Lecture Notes of Advanced Machine Learning (CS 7140)

Hongyang R. Zhang

January 15, 2025

Contents

1 Overview

1.1 What is this course about? (Lecture 1)

Machine learning has been increased used in technology platforms and products, affecting our daily lives.[1] Machine learning involves a collection of models, algorithms, and engineering frameworks:

- Regression and classification: least squares estimation, logistic regression, LDA, bias-variance tradeoff, cross-validation.
- Neural networks and deep learning: CNNs, backpropagation, foundation models, language modeling.
- Unsupervised learning: dimension reduction (e.g., PCA), clustering, contrastive learning.
- Causal machine learning: study the cause-and-effect with a powerful machine learning model.
- Generative AI: diffusion models, multi-modal learning.
- NumPy, Sklearn, PyTorch, TensorFlow, Hugging Face.

This course aims to uncover the common statistical principles underlying the diverse array of methods. This class is mostly about the theoretical analysis of learning algorithms and models. Many of the techniques introduced in this course—which involve a beautiful blend of probability, linear algebra, and optimization—are separate fields in their respective discipline with independent interests outside of machine learning. For example, we will study the supreme of a complex random variable corresponding to the outcome of a learning algorithm applied to train a neural network model. We will show how to design estimation algorithms when we are working under distribution shifts between training and test datasets.

From a practical point of view, studying the underlying working mechanisms of a learning algorithm can deepen our understanding of how things work. For example, suppose we want to build a classifier to predict the topic of a document (e.g., sports, politics, technology, etc). We train a logistic regression model with word frequencies as features and obtain a training accuracy of 90% on 1000 training documents and a test accuracy of 85% on 1000 test documents.

- How reliable are these numbers? If we resample the training data, can we expect the same results?
- How much will the training and test accuracies increase if we double the number of training documents?
- What if we increase the number of features (e.g., use tri-occurrence of words)? Does regularization help?

[1] ChatGPT reportedly has 300 million weekly active users: CNCB news, 2024; Claude reportedly has 4.5 million monthly active users, anthropic.

There is obviously a clear gap between theoretical analysis and the practical performance of an algorithm. For instance, theoretical analysis is usually conducted under strong assumptions, which limit the implications one could draw from the theoretical results. The goal, instead, is to build a deeper understanding through theoretical analysis.

The course materials are divided into three parts: *fundamental concepts of statistical learning* (January), *generalization of neural networks and deep learning* (February), *statistical modeling of representation learning, reinforcement learning, and beyond* (March).[2]

1.2 Basic setup of supervised learning (Lecture 1)

Central questions: *Does minimizing training error lead to low test error? How does the generalization ability depend on the model architecture and the training algorithm?* It turns out that answering these questions is highly non-trivial as it also depends on the underlying data distribution.[3]

To formally study these questions, let us first describe the mathematical setup:

- Let $\mathcal{X}$ denote the feature space. Let $\mathcal{Y}$ denote the space of all possible outcomes. Binary classification example: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{+1, -1\}$
- Consider the problem of predicting an output $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$.
- Let $\mathcal{H}$ be a set of hypotheses. Linear model example:

$$\mathcal{H} = \left\{ x \to \beta^\top x + \eta : \forall \beta \in \mathbb{R}^d, \eta \in \mathbb{R} \right\}$$

- Let $\ell : (\mathcal{X}, \mathcal{Y}) \times \mathcal{H} \to \mathbb{R}$ be a loss function. For example, the mean squared error (MSE) applied to linear models is

$$\ell((x, y), \beta) = \left(\beta^\top x + \eta - y \right)^2, \forall \beta \in \mathbb{R}^d, \forall \eta \in \mathbb{R}$$

- Given n training data samples, denoted by $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, the training loss (or empirical risk) of a hypothesis $h \in \mathcal{H}$ is defined as

$$\hat{L}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i), \forall h \in \mathcal{H} \tag{1}$$

We make a critical assumption about the data-generating process. We assume that every x_i, y_i pair is drawn independently and identically from an unknown distribution $\mathbb{P}^\star$, supported on $\mathcal{X} \times \mathcal{Y}$.

The test loss (or expected risk) of a hypothesis $h \in \mathcal{H}$ is then given by

$$L(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathbb{P}^\star} \left[\ell(h(x), y) \right]. \tag{2}$$

[2] April will be dedicated to course project presentations.

[3] A recent empirical study highlights empirical scaling laws as key metrics for training large language models: paper (see also openai page).

Remarks:

- We have assumed the training and test distributions are the same. While this assumption does not hold exactly in practice, morally speaking, the training and test distributions have to be related.
- Formulating what it means to be related and not related, and dealing with the discrepancy between training and test data is studied under the area of domain adaptation or transfer learning.
- The independence assumption, which also does not hold exactly in practice, ensures that more training data gives us more information.

Empirical risk minimization: Consider minimizing the training loss

$$\hat{h}_{\text{ERM}} \leftarrow \operatorname*{argmin}_{h \in \mathcal{H}} \hat{L}(h). \tag{3}$$

What can say that the relationship between $\hat{L}(\hat{h}_{\text{ERM}})$ and $L(\hat{h}_{\text{ERM}})$? A key challenge is that the randomness of $\hat{h}_{\text{ERM}}$ now depends on $\hat{L}$. Thus, $\hat{L}(\hat{h}_{\text{ERM}})$ involves a correlation between the training data samples and the minimizing hypothesis. A central aspect we will tackle in the first part of the course is to develop the machinery to tackle this challenge.

