Mathematical Foundations of Machine Learning(CS 4783/5783)

Lecture 2: Statistical Learning, Generalization and Uniform Convergence

1 Statistical Learning Framework

We already set up the basic notation for our learning problem. We used the notation $\mathcal{X}$ to indicate the input or instance space, the space $\mathcal{Y}$ to indicate the space of all outcomes, and the loss function $\ell : \mathcal{Y} \times \mathcal{Y}$ that evaluates the performance of our model when it predicts $y' \in \mathcal{Y}$ when the desired outcome is $y \in \mathcal{Y}$ as $\ell(y', y)$. In the first lecture we also considered two scenarios and while introducing those scenarios, I mentioned this set U and specifically how it was different from the input space $\mathcal{X}$, especially in scenario one. To make this difference clear, say you were building a face recognition application for your smart phone. $\mathcal{X}$ would be all possible images, meaning all possible pixel values for each pixel. However, if you have a good face detection software available (which most phones now do), then most images that are input to your face recognition system will be faces, with a small number of non-face images due to errors in face detection. You can think of U as this subset of images. Further, in the scenario one in the first lecture, we drew training samples by randomly (uniformly at random) drawing an image from this set U. We also only considered giving probabilistic guarantees for future instances drawn the same way. Of course in reality one does not have access to this set U. However, the process of drawing from this special set U can be closely (and in fact in a more general and elegant way) captured by the framework called Statistical Learning framework. In this framework, we assume that there exists a distribution $\mathbf{D}$ on $\mathcal{X} \times \mathcal{Y}$ that is unknown to the learner. We further assume that the training sample $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is obtained by repeatedly sampling instance labels from this same fixed distribution $\mathbf{D}$. We also assume that future instances are obtained by drawing from this distribution repeatedly. One can think of the set U relative to $\mathcal{X}$ as capturing this distribution over just the instance space $\mathcal{X}$. That is, $\mathbf{D}$ can be thought of as distribution over images and corresponding labels that your face recognition software might receive. Of course, it is worth noting that such an assumption that instances come i.i.d. (independently and identically drawn) from a fixed distribution is not always right. For instance, even in the face recognition application, in winter you might have more white background due to snow and more people wearing warm hats whereas in summer background might be much more greener and there might be lesser faces with hats. That said, while the assumption might not be exactly correct, often the assumption is not too bad an approximation for the tasks we would like. In any case, formally, in the statistical learning problem, there is a distribution $\mathbf{D}$ over $\mathcal{X} \times \mathcal{Y}$ and training sample is obtained by drawing iid samples from this distribution and at deployment time, instances are still drawn iid from this distribution. To make our lives simpler, let us introduce some notations and terminology. Since at test time we obtain instances iid from the distribution $\mathbf{D}$, our hope is to ensure that expected loss over future draws is small. We will refer to this expected loss as "Risk" and given a model $g : \mathcal{X} \mapsto \mathcal{Y}$ denote its risk as:

$$L_{\mathbf{D}}(g) = \mathbb{E}_{(x,y)\sim\mathbf{D}} \left[\ell(g(x), y)\right]$$

In the example we considered in the first lecture, we assumed that there was a set of models $\mathcal{F} = \{f_1, \ldots, f_N\}$ and that one of the models in this class was perfect with zero loss. This assumption is often too strong, for instance this assumes something strong about $P(Y|X)$ and hence is a strong restriction on $\mathbf{D}$. We would like to relax this assumption. In fact, we will make no explicit assumption but instead modify our goal as follows. Given some fixed class of models $\mathcal{F}$, we would like our algorithm to have risk comparable to the best amongst this class of models $\mathcal{F}$. That is, we would like to ensure that our algorithm has low excess risk. By excess risk of a model g compared to a class of models $\mathcal{F}$ we will refer to the quantity:

$$\text{Excess risk of } g \text{ w.r.t. model class } \mathcal{F} = L_{\mathbf{D}}(g) - \min_{f \in \mathcal{F}} L_{\mathbf{D}}(f)$$

Now with all of this terminology, let us define the goal of statistical learning. Given a fixed class of models $\mathcal{F}$, and training sample $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn iid from a fixed but unknown distribution $\mathbf{D}$, the goal of statistical learning is to return a model $\hat{f}_S$ for which with high probability over draw of samples S, the excess risk of the model $L_{\mathbf{D}}(\hat{f}_S) - \min_{f \in \mathcal{F}} L_{\mathbf{D}}(f)$ is small.

