

Some bibliography:

- Lecture notes in cs.huji.ac.il/~shais/
- Kearns and Vazirani - An Introduction to Computational Learning Theory
- Anothony and Bartlett - Neural Network Learning: Theoretical Foundations
- Vapnik - Statistical learning theory

1 Definition of the learning problem

Thus far we focused mostly on learning distributions parameterized by a vector θ . Our interest there was to recover the vector θ itself. In the next few classes we will focus on a related problem, with slightly different goals. In many cases we are interested in learning a classification rule. For example mapping between handwritten letters and the letters they represent, images and the identity of the face in the image, the topic of a document, email (spam vs not-spam) etc.

We will be interested in learning such rules from examples. And, we would like to be able to say something about how well we can generalize. Some definitions:

- The following constitute the input to the learning algorithm:
 - The input (or feature) space \mathcal{X} . These will be the objects that we classify (images, text etc). Typically we will assume $\mathcal{X} = \mathbb{R}^d$.
 - The output (or label) space \mathcal{Y} . The set of different classes that a \mathcal{X} can belong to (spam vs no spam, $\{0, \dots, 9\}$ for digits, etc). We will focus on the case $\mathcal{Y} \in \{-1, 1\}$ for simplicity of analysis.
 - A training sample $\mathcal{S} = (x_1, y_1), \dots, (x_n, y_n)$ generated IID from an unknown distribution $p(x, y)$. Note that we generally place no restrictions on p so that a given x might be sampled with different y . This is certainly possible since \mathcal{X} does not always have enough information to determine \mathcal{Y} .
- The output of the learning algorithm is a classification rule $h : \mathcal{X} \rightarrow \mathcal{Y}$. We denote this output by $h_{\mathcal{S}}$ to highlight the dependence on the sample \mathcal{S} .

It's intuitively clear that we should not consider all possible h functions. As doing so will almost certainly result in over-fitting. We thus need the notion of a hypothesis class \mathcal{H} which is a set of hypotheses $h : \mathcal{X} \rightarrow \mathcal{Y}$. Some examples are (for binary \mathcal{Y}):

- For $\mathcal{X} = \mathbb{R}^2$. The set $h(x; R)$ such that $h(x; R) = 1$ if x is inside a rectangle R and zero otherwise
- For $\mathcal{X} = \mathbb{R}^d$. The set $h(x; w, b)$ such that $h(x; w, b) = \text{sgn}(x \cdot w - b)$
- The set of neural networks with an output threshold
- For $\mathcal{X} \in \{0, 1\}^d$. The set h where each h is defined by a boolean formula of a 3 CNF form on x : e.g., $h = (x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee x_4 \vee x_6)$
- For $\mathcal{X} = \mathbb{R}^d$. The set of all possible classification rules $h(x)$.

Say we obtain an output hypothesis h . How do we determine how good it is? The natural choice is to measure the probability of error taken over the generating distribution $p(x, y)$:

$$e_p(h) = \sum_{x,y} p(x, y) I(h(x) \neq y) = \mathbb{P}_p[I(h(x) \neq y)] \quad (1)$$

(this can also be viewed as the risk $R[h|p]$). If we had a way of estimating this, we could just choose the h that minimizes it. Note that this will be $h(x) = \arg \max_y p(y|x)$, sometimes called the "Bayes optimal" classifier. But we only have the sample S . We could instead consider the *empirical (or training) error* or *empirical risk*:

$$e_S(h) = \frac{1}{n} \sum_i I(h(x_i) \neq y_i) = \sum_{x,y} \hat{p}(x, y) I(h(x) \neq y) \quad (2)$$

This suggests the following scheme for generating classifiers, also called **Empirical Risk Minimization**:

$$h_S^{ERM} \in \arg \min_{h \in \mathcal{H}} e_S(h) \quad (3)$$

When is this a good approach? Clearly this depends on the nature of the class \mathcal{H} . But how? Here are some key questions we would like to answer:

- How many samples n are needed to reach a classifier with low error?
- How should we choose the class \mathcal{H} ? What are the tradeoff involved?
- When does ERM work?
- What is the relation between the empirical (training) and generalization error?