Uniform convergence: We show provide statements of the following flavor

> With probability at least $1-\delta$, the gap between test loss and training loss of any hypothesis is upper bounded by some small ϵ, that is, $L(h) - \hat{L}(h) \leq \epsilon$, where the ϵ is generally a function that depends on δ and other aspects of the learning algorithm/model

More rigorously, we would like to show statements of the following:

$$\Pr\left[L(h) - \hat{L}(h) > \epsilon\right] \leq 1 - \delta, \tag{4}$$

where the randomness is on the training data samples drawn from $\mathbb{P}^\star$.

Equipped with such a statement, we will then apply the statement to the empirical risk minimizer $\hat{h}_{\text{ERM}}$, since the result essentially holds for any $h \in \mathcal{H}$, which also subsumes $\hat{h}_{\text{ERM}}$ as a special case.

1.3 Basic setup of neural networks (Lecture 1)

Consider the case of a basic one-layer network:

$$x \to \sum_{i=1}^{m} a_i \sigma(w_i^\top x + b_i), \text{ where} \tag{5}$$

- σ is the nonlinear activation function. Typical choices of σ: ReLU $x \to \max(0, x)$, sigmoid $x \to \frac{1}{1+\exp(-x)}$
- $Z = \{\alpha = (a_i, w_i, b_i)\}_{i=1}^m$ are trainable parameters of the network. By varying them we could define the function class $\mathcal{H}$ as

$$\mathcal{H} = \{f_\alpha : \forall \alpha \in Z\}$$

- $\mathcal{H}$ essentially represents a set of one-hidden-layer neural networks with m neurons
- We can define the weight matrix $W = [w_1, w_2, \ldots, w_m]$, and the bias vector $b = [b_1, b_2, \ldots, b_m]$. Thus, we may write map (5) as $x \to a^\top \sigma(Wx + b)$.

By extending the above setup, we may write a deep network as

$$f_\alpha(x) = \sigma_L \left(W_L \sigma_{L-1} \left(\cdots \sigma_2 \left(W_2 \sigma_1 \left(W_1 x + b_1 \right) + b_2 \right) \cdots \right) \right) \tag{6}$$

The depth of the network is given by L. The width is given by $\max(m_1, m_2, \ldots, m_l)$, i.e., the layer with the most neurons in the layer.

Motivating questions: *How could we analyze the training and test losses of a deep network? How well does a deep network generalize, and how does it depend on its depth and width? How does this ability to learn and to generalize rely on the data distributions, and what is the role of optimization algorithms used to train the network?*

Next lecture: In the next lecture, we will wrap up this overview by giving a setup about how to rigorously model transfer learning, and reason about estimation procedures whose test data is different from the training data. Then, we will dive deeper into the uniform convergence framework.

1.4 Statistical transfer learning (Lecture 2)

An important learning paradigm that has emerged in the past few years is transfer learning—transferring the knowledge from one task to help solve another task. How could we develop a more rigorous statistical modeling of transfer learning? A better understanding of this question has applications in NLP and language modeling, CV, robotics, to name a few.

Linear regression: Perhaps the simplest modeling framework is to examine transfer learning in linear regression tasks. For example, we may consider the case of two linear regression tasks, one called the source task and the other called the target task.

Suppose we have n_1 samples from the source task. We have n_2 samples from the target task. How could we use the samples from the source task to help estimate the target task? Concretely, let the samples of the source task be denoted by $(x_1^{(1)}, y_1^{(1)}), (x_2^{(1)}, y_2^{(1)}), \ldots, (x_{n1}^{(1)}, y_{n1}^{(1)})$, where every $x_i^{(1)}$ is a p-dimensional vector and $y_i^{(1)}$ is a real-valued outcome. We shall assume that they follow a linear relation specified by an unknown parameter $\beta^{(1)} \in \mathbb{R}^p$:

$$y_i^{(1)} = {x_i^{(1)}}^\top \beta^{(1)} + \epsilon_i^{(1)}, \text{ for all } i = 1, 2, \ldots, n_1 \tag{7}$$

Similarly, we denote the samples of the target task as $(x_1^{(2)}, y_1^{(2)}), (x_2^{(2)}, y_2^{(2)}), \ldots, (x_{n2}^{(2)}, y_{n2}^{(2)})$. In addition, they follow a linear relation specified by another unknown parameter $\beta^{(2)} \in \mathbb{R}^p$, which can be different from that of the source task:

$$y_i^{(2)} = {x_i^{(2)}}^\top \beta^{(2)} + \epsilon_i^{(2)}, \text{ for all } i = 1, 2, \ldots, n_2 \tag{8}$$

Now we can ask a few more concrete questions:

- How does the difference between $\beta^{(1)}$ and $\beta^{(2)}$ affect transfer learning performance?
- How does the difference between the feature vectors of source task and target task affect transfer learning?

More generally, we may say that the source task and the target task involve a distribution shift between their them. In the area of domain adaptation (Kouw and Loog, 2018):

- Covariate shift refers to scenarios where both tasks follow the same model conditioned on the features (i.e., $\beta^{(1)} = \beta^{(2)}$), but they have different feature distributions.
- Model shift refers to scenarios where the two tasks follow different models conditioned on the same features, that is $\beta^{(1)} \neq \beta^{(2)}$.

We may now ask, how does covariate shift and model shift affect transfer learning performance?

Transfer learning estimator: Typically, there are two strategies for transfer learning, one called hard transfer, where we hard-code the shared component across tasks, the other called soft transfer, where we use separate components for task, and encourage the separate components to be close to each other (Ruder, 2017).

