Lecture Notes for Machine Learning Theory (CS229M/STATS214)

Instructor: Tengyu Ma

September 26, 2021

Contents

Acknowledgments

This monograph is a collection of scribe notes for the course CS229M/STATS214 at Stanford University. The materials in Chapter 1–5 are mostly based on Percy Liang's lecture notes [Liang, 2016], and Chapter 10 is largely based on Haipeng Luo's lectures [Luo, 2017]. Kenneth Tay contributed significantly to the revision of these notes as a teaching assistant for the course. The original contributor to the scribe notes are Stanford students including but not limited to Anusri Pampari, Gabriel Poesia, Alexander Ke, Trenton Chang, Brad Ross, Robbie Jones, Yizhou Qian, Will Song, Daniel Do, Spencer M. Richards, Thomas Lew, David Lin, Jinhui Wang, Rafael Rafailov, Aidan Perreault,Kevin Han, Han Wu, Andrew Wang, Rohan Taori, Jonathan Lee, Rohith Kuditipudi, Kefan Dong, Roshni Sahoo, Sarah Wu, Tianyu Du, Xin Lu, Soham Sinha, Kevin Guo, Jeff Z. HaoChen, Carrie Wu, Kaidi Cao, and Ruocheng Wang. The notes will be updated every year with new materials. The reference list is far from complete.

Chapter 1

Supervised Learning Formulations

In this chapter, we will set up the standard theoretical formulation of supervised learning and introduce the *empirical risk minimization* (ERM) paradigm. The setup will apply to almost the entire monograph and the ERM paradigm will the man focus of Chapter 2, 3, and 4.

1.1 Supervised learning

In supervised learning, we have a dataset where each data point is associated with a label, and we aim to learn from the data a function that maps data points to their labels. The learned function can be used to infer the labels of test data points. More formally, suppose the data points, also called inputs, belong to some input space $\mathcal{X}$ (e.g. images of birds), and labels belong to the output space $\mathcal{Y}$ (e.g. bird species). Suppose we are interested in a specific joint probability distribution P over $\mathcal{X} \times \mathcal{Y}$ (e.g. images of birds in North America), from which we draw a *training set*, i.e we draw a a set of n independent and identically distributed (i.i.d.) data points $\{(x^{(i)}, y^{(i)}\}_{i=1}^{n}$ from P. The goal of supervised learning is to learn a mapping (i.e. a function) from $\mathcal{X}$ to $\mathcal{Y}$ using the training data. Any such function $h : \mathcal{X} \to \mathcal{Y}$ is called a *predictor* (also *hypothesis* or *model*).

Given two predictors, how do we decide which is better? For that, we define a *loss function* over the predictions. There are several ways to define loss functions: for now, define a loss function ℓ as a function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Intuitively, the loss function takes two labels, the prediction made by a model $\hat{y}$ and the true label y, and gives a number that captures how different the two labels are. We assume ℓ is non-negative, i.e $\ell(\hat{y}, y) \geq 0$. Then, the loss of a model h on an example (x, y) is $\ell(h(x), y)$, i.e. the difference (as measured by ℓ) between the prediction made by h and the true label.

With these definitions, we are able to formalize the problem of supervised learning. Precisely, we seek to find a model h that minimizes what we call the expected loss (or population loss or expected risk or population risk):

$$L(h) \triangleq \mathop{\mathbb{E}}_{(x,y)\sim p}[\ell(h(x), y)]. \tag{1.1}$$

Note that L is nonnegative because ℓ is nonnegative. Typically, the loss function is designed so that the best possible loss is zero when $\hat{y}$ matches y exactly. Therefore, the goal is to find h such that $L(h)$ is as close to zero as possible.

Examples: regression and classification problems. Here are two standard types of supervised learning problems based on the properties of the output space:

- In the problem of *regression*, predictions are real numbers ($\mathcal{Y} = \mathbb{R}$). We would like predictions to be as close as possible to the real labels. A classical loss function that captures this is the squared error, $\ell(\hat{y}, y) = (\hat{y} - y)^2$.

- In the problem of *classification*, predictions are in a discrete set of k unordered classes $\mathcal{Y} = [k] = \{1, \cdots, k\}$. One possible classification loss is the $0-1$ loss: $\ell(\hat{y}, y) = \mathbb{1}(\hat{y} \neq y)$, i.e. 0 if the prediction is equal to the true label, and 1 otherwise.

Hypothesis class. So far, we said we would like to find *any function* that minimizes population risk. However, in practice, we do not have a way of optimizing over arbitrary functions. Instead, we work within a more constrained set of functions $\mathcal{H}$, which we call the *hypothesis family* (or *hypothesis class*). Each element of $\mathcal{H}$ is a function $h : \mathcal{X} \to \mathcal{Y}$. Usually, we choose a set $\mathcal{H}$ that we know how to optimize over (e.g. linear functions, or neural networks).

Given one particular function $h \in \mathcal{H}$, we define the *excess risk* of h with respect to $\mathcal{H}$ as the difference between the population risk of h and the best possible population risk inside $\mathcal{H}$:

$$E(h) \triangleq L(h) - \inf_{g \in \mathcal{H}} L(g).$$

Generally we need more assumptions about a specific problem and hypothesis class to bound absolute population risk, hence we focus on bounding the excess risk.

Usually, the family we choose to work with can be parameterized by a vector of parameters $\theta \in \Theta$. In that case, we can refer to an element of $\mathcal{H}$ by h_θ, making that explicit. An example of such a parametrization of the hypothesis class is $\mathcal{H} = \{h : h_\theta(x) = \theta^\top x, \theta \in \mathbb{R}^d\}$.

1.2 Empirical risk minimization

Our ultimate goal is to minimize population risk. However, in practice we do not have access to the entire population: we only have a *training set* of n data points, drawn from the same distribution as the entire population. While we cannot compute population risk, we can compute *empirical risk*, the loss over the training set, and try to minimize that. This is, in short, the paradigm known as *empirical risk minimization* (ERM): we optimize the training set loss, with the hope that this leads us to a model that has low population loss. From now on, with some abuse of notation, we often write $\ell(h_\theta(x), y)$ as $\ell((x, y), \theta)$ and use the two notations interchangeably. Formally, we define the empirical risk of a model h as:

$$\widehat{L}(h_\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{n} \sum_{i=1}^{n} \ell((x^{(i)}, y^{(i)}), \theta). \tag{1.2}$$

Empirical risk minimization is the method of finding the minimizer of $\widehat{L}$, which we call $\hat{\theta}$:

$$\hat{\theta} \triangleq \operatorname*{argmin}_{\theta \in \Theta} \widehat{L}(h_\theta). \tag{1.3}$$

Since we are assuming that our training examples are drawn from the same distribution as the whole population, we know that empirical risk and population risk are equal *in expectation* (over the randomness of the training dataset):

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x^{(i)}, y^{(i)}) \overset{\text{iid}}{\sim} P} \widehat{L}(h_\theta) &= \mathop{\mathbb{E}}_{(x^{(i)}, y^{(i)}) \overset{\text{iid}}{\sim} P} \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(x^{(i)}), y^{(i)}) && (1.4)\\
&= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{(x^{(i)}, y^{(i)}) \overset{\text{iid}}{\sim} P} \ell(h_\theta(x^{(i)}), y^{(i)}) && (1.5)\\
&= \frac{1}{n} \cdot n \cdot \mathop{\mathbb{E}}_{(x^{(i)}, y^{(i)}) \overset{\text{iid}}{\sim} P} \ell(h_\theta(x^{(i)}), y^{(i)}) && (1.6)\\
&= L(h_\theta). && (1.7)
\end{aligned}$$

This is one reason why it makes sense to use empirical risk: it is an unbiased estimator of the population risk.

The key question that we seek to answer in the first part of this course is: **what guarantees do we have on the excess risk for the parameters learned by ERM?** The hope with ERM is that minimizing the training error will lead to small testing error. One way to make this rigorous is by showing that the ERM minimizer's excess risk is bounded.

Chapter 2

Asymptotic Analysis

In this chapter, we use an asymptotic approach (i.e assuming number of training samples $n \to \infty$) to achieve a bound on the ERM. We then instantiate these results to the case where the loss function is the maximum likelihood and discuss the limitations of asymptotics. (In future chapters we will assume finite n and provide a non-asymptotic analysis.)

2.1 Asymptotics of empirical risk minimization

For the asymptotic analysis of ERM, we would like to prove that excess risk is bounded as shown below:

$$L(\hat{\theta}) - \operatorname*{argmin}_{\theta \in \Theta} L(\theta) \leq \frac{c}{n} + o\left(\frac{1}{n}\right). \tag{2.1}$$

Here c is a problem dependent constant that does not depend on n, and $o(1/n)$ hides all dependencies except n. The equation above shows that as we have more training data (i.e as n increases) the excess risk of ERM decrease at the rate of $\frac{1}{n}$.

Let $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(n)}, y^{(n)})\}$ be the training data and let $\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}^p\}$ be the parameterized family of hypothesis functions. Let the ERM minimizer be $\hat{\theta}$ as defined in Equation (1.3). Let θ^* be the minimizer of the population risk L, i.e. $\theta^* = \operatorname{argmin}_\theta L(\theta)$. The theorem below quantifies the excess risk $L(\hat{\theta}) - L(\theta^*)$:

Theorem 2.1 (Informally stated)**.** *Suppose that (a) $\hat{\theta} \xrightarrow{p} \theta^*$ as $n \to \infty$ (i.e consistency of $\hat{\theta}$), (b) $\nabla^2 L(\theta^*)$ is full rank, and (c) other appropriate regularity conditions hold. Then,*

1. *$\sqrt{n}(\hat{\theta} - \theta^*) = O_P(1)$, i.e. for every $\epsilon > 0$, there is an M such that $\sup_n \mathbb{P}(\|\sqrt{n}(\hat{\theta} - \theta^*)\|_2 > M) < \epsilon$. (This means that the sequence $\{\sqrt{n}(\hat{\theta} - \theta^*)\}$ is "bounded in probability".)*
2. *$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{d} \mathcal{N}\left(0, (\nabla^2 L(\theta^*))^{-1} Cov(\ell((x,y),\theta^*))(\nabla^2 L(\theta^*))^{-1}\right)$.*
3. *$n(L(\hat{\theta}) - L(\theta^*)) = O_P(1)$.*
4. *$n(L(\hat{\theta}) - L(\theta^*)) \xrightarrow{d} \frac{1}{2}||S||_2^2$ where $S \sim \mathcal{N}\left(0, (\nabla^2 L(\theta^*))^{-1/2} Cov(\ell((x,y),\theta^*))(\nabla^2 L(\theta^*))^{-1/2}\right)$.*
5. *$\lim_{n\to\infty} \mathbb{E}\left[n(L(\hat{\theta}) - L(\theta^*))\right] = \frac{1}{2} \operatorname{tr}\left(\nabla^2 L(\theta^*)^{-1} \operatorname{Cov}(\nabla \ell((x,y),\theta^*)\right)$.*

Remark: In the theorem above, Parts 1 and 3 only show the rate or order of convergence, while Parts 2 and 4 define the limiting distribution for the random variables.

Theorem 2.1 is a powerful conclusion because once we know that $\sqrt{n}(\hat{\theta}-\theta^*)$ is (asymptotically) Gaussian, we can easily work out the distribution of the excess risk. If we believe in our assumptions and n is large enough such that we can assume $n \to \infty$, this allows us to analytically determine quantities of interest in almost any scenario (for example, if our test distribution changes). The key takeaway is that our parameter error $\hat{\theta} - \theta^*$ decreases in order $1/\sqrt{n}$ and the excess risk decreases in order $1/n$.

2.1.1 Key ideas of proofs

We will prove the theorem above by applying the following main ideas:

1. Obtain an expression for the excess risk by Taylor expansion of the derivative of the empirical risk $\nabla\widehat{L}(\theta)$ around θ^*.
2. By the law of large numbers, we have that $\widehat{L}(\theta) \xrightarrow{p} L(\theta)$, $\nabla\widehat{L}(\theta) \xrightarrow{p} \nabla L(\theta)$ and $\nabla^2\hat{L}(\theta) \xrightarrow{p} \nabla^2 L(\theta)$ as $n \to \infty$.
3. Central limit theorem (CLT).

First, we state the CLT for i.i.d. means and a lemma that we will use in the proof.

Theorem 2.2 (Central Limit Theorem)**.** *Let $X_1, \cdots, X_n$, be i.i.d. random variables, where $\widehat{X} = \frac{1}{n}\sum_{i=1}^n X_i$ and the covariance matrix Σ is finite. Then, as $n \to \infty$ we have*

1. *$\widehat{X} \xrightarrow{p} \mathbb{E}[X]$, and*
2. *$\sqrt{n}(\widehat{X} - \mathbb{E}[X]) \xrightarrow{d} \mathcal{N}(0, \Sigma)$. In particular, $\sqrt{n}(\widehat{X} - \mathbb{E}[X]) = O_P(1)$.*

Lemma 2.3.

1. *If $Z \sim N(0, \Sigma)$ and A is a deterministic matrix, then $AZ \sim N(0, A\Sigma A^T)$.*
2. *If $Z \sim N(0, \Sigma^{-1})$ and $Z \in \mathbb{R}^p$, then $Z^T\Sigma Z \sim \chi^2(p)$, where $\sim \chi^2(p)$ is the chi-squared distribution with p degrees of freedom.*

2.1.2 Main proof

Let us start with heuristic arguments for Parts 1 and 2. First, note that by definition, the gradient of the empirical risk at the empirical risk minimizer, $\nabla\widehat{L}(\hat{\theta})$, is equal to 0. From the Taylor expansion of $\nabla\widehat{L}$ around θ^*, we have that

$$0 = \nabla\widehat{L}(\hat{\theta}) = \nabla\widehat{L}(\theta^*) + \nabla^2\widehat{L}(\theta^*)(\hat{\theta} - \theta^*) + O(\|\hat{\theta} - \theta^*\|_2^2). \tag{2.2}$$

Rearranging, we have

$$\hat{\theta} - \theta^* = -(\nabla^2\widehat{L}(\theta^*))^{-1}\nabla\widehat{L}(\theta^*) + O(||\hat{\theta} - \theta^*||_2^2). \tag{2.3}$$

Multiplying by $\sqrt{n}$ on both sides,

$$\sqrt{n}(\hat{\theta} - \theta^*) = -(\nabla^2\widehat{L}(\theta^*))^{-1}\sqrt{n}(\nabla\widehat{L}(\theta^*)) + O(||\hat{\theta} - \theta^*||_2^2)) \tag{2.4}$$

$$\approx -(\nabla^2\widehat{L}(\theta^*))^{-1}\sqrt{n}(\nabla\widehat{L}(\theta^*)). \tag{2.5}$$

Applying the Central Limit Theorem (Theorem 2.2) using $X_i = \nabla\ell((x^{(i)}, y^{(i)}), \theta^*)$ and $\widehat{X} = \nabla\widehat{L}(\theta^*)$, and noticing that $\mathbb{E}[\nabla\widehat{L}(\theta^*)] = \nabla L(\theta^*)$, we have

$$\sqrt{n}(\nabla\widehat{L}(\theta^*) - \nabla L(\theta^*)) \xrightarrow{d} \mathcal{N}(0, Cov(\ell((x, y), \theta^*))). \tag{2.6}$$

Note that $\nabla L(\theta^*) = 0$ because θ^* is the minimizer of L, so $\sqrt{n}(\nabla \widehat{L}(\theta^*)) \xrightarrow{d} \mathcal{N}(0, Cov(\ell((x,y),\theta^*)))$. By the law of large numbers, $\nabla^2 \widehat{L}(\theta^*) \xrightarrow{p} \nabla^2 L(\theta^*)$. Applying these results to (2.5) (together with an application of Slutsky's theorem),

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{d} \nabla^2 L(\theta^*)^{-1} \mathcal{N}(0, Cov(\ell((x,y),\theta^*)) \tag{2.7}$$

$$\stackrel{d}{=} \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1} Cov(\ell((x,y),\theta^*)) \nabla^2 L(\theta^*)^{-1}\right), \tag{2.8}$$

where the second step is due to Lemma 2.3. This proves Part 2 of Theorem 2.1.

Part 1 follows directly from Part 2 by the following fact: If $X_n \xrightarrow{d} P$ for some probability distribution P, then $X_n = O_P(1)$.

We now turn to proving Parts 3 and 4. Using a Taylor expansion of L with respect to θ at θ^*, we find

$$L(\hat{\theta}) = L(\theta^*) + \langle \nabla L(\theta^*), \hat{\theta} - \theta^* \rangle + \frac{1}{2} \langle \hat{\theta} - \theta^*, \nabla^2 L(\theta^*)(\hat{\theta} - \theta^*) \rangle + o(\|\hat{\theta} - \theta^*\|_2^2). \tag{2.9}$$

Since θ^* is the minimizer of the population risk L, we know that $\nabla L(\theta^*) = 0$ and the linear term is equal to 0. Rearranging and multiplying by n, we can write

$$n(L(\hat{\theta}) - L(\theta^*)) = \frac{n}{2} \langle \hat{\theta} - \theta^*, \nabla^2 L(\theta^*)(\hat{\theta} - \theta^*) \rangle + o(\|\hat{\theta} - \theta^*\|_2^2) \tag{2.10}$$

$$\approx \frac{1}{2} \langle \sqrt{n}(\hat{\theta} - \theta^*), \nabla^2 L(\theta^*) \sqrt{n}(\hat{\theta} - \theta^*) \rangle \tag{2.11}$$

$$= \frac{1}{2} \left\| \nabla^2 L(\theta^*)^{1/2} \sqrt{n}(\hat{\theta} - \theta^*) \right\|_2^2, \tag{2.12}$$

where the last equality follows by the fact that for any vector v and square matrix A of appropriate dimensions, the inner product $\langle v, Av \rangle = v^T A v = \|A^{1/2} v\|_2^2$. Let $S = \nabla^2 L(\theta^*)^{1/2} \sqrt{n}(\hat{\theta} - \theta^*)$, i.e. the random vector inside the norm. By Part 2, we know the asymptotic distribution of $\sqrt{n}(\hat{\theta} - \theta^*)$ is Gaussian. Thus as $n \to \infty$, $n(L(\hat{\theta}) - L(\theta^*)) \xrightarrow{d} \frac{1}{2}\|S\|_2^2$ where

$$S \sim \nabla^2 L(\theta^*)^{1/2} \cdot \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1} \operatorname{Cov}(\nabla \ell((x,y),\theta^*) \nabla^2 L(\theta^*)^{-1}\right) \tag{2.13}$$

$$\stackrel{d}{=} \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1/2} \operatorname{Cov}(\nabla \ell((x,y),\theta^*) \nabla^2 L(\theta^*)^{-1/2}\right). \tag{2.14}$$

This proves Part 4, and Part 3 follows directly from the definition of the O_P notation.

Finally, for Part 5, using the fact that the trace operator is invariant under cyclic permutations, the fact that $\mathbb{E}[S] = 0$, and some regularity conditions,

$$\lim_{n \to \infty} \mathbb{E}\left[n(L(\hat{\theta}) - L(\theta^*))\right] = \frac{1}{2} \mathbb{E}\left[\|S\|_2^2\right] = \frac{1}{2} \mathbb{E}\left[\operatorname{tr}(S^\top S)\right] \tag{2.15}$$

$$= \frac{1}{2} \mathbb{E}\left[\operatorname{tr}(S S^\top)\right] = \frac{1}{2} \operatorname{tr}\left(\mathbb{E}[S S^\top]\right) \tag{2.16}$$

$$= \frac{1}{2} \operatorname{tr}\left(\operatorname{Cov}(S)\right) \tag{2.17}$$

$$= \frac{1}{2} \operatorname{tr}\left(\nabla^2 L(\theta^*)^{-1} \operatorname{Cov}(\nabla \ell((x,y),\theta^*)\right). \tag{2.18}$$

2.1.3 Well-specified case

Theorem 2.1 is powerful because it is general, avoiding any assumptions of a probabilistic model of our data. However in many applications, we assume a model of our data and we define the log-likelihood with respect to this model. Formally, suppose that we have a family of probability distributions P_θ, parameterized by $\theta \in \Theta$, such that P_{θ_*} is the true data-generating distribution. This is known as the well-specified case. To make the results of Theorem 2.1 more applicable, we derive analogous results for this well-specified case in Theorem 2.4.

Theorem 2.4. *In addition to the assumptions of Theorem 2.1, suppose there exists a parametric model $P(y \mid x;\theta)$, $\theta \in \Theta$, such that $\{y^{(i)} \mid x^{(i)}\}_{i=1}^n \sim P(y^{(i)} \mid x^{(i)};\theta_*)$ for some $\theta_* \in \Theta$. Assume that we performing maximum likelihood estimation (MLE), i.e. our loss function is the negative log-likelihood $\ell((x^{(i)}, y^{(i)}), \theta) = -\log P(y^{(i)} \mid x^{(i)};\theta)$. As before, let $\hat{\theta}$ and θ^* denote the minimizers of empirical risk and population risk respectively. Then*

$$\theta^* = \theta_*, \tag{2.19}$$

$$\mathbb{E}\left[\nabla \ell((x,y),\theta^*)\right] = 0, \tag{2.20}$$

$$\mathrm{Cov}\left(\nabla \ell((x,y),\theta^*)\right) = \nabla^2 L(\theta^*), \text{ and} \tag{2.21}$$

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{d} \mathcal{N}(0, \nabla^2 L(\theta^*)^{-1}). \tag{2.22}$$

Remark 1: You may also have seen (2.22) in the following form: under the maximum likelihood estimation (MLE) paradigm, the MLE is asymptotically efficient in the Cramer-Rao lower bound. That is, the parameter error of the MLE estimate converges in distribution to $\mathcal{N}(0, \mathcal{I}(\theta)^{-1})$, where $\mathcal{I}(\theta)$ is the Fisher information matrix (in this case, equivalent to the risk Hessian $\nabla^2 L(\theta^*)$) [Rice, 2006].

Remark 2: (2.21) is also known as Bartlett's identity [Liang, 2016].

Although the proofs were not presented in live lecture, we include them here.

Proof. From the definition of the population loss,

$$\begin{aligned} L(\theta) &= \mathbb{E}\left[\ell((x^{(i)}, y^{(i)}), \theta)\right] && (2.23)\\ &= \mathbb{E}\left[-\log P(y \mid x;\theta)\right] && (2.24)\\ &= \mathbb{E}\left[-\log P(y \mid x;\theta) + \log P(y \mid x;\theta_*)\right] + \mathbb{E}\left[-\log P(y \mid x;\theta_*)\right] && (2.25)\\ &= \mathbb{E}\left[\log \frac{P(y \mid x;\theta_*)}{P(y \mid x;\theta)}\right] + \mathbb{E}\left[-\log P(y \mid x;\theta_*)\right]. && (2.26) \end{aligned}$$

Notice that the second term is a constant which we will express as $\mathcal{H}(y \mid x;\theta_*)$. We expand the first term using the tower rule (or law of total expectation):

$$L(\theta) = \mathbb{E}\left[\mathbb{E}\left[\log \frac{P(y \mid x;\theta_*)}{P(y \mid x;\theta)} \middle| x\right]\right] + \mathcal{H}(y \mid x;\theta_*). \tag{2.27}$$

The term in the expectation is just the KL divergence between the two probabilities, so

$$\begin{aligned} L(\theta) &= \mathbb{E}\left[\mathrm{KL}\left(y \mid x;\theta_* \| y \mid x;\theta\right)\right] + \mathcal{H}(y \mid x;\theta_*) && (2.28)\\ &\geq \mathcal{H}(y \mid x;\theta_*), && (2.29) \end{aligned}$$

since KL divergence is always non-negative. Since θ_* makes the KL divergence term 0, it minimizes $L(\theta)$ and so $\theta_* \in \mathrm{argmin}_\theta\, L(\theta)$. However, the minimizer of $L(\theta)$ is unique because of consistency, so we must have $\mathrm{argmin}_\theta\, L(\theta) = \theta^*$ which proves (2.19).

For (2.20), recall $\nabla L(\theta^*) = 0$, so we have

$$0 = \nabla L(\theta^*) = \nabla \mathbb{E}\left[\ell((x^{(i)}, y^{(i)}), \theta^*)\right] = \mathbb{E}\left[\nabla \ell((x^{(i)}, y^{(i)}), \theta^*)\right], \tag{2.30}$$

where we can switch the gradient and expectation under some regularity conditions.

To prove (2.21), we first expand the RHS using the definition of covariance and express the marginal distributions as integrals:

$$\begin{aligned}\operatorname{Cov}\left(\nabla\ell((x,y),\theta^*)\right) &= \mathbb{E}\left[\nabla\ell((x,y),\theta^*)\nabla\ell((x,y),\theta^*)^\top\right] && (2.31)\\ &= \int P(x)\left(\int P(y\mid x;\theta^*)\nabla\log P(y^{(i)}\mid x^{(i)};\theta^*)\nabla\log P(y^{(i)}\mid x^{(i)};\theta^*)^\top dy\right)dx && (2.32)\\ &= \int P(x)\left(\int \frac{\nabla P(y\mid x;\theta^*)\nabla P(y\mid x;\theta^*)^\top}{P(y\mid x;\theta^*)}dy\right)dx. && (2.33)\end{aligned}$$

Now we expand the LHS using the definition of the population loss and differentiate repeatedly:

$$\begin{aligned}\nabla^2 L(\theta^*) &= \mathbb{E}\left[\nabla^2\log P(y\mid x;\theta^*)\right] && (2.34)\\ &= \int P(x)\left(\int -\nabla^2 P(y\mid x;\theta^*) + \frac{\nabla P(y\mid x;\theta^*)\nabla P(y\mid x;\theta^*)^\top}{P(y\mid x;\theta^*)}dy\right)dx. && (2.35)\end{aligned}$$

Note that we can express

$$\int \nabla^2 P(y\mid x;\theta^*)dy = \nabla^2\int P(y\mid x;\theta^*)dy = \nabla 1 = 0 \tag{2.36}$$

so we find

$$\nabla^2 L(\theta^*) = \int P(x)\left(\int \frac{\nabla P(y\mid x;\theta^*)\nabla P(y\mid x;\theta^*)^\top}{P(y\mid x;\theta^*)}dy\right)dx = \operatorname{Cov}\left(\nabla\ell((x,y),\theta^*)\right). \tag{2.37}$$

Finally, (2.22) follows directly from Part 2 of Theorem 2.1 and (2.21). □

Using similar logic to our proof of Part 4 and 5 of Theorem 2.1, we can see that $n(L(\hat{\theta})-L(\theta^*)) \xrightarrow{d} \frac{1}{2}\|S\|_2^2$ where $S\sim N(0,I)$. Since a chi-squared distribution with p degrees of freedom is defined as a sum of the squares of p independent standard normals, it quickly follows that $2n(L(\hat{\theta})-L(\theta^*))\sim\chi^2(p)$, where $\theta\in\mathbb{R}^p$ and $n\to\infty$. We can thus characterize the excess risk in this case using the properties of a chi-squared distribution:

$$\lim_{n\to\infty}\mathbb{E}\left[L(\hat{\theta})-L(\theta^*)\right] = \frac{p}{2n}. \tag{2.38}$$

2.2 Limitations of asymptotic analysis

One limitation of asymptotic analysis is that our bounds often obscure dependencies on higher order terms. As an example, suppose we have a bound of the form

$$\frac{p}{2n} + o\left(\frac{1}{n}\right). \tag{2.39}$$

(Here $o(\cdot)$ treats the parameter p as a constant as n goes to infinity.) We have no idea how large n needs to be for asymptotic bounds to be "reasonable." Compare two possible versions of (2.39):

$$\frac{p}{2n}+\frac{1}{n^2} \quad \text{vs.} \quad \frac{p}{2n}+\frac{p^{100}}{n^2}. \tag{2.40}$$

Asymptotic analysis treats both of these bounds as the same, hiding the polynomial dependence on p in the second bound. Clearly, the second bound is significantly more data-intensive than the first: we would need $n > p^{50}$ before $\frac{p^{100}}{n^2}$ is less than one. Since p represents the dimensionality of the data, this may be an unreasonable assumption.

This is where non-asymptotic analysis can be helpful. Whereas asymptotic analysis uses large-sample theorems such as the central limit theorem and the law of large numbers to provide convergence guarantees, non-asymptotic analysis relies on concentration inequalities to develop alternative techniques for reasoning about the performance of learning algorithms.

Chapter 3

Concentration Inequalities

In this chapter, we take a little diversion and develop the notion of *concentration inequalities*. Assume that we have independent random variables $X_1, \ldots, X_n$. We will develop tools to show results that formalize the intuition for these statements:

1. $X_1 + \ldots + X_n$ concentrates around $\mathbb{E}[X_1 + \ldots + X_n]$.
2. More generally, $f(X_1, \ldots, X_n)$ concentrates around $\mathbb{E}[f(X_1, \ldots, X_n)]$.

These inequalities will be used in subsequent chapters to bound several key quantities of interest.

As it turns out, the material from this chapter constitutes arguably the important mathematical tools in the entire course. No matter what area of machine learning one wants to study, if it involves sample complexity, some kind of concentration result will typically be required. Hence, concentration inequalities are some of the most important tools in modern statistical learning theory.

3.1 The big-O notation

Throughout the rest of this course, we will use "big-O" notation in the following sense: every occurrence of $O(x)$ is a placeholder for some function $f(x)$ such that for every x, $|f(x)| \leq Cx$ for some absolute/universal constant C. In other words, suppose $O(n_1), \ldots, O(n_k)$ occur in a statement, it means that **there exists** absolute constants $C_1, \ldots, C_k > 0$ and functions $f_1, \ldots, f_k$ satisfying $|f_i(x)| \leq C_i x$ for all x, such that after replacing each occurrence $O(n_i)$ by $f_i(n_i)$, the statement is true. (The difference with traditional "big-O" notation is that we do not need to send $n \to \infty$ in order to define "big-O".)

Also, for any $a, b \geq 0$, we will let $a \lesssim b$ mean that there is some absolute constant $c > 0$ such that $a \leq cb$.

3.2 Hoeffding's inequality

We provide a brief overview of Hoeffding's inequality, an important concentration inequality for bounded random variables:

Theorem 3.1 (Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be i.i.d. real-valued random variables drawn from some distribution, such that $a_i \leq X_i \leq b_i$ almost surely. Define $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$, and let $\mu = \mathbb{E}[\bar{X}]$. Then for any $\varepsilon > 0$,*

$$\Pr\left[|\bar{X} - \mu| \leq \varepsilon\right] \geq 1 - 2\exp\left(\frac{-2n^2\varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right). \tag{3.1}$$

Note that the demoninator within the exponential term, $\sum_{i=1}^{n}(b_i - a_i)^2$, can be thought of as an upper bound or proxy for the variance $\mathrm{Var}(X_i)$. In fact, under the i.i.d. assumption, we can show

$$\mathrm{Var}\left(\bar{X}\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(x_i) \leq \frac{1}{n^2}\sum_{i=1}^{n}(b_i - a_i)^2. \tag{3.2}$$

Let $\sigma^2 = \frac{1}{n^2}\sum_{i=1}^{n}(b_i - a_i)^2$. If we take $\varepsilon = O(\sigma\sqrt{\log n}) = \sigma\sqrt{c\log n}$; i.e. ε bounded above by some large (i.e., $c \geq 10$) multiple of the standard deviation of the x_i's times $\sqrt{\log n}$, we can substitute this value of ε into (3.1) to reach the following conclusion:

$$\Pr\left[|\bar{X} - \mu| \leq \varepsilon\right] \geq 1 - 2\exp\left(\frac{-2\varepsilon^2}{\sigma^2}\right) \tag{3.3}$$

$$= 1 - 2\exp(-2c\log n) \tag{3.4}$$

$$= 1 - 2n^{-2c} \tag{3.5}$$

We can see that as n grows, the right-most term tends to zero such that $\Pr[|\bar{X} - \mu| \leq \varepsilon]$ very quickly approaches 1. Intuitively, this result tells us that, with high probability, the sample mean $\bar{X}$ will not be "much farther" from the population mean μ by some sublogarithmic ($\sqrt{c\log n}$) factor of the standard deviation.[1] Thus, we can restate the above claim we reached as follows:

Remark 3.2. For sufficiently large n, $|\bar{X} - \mu| \leq O(\sigma\sqrt{\log n})$ with high probability.

Remark 3.3. If, in addition, we have $a_i = -O(1)$ and $b_i = O(1)$, then $\sigma^2 = O\left(\frac{1}{n}\right)$, and $|\bar{X} - \mu| \leq O\left(\sqrt{\frac{\log n}{n}}\right) = \widetilde{O}\left(\frac{1}{\sqrt{n}}\right)$.[2]

Remark 3.3 provides a compact form of the Hoeffding bound that we can use when the X_i are bounded almost surely.

3.3 Chebyshev's inequality

Consider an arbitrary random variable Z with finite variance. One of the most famous results characterizing its tail behavior is the following theorem:

Theorem 3.4 (Chebyshev's inequality)**.** *Let Z be a random variable with finite expectation and variance. Then*

$$\Pr[|Z - \mathbb{E}[Z]| \geq t] \leq \frac{\mathrm{Var}(Z)}{t^2}, \quad \forall t > 0. \tag{3.6}$$

Intuitively, this means that as we approach the tails of the distribution of Z, the density decreases at a rate of at least $1/t^2$. Moreover, for any $\delta \in (0, 1]$, by plugging in $t = \mathrm{sd}(Z)/\sqrt{\delta}$ to (3.6) we see that

$$\Pr\left[|Z - \mathbb{E}[Z]| \leq \frac{\mathrm{sd}(Z)}{\sqrt{\delta}}\right] \geq 1 - \delta. \tag{3.7}$$

Unfortunately, it turns out that Chebyshev's inequality is a rather weak concentration inequality. To illustrate this, assume $Z \sim \mathcal{N}(0, 1)$. We can show (using the Gaussian tail bound derived in Problem 3(c) in Homework 0) that

$$\Pr\left[|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z)\sqrt{2\log(2/\delta)}\right] \geq 1 - \delta. \tag{3.8}$$

[1] This is with the caveat, of course, that σ is not exactly standard deviation but a loose upper bound on standard deviation.

