# Lecture Notes on Statistical/Theoretical Machine Learning and Its Applications: CS7140 Spring 2026

Hongyang R. Zhang
Northeastern University

January 22, 2026

## Contents

# 1 Overview

**Background.** Machine learning has been increasingly used in technology platforms and products, affecting our daily lives. Machine learning involves a collection of models, algorithms, and engineering frameworks:

- Regression and classification: least squares estimation, logistic regression, $\ell_1/\ell_2$-regularization, bias-variance tradeoff, cross-validation.
- Neural networks and deep learning: convolutional neural networks, backpropagation, foundation models, language modeling.
- Unsupervised learning: dimension reduction such as principal component analysis, clustering, contrastive learning.
- Reinforcement learning and sequential decision-making: robotics, reinforcement learning from human feedback.
- Generative AI: large language models, diffusion models, multi-modal data.
- Causal machine learning: study the cause-and-effect to estimate the counterfactual.
- Machine learning libraries: numpy, sklearn, pytorch, tensorflow, huggingface...

## 1.1 What is this course about? (Lecture 1)

This course aims to uncover the common mathematical and statistical principles underlying the diverse array of machine learning models and algorithms. This class primarily focuses on 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 supremum 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 working under distribution shifts between the 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 machine learning models work. For example, suppose we want to design a neural network classifier to predict the sentiment of a document. We train a regression model using word frequencies as features and achieve 100% training accuracy on 1000 training documents and 85% test accuracy on 1000 test documents. How can we reduce the gap between training and test accuracy? Further, what happens if the word frequencies between the training corpus and the test corpus are different? It is possible to answer these questions from an engineering perspective; instead, this course will mostly focus on the mathematical analysis underlying these procedures, although we will provide computational examples from time to time to help you understand the theoretical concepts.

There is a clear gap between theoretical analysis and an algorithm's practical performance. For instance, theoretical analysis is usually conducted under standard technical assumptions that are often made to simplify the analysis. The goal, instead, is to *build a deeper understanding of the underlying key concepts through mathematical modeling and theoretical analysis*. We will see if we succeed in accomplishing this objective by the end of this semester.

The course materials are divided into three parts: *fundamental concepts of statistical learning theory* (January), *generalization and optimization of neural networks and deep learning* (February), and *statistical modeling of emerging learning paradigms* (March).[1]

### 1.2 Supervised prediction (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.

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:

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

- 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

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

- 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

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

**Example 1.1** (Linear regression)**.** *To make the above setup more concrete, perhaps the best example would be linear regression. There are many standard texts on this topic; see, e.g., Wainwright, 2019. In a standard parametric regression setup, we have $n$ samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where every $x_i \in \mathbb{R}^p$ is a $p$-dimensional feature vector drawn from some unknown distribution $\mathcal{D}$, and $y_i \in \mathbb{R}$, for every $i = 1, 2, \ldots, n$. In addition, suppose that there exists an unknown $\beta \in \mathbb{R}^p$ such that*

$$y_i = x_i^\top \beta + \varepsilon_i, \text{ for every } i = 1, 2, \ldots, n, \tag{3}$$

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

*where* $x_i^\top \beta = \sum_{j=1}^p x_{i,j}\beta_j$*, and* $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ *is a noise random variable with mean zero and variance* $\sigma^2$*.*

*Given* $n$ *samples, the goal of this problem is to learn a linear model parameterized by* $\hat{\beta}$ *that achieves the lowest mean-squared error (MSE) on an unseen sample.*

**Remark 1.2.** *We have assumed the training and test distributions are the same. While this assumption does not hold exactly in practice, morally, the training and test distributions must be related.*

*Formulating what it means to be related and not related, and addressing the discrepancy between training and test data, are studied in 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.*

### 1.2.1 Empirical risk minimization

Consider minimizing the training loss

$$\hat{h}_{\mathrm{ERM}} \leftarrow \operatorname*{argmin}_{h \in \mathcal{H}} \hat{L}(h). \tag{4}$$

What can say that the relationship between $\hat{L}(\hat{h}_{\mathrm{ERM}})$ and $L(\hat{h}_{\mathrm{ERM}})$? A key challenge is that the randomness of $\hat{h}_{\mathrm{ERM}}$ now depends on $\hat{L}$. Thus, $\hat{L}(\hat{h}_{\mathrm{ERM}})$ 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 developing the machinery to address this challenge.