2 Training Loss, Test Loss and Generalization

Perhaps the most well known terminology in ML is training loss, test loss (and sometimes validation error). The training loss of a model g on training sample S is simply the average loss of the model on the training sample which we denote as

$$\widehat{L}_S(g) = \frac{1}{|S|} \sum_{(x,y) \in S} \ell(g(x), y)$$

when we collect data and set aside a subset of it that we don't use in training phase in any way at all, such a set is referred to as test set and test loss is simply the average loss of a model on this test set. Since the test set is never used while training a model, due to law of large numbers, the test loss of our learnt model is a good proxy for the risk of the trained model when the size of the test set is large.

A common fallacy: A false line of reasoning that is often made is the following. Given a any model $f \in \mathcal{F}$, the law of large numbers tells us that when traing set is large enough, $|\widehat{L}_S(f) - L_{\mathbf{D}}(f)|$ is small (with high probability or in expectation). Using concentration inequalities, the rate of convergence can be made precise. Hence if I train a model $\hat{f}_S$ on sampling set S and it has a small training error, then since $|\widehat{L}_S(\hat{f}_S) - L_{\mathbf{D}}(\hat{f}_S)|$ is small, its risk or test error is also small. THIS STATEMENT IS FALSE. The key thing to remember is that while for any $f \in \mathcal{F}$, $|\widehat{L}_S(f) - L_{\mathbf{D}}(f)|$ is small, this statement requires f to be picked without looking at the sample. If an f is picked looking at the sample, the statement need not be true. As an example, let $\mathcal{F}$ in fact be all possible models. In this case, think of an algorithm which looks at sample and picks a model that perfectly fits exactly the test sample and elsewhere just returns a label of 0 always. Such an algorithm while will have 0 training loss will invariably have a very large Risk. This is immediately clear when one looks at test loss which will also be large. Thus it is important to recognize the order of quantifiers. While it is true that for all f,

$$P\left(|\widehat{L}_S(f) - L_{\mathbf{D}}(f)| \text{ is large}\right) \text{ is small}$$

It need not be true that

$$P\left(\exists f \text{ s.t. } |\widehat{L}_S(f) - L_{\mathbf{D}}(f)| \text{ is large}\right) \text{ is small}$$

While a blatant version of this fallacy is not common in ML research, subtle version of this fallacy occurs all the time!

Think about your favorite ML benchmark dataset. Say CIFAR100 or imagenet. The first paper that used these dataset, the test set would have been truly blind. But for every subsequent paper written that uses these data sets, the authors of those paper would have read previous paper that talk about test performance of each of their models on the so called test set. Hence, clearly for the n'th paper that uses these dataset, the test set is not completely blind. For instance the authors might already know that model 1 did better than models 2, 3 and 4 on these datasets (meaning on the test set). This does mean that the more we use these standard datasets the more we overfit to them. Of course this is a double edged sword in practical ML. On the one hand benchmark datasets are very valuable assets that have helped drive development of some of the state of the art models. On the other hand, the more we use them the more we might overfit.

In any case, let us for now leave this issue aside. The key point for us to take home is that training error can be very different from testing error depending on our training algorithm. One would reasonable think that for simpler algorithms the difference between training and test error or risk is smaller than for more complicated one. Indeed, if our training algorithm was a dumb one that just returned a fixed hypothesis irrespective of the training set, then the deviation between test and training error is simply given by concentration inequalities. But the more complicated our algorithm is, the larger this deviation could be. We say an algorithm generalizes well if its training and test losses (risk) are close.

3 Empirical Risk Minimization and Uniform Convergence

3.1 Empirical Risk Minimizer (ERM)

Since we are interested in minimizing excess risk w.r.t. a set of models $\mathcal{F}$, it is reasonable to expect that our algorithm returns a model in $\mathcal{F}$. Perhaps the most straightforward scheme to pick such a model is to simply pick that model in $\mathcal{F}$ that has the lowest training loss. Such an algorithm is referred to as the Empirical Risk Minimization (ERM) algorithm. The algorithm simply returns

$$\hat{f}_{\text{ERM}} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_S(f)$$

We use belongs to above as there could be multiple minimizers in which case we pick any one of them. This algorithm is one of the most well studied algorithm in statistical learning theory. One would hope, that at least for simple models, since the ERM minimizes training loss, its test loss should also be as small as the best model. Indeed, if $\mathcal{F}$ had only one function this is true by law of large numbers (concentration). But what about in the more general case?