[2] $\widetilde{O}$ is analogous to Big-O notation, in that $\widetilde{O}$ hides logarithmic factors. That is; if $f(n) = O(\log n)$, then $f(n) = \widetilde{O}(1)$.

for any $\delta \in (0, 1]$. In other words, the density at the tails of the normal distribution is decreasing at an exponential rate, while Chebyshev's inequality only gives a quadratic rate. The discrepancy between (3.7) and (3.8) is made more apparent when we consider inverse-polynomial $\delta = \frac{1}{n^c}$ for some parameter n and degree c (we will see concrete instances of this setup in future chapters). Then the tail bound for the normal distribution (3.8) implies that

$$|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z) \cdot \sqrt{\log O\left(n^c\right)} = \mathrm{sd}(Z) \cdot O\left(\sqrt{\log n}\right) \quad w.p.\ 1 - \delta, \tag{3.9}$$

while Chebyshev's inequality gives us the weaker result

$$|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z) \cdot \sqrt{O(n^c)} = \mathrm{sd}(Z) \cdot O(n^{c/2}) \quad w.p.\ 1 - \delta. \tag{3.10}$$

Despite the previous example, Chebyshev's inequality is actually optimal without further assumptions, in the sense that there exist distributions with finite variance for which the bound is tight. With that in mind, we will need to assume more about our random variables if we want to improve upon the Chebyshev inequality's $1/t^2$ rate of tail decay. As an example, recall that when $0 \leq X_i \leq 1$ for $i = 1, \ldots, n$, Hoeffding's inequality is applicable:

$$\Pr[|Z - \mathbb{E}[Z]| \geq t] \leq 2\exp(-2t^2/n). \tag{3.11}$$

This tail probability is exponentially decaying in t instead of polynomially decaying as in Chebyshev's inequality! Certainly requiring boundedness in $[0, 1]$ (or $[a, b]$ more generally) is limiting, so it is worth asking what types of distributions permit such an exponential tail bound. The following section will explore such a class of random variables: *sub-Gaussian* random variables.

3.4 Sub-Gaussian random variables

We begin by defining the class of sub-Gaussian random variables by way of a bound on their moment generating functions, after which we will see how this bound guarantees the exponential tail decay we are after.

Definition 3.5 (Sub-Gaussian Random Variables)**.** A random variable X with finite mean μ is *sub-Gaussian* with parameter σ if

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\sigma^2\lambda^2/2}, \quad \forall \lambda \in \mathbb{R}. \tag{3.12}$$

We say that X is σ-sub-Gaussian and say it has *variance proxy* σ^2.

Remark 3.6. As it turns out, (3.12) is quite a strong condition, requiring that infinitely many moments of X exist and do not grow too quickly. To see why, assume without loss of generality that $\mu = 0$ and take a power series expansion of the moment generating function:

$$\mathbb{E}[\exp(\lambda X)] = \mathbb{E}\left[\sum_{k=0}^{\infty} \frac{(\lambda X)^k}{k!}\right] = \sum_{k=0}^{\infty} \frac{\lambda^k}{k!}\,\mathbb{E}[X^k]. \tag{3.13}$$

A bound on the moment generating function then is a bound on infinitely many moments of X, i.e. a requirement that the moments of X are all finite and grow slowly enough to allow the power series to converge.

Although (3.12) is not a particularly intuitive definition, it turns out to imply exactly the type of exponential tail bound we want:

Theorem 3.7 (Tail bound for sub-Gaussian random variables)**.** *If a random variable X with finite mean μ is σ-sub-Gaussian, then*

$$\Pr[|X - \mu| \geq t] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right), \quad \forall t \in \mathbb{R}. \tag{3.14}$$

Proof. Fix $t > 0$. For any $\lambda > 0$,

$$\begin{aligned}
\Pr[X - \mu \geq t] &= \Pr[\exp(\lambda(X-\mu)) \geq \exp(\lambda t)] && (3.15)\\
&\leq \exp(-\lambda t)\,\mathbb{E}[\exp(\lambda(X-\mu))] && \text{(by Markov's inequality)} \quad (3.16)\\
&\leq \exp(-\lambda t)\exp(\sigma^2\lambda^2/2) && \text{(by (3.12))} \quad (3.17)\\
&= \exp(-\lambda t + \sigma^2\lambda^2/2). && (3.18)
\end{aligned}$$

Because the bound (3.18) holds for any choice of λ and $\exp(\cdot)$ is monotonically increasing, we can optimize the bound (3.18) by finding λ which minimizes the exponent $-\lambda t + \sigma^2\lambda^2/2$. Differentiating and setting the derivative equal to zero, we find that the optimal choice is $\lambda = t/\sigma^2$, yielding the one-sided tail bound

$$\Pr[X - \mu \geq t] \leq \exp\left(-\frac{t^2}{2\sigma^2}\right). \tag{3.19}$$

Going through the same line of reasoning but for $-X$ and $-t$, we can also show that for any $t > 0$,

$$\Pr[X - \mu \leq -t] \leq \exp\left(-\frac{t^2}{2\sigma^2}\right). \tag{3.20}$$

We can then obtain (3.14) by applying the union bound:

$$\Pr[|X - \mu| \geq t] = \Pr[X - \mu \geq t] + \Pr[X - \mu \leq -t] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right). \tag{3.21}$$

□

Remark 3.8 (Tail bound implies sub-Gaussianity). In addition to being a necessary condition for sub-Gaussianity (Theorem 3.7), the tail bound (3.14) for sub-Gaussian random variables is also a sufficient condition up to a constant factor. In particular, if a random variable X with finite mean μ satisfies (3.14) for some $\sigma > 0$, then X is $O(\sigma)$-sub-Gaussian. Unfortunately, the proof of this reverse direction is somewhat more involved, so we refer the interested reader to Theorem 2.6 and its proof in Section 2.4 of [Wainwright, 2019] and Proposition 2.5.2 in [Vershynin, 2018] for details. While the tail bound is the property we ultimately care about most when studying sub-Gaussian random variables, the definition in (3.12) is a more technically convenient characterization, as we will see in the proof of Theorem 3.10.

Remark 3.9. Note that in light of Remark 3.6, the tail bound (3.8) requires all central moments of X to exist and not grow too quickly. In contrast, Chebyshev's inequality (and more generally any polynomial variant of Markov's inequality $\Pr[|X - \mu| \geq t] = \Pr[|X - \mu|^k \geq t^k] \leq t^{-k}\mathbb{E}[|X - \mu|^k]$) only requires that the second central moment $\mathbb{E}[(X-\mu)^2]$ (more generally, the kth central moment $\mathbb{E}[|X-\mu|^k]$) is finite to yield a tail bound. If infinite moments exist however, it turns out that $\inf_{k\in\mathbb{N}} t^{-k}\mathbb{E}[|X-\mu|^k] \leq \inf_{\lambda>0}\exp(-\lambda t)\,\mathbb{E}[\exp(\lambda(X-\lambda))]$, i.e. the optimal polynomial tail bound is tighter than the optimal exponential tail bound (see Exercise 2.3 in [Wainwright, 2019]). As we will see shortly though, using exponential functions of random variables allows us to prove results about sums of random variables more conveniently, which is why most researchers use exponential tail bounds in practice.

Having defined and derived exponential tail bounds for sub-Gaussian random variables, we can now accomplish the first of the goals we set out at the beginning of the chapter: to show that under certain conditions, namely independence and sub-Gaussianity of $X_1, \ldots, X_n$, the sum $Z = \sum_{i=1}^n X_i$ concentrates around $\mathbb{E}[Z] = \mathbb{E}[\sum_{i=1}^n X_i]$.

Theorem 3.10 (Sum of sub-Gaussian random variables is sub-Gaussian)**.** *If $X_1, \ldots, X_n$ are independent sub-Gaussian random variables with variance proxies $\sigma_1^2, \ldots, \sigma_n^2$, then $Z = \sum_{i=1}^n X_i$ is sub-Gaussian with variance proxy $\sum_{i=1}^n \sigma_i^2$. As a consequence, we have the tail bound*

$$\Pr[|Z - \mathbb{E}[Z]| \geq t] \leq 2\exp\left(-\frac{t^2}{2\sum_{i=1}^n \sigma_i^2}\right), \tag{3.22}$$

for all $t \in \mathbb{R}$.

Proof. Using the independence of $X_1, \ldots, X_n$, we have that for any $\lambda \in \mathbb{R}$:

$$\mathbb{E}\left[\exp\left\{\lambda(Z - \mathbb{E}[Z])\right\}\right] = \mathbb{E}\left[\prod_{i=1}^{n} \exp\left\{\lambda(X_i - \mathbb{E}[X_i])\right\}\right] \tag{3.23}$$

$$= \prod_{i=1}^{n} \mathbb{E}\left[\exp\left\{\lambda(X_i - \mathbb{E}[X_i])\right\}\right] \tag{3.24}$$

$$\leq \prod_{i=1}^{n} \exp\left(\frac{\lambda^2 \sigma_i^2}{2}\right) \tag{3.25}$$

$$= \exp\left(\frac{\lambda^2 \sum_{i=1}^{n} \sigma_i^2}{2}\right), \tag{3.26}$$

so Z is sub-Gaussian with variance proxy $\sum_{i=1}^{n} \sigma_i^2$. The tail bound then follows immediately from (3.14). □

The proof above demonstrates the value of the moment generating functions of sub-Gaussian random variables: they factorize conveniently when dealing with sums of independent random variables.

3.4.1 Examples of sub-Gaussian random variables

We now provide several examples of classes of random variables that are sub-Gaussian, some of which will appear repeatedly throughout the remainder of the course.

Example 3.11 (Rademacher random variables)**.** A *Rademacher random variable* ϵ takes a value of 1 with probability 1/2 and a value of -1 with probability 1/2. To see that ϵ is 1-sub-Gaussian, we follow Example 2.3 in [Wainwright, 2019] and upper bound the moment generating function of ϵ by way of a power series expansion of $\exp(\cdot)$:

$$\mathbb{E}[\exp(\lambda\epsilon)] = \frac{1}{2}\left\{\exp(-\lambda) + \exp(\lambda)\right\} \tag{3.27}$$

$$= \frac{1}{2}\left\{\sum_{k=0}^{\infty} \frac{(-\lambda)^k}{k!} + \sum_{k=0}^{\infty} \frac{\lambda^k}{k!}\right\} \tag{3.28}$$

$$= \sum_{k=0}^{\infty} \frac{\lambda^{2k}}{(2k)!} \qquad \text{(for odd } k\text{, } (-\lambda)^k + \lambda^k = 0\text{)} \tag{3.29}$$

$$\leq 1 + \sum_{k=1}^{\infty} \frac{\left(\lambda^2\right)^k}{2^k k!} \qquad \text{($2^k k!$ is every other term of $(2k)!$)} \tag{3.30}$$

$$= \exp(\lambda^2/2), \tag{3.31}$$

which is exactly the moment generating function bound (3.12) required for 1-sub-Gaussianity.

Example 3.12 (Random variables with bounded distance to mean)**.** Suppose a random variable X satisfies $|X - \mathbb{E}[X]| \leq M$ almost surely for some constant M. Then X is $O(M)$-sub-Gaussian.

We now provide an even more general class of sub-Gaussian random variables that subsume the random variables in Example 3.12:

Example 3.13 (Bounded random variables)**.** If X is a random variable such that $a \leq X \leq b$ almost surely for some constants $a, b \in \mathbb{R}$, then

$$\mathbb{E}\left[e^{\lambda(X - \mathbb{E}[X])}\right] \leq \exp\left[\frac{\lambda^2 (b-a)^2}{8}\right],$$

i.e., X is sub-Gaussian with variance proxy $(b-a)^2/4$. (We will prove this in Question 2(a) of Homework 1.) Note that combining the $(b-a)/2$-sub-Gaussianity of i.i.d. bounded random variables $X_1, \ldots, X_n$ and Theorem 3.10 yields a proof of Hoeffding's inequality.

Example 3.14 (Gaussian random variables)**.** If X is Gaussian with variance σ^2, then X satisfies (3.12) and (3.14) with equality. In this special case, the variance and the variance proxy are the same.

3.5 Concentrations of functions of random variables

We now introduce some important inequalities related to the second of our two goals, namely showing that for independent $X_1, \ldots, X_n$ and certain functions f, $f(X_1, \ldots, X_n)$ concentrates around $\mathbb{E}[f(X_1, \ldots, X_n)]$.

Theorem 3.15 (McDiarmid's inequality)**.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ satisfies the* bounded difference condition*: there exist constants $c_1, \ldots, c_n \in \mathbb{R}$ such that for all real numbers $x_1, \ldots, x_n$ and x_i',*

$$|f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_n)| \leq c_i. \tag{3.32}$$

(Intuitively, (3.32) *states that f is not overly sensitive to arbitrary changes in a single coordinate.) Then, for any independent random variables $X_1, \ldots, X_n$,*

$$\Pr\left[f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1, \ldots, X_n)] \geq t\right] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right). \tag{3.33}$$

Moreover, $f(X_1, \ldots, X_n)$ is $O\left(\sqrt{\sum_{i=1}^n c_i^2}\right)$-sub-Gaussian.

Remark 3.16. Note that McDiarmid's inequality is a generalization of Hoeffding's inequality with $f(x_1, \ldots, x_n) = \sum_{i=1}^n \min\{\max\{x_i, b\}, a\}$.

Proof. See the proof of Corollary 2.21 in [Wainwright, 2019], which relies on the Azuma-Hoeffding inequality for martingale difference sequences. □

A more general version of McDiarmid's inequality comes from Theorem 3.18 in [van Handel, 2016]. The setup for this theorem requires defining the *one-sided differences* of a function $f : \mathbb{R}^n \to \mathbb{R}$:

$$D_i^- f(x) = f(x_1, \ldots, x_n) - \inf_z f(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_n) \tag{3.34}$$

$$D_i^+ f(x) = \sup_z f(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_n). \tag{3.35}$$

These two quantities are functions of $x \in \mathbb{R}^n$, and hence can be interpreted as describing the sensitivity of f *at a particular point.* (Contrast this with the bounded difference condition (3.32), which bounds the sensitivity of f universally over all points.) For convenience, define

$$d^+ = \left\| \sum_{i=1}^n |D_i^+ f|^2 \right\|_\infty = \sup_{x_1, \ldots, x_n} \sum_{i=1}^n [|D_i^+ f(x_1, \ldots, x_n)]^2 \tag{3.36}$$

$$d^- = \left\| \sum_{i=1}^n |D_i^- f|^2 \right\|_\infty = \sup_{x_1, \ldots, x_n} \sum_{i=1}^n [D_i^- f(x_1, \ldots, x_n)]^2. \tag{3.37}$$

Theorem 3.17 (Bounded difference inequality, Theorem 3.18 in [van Handel, 2016])**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$, and let $X_1, \ldots, X_n$ be independent random variables. Then, for all $t \geq 0$,*

$$\Pr[f(X_1, \ldots, X_n) \geq \mathbb{E}[f(X_1, \ldots, X_n)] + t] \leq \exp\left(-\frac{t^2}{4d^-}\right) \tag{3.38}$$

$$\Pr[f(X_1, \ldots, X_n) \leq \mathbb{E}[f(X_1, \ldots, X_n)] - t] \leq \exp\left(-\frac{t^2}{4d^+}\right). \tag{3.39}$$

3.5.1 Bounds for Gaussian random variables

Unfortunately, the bounded difference condition (3.32) is often only satisfied by bounded random variables or a bounded function. To get similar concentration inequalities for unbounded random variables, we need some other special conditions. The following inequalities assume that the random variables have the standard normal distribution.

Theorem 3.18 (Gaussian Poincaré inequality, Corollary 2.27 in [van Handel, 2016])**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be smooth. If $X_1, \ldots, X_n$ are independently sampled from $\mathcal{N}(0,1)$, then*

$$\operatorname{Var}(f(X_1, \ldots, X_n)) \leq \mathbb{E}\left[\|\nabla f(X_1, \ldots, X_n)\|_2^2\right]. \tag{3.40}$$

Before introducing the next theorem, we recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is *L-Lipschitz* with respect to the ℓ_2-norm if there exists a non-negative constant $L \in \mathbb{R}$ such that for all $x, y \in \mathbb{R}^n$,

$$|f(x) - f(y)| \leq L\|x - y\|_2. \tag{3.41}$$

We emphasize that L is universal for all points in $\mathbb{R}^n$.

Theorem 3.19 (Theorem 2.26 in [Wainwright, 2019])**.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz with respect to Euclidean distance, and let $X = (X_1, \ldots, X_n)$, where $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. Then for all $t \in \mathbb{R}$,*

$$\Pr[|f(X) - \mathbb{E}[f(X)]| \geq t] \leq 2\exp\left(-\frac{t^2}{2L^2}\right). \tag{3.42}$$

In particular, $f(X)$ is sub-Gaussian.

Chapter 4

Generalization Bounds via Uniform Convergence

In Chapter 2, we pointed out some limitations of asymptotic analysis. In this chapter, we will turn our focus to *non-asymptotic analysis*, where we provide convergence guarantees without having the number of observations n go off to infinity. A key tool for proving such guarantees is *uniform convergence*, where we have bounds of the following form:

$$\Pr\left[\sup_{h\in\mathcal{H}} |\hat{L}(h) - L(h)| \leq \epsilon\right] \geq 1 - \delta. \tag{4.1}$$

In other words, the probability that the difference between our empirical loss and population loss is larger than ϵ is at most δ. We give motivation for uniform convergence and show how it can give us non-asymptotic guarantees on excess risk.

4.1 Basic concepts

A central goal of learning theory is to bound the *excess risk* $L(\hat{\theta}) - L(\theta^*)$. This is important as we don't want the expected risk of our ERM to be much larger than the expected risk of the best possible model. As we will see in the remainder of this section, uniform convergence is a technique that helps us achieve such bounds.

Uniform convergence is a property of a parameter set Θ, which gives us bounds of the form

$$\Pr\left[|\hat{L}(\theta) - L(\theta)| \geq \varepsilon\right] \leq \delta;\ \forall\theta \in \Theta. \tag{4.2}$$

In other words, uniform convergence tells us that for any choice of θ, our empirical risk is always close to our population risk with high probability. Let's look at a motivating example for why this type of bound is useful.

4.1.1 Motivation: Uniform convergence implies generalization

Consider the standard supervised learning setup where we have some i.i.d. $(x^{(i)}, y^{(i)})$. Furthermore, assume that we have a bounded loss function; specifically, suppose that $0 \leq \ell((x, y); \theta) \leq 1$, as in the case of the zero-one loss function. We show that uniform convergence implies generalization.

First, via telescoping sums, we can decompose the excess risk into three terms:

$$L(\hat{\theta}) - L(\theta^*) = \underbrace{L(\hat{\theta}) - \hat{L}(\hat{\theta})}_{①} + \underbrace{\hat{L}(\hat{\theta}) - \hat{L}(\theta^*)}_{②} + \underbrace{\hat{L}(\theta^*) - L(\theta^*)}_{③}. \tag{4.3}$$

We know that $\hat{L}(\hat{\theta}) - \hat{L}(\theta^*) \leq 0$ since $\hat{\theta}$ is a minimizer of $\hat{L}$. This allows us to write

$$L(\hat{\theta}) - L(\theta^*) \leq |L(\hat{\theta}) - \hat{L}(\hat{\theta})| + \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + |\hat{L}(\theta^*) - L(\theta^*)| \tag{4.4}$$

$$\leq |L(\hat{\theta}) - \hat{L}(\hat{\theta})| + 0 + |\hat{L}(\theta^*) - L(\theta^*)| \tag{4.5}$$

$$\leq 2 \sup_{\theta \in \Theta} |L(\theta) - \hat{L}(\theta)|. \tag{4.6}$$

This result tells us that if $\sup_{\theta \in \Theta} |L(\theta) - \hat{L}(\theta)|$ is small (say, less than $\varepsilon/2$), then excess risk $L(\hat{\theta}) - L(\theta^*)$ is less than ε. But this is exactly in the form of the bound in (4.2). Hence, if we can show that a parameter family exhibits uniform convergence, we can get a bound on excess risk as well.

For future references, Equation (4.6) can be strengthened straightforwardly into the following with slightly more careful treatment of the signs of each term:

$$L(\hat{\theta}) - L(\theta^*) \leq |\hat{L}(\theta^*) - L(\theta^*)| + L(\hat{\theta}) - \hat{L}(\hat{\theta}) \leq |\hat{L}(\theta^*) - L(\theta^*)| + \sup_{\theta \in \Theta} \left(L(\hat{\theta}) - \hat{L}(\hat{\theta}) \right) \tag{4.7}$$

This will make some of our future derivations technically slightly more convenient, but the nuanced difference between Equations (4.6) and (4.7) does not change the fundamental idea and the discussions in this chapter.

Let us try to apply our knowledge of concentration inequalities to this problem. Earlier we assumed that $\ell((x, y); \theta)$ is bounded, so we can bound (3) by $\widetilde{O}\left(\frac{1}{\sqrt{n}}\right)$ via Hoeffding's inequality (Remark 3.3). However, we cannot apply the same concentration inequality to (1): since $\hat{\theta}$ is data-dependent by definition, the i.i.d. assumption no longer holds. (To see this, note that $\hat{\theta}$ depends on the training dataset $(x^{(i)}, y^{(i)})$, so the terms in $\widehat{L}(\theta)$, $\ell((x^{(i)}, y^{(i)}); \hat{\theta})$, all depend on the training dataset too.) This is concerning: it is certainly possible that $L(\hat{\theta}) - \hat{L}(\hat{\theta})$ is large. You've probably encountered this yourself when a model exhibits low training loss, but high validation/testing loss.

4.1.2 Deriving uniform convergence bounds

Uniform convergence is one way we can fix this issue. The high-level idea is as follows:

- Suppose we have a bound of the form $\Pr[|\hat{L}(\theta) - L(\theta)| \geq \varepsilon'] \leq \delta'$ for some single, fixed choice of θ.
- If we know *all possible values of* θ in advance, we can use the above bound to create a more general bound over all values of θ.

In particular, we can use the union-bound inequality to create the general bound described in the second bullet point, using the bound in the first bullet point:

$$\Pr\left[\forall \theta \in \Theta, |\hat{L}(\theta) - L(\theta)| \geq \varepsilon'\right] \leq \sum_{\theta \in \Theta} \Pr\left[|\hat{L}(\theta) - L(\theta)| \geq \varepsilon'\right]. \tag{4.8}$$

We can then use Hoeffding's inequality to deal with the summands as θ there is no longer data-dependent. We will talk more later about proving statements of this form.

4.1.3 Intuitive interpretation of uniform convergence

Since uniform convergence implies generalization, if we know that population risk and empirical risk are always "close," then excess risk is "small" as well (Figure 4.1). In fact, it is possible to show that not only is $L(\theta)$ "close" to $\hat{L}(\theta)$ for sufficiently large data, but that the "shape" of $\hat{L}$ is "close" to the shape of L as well (Figure 4.2). This holds for the convex case; furthermore, there are conditions under which this holds in the non-convex case, for which a rigorous treatment can be found in [Mei et al., 2017]. (*Figure design and some wording in this section was inspired by [Liang, 2016, Liu and Thomas, 2018].*)

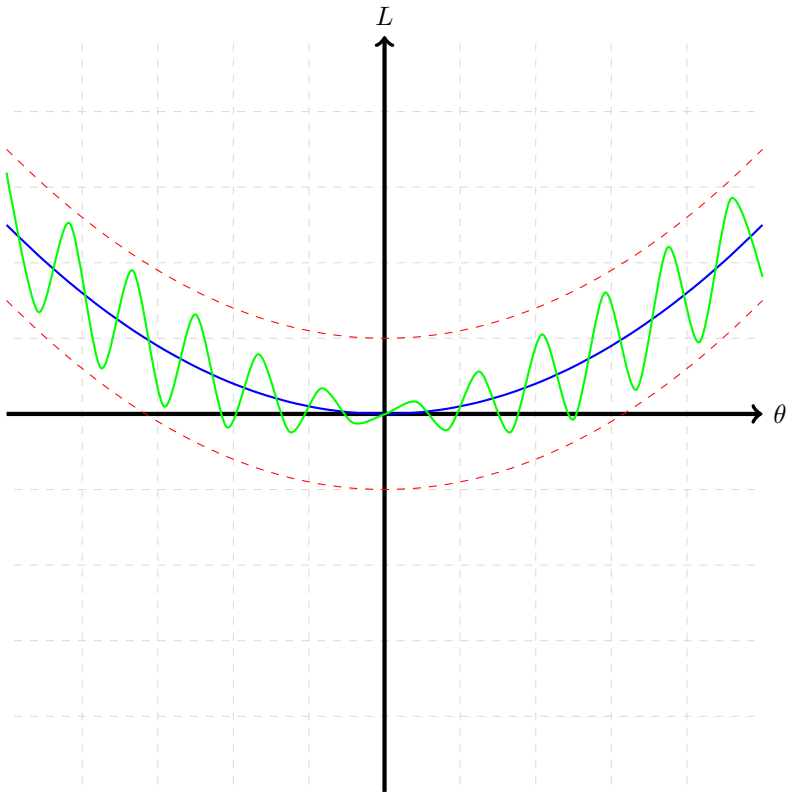

Figure 4.1: Empirical risk landscape under uniform convergence: Green: empirical risk, blue: population risk, red, dashed: ε additive error bounds for excess risk.

4.2 Finite hypothesis class

In this section, assume that $\mathcal{H}$ is finite. The following theorem gives a bound for the excess risk $L(\hat{h}) - L(h^*)$, where $\hat{h}$ and h^* are the minimizers of the empirical loss and population loss respectively.

Theorem 4.1. *Suppose that our hypothesis class $\mathcal{H}$ is finite and that our loss function ℓ is bounded in $[0,1]$, i.e. $0 \leq \ell((x,y),h) \leq 1$. Then $\forall \delta$ s.t. $0 < \delta < \frac{1}{2}$, with probability at least $1-\delta$, we have*

$$|L(h) - \hat{L}(h)| \leq \sqrt{\frac{\ln|\mathcal{H}| + \ln(2/\delta)}{2n}} \qquad \forall h \in \mathcal{H}. \tag{4.9}$$

As a corollary, we also have

$$L(\hat{h}) - L(h^*) \leq \sqrt{\frac{2(\ln|\mathcal{H}| + \ln(2/\delta))}{n}}. \tag{4.10}$$

Proof. We will prove this in two steps:

1. Use concentration inequalities to prove the bound for a fixed $h \in \mathcal{H}$, then
2. Use a union bound across the h's. (Recall that if $E_1, \ldots, E_k$ are a finite set of events, then the union bound states that $\Pr(E_1 \cup \cdots \cup E_k) \leq \sum_{i=1}^{k} \Pr(E_i)$.)

Fix some $\epsilon > 0$. By applying Hoeffding's inequality on the $\ell((x^{(i)}, y^{(i)}), h)$, we know that

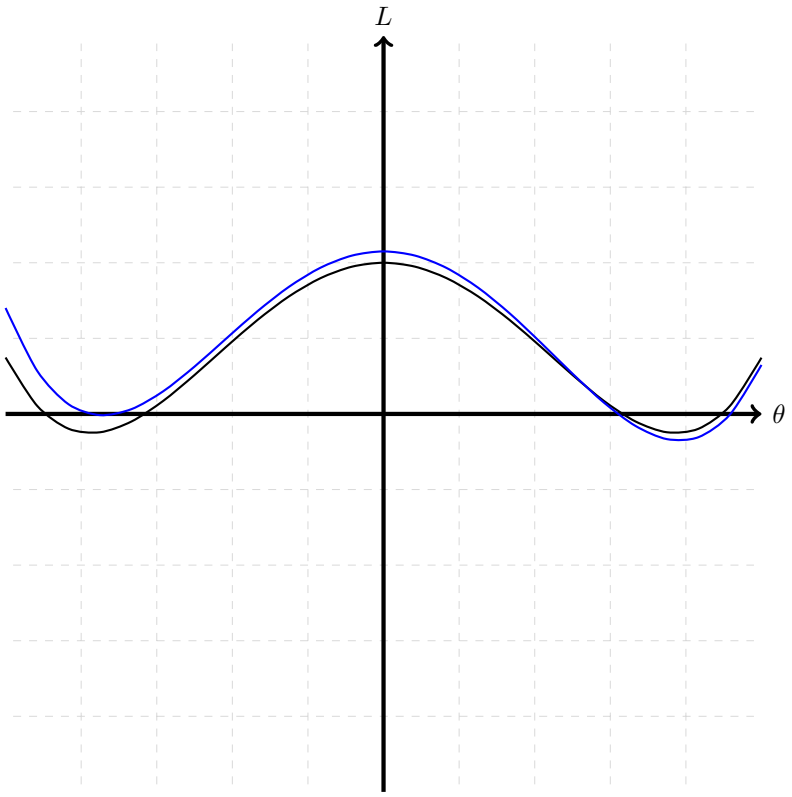

Figure 4.2: Empirical risk landscape under uniform convergence: Blue: empirical risk, black: population risk.

$$\Pr\left(|\hat{L}(h) - L(h)| \geq \epsilon\right) \leq 2\exp\left(-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{4.11}$$

$$= 2\exp\left(-\frac{2n^2\epsilon^2}{n}\right) \tag{4.12}$$

$$= 2\exp(-2n\epsilon^2), \tag{4.13}$$

since we can set $a_i = 0, b_i = 1$. The bound above holds for a single fixed h. To prove a similar inequality that holds for all $h \in \mathcal{H}$, we apply the union bound with $E_h = \{|\hat{L}(h) - L(h)| \geq \epsilon\}$:

$$\Pr\left(\exists h \text{ s.t. } |\hat{L}(h) - L(h)| \geq \epsilon\right) \leq \sum_{h\in\mathcal{H}} \Pr\left(|\hat{L}(h) - L(h)| \geq \epsilon\right) \tag{4.14}$$

$$\leq \sum_{h\in\mathcal{H}} 2\exp(-2n\epsilon^2) \tag{4.15}$$

$$= 2|\mathcal{H}|\exp(-2n\epsilon^2). \tag{4.16}$$

If we take δ such that $2|\mathcal{H}|\exp(-2n\epsilon^2) = \delta$, then it follows that

$$\epsilon = \sqrt{\frac{\ln|\mathcal{H}| + \ln(2/\delta)}{2n}}, \tag{4.17}$$

which proves (4.9). (4.10) follows by the inequality we stated in Section 4.1.1 and taking $\epsilon = \sqrt{\frac{2(\ln|\mathcal{H}|+\ln(2/\delta))}{n}}$:

$$\Pr\left(|L(\hat{h}) - L(h^*)| \geq \epsilon\right) \leq \Pr\left(2\sup_{h\in\mathcal{H}}|L(\hat{h}) - L(h^*)| \geq \epsilon\right) \tag{4.18}$$

$$\leq 2|\mathcal{H}|\exp\left(-\frac{n\epsilon^2}{2}\right). \tag{4.19}$$

□

4.2.1 Comparing Theorem 4.1 with standard concentration inequalities

With standard concentration inequalities, we have the following bound that depends on empirical risk:

$$\forall h \in \mathcal{H}, \quad w.h.p. \quad |\hat{L}(h) - L(h)| \leq \tilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{4.20}$$

The bound here depends on each h. In contrast, the uniform convergence bound we obtain from (4.17) is uniform over all $h \in \mathcal{H}$:

$$w.h.p., \quad \forall h \in \mathcal{H}, \quad |\hat{L}(h) - L(h)| \leq \tilde{O}\left(\frac{\ln|\mathcal{H}|}{\sqrt{n}}\right), \tag{4.21}$$

if we omit the $\ln(1/\delta)$ factor (we can do this since $\ln(1/\delta)$ is small in general and we take $\delta = \frac{1}{poly(n)}$). Hence, the extra $\ln|\mathcal{H}|$ term that depends on the size of our finite hypothesis family $\mathcal{H}$ can be viewed as a trade-off in order to make the bound uniform.

Remark 4.2. There is no standard definition for the term *with high probability* (*w.h.p*). For this class, the term is equivalent to the condition that the probability is higher than $1 - n^{-c}$ for some constant c.

4.2.2 Comparing Theorem 4.1 with asymptotic bounds

We can also compare the bound in Theorem 4.1 with our original asymptotic bound, namely,

$$L(\hat{h}) - L(h^*) \leq \frac{c}{n} + o\left(n^{-1}\right). \tag{4.22}$$

The $o(n^{-1})$ term can vary significantly depending on the problem. For instance, both n^{-2} and $p^{100}n^{-2}$ are $o(n^{-1})$ but the second one converges much more slowly. With the new bound, there are no longer any constants hidden in an $o(n^{-1})$ term (in fact that term is no longer there). However, we now have a slower convergence rate of $O(n^{-1/2})$.

Remark 4.3. $O(n^{-1/2})$ convergence is sometimes known as the *slow rate* while $O(n^{-1})$ convergence is known as the *fast rate*. We were only able to get the slow rate from uniform convergence: we needed asymptotics to get the fast rate. (It is possible to get the fast rate from uniform convergence under certain conditions, e.g. when the population risk on the true h^* is very low.)

4.3 Bounds for infinite hypothesis class via discretization

Unfortunately, we cannot generalize the results from the previous section directly to the case where the hypothesis class $\mathcal{H}$ is infinite, since we cannot apply the union bound to an infinite number of hypothesis functions $h \in \mathcal{H}$. However, if we consider a *bounded* and *continuous* parameterized space of $\mathcal{H}$, then we can obtain a similar uniform bound by applying a technique called *brute-force discretization*.

For this section, assume that our infinite hypothesis class $\mathcal{H}$ can be parameterized by $\theta \in \mathbb{R}^p$ with $\|\theta\|_2 \leq B$ for some fixed $B > 0$. That is, we have

$$\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}, \|\theta\|_2 \leq B\}. \tag{4.23}$$

The intuition behind brute-force discretization is as follows: Let $E_\theta = \{|\widehat{L}(\theta) - L(\theta)| \geq \epsilon\}$ be the "bad" event. We want the bound the probability of any one of these bad events happening (i.e. $\bigcup_\theta E_\theta$). The union bound does not work as we end up with an infinite sum. However, the union bound is very loose: these events can overlap with each other significantly. Instead, we can try to find "prototypical" bad events $E_{\theta_1}, \ldots, E_{\theta_N}$ that are somewhat disjoint so that $\bigcup_\theta E_\theta \approx \bigcup_{i=1}^N E_{\theta_i}$. We can then use the union bound on $\bigcup_{i=1}^N E_{\theta_i}$ to get a non-vacuous upper bound.

We make these ideas precise in the following section.