In the context of linear regression, we can define an hard parameter sharing estimator as follows:

$$\hat{L}^{HPS}(\beta) = \frac{1}{n_1 + n_2} \left(\sum_{i=1}^{n_1} \left({x_i^{(1)}}^\top \beta - y_i^{(1)} \right)^2 + \sum_{j=1}^{n_2} \left({x_j^{(2)}}^\top \beta - y_j^{(2)} \right)^2 \right) \tag{9}$$

We may also elect to use a soft parameter sharing estimator instead:

$$\hat{L}^{SPS}(\beta, z) = \frac{1}{n_1 + n_2} \left(\sum_{i=1}^{n_1} \left({x_i^{(1)}}^\top (\beta + z) - y_i^{(1)} \right)^2 + \sum_{j=1}^{n_2} \left({x_j^{(2)}}^\top \beta - y_j^{(2)} \right)^2 \right) + \lambda \left\| z \right\|^2 \tag{10}$$

Essentially, by adjusting λ, we can adjust the magnitude of z, which then determines how far (and how close) the source and target task models are.

Optimality of the estimator: The above estimation algorithms are based on the best practices of practitioner (see the surveys above). Suppose we analyze their performances. However, how can we know that there are no better estimators out there? How could we understand the fundamental limits of estimation and optimization procedures? These are often called *lower bounds* on the performance of estimators, and it usually falls into the area of information theory (Duchi, 2019). In particular, we will touch on the framework of minimax lower bounds for transfer learning (though the scope of this is much broader than we'll cover in our lectures).

2 Uniform convergence and generalization

Recall that we have introduced the empirical risk and the expected risk of a hypothesis (denoted by $L(h)$ and $\hat{L}(h)$) for some h in a hypothesis class $\mathcal{H}$. Suppose we minimize the empirical risk to get $\hat{h}_{\text{ERM}}$. Two questions:

- Generalization gap: how does the expected and empirical risks compare for ERM, i.e., $L(\hat{h}_{\text{ERM}}) - \hat{L}(\hat{h}_{\text{ERM}})$? This is called the generalization gap.
- Excess risk: how well does ERM do with respect to the best possible hypothesis in the hypothesis class, i.e., $L(\hat{h}_{\text{ERM}}) - \min_{h \in \mathcal{H}} L(h)$? This is also called the excess risk.

A particularly fruitful framework for analyzing learning algorithms is the probably approximately correct (PAC) framework (Valiant, 1984):

> A learning algorithm A PAC learns a hypothesis class $\mathcal{H}$ if
>
> a) For any distribution $\mathbb{P}^\star$ supported over $\mathcal{X} \times \mathcal{Y}$, and any $\epsilon > 0$, $\delta > 0$
>
> b) Upon taking n I.I.D. samples from $\mathbb{P}^\star$, A produces an output $\hat{h} \in \mathcal{H}$ such that with probability at least $1 - \delta$ (over the randomness of the samples)
>
> $$L(\hat{h}) - \hat{L}(\hat{h}) \leq \epsilon$$
>
> c) Further, A runs in time polynomial in $n, d, \epsilon^{-1}, \delta^{-1}$ (where d is the dimension of the input)

Remark: Notice that the running time complexity places a bound on the sample complexity as well. We will assume that the empirical risk minimizer can be computed efficiently. For instance, think of a large neural network whose training loss can be efficiently reduced to reach zero using stochastic gradient descent

2.1 Learning a realizable, finite hypothesis class (Lecture 2)

The ERM framework is very general – we now give a concrete example to illustrate some basic results.

Assumptions (realizable, finite hypothesis): i) The size of the hypothesis space, $\mathcal{H}$, is finite; ii) There exists a hypothesis $h^\star \in \mathcal{H}$ such that $h^\star$ achieves perfect performance, i.e.,

$$L(h^\star) = \mathop{\mathbb{E}}_{(x,y) \in \mathbb{P}^\star} [\ell(h^\star(x), y)] = 0.$$

Under these assumptions, we shall prove the following property of ERM:

Under the above assumptions, with probability $1-\delta$,

$$L(\hat{h}_{\text{ERM}}) \leq \frac{\log(|\mathcal{H}|) + \log(\delta^{-1})}{n} \tag{11}$$

In particular, to reduce the expected risk below ϵ, we want $n \geq \frac{\log(|\mathcal{H}|)+\log(\delta^{-1})}{\epsilon}$. Remarks:

- The excess risk only grows logarithmically with the size of the hypothesis class, so we affort to use an exponentially large hypothesis space.
- The result is independent of $\mathbb{P}^\star$. This is nice because typically we don't know the true distribution.

Proof. We'd like to upper bound the probability of the bad event that $L(\hat{h}_{\text{ERM}}) > \epsilon$:

a) Let $B \subseteq \mathcal{H}$ be the set of bad hypotheses: $\{B \in \mathcal{H} : L(h) > \epsilon\}$

b) We can rewrite our goal as upper bounding the probability of selecting a bad hypothesis $\Pr[L(\hat{h}_{\text{ERM}}) > \epsilon] = \Pr[\hat{h}_{\text{ERM}} \in B]$

c) Recall the empirical risk of ERM is always zero because at least $\hat{L}(h^\star) = 0$

d) Hence for any "bad" hypothesis in B, they must have zero empirical risk

$$\Pr[\hat{h}_{\text{ERM}} \in B] \leq \Pr[\exists h \in B : \hat{L}(h) = 0]$$

e) Now we shall deal with the above in two steps. First, bound $\Pr[\hat{L}(h) = 0]$ for a fixed $h \in B$.

