Informal Notes for Mathematics for Machine Learning
Imperial College London

Lecture 11: Generalization and Cross-Validation

Lecturer: Matthew Wicker

1 Learning Objectives

In this lecture we will continue to expose the mathematics of model comparison and validation. In particular, we start by looking at a few more advanced concentration inequalities than those we were able to prove in the last lecture. We will then provide a first look at generalization bounds for machine learning models which will motivate a more formal discussion of regularization than we have had to this point. We will end with cross-validation as a practical systematic way of making and validating our modelling choices.

2 Recalling Weak Law of Large Numbers

The Weak Law of Large Numbers states that for a sequence of independent and identically distributed random variables $X_1, X_2, \ldots, X_n$ with finite mean μ and variance σ^2, the sample average $\bar{X}_n$ converges in probability to μ as n approaches infinity. Mathematically,

$$\lim_{n\to\infty} P\big(\big|\bar{X}_n - \mu\big| \geq \epsilon\big) = 0 \quad \text{for any} \quad \epsilon > 0.$$

In our previous lecture we more concretely defined this notion of convergence with random variables and how it differs from our previous definition of convergence. The key observation this law makes is that as the sample size increases, the sample mean becomes a more reliable estimator of the population mean. In machine learning, the WLLN is often applied in the context of empirical risk minimization. Let $\mathcal{L}(\theta)$ be a loss function parameterized by θ, and let $Z_1, Z_2, \ldots, Z_n$ be independent and identically distributed random variables that are given by the loss function at particular input samples. Though we have not yet described it as such, the empirical risk is defined as:

$$R_{\text{emp}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\theta; Z_i).$$

By the WLLN, as n increases, $R_{\text{emp}}(\theta)$ converges in probability to the true risk $R(\theta)$. While the WLLN provides theoretical guarantees, practical applications in machine learning may require additional assumptions and considerations. Finite sample sizes, model complexity, and the nature of the data distribution can impact the convergence behavior. We will start this lecture by looking at more advanced concentration inequalities and then will see how this applies more concretely to bounding generalization.

Use of Hoeffdings not on Exam

2.1 Hoeffding's Inequality

Hoeffding's bound is a very strong tool that extends the Markov and Chebychev inequalities we have seen in previous lecture. Though we derived those bounds in previous lectures, we will just state and discuss Hoeffding's inequality here:

Theorem 2.1. ***(Hoeffding's Inequality):*** *Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely for all i, and let $\bar{X}_n$ be their average. Then, for any $\epsilon > 0$,*

$$P\left(\left|\bar{X}_n - \frac{1}{n}\sum_{i=1}^{n} E[X_i]\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-2n\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Typically, when we have our i.i.d. condition $a_i = a_j, \forall i, j$ and $b_i = b_j, \forall i, j$ which greatly simplifies the bound. For example, setting each $a_i = 0$ and $b_i = 1$ we have that the inequality becomes:

$$P\big(\big|\bar{X}_n - E[X]\big| \geq \epsilon\big) \leq 2\exp\big(-2n\epsilon^2\big).$$

Thus $\sum_{i=1}^{n}(b_i - a_i)^2$ plays the same role as our variance in prior concentration inequalities. A proof of Hoeffding's inequality typically requires proof of Chernoff's bound which is a related and useful concentration inequality that requires moment generating functions which is beyond the scope of this class. [1]

$\delta \leq 2e^{-2n\epsilon^2}$

$\ln(\frac{\delta}{2}) \leq -2n\epsilon^2$

$\frac{1}{2\epsilon^2}\ln(\frac{\delta}{2}) \geq n$

Sample complexity Suppose we choose

$$n \geq \frac{1}{2\epsilon^2}\log\frac{2}{\delta}.$$

Then, with probability at least $1 - \delta$, the difference between the empirical mean $\frac{1}{n}\sum_{i=1}^{n} X_i$ and the true mean $E[X]$ is at most ϵ.