**Example 1.3** (Pretraining and fine-tuning)**.** *An emerging learning paradigm that has emerged over the past few years follows a two-stage procedure involving pretraining on a large amount of unlabeled data, followed by fine-tuning on a small amount of labeled data.*

*The pretraining stage usually follows some masked prediction procedure on unlabeled data. The supervised fine-tuning (SFT) procedure can be formulated with the above ERM setup.*

- *Suppose we have some model like a neural network,* $f_{W_0}$*, parameterized by some initialization* $W_0$*.*
- *There is a small amount of training dataset,* $S$*, from which we compute the training loss* $\hat{L}(f_{W_0})$*.*
- *SFT corresponds to minimizing* $\hat{L}(f_{W_0})$*, usually via a stochastic gradient optimization algorithm.*
- *An important consideration in SFT is overfitting, since the model is pretrained on a large amount of unlabeled data. The size of the model is usually much larger than the size of the training dataset* $S$*.*

### 1.2.2 Uniform convergence and generalization gap

In the first part of this course, we will show various "uniform convergence" 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 $\epsilon$, that is, $L(h) - \hat{L}(h) \leq \epsilon$, where the $\epsilon$ is generally a function that depends on $\delta$ and other aspects of the learning algorithm/model

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

$$\Pr\left[\underbrace{L(h) - \hat{L}(h)}_{\text{Generalization gap}} > \epsilon\right] \leq 1 - \delta, \tag{5}$$

where the randomness is on the training data samples drawn from $\mathbb{P}^\star$. This statement essentially quantifies the generalization gap between the training and test losses of the machine learning model.

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 Multi-layer neural networks and generative models (Lecture 2)

Consider the case of a basic one-layer network:

$$f_{a,W,b}(x) := x \to \sum_{i=1}^{m} a_i \sigma(w_i^\top x + b_i), \text{ where} \tag{6}$$

- $\sigma$ is the nonlinear activation function. Typical choices of $\sigma$: ReLU $x \to \max(0, x)$, sigmoid $x \to \frac{1}{1+\exp(-x)}$. Key property: Lipschitz-continuity: A function $f : \mathbb{R} \to \mathbb{R}$ is said to be *$C$-Lipschitz-continuous* if the following is true:

$$|f(x) - f(y)| \leq C \cdot |x - y|.$$

- $Z = \{\alpha = (a_i, w_i, b_i)\}_{i=1}^m$ are trainable parameters of the network. By varying $\alpha$, we 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.

- Let $W = [w_1, w_2, \ldots, w_m]$, and $b = [b_1, b_2, \ldots, b_m]$. We may write $f_{a,W,b}$ equation (6) as $x \to a^\top \sigma(Wx + b)$.

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

$$f_\alpha(x) = \sigma_L\left(W_L \sigma_{L-1}\left(\cdots \sigma_2\left(W_2 \sigma_1\left(W_1 x + b_1\right) + b_2\right) \cdots\right)\right), \tag{7}$$

where $\alpha$ now encodes all the parameters of the network. 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?*

A *language model* specifies a conditional probability distribution $\Pr_\theta(\cdot \mid P)$, given a prompt sequence $P$, produces the next-token according to underlying probability masses.

**Example 1.4** (In-context learning)**.** *To illustrate the concept of a language model, let us consider a few-shot meta-learning problem (Garg et al., 2022). In this problem, each prompt $P_{\theta_i}$ involves a sequence of examples or demonstrations*

$$P_{\theta_i} := (x_1, y_1, x_2, y_2, \ldots, x_{t-1}, y_{t-1}, x_t),$$

*ended with a query example $x_t$. The goal is to predict the correct output $y_t$ corresponding to the query $x_t$.*

*To make this more concrete, suppose that $y_j = \theta_i^\top x_j$, for every $j = 1, 2, \ldots, t-1$. The desired output $y_t = \theta_i^\top x_j$.*

- *At training time, the model sees a sequence of prompt-answer pairs $(P_{\theta_i}, y_t)$.*
- *At test time, the model sees a new prompt $P_\theta$ parameterized by some unknown $\theta$. The model is asked to first "solve" the linear regression from the in-context examples given in $P_\theta$, and then use the "learned" regression model to output the correct answer corresponding to the query.*

## 1.4 Transfer learning and minimax estimation (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 language modeling, computer vision, robotics, to name a few.