Notice that on a random example from $\mathcal{P}^\star$, the accuracy of h should be $1 - L(h)$. Since the training data is drawn IID from $\mathcal{P}^\star$, and $L(h) \geq \epsilon$ for any $h \in B$, we get that

$$\Pr[\hat{L}(h) = 0] \leq (1 - L(h))^n \leq (1-\epsilon)^n \leq \exp^{-\epsilon n},$$

where we use the fact that $1 - x \leq \exp(-x)$.

f) Second, we want the above to hold simultaneously for all $h \in B$. We can apply the union bound to bound the probability of all bad events:

$$\begin{aligned}\Pr[\exists h \in B : \hat{L}(h) = 0] &\leq \sum_{h \in B} \Pr[\hat{L}(h) = 0] \\ &\leq |B| \exp(-\epsilon n) \\ &\leq |\mathcal{H}| \exp(-\epsilon n)\end{aligned}$$

By setting the above at most δ, we conclude that ϵ must be at least $\frac{\log(|\mathcal{H}|)+\log(\delta^{-1})}{n}$.

This concludes the proof for learning finite, realizable hypothesis spaces.

Takeaway: The proof of this result is elementary but illustrates an important pattern that will recur in more complex scenarios. We are interested in the expected risk, but only have access to empirical risk to choose the ERM:

- Step 1 (convergence): for a fixed h, show that $\hat{L}(h)$ is close to $L(h)$ with high probability
- Step 2 (uniform convergence): show that the above holds simultaneously for all hypotheses $h \in \mathcal{H}$

However, the assumptions are restrictive. There exists a perfect hypothesis (realizability). What happens when the problem is not realizable? To answer this, we introduce the tools of concentration estimates.

Second, the hypothesis class is finite. What happens when the number of hypotheses is infinite? To answer this, we need better ways of measuring the "size" of a set – leading to Rademacher complexity, VC, PAC-Bayes, etc.

2.2 Using uniform convergence to reason about generalization (Lecture 2)

We now give a high-level picture of the logic behind how we can use uniform convergence to reason about generalization (in the context of ERM). We'd like to show that ERM's excess risk is small:

$$\Pr\left[L(\hat{h}_{\text{ERM}}) - L(h^\star) \geq \epsilon\right] \leq \delta \tag{12}$$

We can expand the excess risk as

$$L(\hat{h}) - L(h^\star) = \underbrace{\left(L(\hat{h}) - \hat{L}(\hat{h})\right)}_{\text{Uniform convergence}} + \underbrace{\left(\hat{L}(\hat{h}) - \hat{L}(h^\star)\right)}_{\leq 0} + \underbrace{\left(\hat{L}(h^\star) - L(h^\star)\right)}_{\text{Concentration}} \tag{13}$$

We'll see how concentration estimates can be used to control this difference in the third part. However, the same reasoning does not apply to the first part because the ERM $\hat{h}_{\text{ERM}}$ depends on the training examples $\hat{L}$. In particular,

$$\hat{L}(\hat{h}_{\text{ERM}}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{h}_{\text{ERM}}(x_i), y_i). \tag{14}$$

Due to the correlation, the above is not a sum of independent random variables. The central thesis of uniform convergence is that if we could ensure that $L(h)$ and $\hat{L}(h)$ are close for all $h \in \mathcal{H}$, then $L(\hat{h}_{\text{ERM}})$ must be close to $\hat{L}(\hat{h}_{\text{ERM}})$ as well.

In summary, our goal of deriving a uniform convergence result can be stated as

$$\Pr[L(\hat{h}_{\text{ERM}}) - L(h^\star) \geq \epsilon] \leq \Pr\left[\left(\sup_{h\in\mathcal{H}} \left|L(h) - \hat{L}(h)\right|\right) \geq \frac{\epsilon}{2}\right] \leq \delta \tag{15}$$

In particular, the $1/2$ above comes from combining the error terms from the first and third parts together. Put it in words, we'd like to upper bound the probability that the largest difference between the empirical risk and the expected risk is larger than $\epsilon/2$.

2.3 Concentration estimates (Lecture 3)

Concentration inequalities are powerful tools from probability theory that show the average of independent random variables will be concentrated around its expectation. Concentration estimates are the basis of a large branch of learning theory (Bach, 2024) and high-dimensional statistics (Wainwright, 2019). They are one of the most basic tools for studying supervised learning algorithms and models (Zhang, 2023), primarily because much of supervised learning deals with in-distribution samples.

Example (mean estimation): Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables with mean $\mu = \mathbb{E}[X_i]$, for all $i = 1, 2, \ldots, n$. Define the empirical mean as

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

How does $\hat{\mu}_n - \mu$ behave?

Three types of statements from probability:

- Consistency: by law of large numbers,

$$\hat{\mu}_n - \mu \to 0$$

- Asymptotic normality: let the variance of X_i (for all i) be equal to σ^2, by the central limit theorem, we have

$$\sqrt{n}(\hat{\mu}_n - \mu) \sim \mathcal{N}(0, \sigma^2)$$

- Tail estimates: for our purpose, we would like to draw a statement of the following type

$$\Pr[|\hat{\mu}_n - \mu| \geq \epsilon] \leq \delta$$

  For getting such tail estimates, we typically need to study the tail of a distribution, for instance, the tail of a Gaussian distribution, etc.