4.3.1 Discretization of the parameter space by ϵ-covers

We start by defining the notion of an *ϵ-cover* (also *ϵ-net*):

Definition 4.4 (ϵ-cover)**.** Let $\epsilon > 0$. An *ϵ-cover* of a set S with respect to distance metric ρ is a subset $C \subseteq S$ such that $\forall x \in S$, $\exists x' \in C$ such that $\rho(x, x') \leq \epsilon$, or equivalently,

$$S \subseteq \bigcup_{x \in C} \text{Ball}(x, \epsilon, \rho), \quad \text{where} \tag{4.24}$$

$$\text{Ball}(x, \epsilon, \rho) \triangleq \{x' : \rho(x, x') \leq \epsilon\}. \tag{4.25}$$

(We note that in some definitions it is possible for points in C to lie outside of S; we do not worry about this technicality the class.) The following lemma tells us that our parameter space $S = \{\theta \in \mathbb{R}^p : \|\theta\|_2 \leq B\}$ has an ϵ-cover with not too many elements:

Lemma 4.5 (ϵ-cover of ℓ_2 ball)**.** *Let $B, \epsilon > 0$ with $\epsilon \leq B\sqrt{p}$, and let $S = \{x \in \mathbb{R}^p : \|x\|_2 \leq B\}$. Then there exists an ϵ-cover of S with respect to the ℓ_2-norm with at most $\left(\frac{3B\sqrt{p}}{\epsilon}\right)^p$ elements.*

Proof. Set

$$C = \left\{x \in S : x_i = k_i \frac{\epsilon}{\sqrt{p}}, k_i \in \mathbb{Z}, |k_i| \leq \frac{B\sqrt{p}}{\epsilon}\right\}, \tag{4.26}$$

i.e. C is the set of grid points in $\mathbb{R}^p$ of width $\frac{\epsilon}{\sqrt{p}}$ that are contained in S. See Figure 4.3 for an illustration.

We claim that C is an ϵ-cover of S with respect to the ℓ_2-norm: $\forall x \in S$, there exists a grid point $x' \in C$ such that $|x_i - x_i'| \leq \frac{\epsilon}{\sqrt{p}}$ for each i. Therefore,

$$\|x - x'\|_2 = \sqrt{\sum_{i=1}^p |x_i - x_i'|^2} \leq \sqrt{p \cdot \frac{\epsilon^2}{p}} = \epsilon.$$

We now bound the size of C. Since each k_i in the definition of C has at most $2\frac{B\sqrt{p}}{\epsilon} + 1$ choices, we have

$$|C| \leq \left(\frac{2B\sqrt{p}}{\epsilon} + 1\right)^p \leq \left(\frac{3B\sqrt{p}}{\epsilon}\right)^p. \tag{4.27}$$

□

Remark 4.6. If $\epsilon > B\sqrt{p}$, then S is contained in the ball centered at the origin with radius ϵ and the ϵ-cover has size 1.

Remark 4.7. We can actually prove a stronger version of Lemma 4.5: there exists an ϵ-cover of S with at most $\left(\frac{3B}{\epsilon}\right)^p$ elements. We will be using this version of the lemma in the proof below. (We will leave the proof of this stronger version as a homework exercise.)

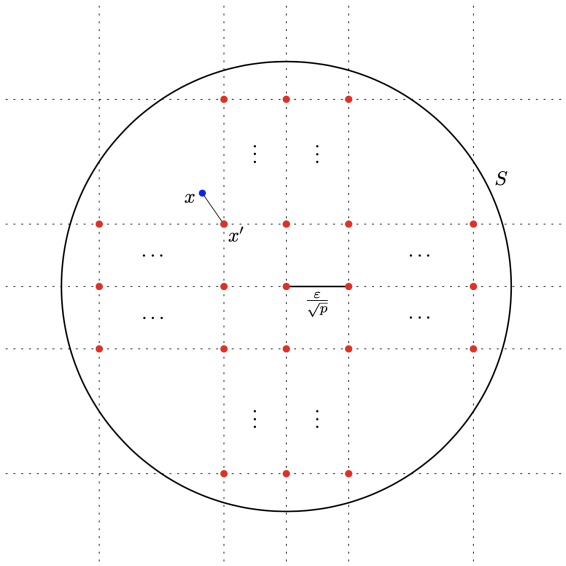

Figure 4.3: Our chosen ϵ-cover (shown in red) of S. For $x \in S$, we choose the grid point x' such that $\|x - x'\|_2 \leq \epsilon$.

4.3.2 Uniform convergence bound for infinite $\mathcal{H}$

Definition 4.8 (κ-Lipschitz functions)**.** Let $\kappa \geq 0$ and $\|\cdot\|$ be a norm on the domain D. A function $L : D \to \mathbb{R}$ is said to be *κ-Lipschitz* with respect to $\|\cdot\|$ if for all $\theta, \theta' \in D$, we have

$$|L(\theta) - L(\theta')| \leq \kappa \|\theta - \theta'\|.$$

Assume that our infinite hypothesis class $\mathcal{H}$ can be parameterized by $\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}, \|\theta\|_2 \leq B\}$. We have the following uniform convergence theorem for our infinite hypothesis class $\mathcal{H}$:

Theorem 4.9. *Suppose $\ell((x, y), \theta) \in [0, 1]$, and $\ell((x, y), \theta)$ is κ-Lipschitz in θ with respect to the ℓ_2-norm for all (x, y). Then, with probability at least $1 - O(\exp(-\Omega(p)))$, we have*

$$\forall \theta, \quad |\hat{L}(\theta) - L(\theta)| \leq O\left(\sqrt{\frac{p \max(\ln(\kappa B n), 1)}{n}}\right). \tag{4.28}$$

Proof of Theorem 4.9. Fix parameters $\delta, \epsilon > 0$ (we will specify their values later). Let C be the ϵ-cover of our parameter space S with respect to the ℓ_2-norm constructed in Lemma 4.5. Define event $E = \left\{\forall \theta \in C,\ |\hat{L}(\theta) - L(\theta)| \leq \delta\right\}$. By Theorem 4.1, we have $\Pr(E) \geq 1 - 2|C| \exp(-2n\delta^2)$.

Now for any $\theta \in S$, we can pick some $\theta_0 \in C$ such that $\|\theta - \theta_0\|_2 \leq \epsilon$. Since L and $\widehat{L}$ are κ-Lipschitz functions (this follows from the Lipschitzness of ℓ), we have

$$|L(\theta) - L(\theta_0)| \leq \kappa \|\theta - \theta_0\|_2 \leq \kappa\epsilon, \text{ and} \tag{4.29}$$

$$|\hat{L}(\theta) - \hat{L}(\theta_0)| \leq \kappa \|\theta - \theta_0\|_2 \leq \kappa\epsilon. \tag{4.30}$$

Therefore, conditional on E, we have

$$|\hat{L}(\theta) - L(\theta)| \leq |\hat{L}(\theta) - \hat{L}(\theta_0)| + |\hat{L}(\theta_0) - L(\theta_0)| + |L(\theta_0) - L(\theta)| \leq 2\kappa\epsilon + \delta. \tag{4.31}$$

It remains to choose suitable parameters δ and ϵ to get the desired bound in Theorem 4.9 while making the failure probability small. First, set $\epsilon = \delta/(2\kappa)$ so that conditional on E,

$$|\hat{L}(\theta) - L(\theta)| \leq 2\delta. \tag{4.32}$$

If we set $\delta = \sqrt{\frac{c_0 p \max(1, \ln(\kappa B n))}{n}}$ with $c_0 = 36$ (see Remark 4.10 for some intuition), then by Remark 4.7,

$$\ln|C| - 2n\delta^2 \le p \ln\left(\frac{6B\kappa}{\delta}\right) - 2n\delta^2 \tag{4.33}$$

$$\le p \ln\left(\frac{6B\kappa\sqrt{n}}{\sqrt{c_0 p \max(1, \ln(\kappa B n))}}\right) - 2n\frac{c_0 p}{n}\ln(\kappa B n) \qquad \text{(dfn of } \delta) \tag{4.34}$$

$$\le p \ln\left(\frac{B\kappa\sqrt{n}}{\sqrt{p}}\right) - 72p\ln(\kappa B n) \qquad (\max(1, \ln(\kappa B n)) \ge 1, c_0 = 36) \tag{4.35}$$

$$\le p\ln(B\kappa n) - 72p\ln(B\kappa n) \qquad (\sqrt{n/p} \le n) \tag{4.36}$$

$$\le -p, \tag{4.37}$$

since $\ln(B\kappa n) \ge 1$ for large enough n. Therefore, with probability greater than $1 - 2|C|\exp(-2n\delta^2) = 1 - 2\exp(\ln|C| - 2n\delta^2) \ge 1 - O(e^{-p})$, we have

$$|\hat{L}(\theta) - L(\theta)| \le 2\delta = O\left(\sqrt{\frac{p}{n}\max(1, \ln(\kappa B n))}\right). \tag{4.38}$$

□

Remark 4.10. Here is the intuition for the choice of δ: The event E happens with probability $1 - 2|C|\exp(-2n\delta^2) = 1 - 2\exp(\ln|C| - 2n\delta^2)$. From Remark 4.7, we know that $\ln|C| \le p\ln(3B/(\delta/2))$. If we ignore the log term and assume $\ln|c| \le p$, then this would give us the high probability bound we want:

$$2|C|\exp(-2n\delta^2) = 2\exp(\ln|C| - 2n\delta^2) \le 2\exp(p - 2p) = 2\exp(-p). \tag{4.39}$$

(At the same time, we see from (4.32) that this choice of δ gives $|\hat{L}(\theta) - L(\theta)| \le 2\sqrt{\frac{p}{n}}$, which is roughly the bound we want.)

Since we cannot drop the log term in the inequality, we need to make δ a little big bigger. δ in the proof was chosen with this intuition in mind to make the subsequent chain of logic work.

Remark 4.11. We bounded the generalization error $|\hat{L}(\theta) - L(\theta)|$ by $\delta + 2\epsilon\kappa \le \sqrt{\frac{\ln|C|}{n}} + 2\epsilon\kappa$. The term $2\epsilon\kappa$ represents the error from our brute-force discretization. It is not a problem is because we can always choose ϵ small enough without worrying about the growth of the first term $\sqrt{\frac{\ln|C|}{n}}$. This in turn is because $\ln|C| \approx p\ln\epsilon^{-1}$, which is very insensitive to ϵ, even if we let $\epsilon = \frac{1}{poly(n)}$. We also observe that both $\sqrt{\frac{\ln|C|}{n}}$ and $\sqrt{\frac{p}{n}}$ are bounds that depend on the "size" of our hypothesis class, in terms of either its total size or dimensionality. This possibly explains why one may need more training samples when the hypothesis class is larger.

4.4 Rademacher complexity

4.4.1 Motivation for a new complexity measure

Recall that our goal is to bound the *excess risk* $L(\hat{h}) - L(h^*)$, where L is the expected loss (or population loss), $\hat{h}$ is our estimated hypothesis and h^* is the hypothesis in the hypothesis class $\mathcal{H}$ which minimizes the expected loss. We previously showed that to do so it suffices to upper bound $\sup_{h\in\mathcal{H}}(L(h) - \widehat{L}(h))$. (Note: we often call $L(\hat{h}) - \widehat{L}(\hat{h})$ the *generalization gap* or *generalization error*.)

In the previous sections, we derived bounds for the generalization gap in two cases:

1. If the hypothesis class $\mathcal{H}$ is finite,

$$L(\hat{h}) - \hat{L}(\hat{h}) \le \tilde{O}\left(\sqrt{\frac{\log|\mathcal{H}|}{n}}\right). \tag{4.40}$$

2. If the hypothesis class $\mathcal{H}$ is p-dimensional,

$$L(\hat{h}) - \hat{L}(\hat{h}) \leq \tilde{O}\left(\sqrt{\frac{p}{n}}\right). \tag{4.41}$$

Both of these bounds have a $\frac{1}{\sqrt{n}}$-dependency on n, which is known as the "slow rate". The terms in the numerator ($\log|\mathcal{H}|$ and p resp.) can be thought of as complexity measure of $\mathcal{H}$.

The bound (4.41) is not precise enough: it depends solely on p and is not always optimal. For example, this would be a poor bound if the hypothesis class $\mathcal{H}$ has very high dimension but small norm. One specific example is for the following two hypothesis classes:

$$\{\theta : \|\theta\|_1 \leq B\} \qquad \text{vs.} \qquad \{\theta : \|\theta\|_2 \leq B\},$$

(4.41) would give both hypothesis classes the same bound of $\tilde{O}\left(\sqrt{\frac{p}{n}}\right)$. Intuitively, we should take into account the norms for a better bound.

With the complexity measure to be introduced, we will prove a bound of the form

$$L(\hat{h}) - \hat{L}(\hat{h}) \leq \tilde{O}\left(\sqrt{\frac{\text{Complexity}(\Theta)}{n}}\right). \tag{4.42}$$

This complexity measure will depend on the distribution p over $\mathcal{X} \times \mathcal{Y}$ (the input and output spaces), and hence takes into account how easy it is to learn p. If p is easy to learn, then this complexity measure will be small even if the hypothesis space is big.

One of the practical implications of having such a complexity measure is that we can restrict the hypothesis space by regularizing the complexity measure (assuming it is something we can evaluate and train with). If we successfully find a low complexity model, then this generalization bound guarantees that we have not overfit.

4.4.2 Definitions

In uniform convergence, we sought a high probability bound for $\sup_{h\in H}(L(h)-\hat{L}(h))$. Here we have a weaker goal: we try to obtain an upper bound for its expectation instead, i.e.

$$\mathbb{E}_{z\sim X\times Y}\left[\sup_{h\in H}(L(h) - \hat{L}(h))\right] \leq \text{ upper bound.} \tag{4.43}$$

The expectation is over the randomness in the training data ($\mathcal{X}\times\mathcal{Y}$). (Note: We cannot just swap the order of $\mathbb{E}$ and sup!)

To do so, we first define *Rademacher complexity*.

Definition 4.12 (Rademacher complexity)**.** Let $\mathcal{F}$ be a family of functions mapping $Z \mapsto \mathbb{R}$, and let P be a distribution over Z. The *(average) Rademacher complexity* of $\mathcal{F}$ is defined as

$$R_n(F) \triangleq \mathbb{E}_{z_1,\ldots,z_n \overset{\text{iid}}{\sim} P}\left[\mathbb{E}_{\sigma_1,\ldots,\sigma_n \overset{\text{iid}}{\sim}\{\pm 1\}}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(z_i)\right]\right], \tag{4.44}$$

where $\sigma_1,\ldots,\sigma_n$ are independent *Rademacher random variables*, i.e. each taking on the value of 1 or -1 with probability 1/2.

Remark 4.13. For our applications we will take $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$. However, Definition 4.12 holds for abstract input spaces $\mathcal{Z}$ as well.

Remark 4.14. Note that $R_n(\mathcal{F})$ is also dependent on the measure P of the space, so technically it should be $R_{n,P}(\mathcal{F})$, but for brevity, we refer to it as $R_n(\mathcal{F})$.

An interpretation is $R_n(\mathcal{F})$ is the maximal possible correlation between outputs of some $f \in \mathcal{F}$ (on points $f(z_1), \ldots, f(z_n)$) and random Rademacher variables $(\sigma_1, \ldots, \sigma_n)$. Essentially, functions with more random sign outputs will better match random patterns of Rademacher variables and have higher complexity (greater ability to mimic or express randomness).

The following theorem is the main theorem involving Rademacher complexity:

Theorem 4.15.

$$\mathbb{E}_{z_1,\ldots,z_n \overset{\text{iid}}{\sim} P}\left[\sup_{f\in F}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}_{z\sim P} f(x)\right]\right] \leq 2R_n(\mathcal{F}). \tag{4.45}$$

Remark 4.16. We can think of $\frac{1}{n}\sum_{i=1}^{n} f(z_i)$ as an empirical average and $\mathbb{E}_{z\sim P} f(x)$ as a population average.

Why is Theorem 4.15 useful to us? We can set $\mathcal{F}$ to be the family of loss functions, i.e.

$$\mathcal{F} = \{z = (x, y) \in \mathcal{Z} \mapsto \ell((x, y), h) \in \mathbb{R} : h \in \mathcal{H}\}. \tag{4.46}$$

This is the family of losses induced from the hypothesis functions in $\mathcal{H}$. Then by Theorem 4.15,

$$\mathbb{E}_{z_i}\left[\sup_{f\in F}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}(f)\right)\right] = \mathbb{E}_{(x^{(i)},y^{(i)})}\left[\sup_{h\in\mathcal{H}}\left[\frac{1}{n}\sum_{i=1}^{n}\ell((x^{(i)}, y^{(i)}, h)) - L(h)\right]\right] \tag{4.47}$$

$$= \mathbb{E}\left[\sup_{h\in\mathcal{H}}\left(\hat{L}(h) - L(h)\right)\right] \tag{4.48}$$

$$\leq 2R_n(\mathcal{F}), \tag{4.49}$$

where $z_i = (x^{(i)}, y^{(i)})$. Thus, $2R_n(\mathcal{F})$ is an upper bound for the generalization error. In this context, $R_n(\mathcal{F})$ can be interpreted as how well the loss sequence $\ell((x^{(1)}, y^{(1)}), h), \ldots \ell((x^{(n)}, y^{(n)}), h)$ correlates with $\sigma_1, \ldots, \sigma_n$.

Example 4.17. Consider the binary classification setting where $y \in \{\pm 1\}$. Let ℓ_{0-1} denote the zero-one loss function. Note that

$$\ell_{0-1}((x, y), h) = \mathbf{1}\{h(x) \neq y\} = \frac{1 - yh(x)}{2}. \tag{4.50}$$

Hence,

$$R_n(\mathcal{F}) = \mathbb{E}_{(x^{(i)},y^{(i)}),\sigma_i}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\ell_{0-1}((x^{(i)}, y^{(i)}), h)\sigma_i\right] \quad \text{(by definition)} \tag{4.51}$$

$$= \mathbb{E}_{(x^{(i)},y^{(i)}),\sigma_i}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\left(\frac{-h(x^{(i)})y^{(i)} + 1}{2}\right)\sigma_i\right] \quad \text{(by (4.50))} \tag{4.52}$$

$$= \frac{1}{2}\mathbb{E}_{(x^{(i)},y^{(i)}),\sigma_i}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i + \sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} -h(x^{(i)})y^{(i)}\sigma_i\right] \quad \text{(sup only over } \mathcal{H}) \tag{4.53}$$

$$= \frac{1}{2}\mathbb{E}_{(x^{(i)},y^{(i)}),\sigma_i}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} -h(x^{(i)})y^{(i)}\sigma_i\right] \quad (\mathbb{E}[\sigma_i] = 0) \tag{4.54}$$

$$= \frac{1}{2}\mathbb{E}_{(x^{(i)},y^{(i)}),\sigma_i}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} h(x^{(i)})\sigma_i\right] \quad (-y_i\sigma_i \overset{d}{=} \sigma_i) \tag{4.55}$$

$$= \frac{1}{2}R_n(\mathcal{H}). \quad \text{(by definition)} \tag{4.56}$$

In this setting, $R_n(\mathcal{F})$ and $R_n(\mathcal{H})$ are the same (except for the factor of 2). $R_n(\mathcal{H})$ has a slightly more intuitive interpretation: it represents how well $h \in \mathcal{H}$ can fit random patterns.

Warning! $R_n(\mathcal{F})$ is not always the same as $R_n(\mathcal{H})$ in other problems.

Remark 4.18. Rademacher complexity is invariant to translation. One example of this in play is in how the $+1$ in the $\left(\frac{-h(x^{(i)})y^{(i)}+1}{2}\right)$ term essentially vanishes in the computation.

Let us now prove Theorem 4.15.

Proof of Theorem 4.15. We use a technique called *symmetrization*, which is a very important technique in probability theory. We first fix $z_1, \ldots, z_n$and draw $z_1', \ldots z_n' \overset{\text{iid}}{\sim} P$. Then we can rewrite the term in the expectation on the LHS of (4.45):

$$\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}[f]\right) = \sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}_{z_1',\ldots,z_n'}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i')\right]\right) \tag{4.57}$$

$$= \sup_{f\in\mathcal{F}}\left(\mathbb{E}_{z_1',\ldots,z_n'}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \frac{1}{n}\sum_{i=1}^{n} f(z_i')\right]\right) \tag{4.58}$$

$$\leq \mathbb{E}_{z_1',\ldots,z_n'}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \frac{1}{n}\sum_{i=1}^{n} f(z_i')\right)\right]. \tag{4.59}$$

The last inequality is because in general,

$$\sup_u\left(\mathbb{E}_v[g(u,v)]\right) \leq \sup_u\left(\mathbb{E}_v\left[\sup_{u'} g(u',v)\right]\right) = \mathbb{E}_v\left[\sup_u(g(u,v))\right] \tag{4.60}$$

since the outer sup becomes vacuous.

Now, if we take the expectation over $z_1, \ldots, z_n$ for both sides of (4.59),

$$\mathbb{E}_{z_1,\ldots,z_n}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}[f]\right)\right] \leq \mathbb{E}_{z_i}\left[\mathbb{E}_{z_i'}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\left(f(z_i) - f(z_i')\right)\right)\right]\right] \tag{4.61}$$

$$= \mathbb{E}_{z_i,z_i'}\left[\mathbb{E}_{\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i\left(f(z_i) - f(z_i')\right)\right)\right]\right] \tag{4.62}$$

$$\leq \mathbb{E}_{z_i,z_i',\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(z_i)\right) + \sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} -\sigma_i f(z_i')\right)\right] \tag{4.63}$$

$$= 2R_n(\mathcal{F}), \tag{4.64}$$

where (4.62) is because $\sigma_i(f(z_i) - f(z_i')) \overset{d}{=} f(z_i) - f(z_i')$ since $f(z_i) - f(z_i')$ has a symmetric distribution. The last equality holds since $-\sigma_i \overset{d}{=} \sigma_i$ and z_i, z_i' are drawn iid from the same distribution. □

Here is an intuitive understanding of what Theorem 4.15 achieves. Consider the quantities on the LHS and RHS of (4.45):

$$\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}[f]\right) \qquad \text{v.s.} \qquad \sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(z_i)\right).$$

First, we removed $\mathbb{E}[f]$, which is hard to control quantitatively since it is deterministic. Second, we added more randomness in the form of Rademacher variables. This will allow us to shift our focus from the randomness in the z_i's to the randomness in the σ_i's. In the future, our bounds on the Rademacher complexity will typically only depend on the randomness from the σ_i's.

4.4.3 Dependence of Rademacher complexity on P

For intuition on how Rademacher complexity depends on the distribution P, consider the extreme example where P is a point mass, i.e. $z = z_0$ almost surely. Assume that $-1 \le f(z_0) \le 1$ for all $f \in \mathcal{F}$. Then

$$\mathbb{E}_{z_1,\dots,z_n \sim P}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right] = \mathbb{E}_{\sigma_1,\dots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n} f(z_0)\sum_{i=1}^n \sigma_i\right] \tag{4.65}$$

$$\le \mathbb{E}_{\sigma_1,\dots,\sigma_n}\left[\left|\frac{1}{n}\sum_{i=1}^n \sigma_i\right|\right] \qquad \text{(since } f(z_0) \in [-1,1]\text{)} \tag{4.66}$$

$$\le \mathbb{E}_{\sigma_i}\left[\left(\frac{1}{n}\sum_{i=1}^n \sigma_i\right)^2\right]^{\frac{1}{2}} \qquad \text{(Jensen's Inequality)} \tag{4.67}$$

$$= \frac{1}{n}\left(\mathbb{E}_{\sigma_i,\sigma_j}\left[\sum_{i,j=1}^n \sigma_i\sigma_j\right]\right)^{\frac{1}{2}} \tag{4.68}$$

$$= \frac{1}{n}\left(\mathbb{E}_{\sigma_i}\left[\sum_{i=1}^n \sigma_i^2\right]\right)^{\frac{1}{2}} \tag{4.69}$$

$$= \frac{1}{n}\cdot\sqrt{n} = \frac{1}{\sqrt{n}}. \tag{4.70}$$

This bound does not depend on $\mathcal{F}$ (except that it is bounded). This example illustrates that sometimes we only need to depend on the distribution of Rademacher random variables.

4.5 Empirical Rademacher complexity

In the previous section, we bounded the expectation of $\sup_{f\in F}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}_{z\sim P} f(x)\right]$. This expectation is taken over the training examples $z_1,\dots,z_n$. In many instances we only have one training set, and do not have access to many training sets. Thus, the bound on the expectation does not give a guarantee for the one training set that we have. In this section, we seek to bound the quantity itself with high probability.

Definition 4.19 (Empirical Rademacher complexity)**.** Given a dataset $S = \{z_1,\dots,z_n\}$, the *emprical Rademacher complexity* is defined as

$$R_S(\mathcal{F}) \triangleq \mathbb{E}_{\sigma_1,\dots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right]. \tag{4.71}$$

$R_S(\mathcal{F})$ is a function of both the function class $\mathcal{F}$ and the dataset S.

Note that, as the name suggests, the expectation of the empirical Rademacher complexity is the Rademacher complexity:

$$R_n(\mathcal{F}) = \mathop{\mathbb{E}}_{\substack{z_1,\dots,z_n \overset{\text{iid}}{\sim} P \\ S=\{z_1,\dots,z_n\}}}\left[R_S(\mathcal{F})\right]. \tag{4.72}$$

Here is the theorem involving empirical Rademacher complexity:

Theorem 4.20. *Suppose for all $f \in \mathcal{F}$, $0 \le f(x) \le 1$. Then, with probability at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}[f]\right] \le 2R_S(F) + 3\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{4.73}$$

Proof. For conciseness, define

$$g(z_1, \ldots, z_n) \triangleq \sup_{f \in F} \left[\frac{1}{n} \sum_{i=1}^{n} f(z_i) - \mathbb{E}[f] \right]. \tag{4.74}$$

We prove the theorem in 4 steps.

Step 1: We bound g using McDiarmid's Inequality. To use McDiarmid's inequality, we check that the bounded difference condition holds:

$$\begin{aligned} g(z_1, \ldots, z_n) - g(z_1, \ldots, z_i', \ldots, z_n) &\leq \sup_{f \in \mathcal{F}} \left[\frac{1}{n} \sum_{j=1}^{n} f(z_j) \right] - \sup_{f \in \mathcal{F}} \left[\left(\frac{1}{n} \sum_{j=1, j \neq i}^{n} f(z_j) \right) + \frac{f(z_i')}{n} \right] && (4.75) \\ &\leq \sup_{f \in \mathcal{F}} \left[\frac{1}{n} \left(f(z_i) - f(z_i') \right) \right] && (4.76) \\ &\leq \frac{1}{n}. && (4.77) \end{aligned}$$

(4.76) holds because in general, $\sup_f A(f) - \sup_f B(f) \leq \sup_f [A(f) - B(f)]$, and (4.77) holds since f is bounded by $[0, 1]$. We can thus apply McDiarmid's Inequality with parameters $c_1 = \cdots = c_n = 1/n$:

$$\mathbb{P}\left[g(z_1, \ldots, z_n) \geq \mathbb{E}_{z_1, \ldots, z_n \overset{\text{iid}}{\sim} P}[g] + \epsilon \right] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{n} c_i^2} \right) = \exp(-2n\epsilon^2). \tag{4.78}$$

Step 2: We apply Theorem 4.15 to get

$$\mathbb{E}_{z_1, \ldots, z_n \overset{\text{iid}}{\sim} P}[g] \leq 2R_n(\mathcal{F}). \tag{4.79}$$

Step 3: Define

$$\tilde{g}(z_1, \ldots, z_n) = R_S(\mathcal{F}) \triangleq \mathbb{E}_{\sigma_i} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(z_i) \right]. \tag{4.80}$$

Using a similar argument like that in Step 1, we show that $\tilde{g}$ satisfies the bounded difference condition:

$$\begin{aligned} &\tilde{g}(z_1, \ldots, z_n) - \tilde{g}(z_1, \ldots, z_i', \ldots, z_n) \\ &\leq \mathbb{E}_{\sigma_i} \left[\sup_{f \in F} \left[\frac{1}{n} \sum_{j=1}^{n} \sigma_j f(z_j) \right] - \sup_{f \in F} \left[\left(\frac{1}{n} \sum_{j=1, j \neq i}^{n} \sigma_j f(z_j) \right) + \frac{1}{n} \sigma_i f(z_i') \right] \right] && (4.81) \\ &\leq \mathbb{E}_{\sigma_i} \left[\sup_{f \in F} \left(\frac{1}{n} \sigma_i (f(z_i) - f(z_i')) \right) \right] && (4.82) \\ &\leq \frac{1}{n}, && (4.83) \end{aligned}$$

since the term inside the sup is always upper bounded by 1. We can thus apply McDiarmid's Inequality with parameters $c_1 = \cdots = c_n = 1/n$:

$$\mathbb{P}\left[\tilde{g} - \mathbb{E}[\tilde{g}] \geq \epsilon \right] \leq \exp(-2n\epsilon^2), \quad \text{and} \quad \mathbb{P}\left[\tilde{g} - \mathbb{E}[\tilde{g}] \leq -\epsilon \right] \leq \exp(-2n\epsilon^2). \tag{4.84}$$

Step 4: We set δ such that $\exp(-2n\epsilon^2) = \delta/2$. (This implies that $\epsilon = \sqrt{\frac{\log(2/\delta)}{2n}}$.) Then, with probability $\geq 1 - \delta$,

$$\begin{aligned} \sup_{f \in \mathcal{F}} \left[\frac{1}{n} \sum_{i=1}^{n} f(z_i) - \mathbb{E}[f] \right] = g &\leq \mathbb{E}[g] + \epsilon && \text{(Step 1)} && (4.85) \\ &\leq 2R_n(\mathcal{F}) + \epsilon && \text{(Step 2)} && (4.86) \\ &\leq 2(R_S(\mathcal{F}) + \epsilon) + \epsilon && \text{(Step 3)} && (4.87) \\ &= 2R_S(\mathcal{F}) + 3\epsilon, && && (4.88) \end{aligned}$$

as required. □

Setting $\mathcal{F}$ to be a family of loss functions bounded by $[0,1]$ in Theorem 4.20 gives the following corollary:

Corollary 4.21. Let $\mathcal{F}$ to be a family of loss functions $\mathcal{F} = \{(x,y) \mapsto \ell((x,y),h) : h \in \mathcal{H}\}$ with $\ell((x,y),h) \in [0,1]$ for all ℓ, (x,y) and h. Then, with probability $1-\delta$, the generalization gap is

$$L(h) - \hat{L}(h) \leq 2R_S(F) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \quad \text{for all } h \in \mathcal{H}. \tag{4.89}$$

Remark 4.22. If we want to bound the generalization gap by the average Rademacher complexity instead, we can replace the RHS of (4.89) with $2R_n(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2n}}$.

Interpretation of Corollary 4.21. It is typically the case that $O\left(\sqrt{\frac{\log(2/\delta)}{n}}\right) \ll R_S(\mathcal{F})$ and $O\left(\sqrt{\frac{\log(2/\delta)}{n}}\right) \ll R_n(\mathcal{F})$. This is the case because $R_S(\mathcal{F})$ and $R_n(\mathcal{F})$ often take the form $\frac{c}{\sqrt{n}}$ where c is a big constant depending on the complexity of $\mathcal{F}$, whereas we only have a logarithmic term in the numerator of $O\left(\sqrt{\frac{\log(2/\delta)}{n}}\right)$. As a result, we can view the $3\sqrt{\frac{\log(2/\delta)}{n}}$ term in the RHS of Corollary 4.21 as negligible. Another way of seeing this is noting that a $\widetilde{O}\left(\frac{1}{\sqrt{n}}\right)$ term is necessary even for the concentration bound of a single function $h \in \mathcal{H}$. Previously, we bounded $L(h) - \hat{L}(h)$ using a union bound over $h \in \mathcal{H}$, which necessarily needs to be larger than $\widetilde{O}\left(\frac{1}{\sqrt{n}}\right)$. As a result, the $O\left(\sqrt{\frac{\log(2/\delta)}{n}}\right)$ term is not significant.

4.5.1 Empirical Rademacher complexity viewed in the output/function space

Assume we have a fixed dataset $S = \{z_1, \ldots, z_n\}$. Since $z_1, \ldots, z_n$ is fixed, each function $f \in \mathcal{F}$ corresponds to a single output $(f(z_1), \ldots, f(z_n)) \in \mathbb{R}^n$. Hence, we can express the set of outputs for every function $f \in \mathcal{F}$ as

$$Q_{\mathcal{F}} = \left\{ \big(f(z_1), \ldots, f(z_n)\big) \mid f \in \mathcal{F} \right\}. \tag{4.90}$$

Now we can mathematically re-express the empirical Rademacher complexity as an inner product:

$$R_S(\mathcal{F}) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right] \tag{4.91}$$

$$= \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{v\in Q} \frac{1}{n}\langle \sigma, v\rangle\right], \tag{4.92}$$

where $\sigma = (\sigma_1, \ldots, \sigma_n)$. (See Figure 4.4 for an illustration of this idea.) This perspective will be helpful later on when proving bounds on the empirical Rademacher complexity.

Another corollary of this is that the empirical Rademacher complexity doesn't depend on the exact parameterization of $\mathcal{F}$. For example, suppose we have two parameterizations $\mathcal{F} = \left\{f(x) = \sum \theta_i x_i \mid \theta \in \mathbb{R}^d\right\}$ and $\mathcal{F}' = \left\{f(x) = \sum \theta_i^3 \cdot w_i x_i \mid \theta \in \mathbb{R}^d, w \in \mathbb{R}^d\right\}$. Since $Q_{\mathcal{F}}$ and $Q_{\mathcal{F}'}$ are the same, we see that $R_S(\mathcal{F}) = R_S(\mathcal{F}')$ since our earlier expression for $R_S(\mathcal{F})$ only depends on $\mathcal{F}$ through $Q_{\mathcal{F}}$.

4.5.2 Rademacher complexity is translation invariant

A useful fact is that both empirical Rademacher complexity and average Rademacher complexity are translation invariant. (This is not obvious when thinking of how translation affects the picture in Figure 4.4.)