### 1.4.1 Transfer learning setup

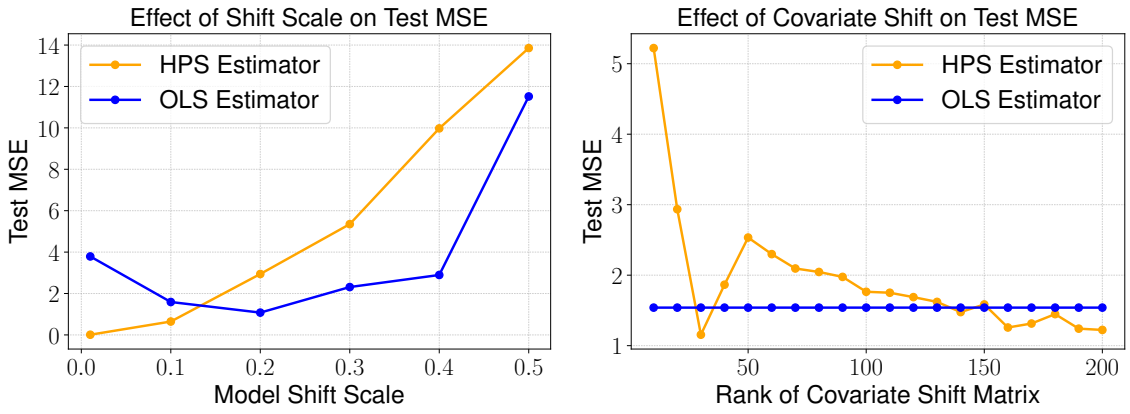
Perhaps the simplest modeling framework is to examine transfer learning in linear regression tasks. For example, we may consider the case of two linear regression tasks, one called the source task and the other called the target task.

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 $(x_1^{(1)}, y_1^{(1)}), (x_2^{(1)}, y_2^{(1)}), \ldots, (x_{n_1}^{(1)}, y_{n_1}^{(1)})$, where every $x_i^{(1)}$ is a $p$-dimensional vector and $y_i^{(1)}$ is a real-valued outcome. Similarly, we denote the samples of the target task as $(x_1^{(2)}, y_1^{(2)}), (x_2^{(2)}, y_2^{(2)}), \ldots, (x_{n_2}^{(2)}, y_{n_2}^{(2)})$.

Now we can ask a few more concrete questions:

- How does the difference between the $x \to y$ mappings affect transfer learning performance?
- How does the covariance between the feature vectors of source and target tasks 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):




(a) Model shift (b) Covariate shift

Figure 1: Illustrating the effects of model shift and covariate shift in transfer learning linear regression.

- Covariate shift refers to scenarios where both tasks follow the same model conditioned on the features, but they have different feature distributions.
- Model shift refers to scenarios where the two tasks follow different models conditioned on the same features.

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

### 1.4.2 Transfer learning estimators

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; Dhifallah and Lu, 2021).

**Example 1.5** (Illustration of model and covariate shifts in linear regression)**.** *We shall assume that the source task follows a linear relation specified by an unknown parameter* $\beta^{(1)} \in \mathbb{R}^p$*:*

$$y_i^{(1)} = {x_i^{(1)}}^\top \beta^{(1)} + \epsilon_i^{(1)}, \text{ for all } i = 1, 2, \ldots, n_1, \tag{8}$$

*where* $\epsilon_i^{(1)}$ *is a white noise with mean zero and variance* $\sigma_1^2$*.*

*We further assume that the target task follow another linear relation specified by an unknown parameter* $\beta^{(2)} \in \mathbb{R}^p$*:*

$$y_i^{(2)} = {x_i^{(2)}}^\top \beta^{(2)} + \epsilon_i^{(2)}, \text{ for all } i = 1, 2, \ldots, n_2, \tag{9}$$

*where* $\epsilon_i^{(2)}$ *is a white noise with mean zero and variance* $\sigma_2^2$*.*

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

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

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

$$\hat{L}^{\text{SPS}}(\beta, z) = \frac{1}{n_1 + n_2} \left( \sum_{i=1}^{n_1} \left( {x_i^{(1)}}^\top (\beta + z) - y_i^{(1)} \right)^2 + \sum_{j=1}^{n_2} \left( {x_j^{(2)}}^\top \beta - y_j^{(2)} \right)^2 \right) + \lambda \|z\|^2 \tag{11}$$

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

*A natural baseline is when we do not use the source task data at all. That is, perform least squares regression using target task data alone.*

*In Figure 1, we illustrate the effects incurred from model shifts and covariate shifts in transfer learning linear regression, comparing between HPS and OLS. In particular, we capture model shift as the distance error between $\beta^{(1)}$ and $\beta^{(2)}$, and we capture covariate shift as the difference in the population covariance matrix between task one and task two. To generate the condition matrix, we use a rank-r matrix whose trace is equal to p, and set its nonzero eigenvalues as $p/r$.*

### 1.4.3 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 *minimax 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 loss and expected loss 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)
>
> $$L(\hat{h}) - \min_{h\in\mathcal{H}} L(h) \le \epsilon$$
>
> c) Further, $n$ is a polynomial function of $\epsilon^{-1}, \delta^{-1}, d, |\mathcal{H}|$, and $A$ 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.