Markov's inequality: Let $Z \geq 0$ be a non-negative random variable. Then

$$\Pr[Z \geq t] \leq \frac{\mathbb{E}[Z]}{t}$$

Proof: Since Z is a non-negative quantity, we always have the condition that

$$t \mathbb{1}_{Z \geq t} \leq Z$$

To see this, notice that if $Z \geq t$, then the above is true. On the other hand, if $Z < t$, then the left-hand side above is zero, whereas $Z \geq 0$. Next, take the expectation over Z on both sides, we get

$$\mathbb{E}[t \mathbb{1}_{Z \geq t}] \leq \mathbb{E}[Z] \Rightarrow \mathbb{E}[\mathbb{1}_{Z \geq t}] \leq \frac{\mathbb{E}[Z]}{t}$$

Notice that $\mathbb{E}[\mathbb{1}_{Z \geq t}] = \Pr[Z \geq t]$. Thus, we have shown that Markov's inequality is true.

Chebyshev's inequality: Let X be a random variable with mean equal to μ. Then

$$\Pr[|X - \mu| \geq \epsilon] \leq \frac{\mathrm{Var}[X]}{\epsilon^2}$$

Proof: We will use Markov's inequality to get this result. Let $Z = (X - \mu)^2$ and let $t = \epsilon^2$. Notice that $Z \geq 0$. Thus, based on Markov's inequality

$$\Pr[Z \geq t] \leq \frac{\mathbb{E}[Z]}{t} = \frac{\mathbb{E}\left[(Z-\mu)^2\right]}{t} = \frac{\mathrm{Var}[Z]}{t},$$

which completes the proof.

Hoeffding's inequality: Let $Z_1, Z_2, \ldots, Z_n$ be n independent and identically distributed random variables drawn from a distribution with expectation μ and whose values are bounded from above by one.

Let $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$ denote the mean of the n random variables. Then, for any $\epsilon \in (0, 1)$, we have

$$\Pr[|\hat{\mu}_n - \mu| > \epsilon] \leq 2\exp(-2\epsilon^2 n)$$

The Hoeffding's inequality is a very powerful result when we work with the average of n random variables. Variants of this inequality (which is restricted to bounded random variables) are also called Chernoff bound.[4] Next, we shall see a proof through the use of moment generating functions (MGF).

Definition (Moment generating function): For a random variable X, its MGF is given by

$$M_X(t) := \mathbb{E}[\exp(tX)]$$

One can also think of the MGF in terms of Taylor's expansion of $\exp(tX)$ as

$$M_X(t) = 1 + t\,\mathbb{E}[X] + \frac{t^2}{2}\,\mathbb{E}\left[X^2\right] + \frac{t^3}{6}\,\mathbb{E}\left[X^3\right] + \cdots$$

Thus, the first moment is the mean of X. The second moment is the variance of X (assuming the mean of X is zero). And so on.

Property: The MGF of the sum of two independent random variables X_1 and X_2 is the product of the MGF of X_1 and X_2, respectively.

- To see this, notice that

$$M_{X+Y}(t) = \mathbb{E}\left[e^{t(X+Y)}\right] = \mathbb{E}\left[e^{tX}e^{tY}\right] = \mathbb{E}\left[e^{tX}\right]\mathbb{E}\left[e^{tY}\right] = M_X(t)M_Y(t)$$

- Here we have used that X and Y are independent, and hence e^{tX} and e^{tY} are independent, to conclude that $\mathbb{E}\left[e^{tX}e^{tY}\right] = \mathbb{E}\left[e^{tX}\right]\mathbb{E}\left[e^{tY}\right]$

The high-level idea for showing the Hoeffding's inequality is obtained by applying Markov's inequality to e^{tX} for some well-chosen value t. From Markov's inequality, we can derive the following useful inequality: for any $t > 0$,

$$\Pr[X \geq a] = \Pr\left[e^{tX} \geq e^{ta}\right] \leq \frac{\mathbb{E}\left[e^{tX}\right]}{e^{ta}}$$

[4] https://en.wikipedia.org/wiki/Chernoff_bound

In particular,

$$\Pr[X \geq a] \leq \min_{t>0} \frac{\mathbb{E}\left[e^{tX}\right]}{e^{ta}} \tag{16}$$

Similarly, for any $t < 0$,

$$\Pr[X \leq a] = \Pr[e^{tX} \geq e^{ta}] \leq \min_{t<0} \frac{\mathbb{E}\left[e^{tX}\right]}{e^{ta}}$$

Bounds for specific distributions are obtained by choosing appropriate values for t.

2.3.1 Chernoff bounds for the sum of Poisson trials

We now develop the most commonly used version of the Chernoff bound for the tail distribution of a sum of independent 0-1 random variables, which are also known as Poisson trials.[5]

Let $X_1, \ldots, X_n$ be a sequence of independent Poisson trials with $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$, and let

$$\mu = \mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} p_i$$

For a given $\delta > 0$, we are interested in bounds on $\Pr[X \geq (1+\delta)\mu]$ and $\Pr[X \leq (1-\delta)\mu]$, that is, the probability that X deviates from its expectation μ by $\delta\mu$ or more. To develop a Chernoff bound, we need to compute the moment generating function of X. We start with the MGF of each X_i:

$$\begin{aligned} M_{X_i}(t) = \mathbb{E}\left[e^{tX_i}\right] = p_i e^t + (1 - p_i) &= 1 + p_i(e^t - 1) \\ &\leq e^{p_i(e^t - 1)}, \end{aligned}$$

where in the last step, we have used the fact that, for any y, $1 + y \leq e^y$. Since the X_i's are independent from each other, we take the product of the n MGF to obtain

$$\begin{aligned} M_X(t) = \prod_{i=1}^{n} M_{X_i}(t) &\leq \prod_{i=1}^{n} e^{p_i(e^t - 1)} \\ &= \exp\left(\sum_{i=1}^{n} p_i(e^t - 1)\right) = e^{(e^t - 1)\mu} \end{aligned}$$

Based on this result, we now apply Markov's inequality: for any $t > 0$, we have

$$\begin{aligned} \Pr[X \geq (1+\delta)\mu] &= \Pr[e^{tX} \geq e^{t(1+\delta)\mu}] \\ &\leq \frac{\mathbb{E}\left[e^{tX}\right]}{e^{t(1+\delta)\mu}} \\ &\leq \frac{e^{(e^t - 1)\mu}}{e^{t(1+\delta)\mu}} \end{aligned}$$

For any $\delta > 0$, we can set $t = \ln(1+\delta) > 0$ to get

$$\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$$

[5] Poisson trials differ from Poisson random variables.