As we shall see, these questions are closely linked. Intuitively if the training error is a good approximation of the generalization error, it would make sense to minimize it. Here are some intuitive observations:

- As n grows the training error should get closer to the generalization error
- As the hypothesis class "grows" the training error should become more different from the generalization error
- As the hypothesis class "grows" the minimal training error should decrease

We will now try to make these statements more exact. The field that studies such questions is known as "Computational Learning Theory". Its pioneers are Vladimir Vapnik, Alexey Chervonenkis, Leslie Valiant, David Pollard, Michael Kearns, Manfred Warmuth and many others.

2 Learning in the realizable case: the PAC framework

Assume that our hypothesis class \mathcal{H} matches the generating distribution p in the following sense: samples are generated by first sampling x and then choosing y deterministically according to some hypothesis $h_0 \in \mathcal{H}$. This means that $e_p(h_0) = 0$. In other words our hypothesis class \mathcal{H} contains the true hypothesis (e.g., see the rectangle example).

Some observations are important here:

- For every sample S there will be at least one (but typically many more) hypotheses $h \in \mathcal{H}$ such that $e_S(h) = 0$.
- Thus ERM will always choose a hypothesis with this property.
- The generalization error of the ERM hypothesis will typically not be zero. Because a finite sample is almost never enough information to determine a zero error hypothesis.
- There could be very "bad" samples, in the sense that they tell us very little about the p . For example, in the rectangle case we could get a cluster of points in the middle of the rectangle. It is thus reasonable to have a theory that will not work for such "unlikely" samples (as long as they have a small probability).
- The requirement that p is realizable can hurt us, because we might need to grow \mathcal{H} until it contains the true hypothesis. This might hurt our generalization. More on this later.

We now want to build an algorithm with "good" generalization properties. Say we require a generalization error of at most ϵ for some $\epsilon > 0$. We'd like a procedure that will always generate a h_S with this generalization error if

we give it a large enough sample. As mentioned above, this is not likely to be possible because of bad samples. So, we should be ready to accept that with probability δ we will get a bad sample.

We now ask the following question: Given $\epsilon, \delta > 0$: How many samples does an ERM algorithm need in such that for any realizable p it obtains a generalization error $e_p(h_S) < \epsilon$ with probability larger than $1 - \delta$. In other words how many samples do we need to guarantee:

$$\mathbb{P}_{S_n \sim p}[e_p(h_{S_n}^{ERM}) \geq \epsilon] \leq \delta$$

Theorem: For a finite hypothesis class \mathcal{H} , and any ϵ, δ define:

$$n(\epsilon, \delta) = \frac{\ln[|\mathcal{H}|/\delta]}{\epsilon} \quad (4)$$

Then for any realizable p , an ERM algorithm that takes $n > n(\epsilon, \delta)$ samples as input, will have a generalization error $e_p(h_S) < \epsilon$ with probability at least $1 - \delta$ over the choice of sample S .

This gives us an upper bound on generalization error which is true when the training error is zero (because this is ERM for the realizable case).

Proof: We will derive n by requiring a generalization error of ϵ . ERM returns a hypothesis h_S^{ERM} that has $e(h_S^{ERM}) = 0$. Let us call hypotheses with $e_p(h) > \epsilon$ bad hypotheses. What is the probability that ERM returns a bad hypothesis? We will do this only if our sample is such that it is consistent with a bad hypothesis. So:

$$\begin{aligned} \mathbb{P}_{S_n \sim p}[e_p(h_{S_n}^{ERM}) \geq \epsilon] &= \mathbb{P}_{S_n \sim p}[h_{S_n}^{ERM} \text{ is a bad hypothesis}] \\ &\leq \mathbb{P}_{S_n \sim p}[S_n \text{ is consistent with some bad hypothesis}] \\ &= \mathbb{P}_{S_n \sim p}[\cup_{h \in \mathcal{H}_{bad}^\epsilon(p)} S_n \text{ is consistent with } h] \\ &\leq \sum_{h \in \mathcal{H}_{bad}^\epsilon(p)} \mathbb{P}_{S_n \sim p}[S_n \text{ is consistent with } h] \end{aligned}$$

The first inequality is because it could be that S_n is consistent with a bad hypothesis, but also with a good one and we happen to choose a good hypothesis.