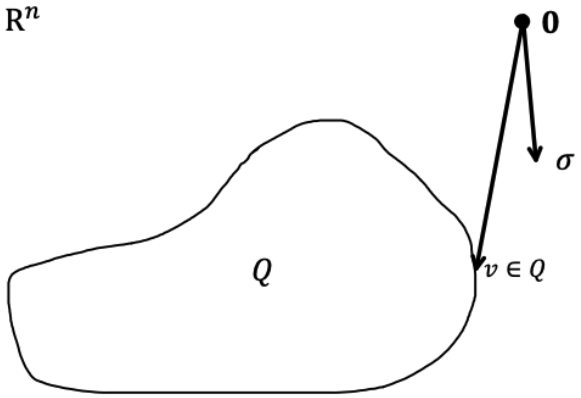

Figure 4.4: We can view empirical Rademacher complexity as the expectation of the maximum inner product between σ and $v \in Q$.

Proposition 4.5.1. Let $\mathcal{F}$ be a family of functions mapping $Z \mapsto \mathbb{R}$ and define $\mathcal{F}' = \{f'(z) = f(z) + c_0 \mid f \in \mathcal{F}\}$ for some $c_0 \in \mathbb{R}$. Then $R_S(\mathcal{F}) = R_S(\mathcal{F}')$ and $R_n(\mathcal{F}) = R_n(\mathcal{F}')$.

Proof. We will prove here that empirical Rademacher complexity is translation invariant.

$$R_S(\mathcal{F}') = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f' \in \mathcal{F}'} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right] \tag{4.93}$$

$$= \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i (f(z_i) + c_0)\right] \tag{4.94}$$

$$= \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i c_0 + \sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right] \tag{4.95}$$

$$= \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right] = R_S(\mathcal{F}), \tag{4.96}$$

where (4.96) follows because $\mathbb{E}_{\sigma_1,\ldots,\sigma_n} \frac{1}{n}\sum_{i=1}^{n} \sigma_i c_0 = 0$, since the σ_i's are Rademacher random variables. □

4.6 Covering number upperbounds Rademacher complexity

In Chapter 5, we will prove Rademacher complexity bounds that hinge on elegant, ad-hoc algebraic manipulations that may not extend to more general settings. Here, we consider a more fundamental approach for proving empirical Rademacher complexity bounds based on coverings of the output space. The trade-off is generally more tedium.

The first important observation is that for purposes of computing the **empirical** Rademacher complexity on samples $z_1, ..., z_n$,

$$R_S(\mathcal{F}) = \mathbb{E}_{\sigma}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(z_i)\right], \tag{4.97}$$

we only care about each function's $f \in \mathcal{F}$ behavior on $\{z_1, ..., z_n\}$. Hence, we can forget the rest of the input space and characterize $f \in \mathcal{F}$ by its outputs $(f(z_1), \ldots, f(z_n))$. Thus, there is a paradigm shift from the

space of all functions $\mathcal{F}$ to the *output space*

$$\mathcal{Q} \triangleq \left\{ \big(f(z_1), \dots, f(z_n)\big)^\top : f \in \mathcal{F} \right\} \subseteq \mathbb{R}^n, \tag{4.98}$$

which may be drastically smaller than $\mathcal{F}$. Correspondingly, the empirical Rademacher complexity can be rewritten as a maximization over the output space $\mathcal{Q}$ instead of the function space $\mathcal{F}$:

$$R_S(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{v \in \mathcal{Q}} \frac{1}{n} \langle \sigma, v \rangle \right]. \tag{4.99}$$

Now, for finite $|\mathcal{Q}|$, we immediately obtain the following bound by Massart's lemma:

$$R_S(\mathcal{F}) \le \sqrt{\frac{2 \log |\mathcal{Q}|}{n}}. \tag{4.100}$$

When $|\mathcal{Q}|$ is infinite, we can use the same discretization trick that we used to prove the generalization bound for an infinite-hypothesis space. Instead of trying to cover the parameter space, we try to cover the output space. To this end, we firstly recall a few definitions concerning ϵ-covers.

Definition 4.23. $\mathcal{C}$ is an *ϵ-cover* of $\mathcal{Q}$ with respect to metric ρ if for all $v \in \mathcal{Q}$, there exists $v' \in \mathcal{C}$ such that $\rho(v, v') \le \epsilon$.

Definition 4.24. The *covering number* is defined as the minimum size of an ϵ-cover, or explicitly:

$$N(\epsilon, \mathcal{Q}, \rho) \triangleq \text{(min size of } \epsilon\text{-cover of } \mathcal{Q} \text{ w.r.t. metric } \rho).$$

The standard metric we will use is $\rho(v, v') = \frac{1}{\sqrt{n}} \|v - v'\|_2$, with the leading coefficient inserted for convenience.

Remark 4.25. While we want to consider ϵ-covers over $\mathcal{Q}$, the notation in the literature refers to them as ϵ-covers of the function class $\mathcal{F}$ using the metric $\rho = L_2(p_n)$, i.e.

$$\rho(f, f') = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f(z_i) - f'(z_i))^2} \tag{4.101}$$

If we take the corresponding $v, v' \in \mathcal{Q}$, this is precisely $\rho(v, v') = \frac{1}{\sqrt{n}} \|v - v'\|_2$.

Equipped with the notion of ϵ-covers, we can prove the following Rademacher complexity bound:

Theorem 4.26. *Let $\mathcal{F}$ be a family of functions $Z \mapsto [-1, 1]$. Then*

$$R_S(\mathcal{F}) \le \inf_{\epsilon > 0} \left(\epsilon + \sqrt{\frac{2 \log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} \right). \tag{4.102}$$

The ϵ term can be thought of as the discretization error, while the latter term is the term from Massart's lemma.

Proof. Fix any $\epsilon > 0$. Let $\mathcal{C}$ be an ϵ-cover $\mathcal{C}$ of $\mathcal{Q}$. Massart's lemma immediately gives the bound

$$R_S(\mathcal{C}) \le \sqrt{\frac{2 \log |\mathcal{C}|}{n}}. \tag{4.103}$$

For every point $v \in \mathcal{Q}$, we can express it as $v = v' + z$, where $v' \in \mathcal{C}$ and z is small (specifically, $\frac{1}{\sqrt{n}} \|z\|_2 \le \epsilon$). This gives

$$\begin{aligned}
\frac{1}{n} \langle v, \sigma \rangle &= \frac{1}{n} \langle v', \sigma \rangle + \frac{1}{n} \langle z, \sigma \rangle && (4.104) \\
&\le \frac{1}{n} \langle v', \sigma \rangle + \frac{1}{n} \|z\|_2 \|\sigma\|_2 && \text{(Cauchy-Schwarz)} \quad (4.105) \\
&\le \frac{1}{n} \langle v', \sigma \rangle + \epsilon. && \text{(since } \|z\|_2 \le \sqrt{n}\epsilon \text{ and } \|\sigma\|_2 \le \sqrt{n}) \quad (4.106)
\end{aligned}$$

Taking the expectation of the supremum on both sides of this inequality gives

$$R_S(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{v \in \mathcal{Q}} \frac{1}{n} \langle v, \sigma \rangle \right] \tag{4.107}$$

$$\leq \mathbb{E}_\sigma \left[\sup_{v' \in \mathcal{C}} \left(\frac{1}{n} \langle v', \sigma \rangle + \epsilon \right) \right] \tag{4.108}$$

$$\leq \sqrt{\frac{2 \log |\mathcal{C}|}{n}} + \epsilon. \qquad \text{(Massart's lemma)} \tag{4.109}$$

Choosing $\mathcal{C}$ to be a minimal ϵ-cover allows us to set $|\mathcal{C}| = N(\epsilon, \mathcal{F}, L_2(p_n))$. Since the argument above holds for any $\epsilon > 0$, we can take the infimum over all ϵ to arrive at Equation (4.102), completing the proof.

□

4.7 Chaining and Dudley's theorem

While Theorem 4.26 is useful, the bound in Equation (4.105) is rarely tight as z might not be perfectly correlated with σ. It is possible to obtain a stronger theorem by constructing a chained ϵ-covering scheme. Specifically, when we decompose $v = v' + z$, we can construct a finer-grained covering of the ball $B(v', \epsilon)$, and then we can decompose z into smaller components and so on (see Figure 4.5 for an illustration).

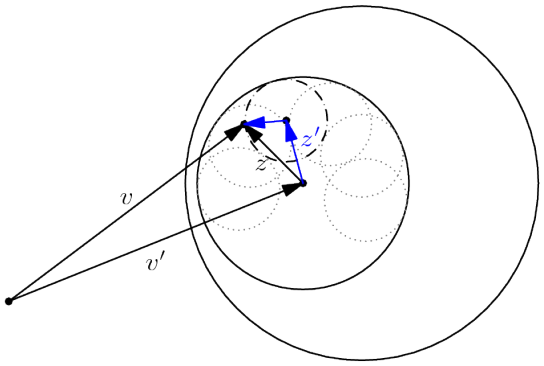

Figure 4.5: Illustration of a chained cover. Within the ϵ-ball containing the discretization error z, we find a finer ϵ'-cover and obtain a smaller error z' from discretizing z.

Using this method of chaining, we can obtain the following (stronger) result:

Theorem 4.27 (Dudley's Theorem)**.** *If $\mathcal{F}$ is a function class from Z to $\mathbb{R}$, then*

$$R_S(\mathcal{F}) \leq 12 \int_0^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{4.110}$$

We can interpret this bound as removing the discretization error term by averaging over different scales of ϵ. For a proof of this theorem, refer to Theorem 15 of [Liang, 2016].

If $\mathcal{F}$ consists of functions bounded in $[-1, 1]$, then, we have that for all $\epsilon > 1, N(\epsilon, \mathcal{F}, L_2(P_n)) = 1$. (To see this choose $\{f \equiv 0\}$, which is a complete cover for $\epsilon > 1$.) Hence, the limits of integration in (4.110) can be truncated to $[0, 1]$:

$$R_S(\mathcal{F}) \leq 12 \int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon, \tag{4.111}$$

since $\log N(\epsilon, \mathcal{F}, L_2(P_n)) = 0$ for $\epsilon > 1$.

4.7.1 Covering number regimes for which Dudley's theorem is finite

Of course, the bound in (4.110) is only useful if the integral on the RHS is finite. Here are some setups where this is the case (we continue to assume that the functions in $\mathcal{F}$ are bounded in $[-1, 1]$):

1. If $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx (1/\epsilon)^R$ (ignoring multiplicative and additive constants), then we have $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx R \log(1/\epsilon)$. We can plug this into the RHS of (4.110) to get

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon = \int_0^1 \sqrt{\frac{R \log(1/\epsilon)}{n}} d\epsilon \approx \sqrt{\frac{R}{n}}. \tag{4.112}$$

2. If the covering number has the form $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx a^{R/\epsilon}$ for some a, then we have $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx \frac{R}{\epsilon} \log a$. The bound in (4.110) becomes

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon \approx \int_0^1 \sqrt{\frac{R}{n\epsilon} \log a}\, d\epsilon \tag{4.113}$$

$$= \sqrt{\frac{R}{n} \log a} \int_0^1 \sqrt{\frac{1}{\epsilon}} d\epsilon \tag{4.114}$$

$$= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). \tag{4.115}$$

3. If the covering number has the form $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx a^{R/\epsilon^2}$, then $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx \frac{R}{\epsilon^2} \log a$. In this case we have:

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon \approx \sqrt{\frac{R}{n} \log a} \underbrace{\int_0^1 \frac{1}{\epsilon} d\epsilon}_{=\infty} = \infty, \tag{4.116}$$

i.e. the bound in (4.110) is vacuous. This is because of the behavior of $\epsilon \mapsto 1/\epsilon^2$ near 0: the function goes to infinity too quickly for us to upper bound its integral. Fortunately, there is an "improved" version of Dudley's theorem that is applicable here:

Theorem 4.28 (Improved Dudley's Theorem)**.** *If $\mathcal{F}$ is a function class from Z to $\mathbb{R}$, then for any fixed cutoff $\alpha \geq 0$ we have the bound*

$$R_S(\mathcal{F}) \leq 4\alpha + 12 \int_\alpha^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{4.117}$$

The proof of this theorem is similar to the proof of the original Dudley's theorem, except that the iterative covering procedure is stopped at the threshold $\epsilon = \alpha$ at the cost of the extra 4α term above.

Theorem 4.28 allows us to avoid the problematic region around $\epsilon = 0$ in the integral in (4.110). If we let $\alpha = 1/poly(n)$, the bound in (4.117) becomes

$$R_S(\mathcal{F}) \leq \frac{1}{poly(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \int_\alpha^1 \frac{1}{\epsilon} d\epsilon \tag{4.118}$$

$$= \frac{1}{poly(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \log(1/\alpha) \tag{4.119}$$

$$= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). \tag{4.120}$$

In summary, we have that $R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{R}{n}}\right)$ for these three dependencies on ϵ: when $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx R\log(1/\epsilon)$, $\frac{R}{\epsilon}\log a$, or $\frac{R}{\epsilon^2}\log a$ for some a. Note that if the dependence on ϵ is $1/\epsilon^c$ for $c > 2$, then even the improved Dudley's theorem does not help us. This is because the $\log(1/\alpha)$ term above becomes $\alpha^{1-c/2}$, and when $\alpha = 1/poly(n)$, this term leads to a bad dependence on n.

4.7.2 Regimes where we can get covering number bounds

The previous remarks discuss how strong our bounds on covering number need to be in order to get a useful result. Here we mention some situations in which we know how to obtain these covering number bounds:

1. Covering number and corresponding Rademacher complexity bounds for linear models are well-known, but fairly technical (see [Zhang, 2002]).

2. Covering numbers interact nicely with composition by Lipschitz functions. If ϕ is a ρ-Lipschitz function, then the following bound holds:

$$N(\epsilon/\rho, \mathcal{F}, L_2(P_n)) \geq N(\epsilon, \phi \circ \mathcal{F}, L_2(P_n)). \tag{4.121}$$

   This result is the analog of Talagrand's lemma for covering numbers. The proof follows easily if one considers ϕ as a change of measure: informally, the Lipschitz condition on ϕ means that a distance of ϵ/ρ in the original space $\mathcal{F}$ can be increased to at most ϵ in the space $\phi \circ \mathcal{F}$.

3. Using these results we can obtain a bound on the Rademacher complexity of a dense neural network. Consider a deep network

$$f(x) = W_r\sigma(W_{r-1}\sigma(\cdots\sigma(W_1x)\ldots), \tag{4.122}$$

   where W_i are layer-wise weights and σ is an activation function which is 1-Lipschitz. For this setup we have the following Rademacher complexity bound:

$$R_S(\mathcal{F}) \leq \underbrace{\left(\prod_{i=1}^{r}\|W_i\|_{\text{op}}\right)}_{\text{relatively large}} \cdot \underbrace{\left(\sum_{i=1}^{r}\frac{\|W_i^T\|_{2,1}^{2/3}}{\|W_i\|_{\text{op}}^{2/3}}\right)^{3/2}}_{\text{relatively small}}. \tag{4.123}$$

   Here $\|W\|_{\text{op}}$ is the operator norm (or spectral norm) of W, and $\|W_i^T\|_{2,1}$ denotes the sum of the l_2 norms of the rows of W_i. The second term is relatively small as it is a sum of matrix norms, and so the bound is dominated by the first term, which is a product of matrix norms. This first term comes from composition of Lipschitz functions as in (4.121) above, since the Lipschitz constant of a linear operator is its spectral norm. The full details are presented in [Bartlett et al., 2017].

4.8 VC dimension and its limitations

We will focus on classification and will be working within the framework of supervised learning stated in Chapter 1. The labels belong to the output space $\mathcal{Y} = \{-1, 1\}$, each classifier is a function $h : \mathcal{X} \to \mathbb{R}$ for all $h \in \mathcal{H}$, and the prediction is the sign of the output, i.e. $\hat{y} = \text{sgn}(h(x))$. We will look at zero-one loss, i.e. $\ell_{0\text{-}1}((x, y), h) = \mathbb{1}(\text{sgn}(h(x)) \neq y)$. Note that we can re-express the loss function as

$$\ell_{0\text{-}1}((x, y), h) = \frac{1 - \text{sgn}(h(x))y}{2}. \tag{4.124}$$

The first approach is to reason directly about the Rademacher complexity of $\ell_{0\text{-}1}$ loss, i.e. considering the family of functions $\mathcal{F} = \{z = (x, y) \mapsto \ell_{0\text{-}1}((x, y), h) : h \in \mathcal{H}\}$. Define Q to be the set of all possible outputs

on our dataset: $Q = \left\{\left(\operatorname{sgn}\left(h\left(x^{(1)}\right)\right), \ldots, \operatorname{sgn}\left(h\left(x^{(n)}\right)\right)\right) \mid h \in \mathcal{H}\right\}$. Then, using our earlier remark about viewing the empirical Rademacher complexity as an inner product between $v \in Q$ and σ, we have

$$R_S(\mathcal{F}) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \frac{1-\operatorname{sgn}(h(x^{(i)}))y_i}{2}\right] \tag{4.125}$$

$$= \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \frac{\operatorname{sgn}(h(x^{(i)}))}{2}\right] \tag{4.126}$$

$$= \frac{1}{2}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{v\in Q} \frac{1}{n}\langle \sigma, v\rangle\right]. \tag{4.127}$$

Notice that the supremum is now over Q instead of $\mathcal{F}$. If n is sufficiently large, then it is typically the case that $|Q| > |\mathcal{F}|$. To see why this is the case, note that each function f corresponds to a single element in Q. However, as n increases, $|Q|$ increases as well. For any particular $v \in Q$, notice that $\langle v, \sigma\rangle$ is a sum of bounded random variables, so we can use Hoeffding's inequality to obtain

$$\Pr\left[\frac{1}{n}(\sigma, v) \geq t\right] \leq \exp(-nt^2/2). \tag{4.128}$$

Taking the union bound over $v \in Q$, we see that

$$\Pr\left[\exists v \in Q \text{ such that } \frac{1}{n}\langle \sigma, v\rangle \geq t\right] \leq |Q|\exp(-nt^2/2). \tag{4.129}$$

Thus, with probability at least $1-\delta$, it is true that $\sup_{v\in Q} \frac{1}{n}\langle v, \sigma\rangle \leq \sqrt{\frac{2(\log|Q|+\log(2/\delta))}{n}}$. Similarly, we can show that $\mathbb{E}\left[\sup_{v\in Q} \frac{1}{n}\langle v, \sigma\rangle\right] \leq O\left(\sqrt{\frac{\log|Q|+\log(2/\delta)}{n}}\right)$ holds.

The key point to notice here is that the upper bound on $R_S(\mathcal{F})$ depends on $\log|Q|$. *VC dimension* is one way that we deal with bounding the size of Q We will not delve into the details of this approach (for those interested, see Section 3.11 of [Liang, 2016]). VC dimension, however, has a number of limitations. For one, we will always end up with a bound that depends somehow on the dimension. For linear models, we obtain a bound $\log|Q| \lesssim d\log n$, corresponding to a bound on Rademacher complexity that looks like

$$R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{d}{n}}\right), \tag{4.130}$$

so we still have a $\sqrt{d}$ term. This will not be a good bound for high-dimensional models. For general models, we will arrive a bound of the form

$$R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{\#\text{ of parameters}}{n}}\right). \tag{4.131}$$

This upper bound only depends on the number of parameters in our model, and does not take into the account the scale and norm of the parameters. Additionally, this doesn't work with kernel methods since the explicit parameterization is possibly infinite-dimensional, and therefore this upper bound becomes useless.

These limitations motivation the use of margin theory, which does take into account the norm of parameters and provides a theoretical basis for regularization techniques such as L_1 and L_2 regularization.

Chapter 5

Rademacher Complexity Bounds for Concrete Models and Losses

In this chapter, we will instantiate Rademacher complexity for two important hypothesis classes: linear models and two-layer neural networks. In the process, we will develop margin theory and use it to bound the generalization gap for binary classifiers.

5.1 Margin theory for classification problems

5.1.1 Intuition

Assume that we are in the same setting as in the previous section. A fundamental problem we face in this setting is that we do not have a continuous loss: everything is discrete in the output space. We need to find a way to reason about the scale of the output. An example of this is logistic regression: the logistic regression model outputs a probability, and while we compare it to the outcome (0 or 1), how close it is to the true output gives us a measure of how confident we are in the prediction.

Figure 5.1 gives similar intuition for linear classifiers. Intuitively, the black line is a "better" decision boundary than the red line because the minimum distance from any point to the black boundary is greater than the minimum distance from any point to the red line. In the next section, we will formalize this intuition by proving that the larger this margin is, the smaller the bound on generalization gap is.

5.1.2 Formalizing margin theory

First, assume that the dataset $\mathcal{D} = ((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}))$ is *completely separable*. In other words, there exists some $h_\theta \in \mathcal{H}$ such that $y^{(i)} = \text{sgn}(h_\theta(x^{(i)}))$ holds for all $(x^{(i)}, y^{(i)}) \in \mathcal{D}$. This is not a necessary condition for our final bound but will make the derivation cleaner.

Definition 5.1 ((Unnormalized) Margin)**.** Fix the hypothesis h_θ. The *(unnormalized) margin* for example (x, y) is defined as $\text{margin}(x) = yh_\theta(x)$. Margin is only defined on examples where $\text{sgn}(h_\theta(x)) = y$. (Note that $\text{margin}(x) \geq 0$ because of our assumption of complete separability.)

Definition 5.2 (Minimum margin)**.** Given a dataset $\mathcal{D} = ((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}))$, the *minimum margin* over the dataset is defined as $\gamma_{\min} \triangleq \min_{i \in \{1,\ldots,|\mathcal{D}|\}} y^{(i)} h_\theta(x^{(i)})$.

Our final bound will have the form (generalization gap)$\leq f(\text{margin}, \text{parameter norm})$. This is very generic since there are many different bounds we could derive based on what margin we use. For this current setting we are using $\gamma_{\min}$, which is the minimum margin, but in other settings could use γ_{average}, which is the average margin of each point in the dataset.

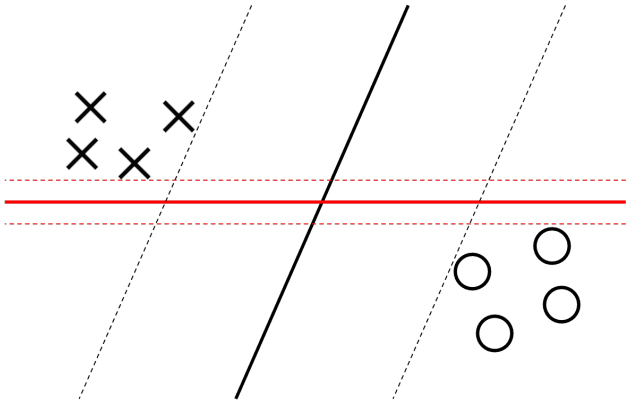

Figure 5.1: The red and black lines are two decision boundaries. The X's are positive examples and the O's are negative examples. The black line has a larger margin than the red line, and is intuitively a better classifier.

We will begin by introducing the idea of a *surrogate loss*, a loss function which approximates zero-one loss but takes the scale of the margin into account. The *margin loss* (also known as *ramp loss*) is defined as

$$\ell_\gamma(t) = \begin{cases} 0 & t \geq \gamma \\ 1 & t \leq 0 \\ 1 - t/\gamma & 0 \leq t \leq \gamma \end{cases} \tag{5.1}$$

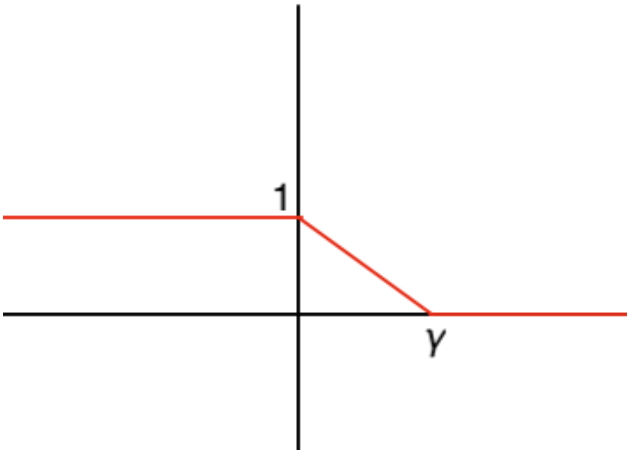

Figure 5.2: Plotted margin loss.

It is plotted in Figure 5.2. For convenience, define $\ell_\gamma((x,y),h) \triangleq \ell_\gamma(yh(x))$. We can view ℓ_γ as a continuous version of $\ell_{0\text{-}1}$ while being more sensitive to the scale of the margin on $[0,\gamma]$. Notice that $\ell_{0\text{-}1}$ is always less than or equal to the ℓ_γ when $\gamma \geq 0$, i.e.

$$\ell_{0\text{-}1}((x,y),h) \leq \ell_\gamma((x,y),h) \tag{5.2}$$

holds for all $(x,y) \sim P$. Taking the expectation over (x,y) on both sides of this inequality, we see that

$$L(h) = \mathbb{E}_{(x,y)\sim P}\left[\ell_{0\text{-}1}((x,y),h)\right] \leq \mathbb{E}_{(x,y)\sim P}\left[\ell_\gamma((x,y),h)\right]. \tag{5.3}$$

Therefore, the population loss is bounded by the expectation of the margin loss, and so it is sufficient to bound the expectation of the margin loss in order to bound the population loss.

Define the population and empirical version of the margin loss:

$$L_\gamma(h) = \mathbb{E}_{(x,y)\sim P}\left[\ell_\gamma((x,y),h)\right], \quad \hat{L}_\gamma(h) = \sum_{i=1}^{n}\left[\ell_\gamma((x^{(i)},y^{(i)}),h)\right]. \tag{5.4}$$

By Corollary 4.21, we see that with probability at least $1-\delta$ that

$$L_\gamma(h) - \hat{L}_\gamma(h) \le 2R_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}, \tag{5.5}$$

where $\mathcal{F} = \{(x,y) \mapsto \ell_\gamma((x,y),h) \mid h \in \mathcal{H}\}$. Note that if we set $\gamma \le \gamma_{\min}$, then $\hat{L}_\gamma(h) = 0$. This follows because by definition of $\gamma_{\min}$, $y^{(i)}h(x^{(i)}) \ge \gamma_{\min}$ for any $(x^{(i)},y^{(i)}) \in \mathcal{D}$. As a result, $\ell_\gamma((x^{(i)},y^{(i)}),h) = \ell_\gamma(y^{(i)}h(x^{(i)})) = 0$ holds. Therefore, it suffices to bound $R_S(\mathcal{F})$.

We will now use *Talagrand's lemma* to bound $R_S(\mathcal{F})$ in terms of $R_S(\mathcal{H})$ to remove any dependence on the loss function from the upper bound.

Lemma 5.3. *(Talagrand's lemma) Let $\phi : \mathbb{R} \to \mathbb{R}$ be a κ-Lipschitz function. Then*

$$R_S(\phi \circ \mathcal{H}) \le \kappa R_S(\mathcal{H}), \tag{5.6}$$

where $\phi \circ \mathcal{H} = \{z \mapsto \phi(h(z)) \mid h \in \mathcal{H}\}$.

We can use Talagrand's lemma directly with $\phi(t) = \ell_\gamma(t)$, which is $\frac{1}{\gamma}$-Lipschitz. We can express $\mathcal{F}$ as $\mathcal{F} = \ell_\gamma \circ \mathcal{H}'$ where $\mathcal{H}' = \{(x,y) \to yh(x) \mid h \in \mathcal{H}\}$. Applying Talagrand's lemma, we see that

$$\begin{aligned}
R_S(\mathcal{F}) &\le \frac{1}{\gamma}R_S(\mathcal{H}') && (5.7)\\
&= \frac{1}{\gamma}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i y^{(i)}h(x^{(i)})\right] && (5.8)\\
&= \frac{1}{\gamma}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(x^{(i)})\right] && (5.9)\\
&= \frac{1}{\gamma}R_S(\mathcal{H}). && (5.10)
\end{aligned}$$

Putting this all together, we have shown that for $\gamma \le \gamma_{\min}$,

$$\begin{aligned}
L_{0\text{-}1}(h) \le L_\gamma(h) &\le 0 + O\left(\frac{R_S(\mathcal{H})}{\gamma}\right) + \widetilde{O}\left(\sqrt{\frac{\log(2/\delta)}{2n}}\right) && (5.11)\\
&= O\left(\frac{R_S(\mathcal{H})}{\min_i y^{(i)}h(x^{(i)})}\right) + \widetilde{O}\left(\sqrt{\frac{\log(2/\delta)}{2n}}\right). && (5.12)
\end{aligned}$$

In other words, for training data of the form $S = \{(x^{(i)},y^{(i)})\}_{i=1}^n \subset \mathbb{R}^d \times \{-1,1\}$, a hypothesis class $\mathcal{H}$ and 0-1 loss, we can derive a bound of the form

$$\text{generalization loss} \le \frac{2R_S(\mathcal{H})}{\gamma_{\min}} + \text{low-order term}, \tag{5.13}$$

where $\gamma_{\min}$ is the minimum margin achievable on S over those hypotheses in $\mathcal{H}$ that separate the data, and $R_S(\mathcal{H})$ is the empirical Rademacher complexity of $\mathcal{H}$. Such bounds state that simpler models will generalize better beyond the training data, particularly for data that is strongly separable.

Remark 5.4. Note there is a subtlety here. If we think of the dataset as random, it follows that $\gamma_{\min}$ is a random variable. Consequently, the γ we choose to define the hypothesis class is random, which is not a valid choice when thinking about Rademacher complexity! Technically we cannot apply Talagrand's lemma with a random κ (which we took to be $1/\gamma$). Also, when we used concentration inequalities, we implicitly assume that the $\ell_\gamma((x^{(i)}, y^{(i)}), h)$ are independent of each other. That is not the case if γ is dependent on the data.

How can we address this? The idea is to do another union bound over γ. Choose a family $\Gamma = \left\{2^k : k \in [-B, B]\right\}$ for some B. For every fixed $\gamma \in \Gamma$, we prove the theorem that

$$L_{0\text{-}1}(h) \leq \widehat{L}_\gamma(h) + O\left(\frac{R_S(\mathcal{H})}{\gamma}\right) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{5.14}$$

We can then take a union bound over all $\gamma \in \Gamma$. Next, choose the largest $\gamma \in \Gamma$ such that $\gamma \leq \gamma_{\min}$. For this value of γ we have $\widehat{L}_\gamma(h) = 0$ and $O\left(\frac{R_S(\mathcal{H})}{\gamma}\right) = O\left(\frac{R_S(\mathcal{H})}{\gamma_{\min}}\right)$.

5.2 Linear models

5.2.1 Linear models with weights bounded in ℓ_2 norm

We begin with the Rademacher complexity of linear models using weights with bounded ℓ_2 norm.

Theorem 5.5. *Let $\mathcal{H} = \{x \mapsto \langle w, x\rangle \mid w \in \mathbb{R}^d, \|w\|_2 \leq B\}$ for some constant $B > 0$. Moreover, assume $\mathbb{E}_{x\sim P}\left[\|x\|_2^2\right] \leq C^2$, where P is some distribution and $C > 0$ is a constant. Then*

$$R_S(\mathcal{H}) \leq \frac{B}{n}\sqrt{\sum_{i=1}^{n} \left\|x^{(i)}\right\|_2^2}, \tag{5.15}$$

and

$$R_n(\mathcal{H}) \leq \frac{BC}{\sqrt{n}}. \tag{5.16}$$

Generally speaking, there are two methods with which we can bound the Rademacher complexity of a model. The first method, which we used in Chapter 4, consists of discretizing the space of possible outputs from our hypothesis class, then using a union bound or covering number argument to bound the Rademacher complexity of the model. While this method is powerful and generally applicable, it yields bounds that depend on the logarithm of the cardinality of this discretized output space, which in turn depends on the number of data points n. In the proof below, we will instead use a more elegant, albeit limited technique which does not rely on discretization of the output space.

Proof. We start with the proof of (5.15). By definition,

$$R_S(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{\|w\|_2 \le B} \frac{1}{n} \sum_{i=1}^n \sigma_i \left\langle w, x^{(i)} \right\rangle \right] \tag{5.17}$$

$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\|w\|_2 \le B} \left\langle w, \sum_{i=1}^n \sigma_i x^{(i)} \right\rangle \right] \tag{5.18}$$

$$= \frac{B}{n} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^n \sigma_i x^{(i)} \right\|_2 \right] \qquad (\sup_{\|w\|_2 \le B} \langle w, v \rangle = B \|v\|_2) \tag{5.19}$$

$$\le \frac{B}{n} \sqrt{\mathbb{E}_\sigma \left[\left\| \sum_{i=1}^n \sigma_i x^{(i)} \right\|_2^2 \right]} \qquad (\text{Jensen's ineq. for } \alpha \mapsto \alpha^2) \tag{5.20}$$

$$= \frac{B}{n} \sqrt{\mathbb{E}_\sigma \left[\sum_{i=1}^n \left(\sigma_i^2 \left\| x^{(i)} \right\|_2^2 + \left\langle \sigma_i x^{(i)}, \sum_{j \ne i}^n \sigma_j x^{(j)} \right\rangle \right) \right]} \tag{5.21}$$

$$= \frac{B}{n} \sqrt{\sum_{i=1}^n \left\| x^{(i)} \right\|_2^2}. \qquad (\sigma_i \text{ indep. and } \mathbb{E}[\sigma_i] = 0) \tag{5.22}$$

This completes the proof of (5.15) for the empirical Rademacher complexity. The bound on the average Rademacher complexity in (5.16) follows from taking the expectation of both sides to get

$$R_n(\mathcal{H}) = \mathbb{E}\left[R_S(\mathcal{H})\right] = \frac{B}{n} \mathbb{E}\left[\sqrt{\sum_{i=1}^n \left\| x^{(i)} \right\|_2^2} \right] \le \frac{B}{n} \sqrt{\sum_{i=1}^n \mathbb{E}\left[\left\| x^{(i)} \right\|_2^2 \right]} \le \frac{BC}{\sqrt{n}}, \tag{5.23}$$

where the first inequality is another application of Jensen's inequality, and the second follows from the assumption $\mathbb{E}_{x \sim P}\left[\|x\|_2^2\right] \le C^2$.

□

We observe that both the empirical and average Rademacher complexities scale with the upper ℓ_2-norm bound $\|w\|_2 \le B$ on the parameters w, which motivates regularizing the model. However, smaller weights in the model may reduce the margin $\gamma_{\min}$, which in turn hurts generalization according to (5.13).

