk *i.e.* the set of samples that give a bad hypothesis (prediction function) must a subset of the misleading samples. Therefore,

$$\left\{S \,\middle|\, \mathrm{er}_D^l\left[\, f_S \,\right] > \epsilon\right\} \quad \subseteq \quad M \implies \mathcal{D}^m\left(S \,\middle|\, \mathrm{er}_D^l\left[\, f \,\right] > \epsilon\right) \qquad \leq \quad \mathcal{D}^m\left(M\right)$$

We can also rewrite M as

$$M \quad = \quad \bigcup_{f \in \mathcal{H}} \Big\{ S \,|\, \mathrm{er}^l_S\,[\,f\,] = 0 \Big\}$$

Substituting the above form of M in the previous equation and using the union bound, we get

$$\mathcal{D}^m \Big(S \,|\, \mathrm{er}^l_D\,[\,f\,] > \epsilon \Big) \quad \leq \quad \sum_{f \in \mathcal{H}} \mathcal{D}^m \left(\Big\{ S \,|\, \mathrm{er}^l_S\,[\,f\,] = 0 \Big\} \right)$$

wher $f \in \mathcal{H}_B$ is some fixed bad prediction rule. Since the sample is generated i.i.d., , we can write

$$\begin{aligned}
\mathcal{D}^m \left(\Big\{ S \,|\, \mathrm{er}^l_S\,[\,f\,] = 0 \Big\} \right) \quad &= \quad \mathcal{D}^m \left(\left\{ S \,|\, \forall\, i, f(x^i) = y^i \right\} \right) \\
&= \quad \prod_{i=1}^{n} \mathcal{D} \left(\left\{ x^i \,|\, f(x^i) = y^i \right\} \right) \\
&= \quad \prod_{i=1}^{n} 1 - \mathrm{er}^l_D\,[\,f\,] \\
&\leq \quad (1 - \epsilon)^n \\
&\leq \quad e^{-\epsilon n}
\end{aligned}$$

Hence, we can combine all the equations and write

$$\mathcal{D}^m \left(\Big\{ S \,|\, \mathrm{er}^l_D\,[\,f_S\,] > \epsilon \Big\} \right) \quad \leq \quad |\,\mathcal{H}_B\,|\, e^{-\epsilon n} \quad \leq \quad |\,\mathcal{H}\,|\, e^{-\epsilon n}$$

We can now define the bound on the size of the training sample S

Theorem 3.1. For some finite hypothesis class, $\mathcal{H}$, let $\delta, \epsilon \in (0, 1)$ and let m be an integer that satisfies

$$m \quad \geq \quad \frac{\log\left(|\,\mathcal{H}\,| / \delta\right)}{\epsilon}$$

Then for any prediction function f, and for any distribution $\mathcal{D}$, for which the realizability assumption holds, with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m, we have that for every ERM hypothesis class will be probably (with confidence $1 - \delta$) approximately (up to an error ϵ) correct.