To see what the above sum over probabilities is, recall that a bad h satisfies $e_p(h) > \epsilon$. This is exactly the probability of drawing one x_i such that (x_i, y_i) is inconsistent with p . Thus the probability of drawing n consistent points is exactly $(1 - \epsilon)^n$. So:

$$\begin{aligned} \mathbb{P}_{S_n \sim p}[e_p(h_{S_n}^{ERM}) \geq \epsilon] &\leq \sum_{h \in \mathcal{H}_{bad}^\epsilon(p)} (1 - \epsilon)^n \\ &\leq |\mathcal{H}|(1 - \epsilon)^n \leq |\mathcal{H}|e^{-n\epsilon} \end{aligned}$$

Now, we would like to choose n such that this probability is less than δ . So

$$\begin{aligned} |\mathcal{H}|e^{-n\epsilon} &\leq \delta \\ n &\geq \frac{1}{\epsilon} \ln \frac{|\mathcal{H}|}{\delta} \end{aligned}$$

Now suppose we fix n, δ and ask what is the maximum value of the ERM generalization error. We know that

$$\mathbb{P}_{S_n \sim p}[e_p(h_{S_n}^{ERM}) \geq \epsilon] \leq |\mathcal{H}|e^{-n\epsilon} \leq \delta$$

This will hold as long as

$$\epsilon \geq \frac{1}{n} \ln \frac{|\mathcal{H}|}{\delta} \quad (5)$$

And specifically at $\epsilon = \frac{1}{n} \ln \frac{|\mathcal{H}|}{\delta}$. Thus we can conclude that with probability greater than $1 - \delta$ it holds that

$$0 \leq e_p(h_{S_n}^{ERM}) \leq \frac{1}{n} \ln \frac{|\mathcal{H}|}{\delta} \quad (6)$$

This is essentially a $1 - \delta$ confidence interval for the generalization error of ERM.

We have shown that to get generalization less than ϵ with certainty $1 - \delta$ we can use a sample size $n(\epsilon, \delta)$ that will guarantee successful learning regardless of the underlying p . Thus, our hypothesis class can be learned. The following definition makes this explicit.

Definition: A hypothesis class \mathcal{H} is PAC learnable if for any $\epsilon > 0, \delta > 0$ there exists an integer $n(\epsilon, \delta)$ and a learning algorithm such that for any distribution $p(x, y)$, which satisfies the realizability assumption, when running the learning algorithm on $n(\epsilon, \delta)$ i.i.d. examples it returns $h \in \mathcal{H}$ such that with probability of at least $1 - \delta$, it holds that $e_p(h) \leq \epsilon$.

We say that a class is efficiently learnable if n is polynomial in $\frac{1}{\epsilon}, \frac{1}{\delta}$. It follows that finite hypothesis classes are efficiently PAC learnable.

3 Learning in the unrealizable case: Agnostic PAC

As mentioned earlier, assuming realizability is often neither realistic or even worthwhile (due to increased sample size needed). We turn to the learning setting where the distribution p is not necessarily realizable in \mathcal{H} , but we are still learning using the class \mathcal{H} . This is referred to as the agnostic setting (where agnostic indicates that we cannot assume knowledge of the underlying distribution).

Here we will not expect to get an error that approaches zero. The best we can hope for is to approach the best hypothesis in \mathcal{H} . Define:

$$\begin{aligned} h_p^* &= \arg \min_{h \in \mathcal{H}} e_p(h) \\ e_p^* &= e_p(h^*) \end{aligned}$$

This is the best generalization error we can hope for when using \mathcal{H} . As the sample size grows it becomes more likely that we can achieve this error.

Here again we will study the sample complexity of the ERM approach. Note that now h_S^{ERM} is not guaranteed to have zero error on the training set.

We would like the empirical error curve to be guaranteed to be close to the generalization curve. If these are close, then minimizing the empirical error should result in a similar error.

Lemma: For every distribution $p(x, y)$ and sample S_n it holds that:

$$e_p(h_S^{ERM}) - e_p^* \leq 2 \sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| \quad (7)$$

Proof: Write:

$$\begin{aligned} e_p(h_S^{ERM}) - e_p^* &= e_p(h_S^{ERM}) - e_p(h_p^*) \\ &= e_p(h_S^{ERM}) - e_S(h_S^{ERM}) + e_S(h_S^{ERM}) - e_p(h_p^*) \\ &\leq e_p(h_S^{ERM}) - e_S(h_S^{ERM}) + e_S(h_p^*) - e_p(h_p^*) \\ &\leq e_p(h_S^{ERM}) - e_S(h_S^{ERM}) + e_S(h_p^*) - e_p(h_p^*) \\ &\leq 2 \sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| \end{aligned}$$

We can use this to obtain the number of samples needed to reach a generalization error that is close to e_p^* .