Remark 5.6. Note that if we scale the data by some multiplicative factor, the bound on empirical Rademacher complexity $R_S(\mathcal{H})$ will scale accordingly. However, at the same time we expect the margin to scale by the same multiplicative factor, so the bound on the generalization gap in (5.13) does not change. This lines up with our intuition that the bound should not depend on the scaling of the data.

5.2.2 Linear models with weights bounded in ℓ_1 norm

Now, we consider linear models again, except we restrict the ℓ_1-norm of the parameters and assume an ℓ_∞-norm bound on the data.

Theorem 5.7. *Let* $\mathcal{H} = \left\{ x \mapsto \langle w, x \rangle \mid w \in \mathbb{R}^d, \|w\|_1 \le B \right\}$ *for some constant* $B > 0$. *Moreover, assume* $\left\|x^{(i)}\right\|_\infty \le C$ *for some constant* $C > 0$ *and all points in* $S = \{x^{(i)}\}_{i=1}^n \subset \mathbb{R}^d$. *Then*

$$R_S(\mathcal{H}) \le BC \sqrt{\frac{2 \log(2d)}{n}}. \tag{5.24}$$

To prove the theorem, we will need Massart's lemma, which provides a bound for the Rademacher complexity of a finite hypothesis class.

Lemma 5.8 (Massart's lemma)**.** *Suppose $\mathcal{Q} \subset \mathbb{R}^n$ is finite and contained in the ℓ_2-norm ball of radius $M\sqrt{n}$ for some constant $M > 0$, i.e.,*

$$\mathcal{Q} \subset \{v \in \mathbb{R}^n \mid \|v\|_2 \leq M\sqrt{n}\}. \tag{5.25}$$

Then, for Rademacher variables $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathbb{R}^n$,

$$\mathbb{E}_\sigma \left[\sup_{v \in \mathcal{Q}} \frac{1}{n} \langle \sigma, v \rangle\right] \leq M\sqrt{\frac{2 \log |\mathcal{Q}|}{n}}. \tag{5.26}$$

As a corollary, if $\mathcal{F}$ is a set of real-valued functions satisfying

$$\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n f(z^{(i)})^2 \leq M^2, \tag{5.27}$$

over some data $S = \{z^{(i)}\}_{i=1}^n$, then

$$R_S(\mathcal{F}) \leq M\sqrt{\frac{2 \log |\mathcal{F}|}{n}}, \quad \textit{and} \quad R_n(\mathcal{F}) \leq M\sqrt{\frac{2 \log |\mathcal{F}|}{n}}. \tag{5.28}$$

We will not prove Massart's lemma in detail. The intuition is to use concentration inequalities to bound $\frac{1}{n} \langle \sigma, v \rangle$ for fixed v, then to use a union bound over the elements $v \in \mathcal{Q}$.

We will now prove Theorem 5.7:

Proof of Theorem 5.7. By definition,

$$R_S(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{\|w\|_1 \leq B} \frac{1}{n} \sum_{i=1}^n \sigma_i \left\langle w, x^{(i)} \right\rangle\right] \tag{5.29}$$

$$= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\|w\|_1 \leq B} \left\langle w, \sum_{i=1}^n \sigma_i x^{(i)} \right\rangle\right] \tag{5.30}$$

$$= \frac{B}{n} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^n \sigma_i x^{(i)} \right\|_\infty\right], \tag{5.31}$$

where the last equality is because $\sup_{\|w\|_1 \leq B} \langle w, v \rangle = B \|v\|_\infty$, i.e., the ℓ_∞-norm is the dual of the ℓ_1-norm, which is a consequence of Hölder's inequality. However, the ℓ_∞-norm is difficult to simplify further. Instead, we use the fact that $\sup_{\|w\|_1 \leq 1} \langle w, v \rangle$ for any $v \in \mathbb{R}^d$ is always attained at one of the vertices $\mathcal{W} = \bigcup_{i=1}^d \{-e_i, e_i\}$, where $e_i \in \mathbb{R}^d$ is the i-th coordinate unit vector. Defining the restricted hypothesis class $\bar{\mathcal{H}} = \{x \mapsto \langle w, x \rangle \mid w \in \mathcal{W}\} \subset \mathcal{H}$, this yields

$$R_S(\mathcal{H}) = \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{\|w\|_1 \leq B} \left\langle w, \sum_{i=1}^n \sigma_i x^{(i)} \right\rangle\right] \tag{5.32}$$

$$= \frac{B}{n} \mathbb{E}_\sigma \left[\max_{w \in \mathcal{W}} \left\langle w, \sum_{i=1}^n \sigma_i x^{(i)} \right\rangle\right] \tag{5.33}$$

$$= B R_S(\bar{\mathcal{H}}). \tag{5.34}$$

Since $\bar{\mathcal{H}} \subset \mathcal{H}$, necessarily $R_S(\bar{\mathcal{H}}) \leq R_S(\mathcal{H})$. In particular, the model class $\bar{\mathcal{H}}$ is bounded and finite with cardinality $|\bar{\mathcal{H}}| = 2d$. This suggests using Massart's lemma to complete the proof. To do so, we need to confirm that $\bar{\mathcal{H}}$ is bounded with respect to the ℓ_2-metric. Indeed, since the inner product of $x^{(i)}$ with a coordinate vector e_j just selects the j-th coordinate of $x^{(i)}$, for any $w \in \mathcal{W}$ we have

$$\frac{1}{n} \sum_{i=1}^n \left\langle w, x^{(i)} \right\rangle^2 \leq \frac{1}{n} \sum_{i=1}^n \left\| x^{(i)} \right\|_\infty^2 \leq \frac{1}{n} \sum_{i=1}^n C^2 = C^2, \tag{5.35}$$

where the last inequality uses the assumption $\|x_i\|_\infty \le C$. So $\bar{\mathcal{H}}$ is bounded in the ℓ_2-metric and finite, thus by Massart's Lemma we have

$$R_S(\mathcal{H}) = BR_S(\bar{\mathcal{H}}) \le BC\sqrt{\frac{2\log|\bar{\mathcal{H}}|}{n}} = BC\sqrt{\frac{2\log(2d)}{n}}, \tag{5.36}$$

which completes the proof. □

5.2.3 Comparing the bounds for different $\mathcal{H}$

First, we note that for this hypothesis class of linear models, it is possible to obtain an upper bound proportional to $\sqrt{d/n}$ using the VC dimension, which grows quickly with the data dimension d. Our bound is better since it does not have as strong of a dependence on d, and accounts for the norms of our model parameters and the data.

In the two subsections above, we considered two different hypothesis classes of linear models, each restricting different norms. In both cases, the bound on the average Rademacher complexity depended on the product of the norm bound on the parameters w and the norm bound on each data point x. To determine which choice of hypothesis class is better, consider the bounds

$$\|w\|_2 \|x\|_2 \quad \text{vs.} \quad \|w\|_1 \|x\|_\infty$$

and see how they compare in different settings. We consider 3 settings here:

- Suppose w and x are random variables with w_i and x_i close to the set of values $\{-1, 1\}$. Then we have

  $$\sqrt{d} \cdot \sqrt{d} \quad \text{vs.} \quad d \cdot 1.$$

  In this case, there is no difference in using either linear hypothesis class.

- If we additionally suppose w is sparse with at most k non-zero entries, then we have

  $$\sqrt{k} \cdot \sqrt{d} \quad \text{vs.} \quad k \cdot 1.$$

  So for $d \gg k$, we have $\sqrt{kd} \gg k$ and thus ℓ_1-norm regularization leads to a better complexity bound when w is suspected to be sparse. Indeed, $\sqrt{d}\|x\|_\infty \approx \|x\|_2$ when the entries of x are somewhat uniformly distributed, and so in the sparse case we have

  $$\|w\|_2 \|x\|_2 \ge \sqrt{d}\|w\|_2 \|x\|_\infty \ge \|w\|_1 \|x\|_\infty . \tag{5.37}$$

- On the other hand, if w is dense in the sense that $\|w\|_2 \approx \sqrt{d}\|w\|_1$ (i.e., if all entries in w are close to each other in magnitude), then

  $$\|w\|_2 \|x\|_2 \le \frac{1}{\sqrt{d}} \|w\|_1 \cdot \sqrt{d}\|x\|_\infty \le \|w\|_1 \|x\|_\infty . \tag{5.38}$$

  In this case, it makes sense to regularize the ℓ_2-norm instead.

In practice, other multiplicative factors enter the generalization bound, so regularizing both the ℓ_1- and ℓ_2-norms of the model parameters w is preferable.

Continuing with this rough style of analysis, for the hypothesis class with restricted ℓ_2-norm, we can write the bound on the generalization gap in (5.13) as

$$\text{generalization loss} \lesssim \frac{\|w\|_2 \|x\|_2}{\sqrt{n}\gamma_{\min}} + \text{low-order term}. \tag{5.39}$$

The presence of $\|w\|_2 / \gamma_{\min}$ motivates both the minimum norm and the maximum margin formulations of the Support Vector Machine (SVM) problem as good methods to improve generalization performance of binary classifiers.

5.3 Two-layer neural networks

We now compute a bound for the Rademacher complexity of two-layer neural networks. Throughout this section, we use the following notation:

- $\theta = (w, U)$ are the parameters of the model with $w \in \mathbb{R}^m$ and $U \in \mathbb{R}^{m \times d}$, where m denotes the number of hidden units. We use $u_i \in \mathbb{R}^d$ to denote the i-th row of U (written as a column vector).
- $\phi(z) = \max(z, 0)$ is the ReLU activation function applied element-wise.
- $f_\theta(x) = \langle w, \phi(Ux) \rangle = w^\top \phi(Ux)$ is the model.
- $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ is the training set, with $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$.

We start with a somewhat weak bound which introduces the technical tools we need to derive tighter bounds subsequently.

Theorem 5.9. *For some constants $B_w > 0$ and $B_u > 0$, let*

$$\mathcal{H} = \{f_\theta \mid \|w\|_2 \le B_w,\ \|u_i\|_2 \le B_u,\ \forall i \in \{1, 2, \dots, m\}\}, \tag{5.40}$$

and suppose $\mathbb{E}\left[\|x\|_2^2\right] \le C^2$. Then

$$R_n(\mathcal{H}) \le 2B_w B_u C \sqrt{\frac{m}{n}}. \tag{5.41}$$

This bound is not ideal as it depends on the number of neurons m. Empirically, it has been found that the generalization error does *not* increase monotonically with m. As more neurons are added to the model, thereby giving it more expressive power, studies have shown that generalization is improved [Belkin et al., 2019]. This contradicts the bound above, which states that more neurons leads to worse generalization. Nevertheless, we now derive this bound.

Proof. By definition,

$$\begin{aligned}
R_S(\mathcal{H}) &= \mathbb{E}_\sigma \left[\sup_\theta \frac{1}{n} \sum_{i=1}^n \sigma_i \left\langle w, \phi(Ux^{(i)}) \right\rangle \right] && (5.42)\\
&= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{U : \|u_j\|_2 \le B_u} \sup_{\|w\|_2 \le B_w} \left\langle w, \sum_{i=1}^n \sigma_i \phi(Ux^{(i)}) \right\rangle \right] && (5.43)\\
&= \frac{B_w}{n} \mathbb{E}_\sigma \left[\sup_{U : \|u_j\|_2 \le B_u} \left\| \sum_{i=1}^n \sigma_i \phi(Ux^{(i)}) \right\|_2 \right] && (\sup_{\|w\|_2 \le B} \langle w, v \rangle = B \|v\|_2) \quad (5.44)\\
&\le \frac{B_w \sqrt{m}}{n} \mathbb{E}_\sigma \left[\sup_{U : \|u_j\|_2 \le B_u} \left\| \sum_{i=1}^n \sigma_i \phi(Ux^{(i)}) \right\|_\infty \right] && (\|v\|_2 \le m \|v\|_\infty) \quad (5.45)\\
&= \frac{B_w \sqrt{m}}{n} \mathbb{E}_\sigma \left[\sup_{U : \|u_j\|_2 \le B_u} \max_{1 \le j \le m} \left| \sum_{i=1}^n \sigma_i \phi(u_j^\top x^{(i)}) \right| \right] && (5.46)\\
&= \frac{B_w \sqrt{m}}{n} \mathbb{E}_\sigma \left[\sup_{\|u\|_2 \le B_u} \left| \sum_{i=1}^n \sigma_i \phi(u^\top x^{(i)}) \right| \right] && (5.47)\\
&\le \frac{2B_w \sqrt{m}}{n} \mathbb{E}_\sigma \left[\sup_{\|u\|_2 \le B_u} \sum_{i=1}^n \sigma_i \phi(u^\top x^{(i)}) \right] && (\text{by Lemma 5.11}) \quad (5.48)\\
&\le \frac{2B_w \sqrt{m}}{n} \mathbb{E}_\sigma \left[\sup_{\|u\|_2 \le B_u} \sum_{i=1}^n \sigma_i u^\top x^{(i)} \right], && (5.49)
\end{aligned}$$

where the last inequality follows by applying the contraction lemma (Talagrand's lemma) and observing that the ReLU function is 1-Lipschitz. (Observe that the expectation in (5.48) is the Rademacher complexity for $\{x \mapsto \phi(u^\top x) \mid \|u\|_2 \leq B_u\}$: this is the family that we are applying the contraction lemma to.)

We now observe that the expectation in (5.49) is the Rademacher complexity of the family of linear models $\{x \mapsto \langle u, x\rangle \mid \|u\|_2 \leq B_u\}$. Thus, applying Theorem 5.7 yields

$$R_S(\mathcal{H}) \leq \frac{2B_w\sqrt{m}}{n} B_u \sqrt{\sum_{i=1}^{n} \left\|x^{(i)}\right\|_2^2}. \tag{5.50}$$

Taking the expectation of both sides and using similar steps to those in the proof of Theorem 5.7 gives us

$$\begin{aligned} R_n(\mathcal{H}) &= \mathbb{E}\left[R_S(\mathcal{H})\right] && (5.51)\\ &\leq \frac{2B_wB_u\sqrt{m}}{n} \mathbb{E}\left[\sqrt{\sum_{i=1}^{n} \left\|x^{(i)}\right\|_2^2}\right] && (5.52)\\ &\leq \frac{2B_wB_u\sqrt{m}}{n} C\sqrt{n} && (5.53)\\ &= 2B_wB_uC\sqrt{\frac{m}{n}}, && (5.54) \end{aligned}$$

which completes the proof.

□

This upper bound is undesirable since it grows with the number of neurons m, contradicting empirical observations of the generalization error decreasing with m.

5.4 Refined bounds for two-layer neural networks

Next, we look at a finer bound that results from defining a new complexity measure. A recurring theme in subsequent proofs will be the functional invariance of two-layer neural networks under a class of rescaling transformations. The key ingredient will be the *positive homogeneity* of the ReLU function, i.e.

$$\alpha\phi(x) = \phi(\alpha x) \qquad \forall \alpha > 0. \tag{5.55}$$

This implies that for any $\lambda_i > 0$ $(i = 1, \ldots, m)$, the transformation $\theta = \{(w_i, u_i)\}_{1\leq i\leq m} \mapsto \theta' = \{(\lambda_i w_i, u_i/\lambda_i)\}_{1\leq i\leq m}$ has no net effect on the neural network's functionality (i.e. $f_\theta = f_{\theta'}$) since

$$w_i \cdot \phi\left(u_i^\top x^{(i)}\right) = (\lambda_i w_i) \cdot \phi\left(\left(\frac{u_i}{\lambda_i}\right)^\top x^{(i)}\right). \tag{5.56}$$

In light of this, we devise a new complexity measure $C(\theta)$ that is also invariant under such transformations and use it to prove a better bound for the Rademacher complexity. This positive homogeneity property is absent in the (implicit) complexity measure used in the hypothesis class (5.40) of Theorem 5.9.

Theorem 5.10. Let $C(\theta) = \sum_{j=1}^{m} |w_j| \left\|u_j\right\|_2$, *and for some constant* $B_C > 0$ *consider the hypothesis class*

$$\mathcal{H} = \{f_\theta \mid C(\theta) \leq B_C\}. \tag{5.57}$$

If $\left\|x^{(i)}\right\|_2 \leq C$ *for all* $i \in \{1, \ldots, n\}$, *then*

$$R_S(\mathcal{H}) \leq \frac{2B_CC}{\sqrt{n}}. \tag{5.58}$$

Proof. Due to the positive homogeneity of the ReLU function ϕ, it will be useful to define the ℓ_2-normalized weight vector $\bar{u}_j := u_j/\|u_j\|_2$ so that $\phi\left(u_j^T x\right) = \|u_j\|_2 \cdot \phi(\bar{u}_j^T x)$. The empirical Rademacher complexity satisfies

$$
\begin{aligned}
R_S(\mathcal{H}) &= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_\theta \sum_{i=1}^n \sigma_i f_\theta\left(x^{(i)}\right)\right] && (5.59)\\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_\theta \sum_{i=1}^n \sigma_i \left[\sum_{j=1}^m w_j \phi\left(u_j^T x^{(i)}\right)\right]\right] && \text{(by dfn of } f_\theta) \quad (5.60)\\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_\theta \sum_{i=1}^n \sigma_i \left[\sum_{j=1}^m w_j \|u_j\|_2 \phi\left(\bar{u}_j^T x^{(i)}\right)\right]\right] && \text{(by positive homogeneity of } \phi) \quad (5.61)\\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\sup_\theta \sum_{j=1}^m w_j \|u_j\|_2 \left[\sum_{i=1}^n \sigma_i \phi\left(\bar{u}_j^T x^{(i)}\right)\right]\right] && (5.62)\\
&\le \frac{1}{n}\mathbb{E}_\sigma\left[\sup_\theta \sum_{j=1}^m |w_j|\|u_j\|_2 \max_{k\in[n]}\left|\sum_{i=1}^n \sigma_i \phi\left(\bar{u}_k^T x^{(i)}\right)\right|\right] && \left(\because \sum_j \alpha_j\beta_j \le \sum_j |\alpha_j| \max_k |\beta_k|\right) \quad (5.63)\\
&\le \frac{B_C}{n}\mathbb{E}_\sigma\left[\sup_{\theta=(w,U)} \max_{k\in[n]}\left|\sum_{i=1}^n \sigma_i \phi\left(\bar{u}_k^T x^{(i)}\right)\right|\right] && (\because C(\theta) \le B_C) \quad (5.64)\\
&= \frac{B_C}{n}\mathbb{E}_\sigma\left[\sup_{\bar{u}:\|\bar{u}\|_2=1}\left|\sum_{i=1}^n \sigma_i \phi\left(\bar{u}^T x^{(i)}\right)\right|\right] && (5.65)\\
&\le \frac{B_C}{n}\mathbb{E}_\sigma\left[\sup_{\bar{u}:\|\bar{u}\|_2\le 1}\left|\sum_{i=1}^n \sigma_i \phi\left(\bar{u}^T x^{(i)}\right)\right|\right] && (5.66)\\
&\le \frac{2B_C}{n}\mathbb{E}_\sigma\left[\sup_{\bar{u}:\|\bar{u}\|_2\le 1}\sum_{i=1}^n \sigma_i \phi\left(\bar{u}^T x^{(i)}\right)\right] && \text{(see Lemma 5.11)} \quad (5.67)\\
&= 2B_C R_S(\mathcal{H}'), && (5.68)
\end{aligned}
$$

where $\mathcal{H}' = \left\{x \mapsto \phi(\bar{u}^T x) : \bar{u} \in \mathbb{R}^d, \|\bar{u}\|_2 \le 1\right\}$. By Talagrand's lemma, since ϕ is 1-Lipschitz, $R_S(\mathcal{H}') \le R_S(\mathcal{H}'')$ where $\mathcal{H}'' = \left\{x \mapsto \bar{u}^T x : \bar{u} \in \mathbb{R}^d, \|\bar{u}\|_2 \le 1\right\}$ is a linear hypothesis space. Using $R_S(\mathcal{H}'') \le \frac{C}{\sqrt{n}}$ by Theorem 5.5 then concludes the proof.

□

We complete the proof by deriving the Lemma 5.11 used in the second last inequality. Notably, the lemma's assumption holds in the current context, since

$$
\sup_\theta \langle \sigma, f_\theta(x)\rangle = \sup_{\bar{u}:\|\bar{u}\|_2\le 1} \sum_{i=1}^n \sigma_i \phi\left(\bar{u}^T x^{(i)}\right) \ge 0. \tag{5.69}
$$

since one can take $\bar{u} = 0$ for any $\sigma = (\sigma_1, \ldots, \sigma_n)$.

Lemma 5.11. *Let $\sigma = (\sigma_1, ..., \sigma_n)$ and $f_\theta(x) = \left(f_\theta\left(x^{(1)}\right), ..., f_\theta\left(x^{(n)}\right)\right)$. Suppose that for any $\sigma \in \{\pm 1\}^n$, $\sup_\theta \langle \sigma, f_\theta(x)\rangle \ge 0$. Then,*

$$
\mathbb{E}_\sigma\left[\sup_\theta |\langle \sigma, f_\theta(x)\rangle|\right] \le 2\mathbb{E}_\sigma\left[\sup_\theta \langle \sigma, f_\theta(x)\rangle\right]. \tag{5.70}
$$

Proof. Letting ϕ be the ReLU function, the lemma's assumption implies that $\sup_\theta \phi(\langle \sigma, f_\theta(x)\rangle) = \sup_\theta \langle \sigma, f_\theta(x)\rangle$ for any $\sigma \in \{\pm 1\}^n$. Observing that $|z| = \phi(z) + \phi(-z)$,

$$\sup_\theta |\langle \sigma, f_\theta(x)\rangle| = \sup_\theta \left[\phi(\langle \sigma, f_\theta(x)\rangle) + \phi(\langle -\sigma, f_\theta(x)\rangle)\right] \tag{5.71}$$

$$\leq \sup_\theta \phi(\langle \sigma, f_\theta(x)\rangle) + \sup_\theta \phi(\langle -\sigma, f_\theta(x)\rangle) \tag{5.72}$$

$$= \sup_\theta \langle \sigma, f_\theta(x)\rangle + \sup_\theta \langle -\sigma, f_\theta(x)\rangle. \tag{5.73}$$

Taking the expectation over σ (and noting that $\sigma \overset{d}{=} -\sigma$), we get the desired conclusion. □

Remark 5.12. Compared to Theorem 5.9, this bound does not explicitly depend on the number of neurons m. Thus, it is possible to use more neurons and still maintain a tight bound if the value of the new complexity measure $C(\theta)$ is reasonable. In contrast, the bound of Theorem 5.9 only looks at the total number of neurons. With the result above, it is possible to regularize $C(\theta)$ and obtain a tighter bound for any number m of neurons. This would lead to an accurate model with better generalization guarantees for a high number neurons.

For example, consider solving the constrained problem

$$\rho_m = \min_\theta C(\theta) \quad \text{such that} \quad f_\theta \text{ fits the data } \{(x^{(i)}, y^{(i)})\}_{i=1}^n. \tag{5.74}$$

In this case, ρ_m monotonically decreases as the number of neurons m increases. Indeed, models with more parameters necessarily include models with a lower number of parameters and thus those of lower complexity. As a result, it is possible to obtain lower complexity models by increasing the number of parameters m.

5.5 More implications and discussions on neural networks

In this section, we discuss practical implications of the refined neural network bound.

5.5.1 Connection to ℓ_2 regularization

Recall that margin theory yields

$$\text{for all } \theta, \quad L_{0\text{-}1}(\theta) \leq \frac{2R_S(\mathcal{H})}{\gamma_{\min}} + \widetilde{O}\left(\sqrt{\frac{\log(2/\delta)}{n}}\right), \tag{5.75}$$

with probability at least $1-\delta$. Thus, Theorem 5.10 motivates us to minimize $\frac{R_S(\mathcal{H})}{\gamma_{\min}}$ by regularizing $C(\theta)$. Concretely, this can be formulated as the optimization problem

$$\begin{aligned} \text{minimize} \quad & C(\theta) = \sum_{j=1}^m |w_j| \cdot \|u_j\|_2 \\ \text{subject to} \quad & \gamma_{\min}(\theta) \geq 1, \end{aligned} \tag{I}$$

or equivalently,

$$\begin{aligned} \text{maximize} \quad & \gamma_{\min}(\theta) \\ \text{subject to} \quad & C(\theta) \leq 1. \end{aligned} \tag{II}$$

At first glance, the above seems orthogonal to techniques used in practice. However, it turns out that the optimal neural network from (I) is functionally equivalent to that of the new problem:

$$\begin{aligned} \text{minimize} \quad & C_{\ell_2}(\theta) = \frac{1}{2}\sum_{j=1}^m |w_j|^2 + \frac{1}{2}\sum_{j=1}^m \|u_j\|_2^2 \\ \text{subject to} \quad & \gamma_{\min}(\theta) \geq 1. \end{aligned} \tag{I*}$$

This is a simple consequence of the positive homogeneity of ϕ. For any scaling factor $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbb{R}_+^m$, the rescaled neural network $\theta_\lambda := \{(\lambda_i w_i, u_i/\lambda_i)\}$ has the same functionality as the original neural network $\theta = \{w_i, u_i\}$ (i.e. it achieves the same $\gamma_{\min}$). Thus,

$$\min_\theta C_{\ell_2}(\theta) = \min_\theta \min_\lambda \left(\frac{1}{2} \sum_{j=1}^m \lambda_j^2 |w_j|^2 + \frac{1}{2} \sum_{j=1}^m \lambda_j^{-2} \|u_j\|_2^2 \right) \tag{5.76}$$

$$= \min_\theta \sum_{j=1}^m |w_j| \cdot \|u_j\|_2 \tag{5.77}$$

$$= \min_\theta C(\theta) \tag{5.78}$$

where we have used the equality case of the AM-GM inequality, attainable by $\lambda_j^* = \sqrt{\frac{\|u_j\|_2}{|w_j|}}$, in the second step. This equality case also shows that $\theta^* = \{(w_i, u_i)\}$ is the optimal solution of (I) if and only if $\hat{\theta}^* = \theta_{\lambda^*}$ is the optimal solution of (I*)—proving that $\hat{\theta}^*$ and θ^* are functionally equivalent since they only differ by a positive scale factor.

This connects our $C(\theta)$ regularization to ℓ_2-norm penalties that are more prevalent in practice. In retrospect, we see this equivalence is essentially due to the positive homogeneity of the neural network which "homogenizes" any inhomogeneous objective such as C_{ℓ_2}. Hence, we can just deal with $C(\theta)$ which is transparently homogeneous.

5.5.2 Stable generalization bound in m

Next, we show that the generalization bound given by Theorem 5.10 does not deteriorate with the network width (number of neurons) m, which is consistent with experimental results. To this end, the perspective of (II) enables us to isolate all dependencies of m in $\gamma_{\min}$. Letting $\widehat{\theta}_m$ denote the minimizer of program (II) with width m and defining optimal value $\gamma_m^* = \gamma_{\min}\left(\widehat{\theta}_m\right)$, we can rewrite the margin bound (5.75) as

$$L(\widehat{\theta}_m) \leq \frac{4C}{\sqrt{n}} \cdot \frac{1}{\gamma_m^*} + (\text{lower-order terms}), \tag{5.79}$$

where all dependencies on m are now contained in γ_m^*. Hence, to show that this bound does not worsen as m grows, we just have to show that γ_m^* is non-decreasing in m. This is intuitively the case since a neural network of width $m+1$ contains one of width m under the same complexity constraints. The following theorem formalizes this hunch:

Theorem 5.13. *Let γ_m^* be the minimum margin obtained by solving* (II) *with a two-layer neural network of width m. Then $\gamma_m^* \leq \gamma_{m+j}^*$ for all positive integers j.*

Proof. Suppose $\theta = \{(w_i, u_i)\}_{1 \leq i \leq m}$ is a two-layer neural network of width m satisfying $C(\theta) \leq 1$. Then we may construct a neural network $\widetilde{\theta} = \{(\tilde{w}_i, \tilde{u}_i)\}_{1 \leq i \leq m+1}$ of width $m+1$ by simply taking

$$(\widetilde{w}_i, \widetilde{u}_i) = \begin{cases} (w_i, u_i) & i \leq m, \\ (0, 0) & \text{otherwise} \end{cases} \tag{5.80}$$

$\widetilde{\theta}$ is functionally equivalent to θ and $C(\widetilde{\theta}) = C(\theta) \leq 1$. This means maximizing $\gamma_{\min}$ over $\{C(\widetilde{\theta}) : \widetilde{\theta} \text{ of width } m+1\}$ should give no lower of a value than the maximum of $\gamma_{\min}$ over $\{C(\theta) : \theta \text{ of width } m\}$. □

5.5.3 Equivalence to an ℓ_1-SVM in $m \to \infty$ limit

Since γ_m^* is non-decreasing in m, quantity

$$\gamma_\infty^* = \lim_{m \to \infty} \gamma_m^* \tag{5.81}$$

is well-defined. The next interesting fact is that in this $m \to \infty$ limit, γ^*_∞ of the two-layer neural network is equivalent to the minimum margin of an ℓ_1-SVM. As a brief digression, we recap the formulation of ℓ_p-SVMs and discuss the importance of ℓ_1-SVMs in particular.

Since a collection of data points with binary class labels may not be a priori separable, a *kernel model* first transforms an input x to $\varphi(x)$ where $\varphi : \mathbb{R}^d \to \mathcal{G}$ is known as the *feature map*. The model then seeks a separating hyperplane in this new (extremely high-dimensional) feature space $\mathcal{G}$, parameterized by a vector μ pointing from the origin to the hyperplane. The prediction of the model on an input x is then a decision score that quantifies $\varphi(x)$'s displacement with respect to the hyperplane:

$$g_{\mu,\varphi}(x) := \langle \mu, \varphi(x) \rangle . \tag{5.82}$$

Motivated by margin theory, it is desirable to seek the maximum-margin hyperplane under a constraint on μ to guarantee the generalizability of the model. In particular, a kernel model with an ℓ_p-constraint seeks to solve the following program:

$$\begin{aligned} \text{maximize} \quad & \gamma_{min} \coloneqq \min_{i \in [n]} y^{(i)} \langle \mu, \varphi(x^{(i)}) \rangle \\ \text{subject to} \quad & \|\mu\|_p \leq 1. \end{aligned} \tag{5.83}$$

Observe that both the prediction and optimization of the feature model only rely on inner products in $\mathcal{G}$. The ingenuity of the SVM is to choose maps φ such that $K(x, x') = \langle \varphi(x), \varphi(x') \rangle$ can be directly computed in terms of x and x' in the original space $\mathbb{R}^d$, thereby circumventing the need to perform expensive inner products in the large space $\mathcal{G}$. Remarkably, this "kernel trick" enables us to even operate in an implicit, infinite-dimensional $\mathcal{G}$.

The case of $p = 1$ is particularly useful in practice as ℓ_1-regularization generally produces sparse feature weights (the constrained parameter space is a polyhedron and the optimum tends to lie at one of its vertices). Hence, ℓ_1-regularization is an important feature selection method when one expects only a few dimensions of $\mathcal{G}$ to be significant. Unfortunately, the ℓ_1-SVM is not kernelizable due to the kernel trick relying on ℓ_2-geometry, and is hence infeasible to implement. However, our next theorem shows that a two-layer neural network can approximate a particular ℓ_1-SVM in the $m \to \infty$ limit (and in fact, for finite m). For the sake of simplicity, we sacrifice rigor in defining the space $\mathcal{G}$ and convey the main ideas.

Theorem 5.14. *Define the feature map $\phi_{\text{relu}} : \mathbb{R}^d \to \mathcal{G}$ such that x is mapped to $\phi(u^\top x)$ for all vectors u on the $d-1$-dimensional sphere $\mathcal{S}^{d-1}$. Informally,*

$$\phi_{\text{relu}}(x) := \begin{bmatrix} \vdots \\ \phi(u^\top x) \\ \vdots \end{bmatrix}_{u \in S^{d-1}}$$

is an "infinite-dimensional vector" that contains an entry $\phi(u^T x)$ for each vector $u \in \mathcal{S}^{d-1}$, and we let $\phi_{\text{relu}}(x)[u]$ denote the "u"-th entry of this vector. Noting that $\mathcal{G}$ is the space of functions which can be indexed by $u \in S^{d-1}$, the inner product structure on $\mathcal{G}$ is defined by $\langle f, g \rangle = \int_{S^{d-1}} f[u] g[u] du$.

Under this set-up, we have

$$\gamma^*_\infty = \gamma^*_{\ell_1}, \tag{5.84}$$

where $\gamma^*_{\ell_1}$ is the minimum margin of the optimized ℓ_1-SVM with $\varphi = \phi_{\text{relu}}$.

Proof. We will only prove the $\gamma^*_\infty \leq \gamma^*_{\ell_1}$ direction. (The $\gamma^*_\infty \geq \gamma^*_{\ell_1}$ direction requires substantial functional analysis.)

Suppose γ^*_∞ is obtained by network weights $(w_1, w_2, \cdots), (u_1, u_2, \cdots)$ where $w_i \in \mathbb{R}, u_i \in \mathbb{R}^d$ (there is a slight subtlety here to be rectified later). Define renormalized versions of $\{w_i\}$ and $\{u_i\}$:

$$\widetilde{w}_i := w_i \cdot \|u_i\|_2, \qquad \overline{u}_i := \frac{u_i}{\|u_i\|_2}. \tag{5.85}$$

Note that $\{(\widetilde{w}_i, \overline{u}_i)\}$ has the same functionality (and also the same complexity measure $C(\theta)$, margin, etc.) as that of $\{(w_i, u_i)\}$, but now $\overline{u}_i$ has unit ℓ_2-norm (i.e. $\bar{u}_i \in \mathcal{S}^{d-1}$). Thus, $\phi(\overline{u}_i^\top x)$ can be treated as a feature in $\mathcal{G}$ and we can construct an equivalent ℓ_1-SVM (denoted by μ) such that $\widetilde{w}_i$ is the coefficient of μ associated with that feature. Since $\widetilde{w}_i$ must only be "turned on' at $\overline{u}_i$, we have

$$\mu[u] = \sum_{i \in \mathcal{S}^{d-1}} \tilde{w}_i \delta(u - \overline{u}_i), \tag{5.86}$$

where $\delta(u)$ is the Dirac-delta function. Given this μ, we can check that the SVM's prediction is

$$\begin{aligned} g_{\mu,\phi_{\text{relu}}}(x) &= \int_{S^{d-1}} \mu[u]\phi_{\text{relu}}(x)[u]du && (5.87)\\ &= \int_{S^{d-1}} \sum_{i \in \mathcal{S}^{d-1}} \tilde{w}_i \delta(u - \overline{u}_i)\phi\left(\overline{u}^\top x\right) du && (5.88)\\ &= \sum_{i \in \mathcal{S}^{d-1}} \tilde{w}_i \phi\left(\overline{u}_i^\top x\right), && (5.89) \end{aligned}$$

which is identical to the output $f_{\{(\widetilde{w}_i, \overline{u}_i)\}}(x)$ of the neural network. Furthermore,

$$\|\mu\|_1 = \sum_{i=1}^{\infty} |\widetilde{w}_i| = \sum_{i=1}^{\infty} |w_i| \cdot \|u_i\|_2 \leq 1, \tag{5.90}$$

where the last equality holds because $\{(\widetilde{w}_i, \overline{u}_i)\}$ satisfies the constraints of (II). This shows that our constructed μ satisfies the ℓ_1-SVM constraint. Thus, $\gamma^*_\infty \leq \gamma^*_{\ell_1}$ since the functional behavior of the optimal neural network is contained in the search range of the SVM.