For $0 < \delta \leq 1$, we need to show that

$$\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \leq e^{-\delta^2/3}$$

Taking the logarithm of both sides, we obtain

$$f(\delta) = \delta - (1+\delta)\ln(1+\delta) + \frac{\delta^2}{3} \leq 0$$

Computing the derivatives of $f(\delta)$, we have:

$$f'(\delta) = 1 - \frac{1+\delta}{1+\delta} - \ln(1+\delta) + \frac{2\delta}{3}$$
$$f''(\delta) = -\frac{1}{1+\delta} + \frac{2}{3}$$

We see that the second derivative is less than zero if $\delta < 1/2$, and it is positive otherwise. Hence, $f'(\delta)$ first decreases and then increases in the interval $[0, 1]$. Since $f'(0) = 0$ and $f'(1) < 0$, we conclude that $f'(\delta) \leq 0$ in the interval $[0, 1]$. Since $f(0) = 0$, it follows that $f(\delta) \leq 0$, which shows that for any δ between 0 and 1, we have

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$$

2.3.2 Example: Coin flips

Let X be the number of heads in a sequence of n independent fair coin flipts. Applying the Chernoff bound, we have

$$\Pr\left[\left|X - \frac{n}{2}\right| \geq \frac{1}{2}\sqrt{6n\ln n}\right] \leq 2\exp\left(-\frac{1}{3}\frac{n}{2}\frac{6\ln n}{n}\right) = \frac{2}{n}$$

This demonstrates that the concentration of the number of heads around the mean $n/2$ is very tight. Most of the time, the deviations from the mean are on the order of $O(\sqrt{n\ln n})$.

Other use cases of Chernoff bound:

- Suppose we are interested in evaluating the probability that a particular gene mutation occurs in the population. Given a DNA sample, a lab test can determine if it carries the mutation. However, the lab test is expensive and we would like to obtain a relatively reliable estimate from a small number of tests.

2.3.3 Example: Gaussian random variables

In the next example, we look at the MGF of Gaussian random variables. Let $X \sim \mathcal{N}(0, \sigma^2)$. Then, $M_X(t) = e^{\sigma^2 t^2/2}$. To derive this, we use the definition of Gaussian probability density:

$$\begin{aligned} M_X(t) = \mathbb{E}\left[e^{tX}\right] &= \int \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2 - 2\sigma^2 tx}{2\sigma^2}\right) dx \\ &= \int \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\sigma^2 t)^2 - \sigma^4 t^2}{2\sigma^2}\right) dx \\ &= \exp\left(\frac{\sigma^2 t^2}{2}\right) \end{aligned}$$

Based on the above MGF, we can derive a tail bound by plugging the form of MGF to equation (16) to get:

$$\Pr[X \geq \epsilon] \leq \inf_t \exp\left(\frac{\sigma^2 t^2}{2} - t\epsilon\right)$$

The infimum of the RHS is attained by setting $e = \frac{\epsilon}{\sigma^2}$, yielding:

$$\Pr[X \geq \epsilon] \leq \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$$

What about non-Gaussian random variables? Notice that the bound still holds if we replace $M_X(t)$ with an upper bound. This motivates the following definition.

Sub-Gaussian: A mean zero random variable X is sub-Gaussian with parameter σ^2 if its MGF is bounded as follows:

$$M_X(t) \leq \exp\left(\frac{\sigma^2 t^2}{2}\right) \tag{17}$$

It follows that for sub-Gaussian X, we again have that $\Pr[X \geq \epsilon] \leq \exp(-\epsilon^2/(2\sigma^2))$.

2.4 Learning finite hypothesis space (Lecture 4)

Recall from last lecture that we developed the Hoeffding's inequality. Next we use this result to bound the excess risk of learning a finite hypothesis class (without the realizable condition).

Learning finite hypothesis classes: Let $\mathcal{H}$ be a finite hypothesis class. Let ℓ be the zero-one loss function: $\ell(h(x), y) = \mathbb{1}_{h(x)\neq y}$. Suppose we minimize the empirical risk to get the minimizer $\hat{h} \in \mathcal{H}$. Then, with probability at least $1 - \delta$ over the randomness of training samples, the excess risk must be bounded by

$$L(\hat{h}) - L(h^\star) \leq \sqrt{\frac{2(\log(|\mathcal{H}|) + \log(2\delta^{-1}))}{n}} \tag{18}$$

We can contrast this result with (11) (of learning finite, realizable hypothesis space). The difference is that we now get a slower convergence rate (n^{-1} to $n^{-1/2}$).

Proof: Recall that the excess risk can be decomposed to

$$L(\hat{h}_{\text{ERM}}) - L(h^\star) = L(\hat{h}_{\text{ERM}}) - \hat{L}(\hat{h}_{\text{ERM}}) + \underbrace{\hat{L}(\hat{h}_{\text{ERM}}) - \hat{L}(h^\star)}_{\leq 0} + \hat{L}(h^\star) - L(h^\star)$$

Notice that the second term is at most zero. Thus, we focus on the first and the third terms.