3.2 ERM and Uniform Convergence

We already talked about $\forall f, P\left(|\widehat{L}_S(f) - L_{\mathbf{D}}(f)| \text{ is large}\right)$ Vs $P\left(\exists f, |\widehat{L}_S(f) - L_{\mathbf{D}}(f)| \text{ is large}\right)$ and how they can be different. The latter quantity it turns out plays an important role in understanding

the performance of ERM in general. To see this, consider the following simple observation.

$$\begin{aligned}
P\left(L_{\mathbf{D}}(\hat{f}_{\text{ERM}}) - \min_{f\in\mathcal{F}} L_{\mathbf{D}}(f) > 2\epsilon\right) &= P\left(L_{\mathbf{D}}(\hat{f}_{\text{ERM}}) - \widehat{L}_S(\hat{f}_{\text{ERM}}) + \widehat{L}_S(\hat{f}_{\text{ERM}}) - \min_{f\in\mathcal{F}} L_{\mathbf{D}}(f) > 2\epsilon\right) \\
&= P\left(L_{\mathbf{D}}(\hat{f}_{\text{ERM}}) - \widehat{L}_S(\hat{f}_{\text{ERM}}) + \max_{f\in\mathcal{F}}\left(\widehat{L}_S(\hat{f}_{\text{ERM}}) - L_{\mathbf{D}}(f)\right) > 2\epsilon\right) \\
&\leq P\left(L_{\mathbf{D}}(\hat{f}_{\text{ERM}}) - \widehat{L}_S(\hat{f}_{\text{ERM}}) + \max_{f\in\mathcal{F}}\left(\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right) > 2\epsilon\right) \\
&\leq P\left(\max_{f\in\mathcal{F}}\left|\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right| > \epsilon\right) \qquad (1)
\end{aligned}$$

Thus we have shown that the probability that the excess risk of ERM is larger than some 2ϵ is upper bounded by the probability that $\max_{f\in\mathcal{F}}\left|\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right|$ is larger than ϵ. The term on the right is referred to as uniform convergence since we are asking uniformly over the model class, what is the probability that average and expectation deviate by some threshold. Now say for any δ, we are able to find an $\epsilon(\delta, n)$ such that $P\left(\max_{f\in\mathcal{F}}\left|\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right| > \epsilon(\delta,n)\right) \leq \delta$ then we can conclude that for any $\delta > 0$, with probability at least $1-\delta$ over samples,

$$L_{\mathbf{D}}(\hat{f}_{\text{ERM}}) - \min_{f\in\mathcal{F}} L_{\mathbf{D}}(f) \leq 2\epsilon(\delta, n)$$

Hence the whole game now is to bound $P\left(\max_{f\in\mathcal{F}}\left|\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right| > \epsilon\right)$.

3.3 Finite class of Models

Let us first start with a finite class of models $\mathcal{F}$ and see how our bounds on learning depend on n and $|\mathcal{F}|$ and δ. For majority of this course we will assume that the loss function is bounded (in absolute) by some constant. In fact, we may assume w.l.o.g. its bounded by 1 because even if loss is bounded by some other number we simply divide the loss by that number and this modified loss is bounded by 1 (in the absolute). In this case, for any fixed $f \in \mathcal{F}$, $\ell(f(x_t), y_t)$'s are iid random variables whose expected value is $\mathcal{L}_D(f)$ by definition. Now Hoeffding's inequality (see reference material) tells us that for any iid random variables $Z_1, \ldots, Z_n$ that are bounded by 1 (ie. each $|Z_i| \leq 1$), for any $\epsilon > 0$,

$$P\left(\left|\frac{1}{n}\sum_{t=1}^{n} Z_t - \mathbb{E}\left[Z\right]\right| > \epsilon\right) \leq 2\exp(-n\epsilon^2/2)$$