□

Remark 5.15. How well does a finite dimensional neural network approximate the infinite-dimensional ℓ_1 network? Proposition B.11 of [Wei et al., 2020] shows that you only need $n + 1$ neurons. Another way to say this is that $\{\gamma_m\}$ stabilizes once $m = n + 1$:

$$\gamma^*_1 \leq \gamma^*_2 \leq \cdots \leq \gamma^*_{n+1} = \gamma^*_\infty. \tag{5.91}$$

The main idea of the proof is that if we have a neural net θ with $(n + 2)$ neurons, then we can always pick a simplification θ' with $(n + 1)$ neurons such that θ, θ' agree on all n datapoints.

As an aside, this result also resolves the issue in our partial proof. A priori, γ^*_∞ may not necessarily be attained by a set of weights $\{(\widetilde{w}_i, \overline{u}_i)\}$ but the above shows that it is achievable with just $n + 1$ non-zero indices.

Chapter 6

Theoretical Mysteries in Deep Learning

We now turn to a high-level overview of deep learning theory. To begin, we outline a framework for classical machine learning theory, then discuss how the situation is different from deep learning theory.

6.1 Framework for classical machine learning theory

At the risk of oversimplification, we can divide classical machine learning theory into three parts:

1. **Approximation theory** attempts to answer whether there is any choice of parameters θ that achieves low population error. In other words, is the choice of hypothesis class good enough to approximate the ground truth function? Using notation from earlier in this course, the goal is to upper bound $L(\theta^*) = \min_{\theta \in \Theta} L(\theta)$.

2. **Statistical generalization** focuses on bounding the excess risk $L(\hat{\theta}) - L(\theta^*)$. In Chapter 4 we obtained the following bound:

$$L(\hat{\theta}) - L(\theta^*) \leq \underbrace{L(\hat{\theta}) - \hat{L}(\hat{\theta})}_{\text{generalization error}} + |L(\theta^*) - \hat{L}(\theta^*)|. \tag{6.1}$$

   The first term here is the generalization error, which usually has an upper bound of the form $R(\theta)/\sqrt{n}$, where $R(\theta)$ is some complexity measure.[1] This is a demonstration of *Occam's Razor*: the principle that simple (low-complexity) explanations generalize better.

   This statistical approach allows us to define a regularized loss $\hat{L}_{\text{reg}}(\theta) = \hat{L}(\theta) + \lambda R(\theta)$. Minimizing this loss gives us a solution $\hat{\theta}_\lambda$ which simultaneously has low training error and low complexity, which lets us bound both the training error and the generalization error. To summarize, in the classical setting, we can prove statements of the form

$$\text{If } \hat{\theta}_\lambda \text{ minimizes } \hat{L}_{\text{reg}}, \text{ then } L(\hat{\theta}_\lambda) - L(\theta^*) \text{ is small.} \tag{6.2}$$

3. **Optimization** considers how to obtain the minimizer $\hat{\theta}$ or $\hat{\theta}_\lambda$ computationally. This usually involves convex optimization: if $\hat{L}$ or $\hat{L}_{\text{reg}}$ is convex, then we have a polynomial-time algorithm to find the global minimum.

[1] In earlier chapters, we defined the complexity of a hypothesis class, not of a specific parameter value. To reconcile these two approaches, think of R as a measure of complexity (such as a norm) that we can then use to define a hypothesis class Θ, i.e. $\Theta = \{\theta' : R(\theta') \leq R(\theta)\}$.

While there are many tradeoffs to consider between these three components (for example, we may be able to find a loss function for which optimization is easy, but generalization becomes worse), it is still possible to study each area individually, then combine all three to get a result.

6.2 Deep learning theory and its differences

The situation is not so simple for deep learning theory. Let us consider how this is the case for each of the three components described for classical machine learning theory

1. **Approximation theory:** Large neural net models are considered to be very expressive. That is, both the population loss $L(\theta^*)$ and the finite sample loss $\hat{L}(\hat{\theta})$ can be made small. In fact, neural networks are *universal approximators*; see for example [Hornik, 1991]. This can be a somewhat misleading statement as the definition of universal approximator allows for the size of the network to be impracticably large, but morally it seems to hold true in practice anyway.

   This expressivity is possible because neural networks are usually highly *over-parametrized*: they have many more parameters than samples. It is possible to prove that in this regime, the network can "memorize" the entire dataset and achieve approximately zero training error.

2. **Statistical generalization:** Relatively weak regularization is used in practice. In many cases only weak ℓ_2 regularization is used, i.e.
   $$\widehat{L}_{\text{reg}}(\theta) = \hat{L}(\theta) + \lambda\|\theta\|_2^2. \tag{6.3}$$

   The first interesting fact is that this regularized loss does not have a unique (approximate) global minimizer. This is due to overparametrization: there are so many degrees of freedom that there are many approximate global minimizers with approximately the same ℓ_2 norm.

   However, it turns out that these global minimizers are not equally good: many models which achieve zero training error may have very bad test error. Take, for example, using stochastic gradient descent (SGD) to learn a model to classify the dataset CIFAR-10. Consider two instantiations of this: one starting with a large learning rate and slowly decreasing it, and one with a small learning rate throughout. Even though both instantiations result in approximately zero training error, the former leads to much better test performance.

   Therefore, the goal in deep learning theory is not just to find an arbitrary global minimum: we need to find the right global minimum. This contrasts sharply with (6.2) from the classical setting, where achieving a global minimum leads to good guarantees on generalization error. This means that (6.2) is simply not powerful enough to deal with deep learning, because it cannot distinguish between θ's with good test error and bad test error.

3. **Optimization:** The discussion above means that optimization plays a significant role in generalization for deep learning. Different training algorithms have different "implicit biases", causing them to converge to different global minimizers. Understanding the implicit biases of algorithms is thus a central goal of deep learning theory. It is impossible to design a good optimization algorithm without also considering its impact on generalization. In fact, many algorithms for non-convex optimization have been proposed that work well for minimizing training loss, but because their implicit bias is different, they lead to worse test performance and are therefore not too useful.

   Often these implicit biases can be interpreted as encouraging $\hat{\theta}$ to have low complexity in some sense. The deep learning analog of (6.2) is that "low complexity solutions generalize". This means that we end up doing more work on the optimization front in order to understand the implicit bias of our algorithm, and then proving generalization bounds works similarly to the classical setting once we understand how our optimizer finds a low-complexity solution.

To explain the success of deep learning, we will cover three tasks in the next two chapters:

1. Prove that our optimization algorithm converges to an approximate global minimum, even though the objective function is non-convex. Our results here will mostly be for simplified models (e.g. linearized neural nets). (We will also show later that this can be accomplished separately from the other tasks using a special optimization setup (the "NTK approach"). However, the generalization of this approach can be poor.)

2. Show that the solution $\hat{\theta}$ has low complexity $R(\hat{\theta}) \leq C$. We can only answer this question for some special cases of models (e.g. logistic regression, matrix factorization) and optimizers (e.g. gradient descent, label noise in SGD, dropout, learning rate).

3. Show that for all θ such that $R(\theta) \leq C$ with $\hat{L}(\theta) \approx 0$, we have $L(\theta)$ is small. That is, we show that low-complexity solutions to the empirical risk problem generalize well. We will be working with more fine-grained complexity measures, and several of the tools we used in classical machine learning can still apply.

Chapter 7

Nonconvex Optimization

In the previous chapter, we outlined conceptual topics in deep learning theory and how the situation was different from classical machine learning theory. In particular, we described *approximation theory*, *statistical generalization* and *optimization*. In this chapter, we will focus on optimization theory in deep learning. We will introduce some basics about optimization, discuss how we can make the notion "all local minima are global minima" rigorous, and walk through two examples where this is the case. Finally, we introduce the *neural tangent kernel* which allows us to make some characterizations of the loss near a given neural network initialization.

7.1 Optimization landscape

The big question that we have in mind is the following: many existing optimizers are designed for optimizing convex functions. **Why do they still work well for non-convex functions?** We note that it is not true that these optimizers always work well with non-convex functions: there are still some very hard cases that give trouble (e.g. very deep feed-forward networks are still hard to fit because of issues like vanishing and exploding gradients). One possible reason is that the non-convex functions that we are minimizing in deep learning usually have some nice properties: see Figure 7.1 for an illustration.

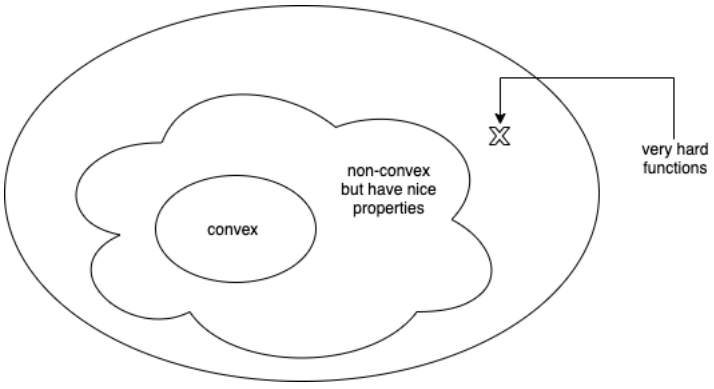

Figure 7.1: Classification of different functions for optimization. The functions we optimize in deep learning seem to fall mostly within the middle cloud.

Before diving into details, we first highlight some observations that will be important to keep in mind when discussing optimization in deep learning. Suppose $g(\theta)$ is the loss function. Recall that the *gradient descent (GD)* algorithm would do the following:

1. $\theta_0 :=$ initialization

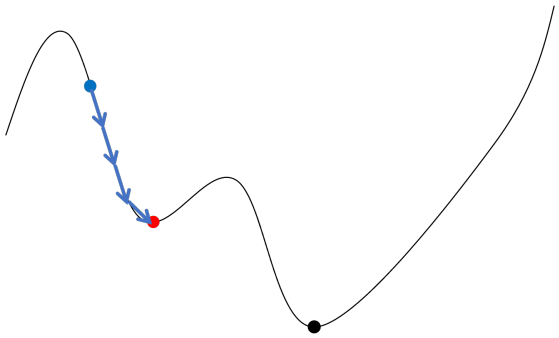

Figure 7.2: Illustration of how gradient descent does not always find the global minimum. In the picture, gradient descent initialized at the blue point only makes it to the local minimum at the red point: it does not find the global minimum at the black point.

2. $\theta_{t+1} = \theta_t - \eta \nabla g(\theta_t)$, where η is the step size.

Here are some observations to :

Observation 1: Gradient descent cannot always find the global minimum (see Figure 7.2 for an illustration).

Observation 2: Finding the global minimum of general non-convex functions is NP-hard.

Observation 3: Gradient descent can find the global minimum for convex functions.

Observation 4: The objective function in deep learning is non-convex.

Observation 5: Gradient descent/stochastic gradient descent typically finds an approximate global minimum of loss function in deep learning.

These observations motivate the following two-step plan:

1. Identify a large set of functions that stochastic gradient descent/gradient descent can solve.
2. Prove that some of the loss functions in machine learning problems belong to this set. (Most of the effort will be spent here.)

Basic idea: Gradient descent can find local minimum + all local minima of f are also global $\Rightarrow$ Gradient descent can find global minima.

7.2 Convergence to local minima

Let f be a twice-differentiable function. We start with the following definition:

Definition 7.1 (Local minimum of a function)**.** We say that x is a *local minimum* of a function f if there exists an open neighborhood N around x such that in N, the function values are at least $f(x)$.

Note that if x is a local minimum of f, then $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$. However, as the next example shows, the reverse is not true. When $\nabla f(x) = 0$ and $\nabla^2 f(x)$ vanishes in some direction (i.e. merely positive semi-definite instead of being strictly positive definite), higher-order derivatives start to matter.

Example 7.2. Consider the function $f(x_1, x_2) = x_1^2 + x_2^3$. $(x_1, x_2) = (0, 0)$ satisfies $\nabla f(x) = 0$ and $\nabla^2 f(x)|_{(x_1,x_2)=(0,0)} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$. However, if we move in the negative direction of x_2, we can decrease the function value. Hence, this example shows why $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ does not imply that x is a local minimum.

It is generally not easy to verify if a point is a local minimum. In fact, we have the following theorem regarding the computational tractability:

Theorem 7.3. *It is NP-hard to check whether a point is a local minimum or not [Murty and Kabadi, 1987]. In addition, Hillar and Lim [Hillar and Lim, 2013] show that a degree four polynomial is NP-hard to optimize.*

7.2.1 Strict-saddle condition

Theorem 7.3 forces us to consider more specific types of functions to be able to obtain computational tractability. To this end, we define the following *strict-saddle condition*:

Definition 7.4 (Strict-saddle condition [Lee et al., 2016])**.** For positive α, β, γ, we say that $f : \mathbb{R}^d \mapsto \mathbb{R}$ is (α, β, γ)-*strict-saddle* if every $x \in \mathbb{R}^d$ satisfies one of the following:

1. $\|\nabla f(x)\|_2 \geq \alpha$.
2. $\lambda_{\min}(\nabla^2 f(x)) \leq -\beta$.
3. x is γ-close to a local minimum x^* in Euclidean distance, i.e. $\|x - x^*\|_2 \leq \gamma$.

Intuitively speaking, this definition is saying if a point has zero gradient and positive semi-definite Hessian, it must be close to a local minimum, i.e. there is no pathological case like Example 7.2.

We have the following theorem for functions that satisfy strict-saddle condition:

Theorem 7.5 (Informally stated)**.** *If f is (α, β, γ)-strict-saddle for some positive α, β, γ, then many optimizers (e.g. gradient descent, stochastic gradient descent, cubic regularization) can converge to a local minimum with ϵ-error in Euclidean distance in time poly $\left(d, \frac{1}{\alpha}, \frac{1}{\beta}, \frac{1}{\gamma}, \frac{1}{\epsilon}\right)$.*

Therefore, if all local minima are global minima and the function satisfies the strict-saddle condition, then optimizers can converge to a global minimum with ϵ-error in polynomial time. (See Figure 7.3 for an example of a function whose local minima are all global minima.) The next theorem expresses this concretely by being explicit about the strict-saddle condition:

Theorem 7.6. *Suppose f is a function that satisfies the following condition: $\exists\ \epsilon_0, \tau_0, c > 0$ such that if $x \in \mathbb{R}^d$ satisfies $\|\nabla f(x)\|_2 \leq \epsilon < \epsilon_0$ and $\nabla^2 f(x) \succeq -\tau_0 I$, then x is ϵ^c-close to a global minimum of f. Then many optimizers can converge to a global minimum of f up to δ-error in Euclidean distance in time poly $\left(\frac{1}{\delta}, \frac{1}{\tau_0}, d\right)$.*

7.3 Two examples where local minima are global minima

So far, we have focused on general results. Next, we give two concrete examples that have the property that all local minima are global minima: (i) principal components analysis (PCA)/matrix factorization/linearized neural nets, and (ii) matrix completion.

7.3.1 Principal components analysis (PCA)

Let matrix $M \in \mathbb{R}^{d \times d}$ be symmetric and positive semi-definite. Consider the problem of finding the best rank-1 approximation of the matrix M. The objective function here is non-convex:

$$\min_{x \in \mathbb{R}^d} g(x) \triangleq \frac{1}{2}\|M - xx^T\|_F^2. \tag{7.1}$$

Theorem 7.7. *All local minima of g are global minima (even though g is non-convex).*

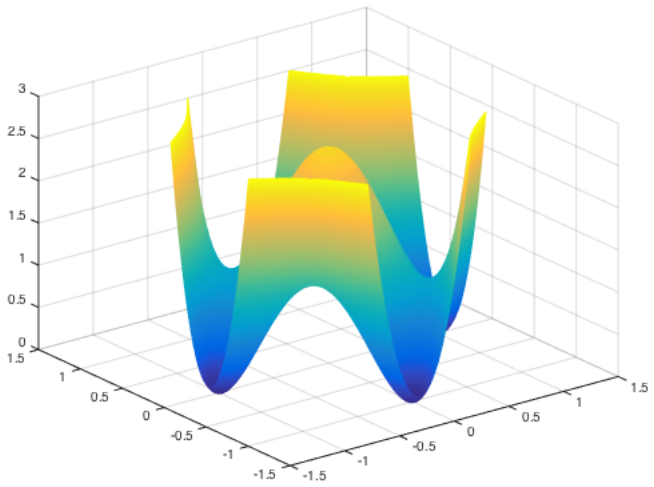

Figure 7.3: A two-dimensional function with the property that all local minima are global minima. It also satisfies the strict-saddle condition because all the saddle points have a strictly negative curvature in some direction.

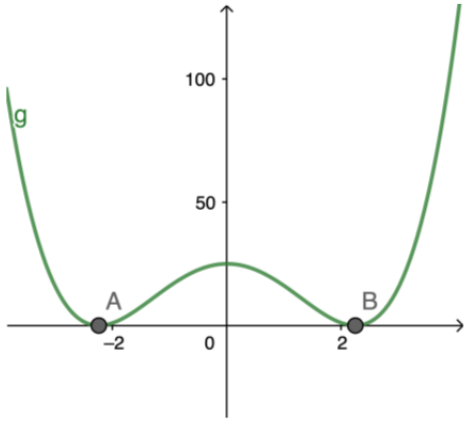

Figure 7.4: Objective function for principal components analysis (PCA) when $d = 1$.

Remark 7.8. For $d = 1$, $g(x) = \frac{1}{2}(m - x^2)^2$ for some constant m. Figure 7.4 below shows such an example. We can see that all local minima are indeed global minima.

Proof. Step 1: Show that all stationary points must be eigenvectors. From HW0, we know that $\nabla g(x) = -(M - xx^T)x$, hence

$$\nabla g(x) = 0 \implies Mx = \|x\|_2^2 \cdot x, \tag{7.2}$$

which implies that x is an eigenvector of M with eigenvalue $\|x\|_2^2$. From the Eckart–Young–Mirsky theorem we know the global minimum (i.e. the best rank-1 approximation) is the eigenvector with the largest eigenvalue.

Step 2: Show that all local minima must be eigenvectors of the largest eigenvalue. We use the second order condition for this. For x to be a local minimum we need $\nabla^2 g(x) \succeq 0$, which means for any $v \in \mathbb{R}^d$,

$$\langle v, \nabla^2 g(x) v \rangle \geq 0. \tag{7.3}$$

To compute $\langle v, \nabla^2 g(x) v \rangle$, we use the following trick: expand $g(x+v)$ into $g(x)$ + linear term in v + quadratic term in v, then the quadratic term will be $\frac{1}{2}\langle v, \nabla^2 g(x) v \rangle$ (see HW0 Problem 2d for an example). Using this trick, we get

$$\begin{aligned} g(x+v) &= \frac{1}{2}\|M - (x+v)(x+v)^T\|_F^2 && (7.4)\\ &= \frac{1}{2}\|M - xx^T\|_F^2 - \langle M - xx^T, xv^T + vx^T \rangle + \frac{1}{2}\langle xv^T + vx^T, xv^T + vx^T \rangle \\ &\quad - \langle M - xx^T, vv^T \rangle + \text{higher order terms in } v. && (7.5) \end{aligned}$$

Hence, we have

$$\begin{aligned} \frac{1}{2}\langle v, \nabla^2 g(x) v \rangle &= \frac{1}{2}\langle xv^T + vx^T, xv^T + vx^T \rangle - \langle M - xx^T, vv^T \rangle && (7.6)\\ &= \langle x, v \rangle^2 + \|x\|_2^2 \|v\|_2^2 - v^T M v + \langle x, v \rangle^2 && (7.7)\\ &= 2\langle x, v \rangle^2 + \|x\|_2^2 \|v\|_2^2 - v^T M v. && (7.8) \end{aligned}$$

Picking $v = v_1$, the unit eigenvector with the largest eigenvalue (denoted λ_1), for x to be a local minimum it must satisfy

$$\langle v_1, \nabla^2 g(x) v_1 \rangle = 2\langle x, v_1 \rangle^2 - v_1^T M v_1 + \|x\|_2^2 \geq 0. \tag{7.9}$$

Note that by (7.2), all our candidates for local minima are eigenvectors of M so naturally we have two cases:

- *Case 1: x has eigenvalue λ_1.* Then x is the global minimum (by the Eckart–Young–Mirsky theorem).
- *Case 2: x has eigenvalue $\lambda < \lambda_1$.* Then we know x and v_1 are orthogonal (eigenvectors with different eigenvalues are always orthogonal), hence

  $$2\langle x, v_1 \rangle^2 - v_1^T M v_1 + \|x\|_2^2 = 0 - \lambda_1 + \lambda \geq 0, \tag{7.10}$$

  which implies $\lambda \geq \lambda_1$, a contradiction.

In summary, if x is a stationary point and x is not a global minimum, then moving in the direction of v_1 would lead to second-order improvement and x cannot be a local minimum. □

7.3.2 Matrix Completion [Ge et al., 2016]

We consider rank-1 matrix completion for simplicity. Let $M = zz^T$ be a rank-1 symmetric and positive semi-definite matrix for some $z \in \mathbb{R}^d$. Given random entries of M, our goal is to recover the rest of entries. Formally, we have the following definitions:

Definition 7.9. Suppose $M \in \mathbb{R}^{d \times d}$ and $\Omega \subseteq [d] \times [d]$, we define $P_\Omega(M)$ to be the matrix obtained by zeroing out every entry outside Ω.

Definition 7.10 (Matrix Completion)**.** Suppose $M \in \mathbb{R}^{d \times d}$ and every entry of M is included in Ω with probability p. The *matrix completion task* is to recover M (with respect to some loss functions) given the observation $P_\Omega(M)$.

A nice real world example of matrix completion is when we have a matrix describing the user ratings for each item. We only observe a small portion of the entries as each customer only buys a small subset of the items. A good matrix completion algorithm is indispensable for a recommendation engine.

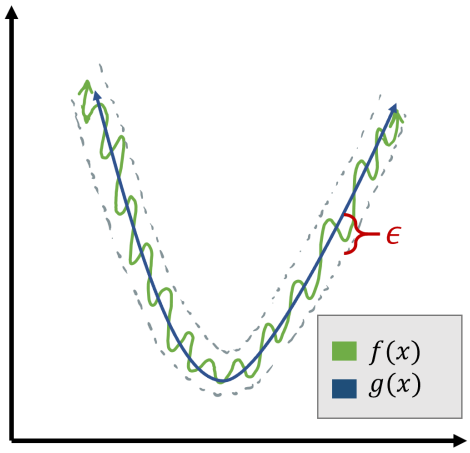

Figure 7.5: Even if $f(x)$ and $g(x)$ are no more than ϵ apart at any given x, the local minima of f may look dramatically different from the local minima of g.

Remark 7.11. We need d parameters to describe a rank-1 matrix M and the number of observations is roughly pd^2. Thus, for identifiability we need to work in the regime where $pd^2 > d$, i.e. $p \gg \frac{1}{d}$.

We define our non-convex loss functions to be

$$\min_{x \in \mathbb{R}^d} f(x) \triangleq \frac{1}{2} \sum_{(i,j) \in \Omega} (M_{ij} - x_i x_j)^2 \tag{7.11}$$

$$= \frac{1}{2} \|P_\Omega(M - xx^T)\|_F^2. \tag{7.12}$$

To really solve our problem we need some regularity condition on the ground truth vector z (recall $M = zz^T$). *Incoherence* is one such condition:

Definition 7.12 (Incoherence)**.** Without loss of generality, assume the ground truth vector $z \in \mathbb{R}^d$ satisfies $\|z\|_2 = 1$. z satisfies the *incoherence condition* if $\|z\|_\infty \leq \frac{\mu}{\sqrt{d}}$, where μ is considered to be a constant or log in dimension d.

Remark 7.13. A nice counterexample to think about why such condition is necessary is when $z = e_1$ and $M = e_1 e_1^T$. All entries of M are 0 except for a 1 in the top-left corner. There is no way to recover M without observing the top-left corner.

The goal is to prove that local minima of this objective function are close to a global minimum:

Theorem 7.14. *Assume $p = \frac{poly(\mu, \log d)}{d\epsilon^2}$ for some sufficient small constant ϵ and assume z is incoherent. Then with high probability, all local minima of f are $O(\sqrt{\epsilon})$-close to $+z$ or $-z$ (the global minima of f).*

Before presenting the proof, we make some observations that will guide the proof strategy.

Remark 7.15. $f(x)$ can be viewed as a sampled version of the PCA loss function $g(x) = \frac{1}{2}\|M - xx^T\|_F^2 = \frac{1}{2}\sum_{(i,j) \in [d] \times [d]} (M_{ij} - x_i x_j)^2$, in which we only observe a subset of the matrix entries. Thus, we would like to claim that $f(x) \approx g(x)$. However, matching the values of f and g is not sufficient to prove the theorem: even a small margin of error between f and g could lead to creation of many spurious local minima (see Figure 7.5 for an illustration). In order to ensure that the local minima of f look like the local minima of g, we will need further conditions like $\nabla f(x) \approx \nabla g(x)$ and $\nabla^2 f(x) \approx \nabla^2 g(x)$.

Remark 7.16. Key idea: concentration for scalars is easy. We can approximate a sum of scalars via a sample:

$$\sum_{(i,j)\in\Omega} T_{ij} \approx p \sum_{(i,j)\in[d]\times[d]} T_{ij}, \tag{7.13}$$

where we use $\approx$ to mean that

$$\Big| \sum_{(i,j)\in\Omega} T_{ij} - p \sum_{(i,j)\in[d]\times[d]} T_{ij} \Big| < \epsilon \tag{7.14}$$

with high probability. This suggests the strategy of casting the estimation of our desired quantities in the form of estimating a scalar sum via a sample. In particular, we note that for any matrices A and B,

$$\langle A, P_\Omega(B)\rangle = \sum_{(i,j)\in\Omega} A_{ij}B_{ij} \approx p\langle A, B\rangle. \tag{7.15}$$

To make use of this observation to understand the quantities of interest ($\nabla f(x)$ and $\nabla^2 f(x)$), we compute the bilinear and quadratic forms for $\nabla f(x)$ and $\nabla^2 f(x)$ respectively:

$$\langle v, \nabla f(x)\rangle = \langle v, P_\Omega(M - xx^T)x\rangle = \langle vx^T, P_\Omega(M - xx^T)\rangle, \tag{7.16}$$

where we have used the fact that $\langle A, BC\rangle = \langle AC^T, B\rangle$. Also note that vx^T is a rank-1 matrix and $M - xx^T$ is a rank-2 matrix.

$$\langle v, \nabla^2 f(x)v\rangle = \|P_\Omega(vx^T + xv^T)\|_F^2 - 2\langle P_\Omega(M - xx^T), vv^T\rangle \tag{7.17}$$

$$= \langle P_\Omega(vx^T + xv^T), vx^T + xv^T\rangle - 2\langle P_\Omega(M - xx^T), vv^T\rangle, \tag{7.18}$$

where we have used the fact that $\|P_\Omega(A)\|_F^2 = \langle P_\Omega(A), P_\Omega(A)\rangle = \langle P(\Omega(A), A\rangle$.

The key lemma that applies the scalar concentration to these matrix quantities is as follows:

Lemma 7.17. *Let $\epsilon > 0$, $p = \dfrac{poly(\mu, \log d)}{d\epsilon^2}$. Given that $A = uu^T, B = vv^T$ for some u, v satisfying $\|u\|_2 \le 1$, $\|v\|_2 \le 1$, $\|u\|_\infty \le \mu/\sqrt{d}$, $\|v\|_\infty \le \mu/\sqrt{d}$, we have $|\langle P_\Omega(A), B\rangle/p - \langle A, B\rangle| \le \epsilon$ w.h.p.*

If we can show that g has no bad local minima via a proof that only uses g via terms of the form $\langle v, \nabla g(x)\rangle$ and $\langle v, \nabla^2 g(x)v\rangle$, then by Lemma 7.17 this proof will automatically generalize to f by concentration.

Next, we prove some facts about g and show the analogous proofs for f that we will use in the proof of Theorem 7.14.

Lemma 7.18 (Connecting inner product and norm for g)**.** *If x satisfies $\nabla g(x) = 0$, then $\langle x, z\rangle^2 = \|x\|_2^4$.*

Proof.

$$\nabla g(x) = 0 \implies \langle x, \nabla g(x)\rangle = 0 \tag{7.19}$$

$$\implies \langle x, (zz^T - xx^T)x\rangle = 0 \qquad (\because \nabla g(x) = (M - xx^T)x) \tag{7.20}$$

$$\implies \langle x, z\rangle^2 = \|x\|_2^4. \tag{7.21}$$

□

Lemma 7.19 (Connecting inner product and norm for f)**.** *Suppose $\|x\|_\infty \le 2\mu/\sqrt{d}$. If x satisfies $\nabla f(x) = 0$, then $\langle x, z\rangle^2 \ge \|x\|_2^4 - \epsilon$ with high probability.*

Proof.

$$\nabla f(x) = 0 \implies \langle x, \nabla f(x)\rangle = 0 \tag{7.22}$$

$$\implies \langle x, \nabla g(x)\rangle \approx \langle x, \nabla f(x)\rangle/p \pm \epsilon \qquad \text{(by Lemma 7.17)} \tag{7.23}$$

$$\implies |\langle x, (zz^T - xx^T)x\rangle| \le \epsilon \qquad \text{w.h.p.} \tag{7.24}$$

$$\implies \langle x, z\rangle^2 \ge \|x\|_2^4 - \epsilon \qquad \text{w.h.p.} \tag{7.25}$$

□

Lemma 7.20 (Bound norm for g)**.** *If $\nabla^2 g(x) \succeq 0$, then $\|x\|_2^2 \geq 1/3$.*

Proof.

$$\begin{aligned}
\nabla^2 g(x) \succeq 0 &\implies \langle z, \nabla^2 g(x) z\rangle \geq 0 && (7.26)\\
&\implies \|zx^T + xz^T\|_F^2 - 2z^T(zz^T - xx^T)z \geq 0 && (7.27)\\
&\implies 1 \leq \|x\|_2^2 + 2\langle x, z\rangle^2 \leq \|x\|_2^2 + 2\|x\|_2^2 = 3\|x\|_2^2 \qquad \text{(by Cauchy-Schwarz)} && (7.28)\\
&\implies \|x\|_2^2 \geq 1/3. && (7.29)
\end{aligned}$$

□

Lemma 7.21 (Bound norm for f)**.** *Suppose $\|x\|_\infty \leq \mu/\sqrt{d}$. If $\nabla^2 f(x) \succeq 0$, then $\|x\|_2^2 \geq 1/3 - \epsilon/3$ with high probability.*

Proof.

$$\begin{aligned}
\nabla^2 f(x) \succeq 0 &\implies \langle z, \nabla^2 f(x) z\rangle \geq 0 && (7.30)\\
&\implies \langle z, \nabla^2 g(x) z\rangle \geq -\epsilon \qquad \text{w.h.p. (by Lemma 7.17)} && (7.31)\\
&\implies 3\|x\|_2^2 \geq 1 - \epsilon \qquad \text{w.h.p.} && (7.32)\\
&\implies \|x\|_2^2 \geq 1/3 - \epsilon/3 \qquad \text{w.h.p.} && (7.33)
\end{aligned}$$

□

Lemma 7.22 (g has no bad local minimum)**.** *All local minima of g are global minima.*

Proof.

$$\begin{aligned}
\nabla g(x) = 0 &\implies \langle z, \nabla g(x)\rangle = 0 && (7.34)\\
&\implies \langle z, (zz^T - xx^T)x\rangle = 0 && (7.35)\\
&\implies \langle x, z\rangle(1 - \|x\|_2^2) = 0. && (7.36)
\end{aligned}$$

Since $|\langle x, z\rangle| \geq 1/3 \neq 0$ (by Lemma 7.20), we must have $\|x\|_2^2 = 1$. But then Lemma 7.18 implies $\langle x, z\rangle^2 = \|x\|_2^4 = 1$, so $x = \pm z$ by Cauchy-Schwarz. □

We now prove Theorem 7.14, restated for convenience:

Theorem 7.23 (f has no bad local minimum)**.** *Assume $p = \dfrac{poly(\mu, \log d)}{d\epsilon^2}$. Then with high probability, all local minima of f are $O(\sqrt{\epsilon})$-close to $+z$ or $-z$.*

Proof. Observe that $\|x - z\|_2^2 = \|x\|_2^2 + \|z\|_2^2 - 2\langle x, z\rangle \leq \|x\|_2^2 + 1 - 2\langle x, z\rangle$. Our goal is to show that this quantity is small with high probability, hence we need to bound $\|x\|_2^2$ and $\langle x, z\rangle$ w.h.p. Note that the following bounds in this proof are understood to hold w.h.p.