3 Generalization Error Bounds

idk if this is examinable

Our concentration inequalities to this point have focused on taking the statements we make about a test-set and ensuring that they hold in general when we deploy our model. However, a critical use of concentration bounds is to the development of generalization error of a class of functions F. In our last lecture, we discussed universal function approximation theorems which state that certain collections of functions are able to approximate continuous functions up to arbitrary precision. It is intuitive that we would use the notion of generalization to compare models, yet our analysis to this point has focused on the generalization statements we can make about a single parameter θ.

[1] Reference for proof of Hoeffding's inequality: https://cs229.stanford.edu/extra-notes/hoeffding.pdf

Consider the entire probability of one parameter from the *hypothesis space* $\theta \in \Theta$ exceeding our desired bound:

$$\mathbb{P}\left[\sup_{\theta\in\Theta}|R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right] = \mathbb{P}\left[\bigcup_{\theta\in\Theta}|R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right]$$

A full discussion of the above equality requires a bit more math (real analysis) than is within the scope of the course. But, to give intuition: the right-hand probability is the probability that the worst-case difference between the true and empirical error exceeds ϵ while the left-hand probability is the probability of the union of of the same event for each parameter. To provide informal intuition: If there exists at least one hypothesis θ with a large deviation between its true risk and empirical risk, then the union over all hypotheses will also include that event. Conversely, if the union over all hypotheses has a large deviation, it means that at least one hypothesis must have a large deviation. The left-hand probability is one we have not yet seen how to deal with, so let's take a tangent to develop some tools to do so.

3.1 Union Bound

The union bound is a fundamental concept in probability theory that provides an upper bound on the probability of the union of multiple events. Let $A_1, A_2, \ldots, A_n$ be events in a probability space. The union bound states that the probability of the union of these events is no greater than the sum of their individual probabilities:

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i)$$

The intuition behind the union bound is straightforward. When considering the probability of the union of events, the probability of the combined event is at most the sum of the probabilities of the individual events. This bound becomes particularly useful when dealing with situations where it is challenging to directly compute the probability of the union. Let's derive the union bound using the inclusion-exclusion principle. The inclusion-exclusion principle provides a systematic way of counting the elements in the union of multiple sets. For two events A and B, the inclusion-exclusion principle is given by:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

It is clear to see that removing the term $P(A \cap B)$ means that we are not considering the double counting and thus no longer have equality, but an upper-bound on the probability of interest. Generalizing this idea to more than two events gives us exactly the union bound.

3.2 Assembling a Generalization Error Bound

Applying the union bound to term on the left-hand probability we get:

$$\mathbb{P}\left[\bigcup_{\theta\in\Theta}|R(\theta) - R_{\text{emp}}(\theta)| > \epsilon\right] \leq \sum_{\theta\in\Theta}|R(\theta) - R_{\text{emp}}(\theta)| > \epsilon$$

Now, we have taken the bound we did not know how to work with and written it in terms of a sum that we can apply our concentration inequalities to; using Hoeffding's inequality, we have:

$$\mathbb{P}\left(\sup_{\theta\in\Theta}|R(h)-R_{\text{emp}}(\theta)|>\epsilon\right)\leq 2|\Theta|\exp\left(-2n\epsilon^2\right)$$

where $|\Theta|$ is the size of the hypothesis space. Denoting the right-hand side of the above inequality by δ, we can say that with confidence $1-\delta$:

$$|R(h)-R_{\text{emp}}(\theta)|\leq\epsilon\Rightarrow R(h)\leq R_{\text{emp}}(\theta)+\epsilon$$

With some basic algebra, we can express ϵ in terms of δ and get:

$$R(\theta)\leq R_{\text{emp}}(\theta)+\sqrt{\frac{\log(|\Theta|)+\log(2/\delta)}{2n}}$$

This is our first generalization bound, stating that the generalization error is bounded by the training error plus a function of the hypothesis space size and the dataset size. We can also see that the bigger the hypothesis space gets, the bigger the generalization error becomes. We will not dive deeper into generalization bounds than this rudimentary bound, but it does give you an important look into the techniques used by statistical learning theory. Additionally, it gives us good intuitive insights into how to improve the generalization performance of our algorithm including limiting the size of the hypothesis space which can be done via regularization.