- Use the Hoeffding's inequality to deal with the third term.
- Apply uniform convergence to upper bound the first term by $\sup_{h\in\mathcal{H}} \left|L(h) - \hat{L}(h)\right|$ (Use the union bound).

Step 1: bound $\Pr[L(h) - \hat{L}(h) \geq \epsilon]$ for a fixed $h \in \mathcal{H}$. For a fixed $h \in \mathcal{H}$, notice that $\hat{L}(h)$ is the averaged loss among n IID samples. Each of the loss term is bounded between zero and one (with expectation equal to $L(h)$). Therefore, by Hoeffding's inequality,

$$\Pr\left[\left|L(h) - \hat{L}(h)\right| \geq \epsilon\right] \leq 2\exp(-2n\epsilon^2)$$

Step 2: apply Step 1 uniformly over all possible $h \in \mathcal{H}$. In particular, we can apply union bound over all the possible bad events to get

$$\Pr\left[\sup_{h\in\mathcal{H}}\left|L(h) - \hat{L}(h)\right| \geq \epsilon\right] \leq |\mathcal{H}| \cdot 2\exp(-2n\epsilon^2) = \delta$$

By setting $\epsilon = \sqrt{\frac{2(\log|\mathcal{H}|+\log(2\delta^{-1}))}{n}}$, we can get the probability set to δ.

Remarks: We have removed the realizable assumption by suffering a $\sqrt{n}$ factor in the generalization bound. The $\sqrt{n}$ factor arises from sampling noise. It makes sense that learning is faster when there is no noise.

2.5 Rademacher complexity (Lecture 4)

Both of our generalization bounds require finite hypothesis classes. What about infinite hypothesis classes?

We can't directly apply the union bound to infinite hypothesis classes. Need a more sophisticated way to measure complexity of a hypothesis class.

With Rademacher complexity, the key idea is symmetrization. Along the way, we need an extension of the Hoeffding's inequality to some function of bounded random variables, which is known as McDiarmid's inequality. Let us first start by motivating why we need these tools.

2.5.1 Motivation

Recall that, within the uniform convergence framework, we want to get a statement of the following

$$\Pr\left[L(\hat{h}) - L(h^\star) \geq \epsilon\right] \leq \Pr\left[\sup_{h\in\mathcal{H}}\left|L(h) - \hat{L}(h)\right| \geq \frac{\epsilon}{2}\right] \leq \delta \tag{19}$$

Since there are two cases here, let us denote

$$G_n := \sup_{h\in\mathcal{H}} L(h) - \hat{L}(h), \quad G_n' := \sup_{h\in\mathcal{H}} \hat{L}(h) - L(h)$$

Then we have that

$$\Pr\left[\sup_{h\in\mathcal{H}}\left|L(h) - \hat{L}(h)\right| \geq \frac{\epsilon}{2}\right] \leq \Pr\left[G_n \geq \frac{\epsilon}{2}\right] + \Pr\left[G_n' \geq \frac{\epsilon}{2}\right]$$

Let us focus on the first case, since the second case will be similar.

Now, our main object becomes G_n: This is a rather non-trivial function, because of taking the supremum over a sum of random variables.

Hence, let's look at its expectation first! Usually, when we encounter a complicated random variable, we start by examining its first moment, then second moment, and so on.

2.5.2 Defintion of Rademacher complexity

Our main object of interest is now $\mathbb{E}[G_n] = \mathbb{E}\left[\sup_{h\in\mathcal{H}}(L(h) - \hat{L}(h))\right]$. Recall that $\hat{L}(h)$ is the empirical risk, and $L(h)$ is the expected risk.

This quantity is still quite difficult because it depends on the expected risk, an expectation over the unknown data distribution. The key idea of symmetrization is to remove this expected risk term with a "simpler" term.

Definition of Rademacher complexity: Let us imagine n data points $Z_1' = (x_1', y_1'), Z_2' = (x_2', y_2'), \ldots, Z_n' = (x_n', y_n')$, sampled from the true data distribution. Then, clearly $L(h) = \mathbb{E}\left[\hat{L}'(h)\right]$, where $\hat{L}'(h)$ is the empirical risk on this "copy" dataset.

Hence,

$$\begin{aligned}
\mathbb{E}[G_n] &= \mathop{\mathbb{E}}_{Z_{1:n}}\left[\sup_{h\in\mathcal{H}}(L(h) - \hat{L}(h))\right] \\
&= \mathop{\mathbb{E}}_{Z_{1:n}}\left[\sup_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{Z'_{1:n}}\left[\hat{L}'(h) - \hat{L}(h)\right]\right] \\
&= \mathop{\mathbb{E}}_{Z_{1:n}}\left[\mathop{\mathbb{E}}_{Z'_{1:n}}\left[\sup_{h\in\mathcal{H}}(\hat{L}'(h) - \hat{L}(h))\right]\right]
\end{aligned}$$

Let us remove the dependence on the copy dataset. To that end, we introduce IID Rademacher random variables $\sigma_1, \sigma_2, \ldots, \sigma_n$ sampled uniformly from $\{+1, -1\}$.