Let x be such that $\nabla f(x) = 0$. For $\epsilon \leq 1/16$,

$$\begin{aligned}
\langle x, z\rangle^2 &\geq \|x\|_2^4 - \epsilon && \text{(by Lemma 7.19)} && (7.37)\\
&\geq (1/3 - \epsilon/3)^2 - \epsilon && \text{(by Lemma 7.21)} && (7.38)\\
&\geq 1/32. && && (7.39)
\end{aligned}$$

With this, we can get a bound on $\|x\|_2^2$:

$$\begin{aligned}
\nabla f(x) = 0 &\implies \langle x, \nabla f(x)\rangle = 0 && && (7.40)\\
&\implies |\langle z, \nabla g(x)\rangle| \leq \epsilon && \text{(by Lemma 7.17)} && (7.41)\\
&\implies |\langle x, z\rangle| \cdot |1 - \|x\|_2^2| \leq \epsilon && \text{(by dfn of } g\text{)} && (7.42)\\
&\implies |1 - \|x\|_2^2| \leq 32\epsilon = O(\epsilon) && \text{(by (7.39))} && (7.43)\\
&\implies \|x\|_2^2 = 1 \pm O(\epsilon). && && (7.44)
\end{aligned}$$

Next, we bound $\langle x, z\rangle$:

$$\langle x, z\rangle^2 \geq \|x\|_2^4 - \epsilon \qquad \text{(by Lemma 7.19)} \tag{7.45}$$
$$\geq (1 - O(\epsilon))^2 - \epsilon \qquad \text{(by (7.44))} \tag{7.46}$$
$$= 1 - O(\epsilon). \tag{7.47}$$

Finally, we put these quantities together to bound $\|x - z\|_2^2$. We have two cases:
Case 1: $\langle x, z\rangle \geq 1 - O(\epsilon)$. Then

$$\|x - z\|_2^2 = \|x\|_2^2 + \|z\|_2^2 - 2\langle x, z\rangle \tag{7.48}$$
$$\leq \|x\|_2^2 + 1 - 2\langle x, z\rangle \tag{7.49}$$
$$\leq 1 + O(\epsilon) + 1 - 2(1 - O(\epsilon)) \tag{7.50}$$
$$\leq O(\epsilon). \tag{7.51}$$

Hence we conclude x is $O(\sqrt{\epsilon})$-close to z.
Case 2: $\langle x, z\rangle \leq -(1 - O(\epsilon))$. Then by an analogous argument, x is $O(\sqrt{\epsilon})$-close to $-z$. □

We have shown above that matrix completion of a rank-1 matrix has no spurious local minima. This proof strategy can be extended to handle higher-rank matrices and noisy matrices [Ge et al., 2016]. The proof also demonstrates a generally useful proof strategy: often, reducing a hard problem to an easy problem results in solutions that do not give much insight into the original problem, because the proof techniques do not generalize. It can often be fruitful to seek a proof in the simplified problem that makes use of a restricted set of tools that could generalize to the harder problem. Here we limited ourselves to only using $\langle v, \nabla g(x)\rangle$ and $\langle v, \nabla^2 g(x)v\rangle$ in the easy case; these quantities could then be easily converted to analogous quantities in f via the concentration lemma (Lemma 7.17).

7.4 Other problems where all local minima are global minima

We have now demonstrated that two classes of machine learning problems, rank-1 PCA and rank-1 matrix completion, have no spurious local minima and are thus amenable to being solvable by gradient descent methods. We now outline some major classes of problems for which it is known that there are no spurious local minima.

- Principal component analysis (covered in previous lecture).
- Matrix completion (and other matrix factorization problems). On a related note, it has also been shown that linearized neural networks of the form $y = W_1 W_2 x$, where W_1 and W_2 are optimized separately, have no spurious local minima [Baldi and Hornik, 1989]. It should be noted that linearized neural networks are not very useful in practice since the advantage of optimizing W_1 and W_2 separately versus optimizing a single $W = W_1 W_2$ is not clear.
- Tensor decomposition. The problem is as follows:

$$\text{maximize} \quad \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d}\sum_{l=1}^{d} T_{ijkl} x_i x_j x_k x_l \quad \text{such that} \quad \|x\|_2 = 1. \tag{7.52}$$

  Additionally, constraints are imposed on the tensor T to make the problem tractable. For example, one condition is that T must be a low-rank tensor with orthonormal components [Ge et al., 2015].

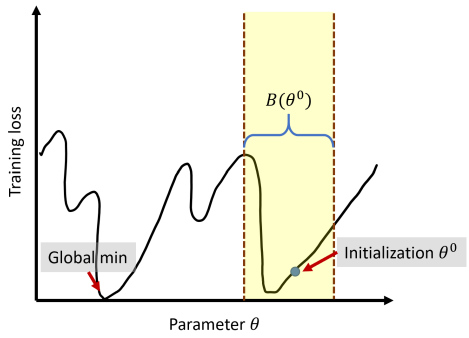

Figure 7.6: The training loss landscape around a given parameter initialization θ^0. We hope that the neighborhood around θ^0 contains a local minimum that is close to the global minimum.

7.5 Neural tangent kernel (NTK) approach

In general, the loss landscapes of neural networks (with nonlinearities) is currently not as well understood. We now introduce the *neural tangent kernel* which allows us to make some characterizations of the loss near a given neural network initialization.

The key insight of the NTK approach is that if we take an appropriate random parameter initialization θ^0 (which we will choose later), we can identify a special neighborhood of θ^0, denoted $B(\theta^0)$, where "everything is nice". That is, the function is convex in $B(\theta^0)$, there is a global minimum in the $B(\theta^0)$, and the algorithm starting at θ^0 will converge to that global minimum. (See Figure 7.6 for an illustration.)

Take a random initialization $\theta = \theta^0$ and Taylor expand the loss around θ^0 w.r.t. θ:

$$f_\theta(x) = \underbrace{f_{\theta^0}(x) + \langle \nabla_\theta f_{\theta^0}(x), \theta - \theta^0 \rangle}_{g_\theta(x)} + O((\theta - \theta^0)^2). \tag{7.53}$$

In other words, we take the tangent plane to f_θ at x (a linear approximation). We observe that g_θ is an affine function of θ. Additionally, defining $\Delta\theta = \theta - \theta^0$, we see that the first term does not depend on $\Delta\theta$ while the second term $\langle \nabla_\theta f_{\theta^0}(x), \theta - \theta^0 \rangle$ is linear in $\Delta\theta$. (For convenience, we will sometimes choose to design θ^0 such that $f_{\theta^0}(x) = 0$ so that g_θ is linear in θ^0. However, the difference is not very important since $f_{\theta^0}(x)$ can simply be subsumed in the training labels y via $y' = y - f_{\theta^0}(x)$.)

Now we have that $y \approx \nabla_\theta f_{\theta^0}(x)^T \Delta\theta$. We can view $\phi(x) \triangleq \nabla_\theta f_{\theta^0}(x)$ as a feature map, i.e. we can rewrite the expression as $\phi(x)^T \Delta\theta$ where $\phi(x)$ is fixed (only depends on θ^0 and the architecture). This observation motivates the definition of the *neural tangent kernel*:

Definition 7.24 (Neural tangent kernel)**.** The *neural tangent kernel* K is defined as the function

$$K(x, x') = \langle \phi(x), \phi(x') \rangle = \langle \nabla f_{\theta^0}(x), \nabla f_{\theta^0}(x') \rangle. \tag{7.54}$$

Suppose we fit $g_\theta(x)$ to y, i.e. we minimize the loss

$$\text{Loss} = \ell(\phi(x)^T \Delta\theta, y), \tag{7.55}$$

where $\phi(x)^T \Delta\theta$ is linear and the loss as a whole is convex. We will show that for a sufficiently wide neural network with proper initialization θ^0, optimizing $f_\theta(x)$ starting from θ^0 never leaves the neighborhood of θ^0, effectively behaving the same as optimizing $g_\theta(x)$. In particular, two questions have to be answered:

1. *Why does there exist a small neighborhood $B(\theta^0)$ such that there exists a global minimum in $B(\theta^0)$?* This is more surprising, and it involves proper design of θ^0. We will spend the rest of this chapter answering this question mathematically.
2. *Does gradient descent on the original loss with respect to $f_\theta(x)$ stay in the neighborhood $B(\theta^0)$?* The answer to this question is "yes". However, more technical machinery is required to prove it. We skip discussion of this as it is the less surprising claim.

7.5.1 The two-layer network case

We demonstrate the NTK approach for the two-layer network setup. For $i \in [m]$, let $a_i \in \mathbb{R}$ be scalars and let $w_i \in \mathbb{R}^d$ be vectors. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function defined as $\sigma(t) = \max\{t, 0\}$. Suppose we have the following two-layer network:

$$\hat{y} = f_\theta(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(w_i^\top x), \tag{7.56}$$

for some input x.

Our weight matrix is $W = \begin{bmatrix} w_1^\top \\ \vdots \\ w_m^\top \end{bmatrix} \in \mathbb{R}^{m \times d}$. We initialize W randomly using $W_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$ for all i and j. We initialize $a_i \in \{\pm 1\}$ and assume that the a_i's are fixed after initialization, i.e. not updated during training. (We fix a_i for simplicity: the results still hold when we are allowed to optimize a_i in training.) We also assume that x has norm on the order of 1, and that the true label y is on the order of 1.

In our analysis, we will assume we have sufficiently large $m = \text{poly}(n, d)$. In other words, the width of the network m is sufficiently large such that $\text{poly}(n, d)$ factors are not important in the analysis. For simplicity, we write $O_{d,n}(1)$ to hide polynomial dependencies on d and n. Thus $O_{d,n}(m^c) = m^c \cdot \text{poly}(n, d)$.

Why do we need the scaling factor $1/\sqrt{m}$ in Equation (7.56)? It is included to prevent the model outputs from blowing up when we increase the number of neurons m. Note that $\sigma(w_i^\top x) \approx O_{d,n}(1)$ since $w_i^\top \approx O_{d,n}(1)$ and $x \approx O_{d,n}(1)$. Since $a_i \in \{\pm 1\}$ for all i, this implies that $\sum_{i=1}^m a_i \sigma(w_i^\top x) \approx O_{d,n}(\sqrt{m})$. Thus, the scaling factor is needed to obtain $\hat{y} = f_\theta(x) = O_{d,n}(1)$.

Next, we introduce some notation that will be helpful for our analysis. Let $\Delta\theta = \theta - \theta^0$. Suppose we have n examples $x^{(1)}, ..., x^{(n)}$ and labels $y^{(1)}, ..., y^{(n)}$. Let $\vec{y} = \begin{bmatrix} y^{(1)} & \dots & y^{(n)} \end{bmatrix}^\top$. Let

$$\vec{y'} = \begin{bmatrix} y^{(1)} - f_{\theta^0}(x^{(1)}) \\ \vdots \\ y^{(n)} - f_{\theta^0}(x^{(n)}) \end{bmatrix} \tag{7.57}$$

be the transformed labels where we subsume the affine term in the label, allowing us to treat this as a purely linear model without loss of generality. Note that $\theta = \text{vec}(W) \in \mathbb{R}^{dm}$ is the vectorized version of the weights. Let $\varphi^{(i)} = \nabla_\theta f_{\theta^0}(x^{(i)}) \in R^{dm}$ be the feature associated with the ith example, and let φ denote the collection of the features across the examples:

$$\varphi = \begin{bmatrix} {\varphi^{(1)}}^\top \\ \vdots \\ {\varphi^{(n)}}^\top \end{bmatrix} \in R^{n \times dm}. \tag{7.58}$$

Recall that we defined

$$g_\theta(x) = f_{\theta^0}(x) + \langle \nabla_\theta f_{\theta^0}, \theta - \theta^0 \rangle, \tag{7.59}$$

which is the linear approximation of $f_\theta(x)$ at θ^0. If we wish to fit $g_\theta(x)$ to y with the ℓ_2-loss, we may consider minimizing the following objective function over θ:

$$\sum_{i=1}^{n} \left(y^{(i)} - f_{\theta^0}(x^{(i)}) - \langle \nabla_\theta f_{\theta^0}(x^{(i)}), \theta - \theta^0 \rangle \right)^2 = \sum_{i=1}^{n} \left(y^{(i)} - f_{\theta^0}(x^{(i)}) - \Delta\theta^\top \varphi^{(i)} \right)^2 \tag{7.60}$$

$$= \|\vec{y'} - \varphi \Delta\theta\|_2^2. \tag{7.61}$$

This is equivalent to the following optimization problem:

$$\min_{\Delta\theta} \quad \|\vec{y'} - \varphi \Delta\theta\|_2^2. \tag{7.62}$$

Since $n \ll dm$, so we have an undetermined linear system. Since our goal is to show that the relevant neighborhood around θ^0 is small, we should choose the minimum norm solution which can be found directly by the pseudoinverse: $\hat{\theta} = \varphi^\dagger \vec{y'}$, where $\varphi^\dagger$ is the pseudoinverse of φ given by $\varphi^\dagger = \varphi^\top(\varphi\varphi^\top)^{-1}$.

It remains to show that the norm of $\hat{\theta}$ is small. Before we can do that, we will prove some useful claims:

Lemma 7.25 (φ is a well-conditioned matrix)**.** *When θ^0 is random, φ is well-conditioned in the sense that*

$$\sigma_{\min}(\varphi) \approx \frac{1}{\sqrt{n}}\|\varphi\|_F \quad \textit{and} \quad \sigma_{\max}(\varphi) \approx \frac{1}{\sqrt{n}}\|\varphi\|_F. \tag{7.63}$$

($\sigma_{\min}(\varphi)$ and $\sigma_{\max}(\varphi)$ denote the smallest and largest singular values of φ respectively.) Specifically, $\sigma_{min}(\varphi) \gtrsim \Omega\left(\frac{1}{\sqrt{n}}\|\varphi\|_F\right)$, and vice-versa for $\sigma_{\max}(\varphi)$.

We omit the proof as it uses tools from random matrix theory that are not required for this course. (The high-level idea is to show that $\varphi\varphi^\top \approx c \cdot I$ for some constant scalar c.)

Remark 7.26. Since $\varphi \in R^{n \times dm}$, φ has at most n singular values $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$. Since $\|\varphi\|_F = \sqrt{\sigma_1^2 + ... + \sigma_n^2}$, the fact that $\sigma_n \approx \frac{1}{\sqrt{n}}\|\varphi\|_F$ means that all the singular values are not very different from each other.

Lemma 7.27 (Frobenius norm of φ is order 1)**.** $\|\varphi\|_F \asymp \Theta_{d,n}(1)$*, which implies that*

$$\|\varphi\|_{op},\ \|\varphi^\dagger\|_{op} \asymp \Theta_{d,n}(1). \tag{7.64}$$

Proof. Applying the definition of $\varphi^{(i)}$, we have

$$\varphi^{(i)} = \text{vec}\left(\frac{\partial f_\theta(x^{(i)})}{\partial W}\right) = \frac{1}{\sqrt{m}}(a \odot \sigma'(wx^{(i)})) \cdot \left(x^{(i)}\right)^\top. \tag{7.65}$$

Thus, its norm can be written as

$$\|\varphi^{(i)}\|_2 = \frac{1}{\sqrt{m}}\|a \odot \sigma'(wx^{(i)})\|_2 \cdot \|x^{(i)}\|_2 \tag{7.66}$$

$$\approx \Theta_{d,n}\left(\frac{1}{\sqrt{m}} \cdot \sqrt{m} \cdot 1\right) \tag{7.67}$$

$$\approx \Theta_{d,n}(1). \tag{7.68}$$

The first equality is because $\|ab^\top\|_2 = \|a\|_2 \cdot \|b\|_2 / \|ab^\top\|_F$ for any vectors a and b, and (7.67) is because $\|x^{(i)}\|_2$ is on order of 1, $a \odot \sigma'(wx^{(i)})$ is a vector of length m with each entry being on the order of 1 (each a_i is either 1 or -1, and σ' is either 0 or 1). Summing up over the $\varphi^{(i)}$'s,

$$\|\varphi\|_F = \sqrt{\sum_{i=1}^{n} \|\varphi^{(i)}\|_2^2} \approx \Theta_{d,n}(1). \tag{7.69}$$

Putting this together with Lemma 7.25, all the singular values of φ are $\Theta_{d,n}(1)$. By extension, $\|\varphi^\dagger\|_{\text{op}} \asymp \Theta_{d,n}(1)$ as well, since if a matrix A has singular values $\sigma_1, \ldots, \sigma_n$, $A^\dagger$ has singular values $1/\sigma_1, \ldots, 1/\sigma_n$. □

Now we can leverage the previous two lemmas to produce a bound on the ℓ_2-norm of the solution of the optimization problem (7.62), $\hat{\theta}$. Recall that $\hat{\theta} = \varphi^\dagger \vec{y'}$. Upper bounding the norm yields

$$\|\hat{\theta}\|_2 \leq \|\varphi^\dagger\|_{\text{op}} \cdot \|\vec{y'}\|_2 \tag{7.70}$$

$$\leq O_{d,n}(1) \cdot \|\vec{y'}\|_2 \qquad \text{(by Lemma 7.27)} \tag{7.71}$$

$$\leq O_{d,n}(1), \tag{7.72}$$

where the last inequality is because $\vec{y'}$ is of dimension n and each entry is on the order of 1 (the original labels are on the order of 1 and the shifts $f_{\theta^0}(x^{(i)})$ are also on the order of 1). Although $\|\hat{\theta}\|_2$ may not appear to be small, since it may still be poly(d, n), we can view it as comparatively small relative to the size of θ^0 since

$$\|\theta^0\|_2 = \|W^0\|_F^2 \tag{7.73}$$

$$\asymp \sqrt{dm} \qquad \text{(because } W^0 \text{ has } dm \text{ entries of order 1)} \tag{7.74}$$

$$= \Theta_{d,n}(\sqrt{m}). \tag{7.75}$$

Thus, the neighborhood size is much smaller than the norm of the initialization in terms of m. Further justification of the ℓ_2-norm as a reasonable metric for defining neighborhood size may require deeper inspection of the higher order terms and their behavior within the neighborhood. The intuition is that one only needs to move a little to reach the solution. Relative to the norm of the initialization, the neighborhood size is shrinking.

Remark 7.28. While we do not cover this in detail, the main takeaway on the optimization front is that the problem of fitting $g_\theta(x)$ is a standard strongly convex optimization problem, which enjoys the geometric rate of convergence.

7.5.2 Limitations of NTK

The NTK approach has its limitations.

- Empirically, optimizing $g_\theta(x)$ as described in the theory does not work as well as state-of-the-art (or even standard) deep learning methods. For example, using the NTK approach (i.e., taking the Taylor expansion and optimizing $g_\theta(x)$) with a ResNet generally does not perform as well as ResNet with best-tuned hyperparameters.
- The NTK approach requires a specific initialization scheme and learning rate which may not coincide with what is commonly used in practice.
- The analysis above was for gradient descent, while stochastic gradient descent is used in practice, introducing noise in the procedure. This means that NTK with stochastic gradient descent requires a small learning rate to stay in the initialization neighborhood. Deviating from the requirements can lead to leaving the initialization neighborhood.

One possible explanation for the gap between theory and practice is because NTK effectively requires a fixed kernel, so there is no incentive to select the right features. Furthermore, the minimum ℓ_2-norm solution is typically dense. This is similar to the difference between sparse and dense combinations of features observed in the ℓ_1-SVM/two-layer network versus the standard kernel method SVM (or ℓ_2-SVM) analyzed previously.

To make these ideas more concrete, consider the following example [Wei et al., 2020].

Example 7.29. Let $x \in \mathbb{R}^d$ and $y \in \{-1, 1\}$. Assume that each component of x satisfies $x_i \in \{-1, 1\}$. Define the output $y = x_1 x_2$, that is, y is only a function of the first two components of x.

This output function can be described exactly by a neural network consisting of a sparse combination of the features (4 neurons to be exact):

$$\hat{y} = \frac{1}{2}\left[\phi_{\text{relu}}(x_1 + x_2) + \phi_{\text{relu}}(-x_1 - x_2) - \phi_{\text{relu}}(x_1 - x_2) - \phi_{\text{relu}}(x_2 - x_1)\right] \tag{7.76}$$

$$= \frac{1}{2}\left(|x_1 + x_2| - |x_1 - x_2|\right) \tag{7.77}$$

$$= x_1 x_2. \tag{7.78}$$

(7.77) follows from the fact that $\phi_{\text{relu}}(t) + \phi_{\text{relu}}(-t) = |t|$ for all t, while (7.78) follows from evaluating the 4 possible values of (x_1, x_2). Thus, we can solve this problem exactly with a very sparse combination of features.

However, if we were to use the NTK approach (kernel method), the network's output will always involve $\sigma(w^\top x)$ where w is random so it includes all components of x (i.e. a dense combination of features), and cannot isolate just the relevant features x_1 and x_2. This is illustrated in the following informal theorem:

Theorem 7.30. *The kernel method with NTK requires $n = \Omega(d^2)$ samples to learn Example 7.29 well. In contrast, the neural network regularized by $\sum_{j=1}^m |u_j| \|w_j\|_2$ only requires $n = O(d)$ samples.*

Chapter 8

Implicit/Algorithmic Regularization Effect

One of the miracles of modern deep learning is the phenomenon of *algorithmic regularization* (also known as *implicit regularization* or *implicit bias*): although the loss landscape may contain infinitely many global minimizers, many of which do not generalize well, in practice our optimizer (e.g. SGD) tends to recover solutions with good generalization properties.

The focus of this chapter will be to illustrate algorithmic regularization in simple settings. In particular, we first show that gradient descent (with the right initialization) identifies the minimum norm interpolating solution in overparametrized linear regression. Next, we show that for a certain non-convex reparametrization of the linear regression task where the data is generated from a sparse ground-truth model, gradient descent (again, suitably initialized) approximately recovers a sparse solution with good generalization. Finally, we discuss algorithmic regularization in the classification setting, and how stochasticity can contribute to algorithmic regularization.

8.1 Algorithmic regularization in overparametrized linear regression

We prove that gradient descent initialized at the origin converges to the minimum norm interpolating solution (assuming such a solution exists). Let $X := \left[x^{(1)}, ..., x^{(n)}\right]^\top \in \mathbb{R}^{n \times d}$ denote our data matrix and $y := \left[y^{(1)}, ..., y^{(n)}\right]^\top \in \mathbb{R}^n$ denote our label vector, where $n < d$. Assume X is full rank. Our goal is to find a weight vector β that minimizes our empirical loss function $\widehat{L}(\beta) := \frac{1}{2}||y - X\beta||_2^2$.

8.1.1 Analysis of algorithmic regularization

As we are in the overparametrized setting with $n < d$ and X full rank, there exist infinitely many global minimizers that interpolate the data and hence achieve zero loss. In fact, the following lemma shows that the set of global minimizers forms a subspace.

Lemma 8.1. *Let $X^\dagger$ denote the pseudoinverse[1] of X. Then β is a global minimizer if and only if $\beta = X^\dagger y + \zeta$ for some ζ such that $\zeta \perp x_1, ..., x_n$.*

Proof. For any $\beta \in \mathbb{R}^d$, we can decompose it as $\beta = X^\dagger + \zeta$ for some $\zeta \in \mathbb{R}^d$. Since

$$X\beta = X(X^\dagger y + \zeta) = y + X\zeta, \tag{8.1}$$

[1]Since X is full rank, $XX^\top$ is invertible and so we have $X^\dagger = X^\top(XX^\top)^{-1}$. Note that $XX^\dagger X = X$.

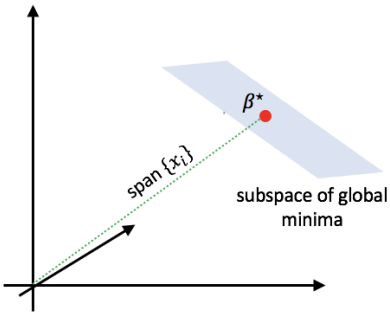

Figure 8.1: Visualization of proof intuition for Theorem 8.3. The solution $\beta^\star$ is the projection of the origin onto the subspace of global minima.

β is a global minimizer if and only if $X\zeta = 0$, which happens if and only if $\zeta \perp x_1, ..., x_n$.

□

From Lemma 8.1, we can derive an explicit formula for the minimum norm interpolant $\beta^\star := \arg\min_{\beta:\widehat{L}(\beta)=0} ||\beta||_2$.

Corollary 8.2. $\beta^\star = X^\dagger y$.

Proof. Take any β such that $\widehat{L}(\beta) = 0$, and write $\beta = X^\dagger y + \zeta$. Then from the definition of $X^\dagger$ and the fact that $X\zeta = 0$ (see the proof of Lemma 8.1), we have

$$\begin{aligned}
||\beta||_2^2 &= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle X^\dagger y, \zeta\rangle && (8.2)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle X^\top (XX^\top)^{-1} y, \zeta\rangle && (8.3)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 + 2\langle (XX^\top)^{-1} y, X\zeta\rangle && (8.4)\\
&= ||X^\dagger y||_2^2 + ||\zeta||_2^2 \qquad \text{(because } X\zeta = 0\text{)} && (8.5)\\
&\geq ||X^\dagger y||_2^2, && (8.6)
\end{aligned}$$

with equality if and only if $\zeta = 0$.

□

Now, suppose we learn β using gradient descent with initialization β^0, where at iteration t we set $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1})$ for some learning rate η. Since $\widehat{L}(\beta)$ is convex, we know from standard results in convex optimization that gradient descent will converge to a global minimizer for a suitably chosen learning rate η (in particular, taking η to be sufficiently small). Assuming $\beta^0 = 0$, we will in fact recover the minimum norm interpolating solution.

Theorem 8.3. *Suppose gradient descent on $\widehat{L}(\beta)$ with initialization $\beta^0 = 0$ converges to a solution $\hat{\beta}$ such that $\widehat{L}(\hat{\beta}) = 0$. Then $\hat{\beta} = \beta^\star$.*

The main idea of the proof is that the iterates of gradient descent always lie in the span of the $x^{(i)}$'s (see Figure 8.1 for an illustration).

Proof. We first show via induction that $\beta^t \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ for all t. For the induction base case, note that $\beta^0 = 0 \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. Now suppose $\beta^{t-1} \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. Recall that $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1})$. Since left-multiplying any vector by $X^\top$ amounts to taking a linear combination of the rows of X, it follows that $\eta\nabla\widehat{L}(\beta^{t-1}) = \eta X^\top(X\beta^{t-1} - y) \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$, and so $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1}) \in \text{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. This proves the induction step.

Next, we show that $\hat{\beta} \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ and $\widehat{L}(\hat{\beta}) = 0$ implies $\hat{\beta} = \beta^\star$. By definition, $\hat{\beta} \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ implies $\hat{\beta} = X^\top v$ for some $v \in \mathbb{R}^n$. Since $\widehat{L}(\hat{\beta}) = 0$, we have $0 = X\hat{\beta} - y = XX^\top v - y$. This implies $v = (XX^\top)^{-1}y$, and so $\hat{\beta} = X^\top v = X^\top (XX^\top)^{-1}y = X^\dagger y = \hat{\beta}^\star$. □

8.2 Algorithmic regularization in non-linear models

We give an example of algorithmic regularization in a non-convex version of the overparametrized linear regression task considered in the previous section.

Take X and y as defined in Section 8.1. This time, our goal is to find a weight vector that minimizes our empirical loss function

$$\widehat{L}(\beta) := \frac{1}{4n} \sum_{i=1}^{n} \left(y^{(i)} - f_\beta(x^{(i)}) \right)^2, \tag{8.7}$$

where $f_\beta(x) := \langle \beta \odot \beta, x \rangle$. (The operation $\odot$ denotes the Hadamard product: for $u, v \in \mathbb{R}^d$, $u \odot v \in \mathbb{R}^d$ is defined by $(u \odot v)_i := u_i v_i$ for $i = 1, \ldots, d$.)

We assume $x^{(1)}, ..., x^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{d\times d})$ and $y^{(i)} = f_{\beta^\star}(x^{(i)})$, where the ground truth vector $\beta^\star$ is r-sparse (i.e. $\|\beta^\star\|_0 = r$). For simplicity, we assume $\beta_i^\star = \mathbf{1}\{i \in S\}$ for some $S \subset [d]$ such that $|S| = r$. We again analyze the overparametrized setting, where this time $n \ll d$ but also $n \geq \widetilde{\Omega}(r^2)$.

8.2.1 Main results of algorithmic regularization

Note that while f_β is still linear over x, our loss is no longer convex over β. (To see this, suppose $\beta \neq 0$ is a global minimizer. Then we have $\widehat{L}(0) > \widehat{L}(\beta) = \widehat{L}(-\beta)$.) Thus, the effect of algorithmic regularization induced by gradient descent will be much different from the overparametrized linear regression setting.

In the previous setting of linear regression, solutions with low ℓ_2 norm are desirable as they tend to generalize well. In the present setting, we know our ground-truth parameter $\beta^\star$ is sparse. Thus, we want to learn a sparse solution $\hat{\beta}$, avoiding non-sparse solutions that may not generalize well. One approach to finding sparse solutions, called *lasso regression*, is to minimize the ℓ_1-regularized proxy loss

$$\sum_{i=1}^{n} \left(\langle \theta, x^{(i)} \rangle - y^{(i)} \right)^2 + \lambda \|\theta\|_1 \tag{8.8}$$

with respect to θ, where $\theta = \beta \odot \beta$. However, it turns out that we can equivalently learn a sparse solution by running gradient descent from a suitable initialization on the original *unregularized* loss.

To be specific, let $\beta^0 = \alpha \mathbf{1} \in \mathbb{R}^d$ be the initialization where α is a small positive number. The update rule of gradient descent algorithm is given by $\beta^{t+1} = \beta^t - \eta \nabla \widehat{L}(\beta^t)$. The next theorem shows that when $n = \widetilde{\Omega}(r^2)$, gradient descent on $\widehat{L}(\beta)$ converges to $\beta^\star$.

Theorem 8.4. *Let c be a sufficiently large universal constant. Suppose $n \geq cr^2 \log^2(d)$ and $\alpha \leq 1/d^c$, then when $\dfrac{\log(d/\alpha)}{\eta} \lesssim T \lesssim \dfrac{1}{\eta\sqrt{d\alpha}}$, we have*

$$\left\| \beta^T \odot \beta^T - \beta^\star \odot \beta^\star \right\|_2^2 \leq O\left(\alpha\sqrt{d}\right). \tag{8.9}$$

(Here, T indexes the gradient descent steps.)

We make several remarks about Theorem 8.4 before presenting the proof.

Remark 8.5. In this problem we do not use $\beta^0 = 0$ as the initialization point because $\beta = 0$ is a critical point, that is, $\nabla \widehat{L}(0) = 0$. Note that the lower bound on T depends logarithimically on $1/\alpha$, so we can take α to be a small inverse polynomial on d and the lower bound won't change much. Also, the upper bound depends polynomially on $1/\alpha$ (which is considered very big when c is sufficiently large), so we do not need to use early stopping in a serious way.

Remark 8.6. Theorem 8.4 is a simplified version of Theorem 1.1 in [Li et al., 2018].

Remark 8.7. $\widehat{L}(\beta)$ has many global minima. To see this, observe that the number of parameters is d and the number of constraints to fit all the examples is $O(n)$ because there are only n examples. Recall that for overparameterized model we have $d \gg n$; consequently, there exists many global minima of $\widehat{L}(\beta)$.

Remark 8.8. $\beta^\star$ is the min-norm solution in this case. That is,

$$\beta^\star = \operatorname{argmin} \|\beta\|_2^2 \qquad \text{s.t. } \widehat{L}(\beta) = 0. \tag{8.10}$$

Informally, this is because we can view $\beta \odot \beta$ as a vector $\theta \in \mathbb{R}^d$, which leads to $\|\beta\|_2^2 = \|\theta\|_1$. Then in the θ space (and with a little abuse of notation), the optimization problem (8.10) becomes

$$\theta^\star = \operatorname{argmin} \|\theta\|_1 \qquad \text{s.t. } \widehat{L}(\theta) = 0, \tag{8.11}$$

which is a lasso regression, whose solution is sparse.

Remark 8.9. In this non-linear case and the linear case before, gradient descent with small initialization converges to minimum ℓ_2-norm solution. Similarly, in the NTK regime, gradient descent converges to a solution that is very close to the initialization. Therefore, it seems conceivable that GD generally prefers global minima nearest to the initialization. However, we do not have a general theorem for this phenomenon (and the instructor also believes that this is not universally true without other conditions).

8.2.2 Ground work for proof and the restricted isometry property

In this section we prepare the ground work for the proof of Theorem 8.4.

We start by showing several basic properties about $\widehat{L}(\beta)$. Note that for any fixed vector $v \in \mathbb{R}^d$ and $x \in \mathbb{R}^d$, when x is drawn from $\mathcal{N}(0, I)$, we have

$$\mathbb{E}\left[\langle x, v\rangle^2\right] = \mathbb{E}\left[v^\top x x^\top v\right] = v^\top \mathbb{E}\left[xx^\top\right] v = \|v\|_2^2. \tag{8.12}$$

It follows that

$$\begin{aligned} L(\beta) &= \frac{1}{4}\mathbb{E}_{x\sim\mathcal{N}(0,I)}\left[(y - \langle \beta \odot \beta, x\rangle^2\right] && (8.13)\\ &= \frac{1}{4}\mathbb{E}_{x\sim\mathcal{N}(0,I)}\left[\langle \beta^\star \odot \beta^\star - \beta \odot \beta, x\rangle^2\right] && \text{(by definition of } y) \quad (8.14)\\ &= \frac{1}{4}\left\|\beta^\star \odot \beta^\star - \beta \odot \beta\right\|_2^2. && \text{(by (8.12))} \quad (8.15) \end{aligned}$$

Note that (8.15) is the metric that we use to characterize how close β is to the ground-truch parameter $\beta^\star$ (see (8.9)).

In the following lemma we show that $\widehat{L}(\beta) \approx L(\beta)$ by uniform convergence. Generally speaking, uniform convergence of the loss function for all β requires $n \geq \Omega(d)$ samples, so in our setting (where $n \ll d$) $\widehat{L}(\beta) \approx L(\beta)$ does not always hold. However, since we assume $\beta^\star$ is sparse, the analysis only requires uniform convergence for sparse vectors.

Lemma 8.10. *Assume $n \geq \widetilde{\Omega}(r^2)$. With high probability over the randomness in $x^{(1)}, \cdots, x^{(n)}$, $\forall v$ such that $\|v\|_0 \leq r$ we have*

$$(1-\delta)\|v\|_2^2 \leq \frac{1}{n}\sum_{i=1}^{n}\langle v, x^{(i)}\rangle^2 \leq (1+\delta)\|v\|_2^2. \tag{8.16}$$

Lemma 8.10 is a special case of Lemma 2.2 in [Li et al., 2018] so the proof is omitted here. We say the set $\left\{x^{(1)}, \cdots, x^{(n)}\right\}$ (or $X = [x^{(1)}, \cdots, x^{(n)}]$) satisfies (r, δ)-*RIP condition* (*restricted isometric property*) if (8.16) holds.