4 Regularization

Above we saw that by simplifying our hypothesis space, we are able to potentially make stronger guarantees about the generalization performance of our machine learning models. But how can we practically go about that?

4.1 Explicit Regularization

One approach is to *explicitly* regularize our loss. That is, we explicitly add a term to our loss function that introduces constraints or limits our hypothesis class. Weight decay is perhaps the most well-studied version of explicit regularization. The ridge regression or ℓ_2 weight decay loss is given as:

$$\mathcal{L}_{\text{ridge}}(\theta):=\mathcal{L}(\theta)+\alpha||\theta||_2$$

The effect of this term is that weight values that have high ℓ_2 magnitude also incur high loss. Thus, in our polynomial regression, all coefficients including those on higher-order terms have reduced magnitude. As practice in deriving maximum likelihood estimates, one potential practice problem is to derive the OLS estimator for ridge regression and show when and where it is equivalent to the MLE of a Gaussian likelihood linear regression and MAP estimator. In our last lecture we will see many forms of explicit regularization for developing more trustworthy machine learning.

4.2 Implicit Regularization

Many modelling choices in machine learning *can* be seen as implicit regularization. However, because implicit regularization occurs as a side-effect of making particular choices (early stopping) it can be difficult to analyze mathematically. However, understanding the regularization effects of model choices in deep learning is a rich area of research that greatly helps us understand our models and even develop better models, e.g., MC dropout.

5 Cross-validation

Having discussed universal approximation and used generalization bounds to motivate regularization we might wonder how we can systematically make such regularization decisions. We have already seen that using a test-set can help us measure generalization; however, it is important to understand why we cannot use the same test-set to measure generalization and make our hyper-parameter choices (discussed in class). Thus, we must further split our data. We divide out not just test and training sets but further split our training set into training and validation sets.[2] In the following sections, we will look at using this additional held-out validation set in a framework known as cross-validation. The most commonly used form of cross-validation is k-fold cross-validation.

5.1 K-Fold Cross-Validation

In k-fold cross-validation, the dataset is divided into k mutually exclusive subsets, or folds, of approximately equal size. The model is trained k times, each time using $k-1$ folds for training and the remaining fold for validation. This process is repeated k times, with each of the k folds used exactly once as the validation data. The performance measures from each run are then averaged to obtain a more robust evaluation of the model. Mathematically, if we denote the dataset as $\mathcal{D}$ and the model as M, the k-fold cross-validation process can be expressed as follows:

$$\text{CV}(\mathcal{D}, M) = \frac{1}{k} \sum_{i=1}^{k} R(\mathcal{D}_{\text{Train},i}, \mathcal{D}_{\text{Valid},i})$$

where $\mathcal{D}_{\text{Train},i}$ and $\mathcal{D}_{\text{Valid},i}$ represent the i-th training and validation subsets, respectively.

5.2 Leave-One-Out Cross-Validation (LOOCV)

A special case of k-fold cross-validation is leave-one-out cross-validation (LOOCV), where k is set to the total number of samples in the dataset. In each iteration, a single data point is used for validation, and the model is trained on the remaining $n-1$ data points. This process is repeated n times, where n is the total number of samples.

[2] Because of our lead into this topic being regularization and hyper-parameter selection, we are starting our cross-validation exposition with what is considered the inner loop of *nested* cross-validation. The distinction is made at the end of this section.

Test your intuition To test your intuition about this, it can be a good exercise to explore the complexity and trade-offs of picking various parameters of k-fold cross validation.

5.3 Full Algorithm

The algorithm for training an ML algorithm (as we have described above) with cross-validation, is then described as taking as input a model M (including a hyper-parameter selection) and our training set split into K folds. We then train the algorithm using $K-1$ folds for training and estimating the generalization with the remaining fold. We can estimate the hyper-parameters performance using the average of K runs where each fold is used as a validation set at some point. Finally, we can use the test set that we isolated to estimate the generalization error of each hyper-parameter.