First, we bound the above sup for the finite class case:

$$\begin{aligned} \mathbb{P}_{S_n \sim p} \left[\sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| > \epsilon \right] &= \mathbb{P}_{S_n \sim p} \left[\bigcup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| > \epsilon \right] \\ &\leq \sum_{h \in \mathcal{H}} \mathbb{P}_{S_n \sim p} \left[|e_S(h) - e_p(h)| > \epsilon \right] \end{aligned}$$

This expression is the difference between the empirical and expected value of a boolean variable with expected value $e_p(h)$. We can use the Hoeffding inequality which says that:

$$\mathbb{P} \left(\left| \frac{1}{n} \sum_{i=1}^n X_i - p \right| \geq \epsilon \right) \leq 2e^{-2n\epsilon^2} \quad (8)$$

To obtain

$$\mathbb{P}_{S_n \sim p} \left[\sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| \geq \epsilon \right] \leq 2|\mathcal{H}|e^{-2n\epsilon^2}$$

This implies that with probability at least $1 - \delta$ the error $e_S(h)$ satisfies for **all** hypotheses h :

$$e_S(h) - \sqrt{\frac{1}{2n} \log \frac{2|\mathcal{H}|}{\delta}} \leq e_p(h) \leq e_S(h) + \sqrt{\frac{1}{2n} \log \frac{2|\mathcal{H}|}{\delta}} \quad (9)$$

The factor $\log \frac{2|\mathcal{H}|}{\delta}$ can be improved to $\log \frac{|\mathcal{H}|}{\delta}$ by using a one-sided inequality.

Theorem: For a finite hypothesis class, for every ϵ, δ and $n \geq \frac{2}{\epsilon^2} \ln \frac{2|\mathcal{H}|}{\delta}$ it holds that for all p :

$$\mathbb{P}_{S_n \sim p} [|e_p(h_S^{ERM}) - e_p^*| \geq \epsilon] \leq \delta \quad (10)$$

Proof:

$$\begin{aligned} \mathbb{P}_{S_n \sim p} [|e_p(h_S^{ERM}) - e_p^*| \geq \epsilon] &\leq \mathbb{P}_{S_n \sim p} [2 \sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| > \epsilon] \\ &= \mathbb{P}_{S_n \sim p} [\sup_{h \in \mathcal{H}} |e_S(h) - e_p(h)| > \frac{\epsilon}{2}] \\ &\leq 2|\mathcal{H}| e^{-\frac{n\epsilon^2}{2}} \end{aligned}$$

We want this to be less than δ . So we obtain:

$$\begin{aligned} 2|\mathcal{H}| e^{-\frac{n\epsilon^2}{2}} &\leq \delta \\ n &\geq \frac{2}{\epsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \end{aligned}$$

Also, for a given δ, n we have the following bound on generalization error (with confidence $1 - \delta$):

$$e_p(h_S^{ERM}) \leq e_p^* + \sqrt{\frac{2}{n} \ln \frac{2|\mathcal{H}|}{\delta}} \quad (11)$$

We conclude that finite hypothesis classes are PAC Agnostic learnable.

4 Infinite Hypothesis Classes - The VC Dimension

Define $\Pi_{\mathcal{H}}(S)$ to be the number of dichotomies that \mathcal{H} induces on a given sample S . Formally:

$$\Pi_{\mathcal{H}}(S) = \{[h(x_1), \dots, h(x_n)] : h \in \mathcal{H}\} \quad (12)$$

Note $\Pi_{\mathcal{H}}(S) \subseteq \{0, 1\}^n$.

If $|\Pi_{\mathcal{H}}(S)| = 2^n$ for a sample of size n we say that S is shattered by \mathcal{H} .

Definition: The VC dimension is the maximum n such that there exists a sample of size n that is shattered by \mathcal{H} .

Examples:

- The VC dimension of the axis-aligned rectangles is four. Show that four can be