Notice that

$$\hat{L}'(h) - \hat{L}(h) = \sum_{i=1}^{n}(\ell(h(x_i'), y_i') - \ell(h(x_i), y_i)),$$

is symmetric around zero. Hence, multiplying each individual term by σ_i does not change its distribution. Thus,

$$\begin{aligned}
\mathbb{E}[G_n] &\le \mathop{\mathbb{E}}_{Z_{1:n}, Z'_{1:n}}\left[\sup_{h\in\mathcal{H}}(\hat{L}'(h) - \hat{L}(h))\right] \\
&= \mathop{\mathbb{E}}_{Z_{1:n}, Z'_{1:n}, \sigma_{1:n}}\left[\sup_{h\in\mathcal{H}} \sigma_i(\ell(h(x_i'), y_i') - \ell(h(x_i), y_i))\right] \\
&\le 2 \mathop{\mathbb{E}}_{Z_{1:n}, \sigma_{1:n}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(h(x_i), y_i)\right]
\end{aligned}$$

The last line (without the factor of 2) is the Rademacher complexity of $\mathcal{H}$.

In summary, let $\mathcal{H}$ be a hypothesis class consisting of a class of real-valued functions. Example: two-layer neural nets, linear models. Define the Rademacher complexity (or Rademacher average) of $\mathcal{H}$ to be

$$R_n(\mathcal{H}) := \mathop{\mathbb{E}}_{Z_{1:n}, \sigma_{1:n}}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(h(x_i), y_i)\right],$$

where $Z_i = (x_i, y_i)$ is a random sample from the underlying data distribution, and σ_i is sampled uniformly from $\{+1, -1\}$.

2.5.3 Generalization bounds based on Rademacher complexity

To see the power of the concept we just introduced, here is a very general statement where we can always rely on when we work with supervised learning algorithms and models.

Generalization bounds based on Rademacher complexity: Define

$$\mathcal{A} = \{(x, y) \to \ell(h(x), y) : h \in \mathcal{H}\}$$

to be the loss function composed with the hypothesis space. Let $\mathcal{R}_n(\mathcal{A})$ denote the Rademacher complexity of the function class $\mathcal{A}$.

With probability at least $1 - \delta$,

$$L(\hat{h}_{\text{ERM}}) - L(h^\star) \leq 4R_n(\mathcal{A}) + \sqrt{\frac{2\log(2\delta^{-1})}{n}} \tag{20}$$

To be clear, recall that $\hat{h}_{\text{ERM}}$ is the empirical risk minimizer (ERM), $h^\star$ is the expected risk minimizer, and n is the size of the training set.

Proof sketch:

- Show that empirical Rademacher complexity is close to the expectation
- Use McDiarmid's inequality, which is essentially a concentration result for functions of IID random variables. We'll introduce this tool shortly
- Show that the Rademacher complexity upper bounds the excess risk. We've seen this logic from the motivation

In more detail, recall that $G_n = \sup_{h\in\mathcal{H}} L(h) - \hat{L}(h)$. Our first claim is to show that G_n is close to $\mathbb{E}[G_n]$:

$$\Pr[G_n \geq \mathbb{E}[G_n] + \epsilon] \leq \exp(-2n\epsilon^2) \tag{21}$$

This shows that G_n is indeed close to its expectation plus a small error. Hence it suffices to upper bound its expectation.

Our second claim is to show that $\mathbb{E}[G_n]$ is upper bounded by the Rademacher complexity.

$$\mathbb{E}[G_n] \leq 2R_n(\mathcal{A}).$$

We have already seen the proof of this claim. Combined together, we can derive (20).

2.5.4 Properties of Rademacher complexity

Boundedness:

$$R_n(\mathcal{H}) \leq \max_{h\in\mathcal{H}} \max_{x,y} \ell(h(x), y)$$

This only shows that the Rademacher complexity is bounded by some constant. Usually, we'd like to show it goes to zero as n goes to infinity.

Monotonicity: If $\mathcal{H}_1 \subseteq \mathcal{H}_2$, then $R_n(\mathcal{H}_1) \leq R_n(\mathcal{H}_2)$.

This is because we now take the supreme over a larger set, hence R_n increases.

Scaling: $R_n(c \cdot \mathcal{H}) = c \cdot R_n(\mathcal{H})$.

Lipschitz composition: Suppose ϕ is a Lipschitz-continuous, bounded by some constant c_ϕ. Recall a function is Lipschitz-continuous if changing x only changes the function value by c_ϕ times.

Let $\phi \circ \mathcal{H} = \{(x, y) \to \phi(h(x), y) : h \in \mathcal{H}\}$, i.e., compose ϕ with h. Then, we have that

$$R_n(\phi \circ \mathcal{H}) \leq c_\phi \cdot R_n(\mathcal{H}).$$

The proof requires going through the definition carefully. This property is useful because we can start by studying a simpler hypothesis class, and then compose more functions with the class without going through the calculation again

Next lecture: more examples of the Rademacher complexity, plus completing the proof.
Suggested readings: Chapter 3.8 of Liang, 2016.

References

Bach, F. (2024). *Learning theory from first principles*. MIT press (page 10).

Duchi, J. (2019). "Lecture notes for statistics 311/electrical engineering 377". In: *URL: http://web.stanford.edu/class/stats311/lecture-notes.pdf* (page 6).

Kouw, W. M. and Loog, M. (2018). "An introduction to domain adaptation and transfer learning". In: *arXiv preprint arXiv:1812.11806* (page 6).

Liang, P. (2016). *CS229T/STAT231: Statistical learning theory (Winter 2016)* (page 18).

Ruder, S. (2017). "An Overview of Multi-Task Learning in Deep Neural Networks". In: *arXiv preprint arXiv:1706.05098* (page 6).

Valiant, L. G. (1984). "A theory of the learnable". In: *Communications of the ACM* 27.11, pp. 1134–1142 (page 7).

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press (page 10).

Zhang, T. (2023). *Mathematical analysis of machine learning algorithms*. Cambridge University Press (page 10).