**Example 2.1** (Policy learning)**.** *Even though the above definition was proposed three decades ago, it remains a very fundamental concept, and applies broadly beyond supervised prediction.*

*Consider a finite Markov decision process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon $H$, and unknown transition and reward dynamics. Let $\Pi$ be a finite class of deterministic policies $\pi : \mathcal{S} \to \mathcal{A}$. For any policy $\pi \in \Pi$, we can define the expected loss as*

$$L(\pi) := \mathbb{E}\left[\sum_{t=1}^{H} \ell(s_t, a_t)\right],$$

*where the trajectory $(s_t, a_t)$ is generated by executing $\pi$ in the MDP, and $\ell$ is the loss measured at each step.*

*Once we collect $n$ trajectories (e.g., via an exploration policy),*[2] *we can write down the empirical loss averaged over the $n$ sampled trajectories. Therefore, a policy learning algorithm*

[2] For instance, a uniform random exploration picks an action $a \in \mathcal{A}$ uniformly at random at any state. This guarantees unbiased coverage of all actions. An $\varepsilon$-greedy exploration instead follows a fixed deterministic policy with probability $1 - \varepsilon$, while choosing a random action with probability $\varepsilon$.

*A PAC-learns the policy space* $\Pi$ *if A can produce a near-optimal policy with excess loss at most* $\epsilon$ *with probability at least* $1-\delta$*, and n only depends polynomially on* $|\mathcal{S}|, |\mathcal{A}|, \delta^{-1}$*.*

## 2.1 Learning a realizable, finite hypothesis class (Lecture 3)

Next, we give a concrete example to illustrate learnability in a finite, realizable hypothesis class.

**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 $1-\delta$,

$$L(\hat{h}_{\text{ERM}}) \leq \frac{\log(|\mathcal{H}|) + \log(\delta^{-1})}{n} \tag{12}$$

In particular, to reduce the expected risk below $\epsilon$, we want

$$n \geq \frac{\log(|\mathcal{H}|) + \log(\delta^{-1})}{\epsilon}.$$

**Example 2.2.** *Imagine a clinic designing a digital diagnostic tool to determine if a patient should be referred to a specialist based on three binary (Yes/No) symptoms: 1. Fever? 2. Cough? 3. Shortness of breath?*

*Since there are only* 3 *binary features, there are only* 8 *possible patient profiles. A hypothesis in this setup is a rule that assigns a "Refer" or "Not Refer" label to each profile. The number of possible ways to assign binary labels to these* 8 *profiles is* $2^8 = 256$*, which is finite. A hypothesis is any rule that assigns a binary labeling to the* 8 *profiles. Thus, the size of the hypothesis space is* 256 *in total.*

*If a medical board has already decided a specific protocol regarding the referral, then there exists a true target function in the* 256 *options. Thus, suppose you train the model on historical data from a clinic, you are guaranteed that at least one hypothesis in your class can achieve zero error, because the data was generated by a specific rule.*

*Proof.* We'd like to upper bound the probability of the bad event that $L(\hat{h}_{\text{ERM}}) > \epsilon$: Let $B \subseteq \mathcal{H}$ be the set of bad hypotheses: $\{B \in \mathcal{H} : L(h) > \epsilon\}$

We can rewrite our goal as upper bounding the probability of selecting a bad hypothesis $\Pr[L(\hat{h}_{\text{ERM}}) > \epsilon] = \Pr[\hat{h}_{\text{ERM}} \in B]$. Recall the empirical risk of ERM is always zero because at least $\hat{L}(h^\star) = 0$. Hence for any "bad" hypothesis in $B$, they must have zero empirical risk

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

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 $1 - x \leq \exp(-x)$.

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 $\delta$, we conclude that $\epsilon$ must be at least $\frac{\log(|\mathcal{H}|)+\log(\delta^{-1})}{n}$. This concludes the proof for learning finite, realizable hypothesis spaces.

Two remarks:

- The excess risk only grows logarithmically with the size of the hypothesis class, so we can afford 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.

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 dimension, and PAC-Bayes bounds (to name a few).

## 2.2 Concentration estimates (Lecture 4)

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 2.3** (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:*

- *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* $\sigma^2$*, 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 $\mu$. 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 $\mu$ 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

$$\Pr[|\hat{\mu}_n - \mu| > \epsilon] \leq 2\exp(-2\epsilon^2 n). \tag{13}$$

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.[3] Next, we shall see a proof through the use of moment generating functions (MGF).