5.4 Without hyper-parameter selection

What we have described above is how one would use cross-validation to perform hyper-parameter selection while performing error estimation separately. However, if one only cares about estimating the error, then one can use the cross validation we described above but without the existence of an additional held-out test set. It should be noted, however, that cross-validation may be the wrong tool to use to estimate generalization error of a given model parameter since the estimand of cross-validation is not the prediction error and the variance of the estimator is poorly understood (i.e., has no unbiased estimator).

A Lecture 10 & 11: Generalisation, Test Sets, Monte Carlo

Question 1 (Independence of Losses)**.** Under the iid assumption, the loss can be seen as a transformation of a random variable. Show that the losses are independent.

Question 2 (Basic Monte Carlo Estimate)**.** Consider the following integral over the indicator function $\mathbb{I}(\cdot)$, which takes value 1 when its argument evaluates to `True`:

$$I = \int_{-1}^{1} \int_{-1}^{1} \mathbb{I}(x^2 + y^2 < 1) \mathrm{d}x \mathrm{d}y \,. \tag{1}$$

a. Write this integral as an expectation.

b. Construct a Monte Carlo estimate $\hat{I}$ for this integral.

c. Bonus: Implement this in Python and verify that the value converges to π.

Question 1 (Independence of Losses)**.** Under the iid assumption, the loss can be seen as a transformation of a random variable. Show that the losses are independent.

y_i: actual $\hat{y}_i$: predicted x_i: input $f^\theta(\cdot)$: model

$\hat{y}_i = f^\theta(x)$

$z_i = \ell(f^\theta, x_i = x_i, y_i = y_i)$

If

Question 2 (Basic Monte Carlo Estimate)**.** Consider the following integral over the indicator function $\mathbb{I}(\cdot)$, which takes value 1 when its argument evaluates to `True`:

$$I = \int_{-1}^{1}\int_{-1}^{1} \mathbb{I}(x^2 + y^2 < 1)\mathrm{d}x\mathrm{d}y\,. \tag{1}$$

a. Write this integral as an expectation.

b. Construct a Monte Carlo estimate $\hat{I}$ for this integral.

c. Bonus: Implement this in Python and verify that the value converges to π.

a. Assume $x, y \in [-1, 1)$ $\rightarrow$ $p(x) = \frac{1}{b-a} = \frac{1}{1-(-1)} = \frac{1}{2} = p(y)$

$E[f(x,y)] = \int_{-1}^{1}\int_{-1}^{1} p(x,y) f(x,y)\,dx\,dy$ x, y indep? $\rightarrow p(x,y) = p(x)p(y) = \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4}$

$= \int_{-1}^{1}\int_{-1}^{1} \frac{1}{4} f(x,y)\,dx\,dy = \frac{1}{4} I$

$\therefore I = 4E[f(x,y)]$

b. Use sample mean instead of integral

$$I = 4E[f(x,y)] \approx 4 \times \frac{1}{N}\sum_{i=1}^{N} I(x_i^2 + y_i^2 < 1) = \hat{I}$$

as $N \to \infty$, converges to actual expectation due to WLLN

$(4 \times 3)^2$

c. ...

d. $E[\hat{I}^2] = E\left[\left(\frac{1}{N}\sum f(x_n)\right)^2\right] = E\left[\left(\frac{1}{N}\sum_m f(x_m)\right)\left(\frac{1}{N}\sum_n f(x_n)\right)\right]$

$= \frac{1}{N^2} E\left[\left(\sum f(x_m)\right)\left(\sum f(x_n)\right)\right] = \frac{1}{N^2} E\left[\left(f x_1 + f x_2 + \cdots\right)\left(f x_1 + f x_2 + \cdots\right)\right]$

$= \frac{1}{N^2}\left(E\left[\sum_{n \neq m} f(x_n) f(x_m)\right] + E\left[\sum_{n=m} f(x_n) f(x_m)\right]\right)$

$= \frac{1}{N^2}\left(\sum_{n \neq m} E[f(x_n) f(x_m)] + \sum_{n} E[f(x_n) f(x_n)]\right)$