Hence, if for any $f \in \mathcal{F}$ we define $Z_t^f = \ell(f(x_t), y_t)$, then noting that $Z_1^f, \ldots, Z_n^f$ are iid random variables bounded by 1, we conclude from Hoeffding's inequality that: for any $f\mathcal{F}$

$$P\left(\left|\widehat{L}_S(f) - L_{\mathbf{D}}(f)\right| > \epsilon\right) \leq 2\exp(-n\epsilon^2/2)$$

This in itself is not enough since this is not a uniform convergence bound. However, since we are considering a finite hypothesis class, lets use the above with union bound. Union bound tells us

that:

$$P\left(\max_{f\in\mathcal{F}}\left|\widehat{L}_S(f)-L_{\mathbf{D}}(f)\right|>\epsilon\right)=P\left(\exists f\in\mathcal{F}\text{ s.t. }\left|\widehat{L}_S(f)-L_{\mathbf{D}}(f)\right|>\epsilon\right)$$
$$\leq\sum_{f\in\mathcal{F}}P\left(\left|\widehat{L}_S(f)-L_{\mathbf{D}}(f)\right|>\epsilon\right)$$
$$\leq\sum_{f\in\mathcal{F}}2\exp(-n\epsilon^2/2)=2|\mathcal{F}|\exp(-n\epsilon^2/2)$$

Now notice that in the above for any $\delta>0$ if we set ϵ to be such that $2|\mathcal{F}|\exp(-n\epsilon^2/2)=\delta$, or in other words, if we set $\epsilon=\sqrt{\frac{2\log(2|F|/\delta)}{n}}$ then we can conclude that:

$$P\left(\exists f\in\mathcal{F}\text{ s.t. }\left|\widehat{L}_S(f)-L_{\mathbf{D}}(f)\right|>\sqrt{\frac{2\log\left(2|F|/\delta\right)}{n}}\right)\leq\delta$$

Plugging this back in Eq. 1 we can conclude that:

$$P\left(L_{\mathbf{D}}(\hat{f}_{\text{ERM}})-\min_{f\in\mathcal{F}}L_{\mathbf{D}}(f)>2\sqrt{\frac{2\log\left(2|F|/\delta\right)}{n}}\right)\leq\delta$$

Rewriting this (in terms of the complement) we get that for any $\delta>0$, with probability at least $1-\delta$ over samples,

$$L_{\mathbf{D}}(\hat{f}_{\text{ERM}})-\min_{f\in\mathcal{F}}L_{\mathbf{D}}(f)\leq\sqrt{\frac{8\log\left(2|F|/\delta\right)}{n}}$$

Thus we see that in the general bounded loss case, we can still conclude that with high probability (at least $1-\delta$), excess risk of ERM w.r.t. model class $\mathcal{F}$ is bounded by $O\left(\sqrt{\frac{\log(|\mathcal{F}|/\delta)}{n}}\right)$. That is a bound that goes to 0 as $1/\sqrt{n}$ and has only a logarithmic dependence on $|\mathcal{F}|$ and on $1/\delta$.

3.4 What about Infinite set of Models $\mathcal{F}$?

In practice, we often are interested not in a finite set of models but rather an infinite set. For example, the set of all K layer neural network models or the set of all halfspaces etc. How should one deal with infinite set of models?

Well a first cut approach is to approximate an infinite set by a finite set. For instance, for a given $\mathcal{F}$, if we are able to come up with a finite $\mathcal{F}'$ such that for any $f\in\mathcal{F}$, there is an $f'\in\mathcal{F}'$ such that for all x,y, $\ell(f(x),y)$ is close to $\ell(f'(x),y')$. Then we can pay an additive factor for approximation and just think of the model class as just being the finite $\mathcal{F}'$. However, think of the simple problem of binary classification where $\mathcal{X}=[0,1]$ (the interval between 0 and 1). and think about $\mathcal{F}$ as being all possible real valued threshold such that anything to the left of the threshold is labeled -1 and anything to the right is labeled $+1$. In this case, its not hard to see that no finite $\mathcal{F}'$ can approximate $\mathcal{F}$ to an approximation error better than $1/2$. However it turns out that using ERM, thresholds can easily be learnt. Why is this? Think about how expressive is the class of thresholds on a given set of n points. Why should this matter?