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

**Definition 2.4** (Moment generating function)**.** *For a random variable $X$, its MGF is given by*

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

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

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

*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.*

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 $t > 0$,

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

In particular,

$$\Pr[X \geq a] \leq \min_{t>0} \frac{\mathbb{E}\left[e^{tX}\right]}{e^{ta}} \tag{14}$$

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 $t$.

### 2.2.1 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.[4]

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

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

[4]Poisson trials differ from Poisson random variables.

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 $\mu$ 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

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

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:

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

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 $\delta$ between 0 and 1, we have

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

**Example 2.5** (Coin flips). *Let $X$ be the number of heads in a sequence of $n$ independent fair coin flips. That is $X = X_1 + X_2 + \cdots + X_n$, where every $X_i$ is a Bernoulli random variable yielding one with probability $1/2$. 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 $O(\sqrt{n \ln n})$.*[5]

**Example 2.6** (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 (14) 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.

**Definition 2.7** (Sub-Gaussian). *A mean zero random variable $X$ is sub-Gaussian with parameter $\sigma^2$ if its MGF is bounded as follows:*

$$M_X(t) \leq \exp\left(\frac{\sigma^2 t^2}{2}\right) \tag{15}$$

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

[5] 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 Using uniform convergence to reason about generalization (Lecture 4)

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. This will lead us to the concept of Rademacher complexity, which is a crucial notion in statistical learning theory. We would like to show that the excess risk of ERM is small:

$$\Pr\left[L(\hat{h}_{\text{ERM}}) - L(h^\star) \geq \epsilon\right] \leq \delta \tag{16}$$

We can expand the excess risk as

$$L(\hat{h}) - L(h^\star) = \underbrace{\left(L(\hat{h}) - \hat{L}(\hat{h})\right)}_{\text{Uniform convergence}} + \underbrace{\left(\hat{L}(\hat{h}) - \hat{L}(h^\star)\right)}_{\leq 0} + \underbrace{\left(\hat{L}(h^\star) - L(h^\star)\right)}_{\text{Concentration}} \tag{17}$$

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,

$$\hat{L}(\hat{h}_{\text{ERM}}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{h}_{\text{ERM}}(x_i), y_i). \tag{18}$$

**Example 2.8** (Stochastic optimization)**.** *In practice,* $\hat{h}_{\text{ERM}}$ *is often computed using* stochastic optimization*, such as stochastic gradient descent (SGD), which updates:*

$$h^{(t+1)} = h^{(t)} - \eta_t \nabla \ell(h^{(t)}(x_{i_t}, y_{i_t})),$$

*where* $i_t$ *is sampled uniformly from* $1, 2, \ldots, n$*.*

*Each SGD step is an unbiased estimate of the gradient of* $\hat{L}(h)$*, but the final iterate* $\hat{h}_{\text{ERM}}$ *depends on the entire dataset through the optimization trajectory.*

Due to the correlation, $\hat{L}$ 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, the goal of uniform convergence can be stated as

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

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 $\epsilon/2$.

**Next lecture:** In the next lecture, we will define *Rademacher complexity*, and we will introduce concentration estimates as a fundamental tool for reasoning about generalization.

**P.S.**

- Have feedback or want to ask a question for the instructor? Leave a note here: https://forms.gle/SCjXUW6dkM8cDi4o9.
- Want to see the latex source code? You can find the tex files here: https://github.com/hongyangsg/cs7140-advanced-ml.

## References


Bach, F. (2024). *Learning theory from first principles*. MIT press (page 11).

Dhifallah, O. and Lu, Y. M. (2021). "Phase transitions in transfer learning for high-dimensional perceptrons". In: *Entropy* 23.4, p. 400 (page 7).

Duchi, J. (2019). "Lecture notes for statistics 311/electrical engineering 377". In: *URL: http://web.stanford.edu/class/stats311/lecture-notes.pdf* (page 8).

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022). "What can transformers learn in-context? a case study of simple function classes". In: *Advances in neural information processing systems* 35, pp. 30583–30598 (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).

Ruder, S. (2017). "An Overview of Multi-Task Learning in Deep Neural Networks". In: *arXiv preprint arXiv:1706.05098* (page 7).

Valiant, L. G. (1984). "A theory of the learnable". In: *Communications of the ACM* 27.11, pp. 1134–1142 (page 9).

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press (pages 3, 11).

Zhang, T. (2023). *Mathematical analysis of machine learning algorithms*. Cambridge University Press (page 11).