By algebraic manipulation, (8.16) is equivalent to

$$(1-\delta)\|v\|_2^2 \le v^\top \left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) v \le (1+\delta)\|v\|_2^2. \tag{8.17}$$

In other words, from the point of view of a sparse vector v we have $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top \approx I$. (Note however that $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is not close to $I_{d\times d}$ in other notions of closeness. For example, $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is not close to $I_{d\times d}$ in spectral norm. Another way to see this is that $\sum_{i=1}^n x^{(i)}(x^{(i)})^\top$ is a $d\times d$ matrix but only has rank $n \ll d$.)

As a result, with the RIP condition we have $\widehat{L}(\beta) \approx L(\beta)$ if β is sparse. With more tools we can also get $\nabla\widehat{L}(\beta) \approx \nabla L(\beta)$. Let us define the set $S_r = \{\beta : \|\beta\|_0 \le O(r)\}$, the set where we have uniform convergence of $\widehat{L}$ and $\nabla\widehat{L}$. Informally, as long as we are in the set S_r, $\widehat{L}$ and $\nabla\widehat{L}$ have similar behavior to their population counterparts. (Note, on the other hand, that there exists a dense $\beta \notin S_r$ such that $\widehat{L}(\beta) = 0$ but $L(\beta) \gg 0$.)

The RIP condition also gives us the following lemma which will be needed for the proof of Theorem 8.4.

Lemma 8.11. *Suppose $x^{(1)}, x^{(2)}, \dots x^{(n)}$ satisfy the (r,δ)-RIP condition. Then, $\forall v, w$ such that $\|v\|_0 \le r$ and $\|w\|_0 \le r$, we have that*

$$\left|\frac{1}{n}\sum_{i=1}^n \langle x^{(i)}, v\rangle\langle x^{(i)}, w\rangle - \langle v, w\rangle\right| = \left|v^T\left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) w - \langle v, w\rangle\right| \tag{8.18}$$

$$\le 4\delta \|v\|_2 \cdot \|w\|_2 . \tag{8.19}$$

Corollary 8.12. Taking $w = e_1, \dots, e_d$ in Lemma 8.11, we can conclude that

$$\left\|\frac{1}{n}\sum_{i=1}^n \langle x^{(i)}, v\rangle x^{(i)} - v\right\|_\infty = \left\|\left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) v - v\right\|_\infty \tag{8.20}$$

$$\le 4\delta \|v\|_2 . \tag{8.21}$$

8.2.3 Warm up: Gradient descent on population loss

The main intuition for proving Theorem 8.4 is to leverage the uniform convergence when β belongs to the set S_r (see Figure 8.2). Note that the initialization β^0 is not exactly r-sparse, but taking α to be sufficiently small, β^0 is approximately 0-sparse. The proof is decomposed into the following steps:

1. Gradient descent on $L(\beta)$ converges to $\beta^\star$ without leaving S_r, and
2. Gradient descent on $\widehat{L}(\beta)$ is similar to gradient descent on $L(\beta)$ inside S_r.

Combining the two steps we can show that gradient descent on $\widehat{L}(\beta)$ does not leave S_r and converges to $\beta^\star$.

As a warm up, we prove the following theorem for gradient descent on $L(\beta)$.

Theorem 8.13. *For sufficiently small η, gradient descent on $L(\beta)$ converges to $\beta^\star$ in $\Theta\left(\frac{\log(1/(\epsilon\alpha))}{\eta}\right)$ iteration with ϵ-error in ℓ_2-distance.*

Proof. Since

$$\nabla L(\beta) = (\beta\odot\beta - \beta^\star\odot\beta^\star)\odot\beta, \tag{8.22}$$

the gradient descent step is

$$\beta^{t+1} = \beta^t - \eta(\beta^t\odot\beta^t - \beta^\star\odot\beta^\star)\odot\beta^t. \tag{8.23}$$

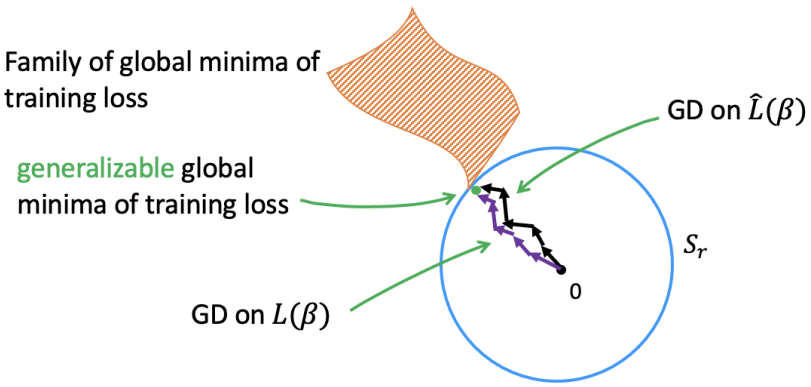

Figure 8.2: Visualization of proof intuition for Theorem 8.4.

Recall that $\beta^\star = \mathbf{1}\{i \in S\}$ and $\beta^0 = \alpha\mathbf{1}$, and the update rule above decouples across the coordinates of β^t. Thus, we only need to show that $|\beta_i^\star - \beta^t| \le \epsilon$ for the number of iterations stated in the Theorem.

Case 1: $i \in S$. For $i \in S$, the update rule for coordinate i is

$$\beta_i^{t+1} = \beta_i^t - \eta(\beta_i^t \cdot \beta_i^t - 1 \cdot 1) \cdot \beta_i^t \tag{8.24}$$

$$= \beta_i^t - \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t. \tag{8.25}$$

Consider the following two cases:

- If $\beta_i^t \le 1/2$, we have

$$\beta_i^{t+1} = \beta_i^t\left[1 + \eta\left(1 - \left(\beta_i^t\right)^2\right)\right] \tag{8.26}$$

$$\ge \beta_i^t\left(1 + \frac{3}{4}\eta\right). \tag{8.27}$$

  Consequently, β_i^{t+1} grow exponentially, and it takes $\Theta\left(\frac{\log(1/\alpha)}{\eta}\right)$ iterations for β_i^t to grow from α to at least $1/2$.[2] This will bring us into the second case.

- if $\beta_i^t \ge 1/2$, we have

$$1 - \beta_i^{t+1} = 1 - \beta_i^t + \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t \tag{8.28}$$

$$= 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\left(1 + \beta_i^t\right)\beta_i^t \tag{8.29}$$

$$\le 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\beta_i^t \qquad \text{(because } 1 + \beta_i^t \ge 1\text{)} \tag{8.30}$$

$$= \left(1 - \beta_i^t\right)\left(1 - \eta\beta_i^t\right) \tag{8.31}$$

$$\le \left(1 - \beta_i^t\right)(1 - \eta/2). \qquad \text{(because } \beta_i^t \ge 1/2\text{)} \tag{8.32}$$

  Therefore it takes $\Theta\left(\frac{\log(1/\epsilon)}{\eta}\right)$ iterations to achieve $1 - \beta_i^t \le \epsilon$.

Case 2: $i \notin S$. For all $i \notin S$, we claim (informally) that it is sufficient to show that when $t \le 1/(10\eta\alpha^2)$, $\beta_i^t \le 2\alpha$. This is because when $i \notin S$, β_i stays small and will take many iterations before it even gets to 2α, which is close to 0 since α is chosen to be small.

[2] This is because $(1+\eta)^{1/\eta} \approx e$, so $(1+\eta)^{c/\eta} \approx e^c$.

For a coordinate $i \notin S$, the gradient descent update for this problem becomes

$$\beta_i^{t+1} = \left[\beta^t - \eta(\beta^t \odot \beta^t - \beta^\star \odot \beta^\star) \odot \beta^t\right]_i \tag{8.33}$$

$$= \beta_i^t - \eta(\beta_i^t \cdot \beta_i^t) \cdot \beta_i^t \qquad (\text{since } \beta_i^\star = 0 \ \forall i \notin S) \tag{8.34}$$

$$= \beta_i^t - \eta(\beta_i^t)^3. \tag{8.35}$$

Since our initialization β^0 was small, the update to these coordinates will be even smaller because $(\beta_i^t)^3$ is small. We can prove the desired claim using strong induction. Suppose $\beta_i^s \le 2\alpha$ for all $s \le t$ and $i \notin S$, and that $t+1 \le 1/(10\eta\alpha^2)$. Then, for all $s \le t$,

$$\beta_i^{s+1} = (1 - \eta(\beta_i^s)^2)\beta_i^s \tag{8.36}$$

$$\le (1 + \eta(\beta_i^s)^2)\beta_i^s \tag{8.37}$$

$$\le (1 + 4\eta\alpha^2)\beta_i^s. \qquad (\text{since } \beta_i^s \le 2\alpha) \tag{8.38}$$

With strong induction, we can repeatedly apply this gradient update starting from $t = 0$ to obtain

$$\beta_i^{t+1} \le \beta_0 \cdot (1 + 4\eta\alpha^2)^t \tag{8.39}$$

$$\le \beta_0 (1 + 4\eta\alpha^2)^{\frac{1}{10\eta\alpha^2}} \tag{8.40}$$

$$\le \beta_0 \exp\left(\frac{4\eta\alpha^2}{10\eta\alpha^2}\right) \tag{8.41}$$

$$= \beta_0 \cdot e^{2/5} \tag{8.42}$$

$$\le 2\alpha, \tag{8.43}$$

which completes the inductive proof of the claim.

□

8.2.4 Proof of main result: Gradient descent on empirical loss

Analyzing gradient descsent on the empirical risk $\widehat{L}(\beta)$ is more complicated than analyzing gradient descent on the population risk, so we focus on the case when $\beta^\star$ is 1-sparse, i.e. $r = 1$. (When $r > 1$, the main idea is the same but requires some more advanced analysis techniques.)

Note that in our setup, i.e. when $x^{(1)} \dots x^{(n)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I_{d\times d})$ and when $n \ge \widetilde{\Omega}(r/\delta^2)$, with high probability the data satisfy the (r, δ)-RIP condition. It follows that when $r = 1$ and $\delta = \widetilde{O}(1/\sqrt{n})$, the data are $(1, \delta)$-RIP. This will allow us to use the lemmas involving the RIP condition for the proof.

We restate the case of $r = 1$ in the following theorem.

Theorem 8.14. *Suppose $\eta \ge \widetilde{\Omega}(1)$. Then, gradient descent on $\widehat{L}(\beta)$ with $t = \Theta\left(\frac{\alpha \log(1/\delta)}{\eta}\right)$ steps satisfies*

$$\left\|\beta^t \odot \beta^t - \beta^\star \odot \beta^\star\right\|_2^2 \le \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{8.44}$$

Remark 8.15. Note that Theorem 8.14 is a slightly weaker version of Theorem 8.4 for $r = 1$, since the bound on the RHS depends on the number of examples and not the initialization α. In Theorem 8.4, we could take α as small as we like to drive the bound to zero; we cannot do this for Theorem 8.14.

We proceed to prove Theorem 8.14 with the follow steps:

1. Computing the gradient update $\nabla\widehat{L}(\beta)$,
2. Dynamics analysis of noise ζ_t,
3. Dynamics analysis of signal r_t, and

4. Putting it all together.

Computing the gradient update $\nabla \widehat{L}(\beta)$

WLOG, assume that $\beta^\star = e_1$. We can decompose the gradient descent iterate β^t as

$$\beta^t = r_t \cdot e_1 + \zeta_t, \tag{8.45}$$

where $\zeta_t \perp e_1$. The idea is to prove convergence to $\beta^\star$ by showing that (i) $r_t \to 1$ as $t \to \infty$, and (ii) $\|\zeta_t\|_\infty \le O(\alpha)$ for $t \le \widetilde{O}(1/\eta)$. In other words, the *signal* r_t converges quickly to 1 while the *noise* ζ_t remains small for some number of initial iterations. One may be concerned that it is possible for the noise to amplify after many iterations, but we will not have to worry about this scenario if we can guarantee that β^t converges to $\beta^\star$ first.

We can compute the gradient of $\widehat{L}(\beta^t)$ as follows. Since $y^{(i)} = \langle \beta^\star \odot \beta^\star, x^{(i)} \rangle$ and $\beta^t = r_t e_1 + \zeta_t = r_t \beta^\star + \zeta_t$,

$$\begin{aligned}
\nabla \widehat{L}(\beta^t) &= \frac{1}{n} \sum_{i=1}^n (\langle \beta^t \odot \beta^t, x^{(i)} \rangle - y^{(i)}) x^{(i)} \odot \beta^t && (8.46)\\
&= \frac{1}{n} \sum_{i=1}^n (\langle \beta^t \odot \beta^t - \beta^\star \odot \beta^\star, x^{(i)} \rangle) x^{(i)} \odot \beta^t && (8.47)\\
&= \frac{1}{n} \sum_{i=1}^n \langle r_t^2 \beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t - \beta^\star \odot \beta^\star, x^{(i)} \rangle x^{(i)} \odot \beta^t && (8.48)\\
&= \underbrace{\frac{1}{n} \sum_{i=1}^n \left\langle \big(r_t^2 - 1\big) \beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t, x^{(i)} \right\rangle x^{(i)}}_{m_t} \odot \beta^t. && (8.49)
\end{aligned}$$

To simplify the analysis, we can rearrange some of the terms that are part of the gradient. Define m_t such that $\nabla \widehat{L}(\beta^t) = m_t \odot \beta^t$. Also, let $X = \frac{1}{n} \sum_{i=1}^n x^{(i)} (x^{(i)})^\top$. Then,

$$\begin{aligned}
m_t &= \frac{1}{n} \sum_{i=1}^n \left\langle \big(r_t^2 - 1\big) \beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t,\ x^{(i)} \right\rangle x^{(i)} && (8.50)\\
&= \left(\frac{1}{n} \sum_{i=1}^n x^{(i)} \big(x^{(i)}\big)^\top \right) \big(r_t^2 - 1\big) \cdot (\beta^\star \odot \beta^\star) + \left(\frac{1}{n} \sum_{i=1}^n x^{(i)} (x^{(i)})^\top \right) (\zeta_t \odot \zeta_t) && (8.51)\\
&= \underbrace{X\big(r_t^2 - 1\big) \cdot \big(\beta^\star \odot \beta^\star\big)}_{\text{part of } u_t} + \underbrace{X\big(\zeta_t \odot \zeta_t\big)}_{v_t}. && (8.52)
\end{aligned}$$

Now, define $u_t := (r_t^2 - 1)(\beta^\star \odot \beta^\star) - X(r_t^2 - 1)(\beta_* \odot \beta_*)$ and $v_t := X\big(\beta_t \odot \beta_t\big)$. Then we can rewrite the gradient as

$$\nabla \widehat{L}(\beta^t) = m_t \odot \beta^t = [(r_t^2 - 1)\beta^\star \odot \beta^\star - u_t + v_t] \odot \beta_t. \tag{8.53}$$

Our goal is to show that both u_t and v_t are small, so that $\nabla \widehat{L}(\beta^t)$ is close to its population version $\nabla L(\beta^t)$. Observe that X appears in both u_t and v_t. This matrix is challenging to deal with mathematically because it does not have full rank (because $n < d$). Instead, we rely on the RIP condition to reason about the behavior of X: the idea is that X behaves like the identity for sparse vector multiplication. Applying Corollary 8.12, we can bound $\|u_t\|_\infty$ as

$$\|u_t\|_\infty \le 4\delta \left\| (r_t^2 - 1)\beta^\star \odot \beta^\star \right\|_2 \le 4\delta ||\beta^\star \odot \beta^\star||_2 \le 4\delta. \tag{8.54}$$

(In the second inequality, we assume that $|r_t| < 1$. We can do this because r_t starts out at α which is small; if $r_t \ge 1$, then we are already in the regime where gradient descent has converged.) We can bound

$\|v_t\|_\infty$ in a similar manner: since Corollary 8.12 implies $\|v_t - \zeta_t \odot \zeta_t\|_\infty \le 4\delta \|\zeta_t \odot \zeta_t\|_2$,

$$\begin{aligned}
\|v_t\|_\infty &\le \|\zeta_t \odot \zeta_t\|_\infty + 4\delta \|\zeta_t \odot \zeta_t\|_2 && \text{(by the triangle inequality)} && (8.55)\\
&\le \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t \odot \zeta_t\|_1 && \text{(since } \zeta_t \text{ very small)} && (8.56)\\
&= \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t\|_2^2 . && && (8.57)
\end{aligned}$$

Note that the size of v_t depends on the size of the noise ζ_t. Thus, by bounding ζ_t (e.g. with a small initialization), we can ensure that v_t is also small. (Ensuring bounds on u_t is more difficult because it depends only on δ.) In the next two subsections, we analyze the growth of ζ_t and r_t.

Dynamics analysis of ζ_t

First, we analyze the dynamics of the noise ζ_t, which we want to ensure does not grow too fast.

Lemma 8.16. *For all $t \le 1/(c\eta\delta)$ with sufficiently large constant c, we have*

$$\|\zeta_t\|_\infty \le 2\alpha, \qquad \|\zeta_t\|_2^2 \le \frac{1}{2}, \qquad \textit{and} \quad \|\zeta_{t+1}\|_\infty \le \big(1 + O(\eta\delta)\big) \|\zeta_t\|_\infty . \tag{8.58}$$

Note that this result is weaker than what we were able to show for the population gradient (exponential growth with a small fixed rate), but we will ultimately show that the growth of the signal will be even faster.

Proof. Recall that the empirical gradient (8.53) is $\nabla \hat{L}(\beta) = \big[(r_t^2 - 1)\beta^\star \odot \beta^\star - u_t + v_t\big] \odot \beta^t$. Hence, the gradient update to β^t is

$$\begin{aligned}
\beta^{t+1} &= \beta^t - \eta \big[\big(r_t^2 - 1\big)\, \beta^\star \odot \beta^\star - u_t + v_t\big] \odot \beta^t && (8.59)\\
&= \underbrace{\beta^t - \eta \big(r_t^2 - 1\big)\, \beta^\star \odot \beta^\star \odot \beta^t}_{\text{GD update for population loss}} - \eta\, (-u_t + v_t) \odot \beta^t . && (8.60)
\end{aligned}$$

Recall that ζ_{t+1} is simply β^{t+1} except for the first coordinate (where it has a zero instead of r_{t+1}), i.e. ζ_{t+1} is the projection of β^{t+1} onto the subspace orthogonal to e_1. Hence,

$$\begin{aligned}
\zeta_{t+1} &= \big(I - e_1 e_1^\top\big)\, \beta^{t+1} && && (8.61)\\
&= \big(I - e_1 e_1^\top\big) \cdot \beta^t - \eta \big(I - e_1 e_1^\top\big)\, (v_t - u_t) \odot \beta^t && \text{(by (8.60), second term = 0)} && (8.62)\\
&= \zeta_t - \eta \big[\big(I - e_1 e_1^T\big)\, (v_t - u_t) \odot \big(I - e_1 e_1^T\big)\, \beta^t\big] && \text{(by distribution law for } \odot) && (8.63)\\
&= \zeta_t - \eta \underbrace{\big[\big(I - e_1 e_1^T\big)\, (v_t - u_t)\big]}_{\rho_t} \odot \zeta_t . && && (8.64)
\end{aligned}$$

If we define ρ_t such that $\zeta_{t+1} = \zeta_t - \eta\rho_t \odot \zeta_t$, then the growth of ζ_t is dictated by the size of ρ_t. We can bound this as

$$\|\zeta_{t+1}\|_\infty \le (1 + \eta \|\rho_t\|_\infty) \|\zeta_t\|_\infty . \tag{8.65}$$

Now, we will prove the lemma by using strong induction on t. Suppose that the first two pieces of (8.58) hold for all iterations up to t. We can show that

$$\begin{aligned}
\|\rho_t\|_\infty &\le \|u_t\|_\infty + \|v_t\|_\infty && && (8.66)\\
&\le 4\delta + \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t\|_2^2 && \text{(by (8.54) and (8.57))} && (8.67)\\
&\le 4\delta + (2\alpha)^2 + 4\delta \cdot \frac{1}{2} && \text{(by the inductive hypothesis)} && (8.68)\\
&\le 8\delta, && && (8.69)
\end{aligned}$$

where the last step holds because we can take α to be arbitrarily small (e.g. $\alpha \le \text{poly}(1/n) \le O(\delta)$). Plugging this into (8.65), we have

$$\|\zeta_{t+1}\|_\infty \le (1 + 8\eta\delta) \|\zeta_t\|_\infty = \big(1 + O(\eta\delta)\big) \|\zeta_t\|_\infty , \tag{8.70}$$

which proves the third piece of the lemma. Using this piece, we can show that

$$\|\zeta_{t+1}\|_\infty \le (1+8\eta\delta)^{t+1}\|\zeta_0\|_\infty \le \big(1+8\eta\delta\big)^{1/(c\eta\delta)}\cdot\alpha \le 2\alpha \tag{8.71}$$

for a sufficiently large constant c, which proves the second piece. Finally, we show that

$$\|\zeta_{t+1}\|_2^2 \le \big(1+8\eta\delta\big)^{t+1}\|\zeta_0\|_2^2 \le \big(1+8\eta\delta\big)^{1/(c\eta\delta)}\cdot\alpha\sqrt{d} \le \frac{1}{2}, \tag{8.72}$$

if $\alpha \le \frac{1}{n^{O(1)}}$, which proves the first piece.

□

Dynamics analysis of r_t

Next, we analyze the dynamics of the signal r_t, which we want to show converges to 1.

Lemma 8.17. *For all $t \le 1/(c\eta\delta)$ with sufficiently large constant c, we have that*

$$r_{t+1} = \big(1+\eta\big(1-r_t^2\big)\big)r_t + O\big(\eta\delta\big)r_t.$$

Note that the first term on the RHS is r_{t+1} during gradient descent on the population loss, and the second term captures the error.

Proof. Recall that the gradient descent update from the empirical gradient (8.53) is

$$\beta^{t+1} = \beta^t - \eta\big[\big(r_t^2-1\big)\beta^\star\odot\beta^\star - u_t + v_t\big]\odot\beta_t. \tag{8.73}$$

We have that

$$\begin{aligned}
r_{t+1} &= \big\langle\beta^{t+1}, e_1\big\rangle && (8.74)\\
&= \big\langle\beta^t, e_1\big\rangle - \eta\big(r_t^2-1\big)\big\langle\beta^t, e_1\big\rangle - \eta\big\langle v_t-u_t, e_1\big\rangle\big\langle\beta^t, e_1\big\rangle && (8.75)\\
&= r_t - \eta\big(r_t^2-1\big)r_t - \eta\big\langle v_t-u_t, e_1\big\rangle r_t && (8.76)\\
&= \Big(1+\eta\big(1-r_t^2\big)\Big)r_t + \eta\big\langle u_t-v_t, e_1\big\rangle r_t && (8.77)
\end{aligned}$$

so all we need to do is bound the second term as follows:

$$\begin{aligned}
|\eta\langle v_t-u_t, e_1\rangle r_t| &\le \eta\cdot r_t\,\|v_t-u_t\|_\infty && (8.78)\\
&\le \eta\cdot r_t\cdot 8\delta \qquad\qquad \text{(by (8.69))} && (8.79)\\
&= O(\eta\delta)\cdot r_t. && (8.80)
\end{aligned}$$

□

Putting it all together Finally, we return to the proof of Theorem 8.14. By Lemma 8.17, we know that as long as $r_t \le 1/2$ it will grow exponentially fast, since

$$r_{t+1} \ge \Big(1+\eta\big(1-r_t^2\big)-O(\eta\delta)\Big)\cdot r_t \ge \left(1+\frac{\eta}{2}\right)\cdot r_t. \tag{8.81}$$

This implies that at some $t_0 = O\Big(\frac{\log(1/\alpha)}{\eta}\Big)$, we'll observe $r_{t_0} > 1/2$ for the first time. Consider what happens after this point.

- When $1/2 < r_t \leq 1$, we have that

$$1 - r_{t+1} \leq 1 - r_t - \eta\big(1 - r_t^2\big) r_t + O(\eta\delta) \cdot r_t \tag{8.82}$$

$$\leq 1 - r_t - \frac{\eta\big(1 - r_t^2\big)}{2} + O(\eta\delta) \tag{8.83}$$

$$\leq 1 - r_t - \frac{\eta\big(1 - r_t\big)}{2} + O(\eta\delta) \tag{8.84}$$

$$= \left(1 - \frac{\eta}{2}\right)(1 - r_t) + O(\eta\delta). \tag{8.85}$$

  Thus, we can achieve $1 - r_{t+1} \leq 2 \cdot \frac{O(n\delta)}{\eta/2} = O(\delta)$ in $\Theta\Big(\frac{\log(1/\delta)}{\eta}\Big)$ iterations.

- When $r_t > 1$, we can show in a similar manner that

$$r_{t+1} - 1 \leq (1 - \eta)(r_t - 1) + O(\eta\delta) \leq O(\delta), \tag{8.86}$$

  implying that r_t remains very close to 1 after the same order of iterations.

This completes the proof of Theorem 8.14, bounding the number of iterations needed for gradient descent on the empirical loss to converge to β^*. □

8.3 Algorithmic regularization for classification

In this section, we will discuss algorithmic regularization for classification problem. In particular, we consider binary classification with logistic loss. Let $\{(x_i, y_i)\}_{i=1}^n$ be a separable dataset with $y_i \in \{\pm 1\}$, $x_i \in \mathbb{R}^d$. We have a linear model $h_w(x) = w^\top x$, and we minimize the empirical logistic loss

$$\widehat{L}(w) = \frac{1}{n} \sum_{i=1}^n \ell(y_i h_w(x_i)) \tag{8.87}$$

$$= \frac{1}{n} \sum_{i=1}^n \ell(y_i w^\top x_i), \tag{8.88}$$

where $\ell(t) = \log(1 + \exp(-t))$ is the logistic loss.

In order to observe algorithmic regularization, we need to ensure that there exist multiple global minima for this setup. This is the case here: because the dataset is linearly separable, there exists some w such that $y_i w^\top x_i > 0$ for all i. Clearly any w' in a small neighborhood of w also classifies all the data correctly; hence, there exists an infinite number of separating classifiers $\overline{w}$ with unit norm. For any of these $\overline{w}$, note that $\widehat{L}(\alpha\overline{w}) \to 0$ as $\alpha \to \infty$, hence intuitively all of "$\infty\overline{w}$" classifiers are global minima.

Having shown the existence of multiple global minima, we now show that gradient descent will actually converge to the solution which maximizes the *margin*. We first define the *normalized margin* for a separating classifier w as

$$\gamma(w) = \frac{\min_{i \in [n]} y_i w^\top x_i}{\|w\|_2}. \tag{8.89}$$

We call $\overline{\gamma} = \max_w \gamma(w)$ the *max margin*. Now we are ready to state the theorem:

Theorem 8.18 ([Soudry et al., 2018])**.** *Gradient descent with iterates w_t converges to the direction of a max-margin solution:*

$$\gamma(w_t) \to \overline{\gamma} \quad \textit{as} \quad t \to \infty. \tag{8.90}$$

In other words, gradient descent on logistic loss is equivalent to the SVM.[3]

Here, we provide the intuition behind the proof. The proof of this theorem follows these steps:

1. By standard convex optimization arguments, $\widehat{L}(w_t) \to 0$ as $t \to \infty$.
2. For sufficiently large t, $\|w_t\|_2 \to \infty$.
3. For sufficiently large t, w_t will separate the data (since the loss goes to 0).
4. As $z \to \infty$, $l(z) = \log(1 + \exp(-z)) \approx \exp(-z)$ (i.e. logistic loss is similar to exponential loss).
5. When $\|w\|_2 = q$ is big, the loss $\widehat{L}(w)$ mainly depends on supporting data $\{(x_i, y_i) : y_i \overline{w}^\top x_i = \gamma(w)\}$.

To see the last bullet point: letting $\overline{w} = w/\|w\|_2$, we notice that

$$\widehat{L} = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i w^\top x_i) \tag{8.91}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q y_i \overline{w}^\top x_i\right) \tag{8.92}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q y_i \overline{w}^\top x_i\right) 1\left[y_i \overline{w}^\top x_i = \gamma(w)\right] \tag{8.93}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q\gamma(w)\right) 1\left[y_i \overline{w}^\top x_i = \gamma(w)\right]. \tag{8.94}$$

Here the first approximation (8.92) is because of the logistic loss vs. exponential loss approximation, while the second approximation (8.93) is because for any data x_i, y_i that is not a support vector, i.e.

$$\overline{w}^\top x_i y_i \geq \gamma(w) + \epsilon, \tag{8.95}$$

for $\epsilon > 0$, then

$$\exp(-q y_i \overline{w}^\top x_i) \leq \exp(-q\gamma(w)) \exp(-q\epsilon), \tag{8.96}$$

and as $q \to \infty$ the term $\exp(-q\epsilon) \to 0$, making such terms negligible.

In conclusion, minimizing the (approximate) loss (8.94) is (informally) equivalent to maximizing the margin. (Note that if you examine (8.94), there are actually two ways to make the loss small: maximizing the margin or making q large. The rigorous proof shows that when q is large, the margin is already very close to the max margin. These are technical details that we will not concern ourselves with.)

8.4 Stochasticity in algorithmic regularization

Finally, we note that in general, when the loss has multiple global minima, any decisions we make about the optimization algorithm will make a difference. Another important source of algorithmic regularization (possibly the most important) comes from the *stochasticity* in the stochastic gradient descent (SGD) algorithm, where the parameters are optimized by updates of the form

$$\theta_{t+1} = \theta_t - \eta(\nabla \hat{L}(\theta_t) + \xi_t), \tag{8.97}$$

where ξ_t is a random noise term, typically with $\mathbb{E}[\xi_t] = 0$ so the noise will not affect the result too much. The variance of ξ_t can sometimes be time-dependent: for example, ξ_t could be dependent on the parameter θ_t.

[3] This result is still very limited it only works without regularization, and one needs to run gradient descent for a long time before this convergence in direction happens. Also, SVM is not always the best possible solution.

In practice, it turns out that larger gradient noise can lead to better generalization performance, as long as the algorithm can optimize under such level of noise. The intuition behind this phenomenon is that SGD converges to a "flat" global minimum, i.e. one with small curvature and small noise covariance. On the other hand, if you have a "sharp" local/global minimum with a large amount of noise, SGD will not converge to it stably. There are a number of works on this topic [HaoChen et al., 2020, Blanc et al., 2020], but a lot of questions in this space remained to be answered.

Chapter 9

Data-dependent generalization bounds

In this chapter, we discuss the Lipschitzness of models and why they seem to generalize better than arbitrary networks. To do so, we introduce a refined notion of uniform convergence that is data-dependent, and use it to derive a generalization bound for generalized margin. We end by introducing the *all-layer margin*, a specific instance of generalized margin that captures model Lipschitzness, thus allowing us to use the data-dependent generalization bound.

9.1 Lipschitzness of models and generalization

It has been found that the Lipschitzness of the model plays an important role in algorithmic regularization. As an illustration, note that the curvature (Hessian) of the loss function, the Lipschitzness of the model, and the noise level in SGD are all closely-related. To give a sense of the connections between them, suppose we have a model $f(x;\theta)$, a single example (x,y), and a loss function $L(\theta) = \ell(f(x),y) = (f(x)-y)^2$. In this setup, we have the decomposition

$$\nabla^2 L(\theta) = \underbrace{\frac{\partial^2 \ell}{\partial f^2}}_{\text{scalar}} \cdot \underbrace{\frac{\partial f}{\partial \theta}}_{\mathbb{R}^p} \cdot \underbrace{\frac{\partial f}{\partial \theta}^\top}_{\mathbb{R}^p} + \frac{\partial \ell}{\partial f} \cdot \underbrace{\frac{\partial^2 f}{\partial \theta^2}}_{\mathbb{R}^{p\times p}}. \tag{9.1}$$

This decomposition is useful because it has been found empirically that the second term is relatively small. This implies that the Hessian is somewhat dominated by the first term. The first term, especially $\frac{\partial f}{\partial \theta}$, relates to the Lipschitzness of the model with respect to the parameter. (There are similar connections between other quantities.)

Our algorithmic choices (e.g. SGD) seem to prefer Lipschitz models[1], which implies that such models generalize better. It remains to answer the question: **Why do Lipschitz models generalize better than arbitrary networks?** We want to theoretically analyze the relationship between Lipschitzness and generalization performance, and derive some generalization bounds w.r.t to the Lipschitzness of the models.

First, we note that the idea of using Lipschitzness to obtain generalization bounds is not new: it is the core of non-parametric statistics. However, such bounds suffer from the "curse of dimensionality", that is, the sample complexity grows exponentially as the data dimension d. Thus, using only Lipschitzness property is not enough to explain the generalization performance of neural networks: we need the help of parameterization.

[1] By this we mean the Lipschitz constant of the model is small. Also, we are not distinguishing between the Lipschitz constant w.r.t to the input and that w.r.t. to the parameter because they are actually related (not covered in the lecture).

Consider a deep neural network $f(x) = \sigma(W_r\sigma(W_{r-1}\cdots\sigma(W_1x)))$ for binary classification. Recall, [Bartlett et al., 2017] showed that:

$$L(\theta) \le \frac{R_S(\mathcal{F})}{\gamma}, \tag{9.2}$$

where γ is the margin of the model, and $R_S(\mathcal{F})$ is some complexity that satisfies

$$R_S(\mathcal{F}) \le \underbrace{\left(\prod_{i=1}^{r}\|W_i\|_{\text{op}}\right)}_{\text{relatively large}} \cdot \underbrace{\left(\sum_{i=1}^{r}\frac{\|W_i^T\|_{2,1}^{2/3}}{\|W_i\|_{\text{op}}^{2/3}}\right)^{3/2}}_{\text{relatively small}}. \tag{9.3}$$

The first term is essentially the upper bound on the Lipschitzness of the model w.r.t. to the input over the entire space.

The limitation of this bound is that if $\|W_i\|_{\text{op}} > 1$, then it grows exponentially in depth. On the other hand, if $\|W_i\|_{\text{op}} < 1$, then the $f_\theta(x)$ is exponentially small. Thus, it is very hard to make the spectral norm small while keeping the margin large. In the typical case, we have

$$\|W_1x\| \approx \|W_1\|_{\text{op}}\|x\|, \tag{9.4}$$

$$\|\sigma(W_1x)\| \approx \frac{1}{\sqrt{2}}\|W_1\|_{\text{op}}\|x\|. \tag{9.5}$$

(The second approximation comes from the heuristic that the ReLU function σ will zero out about half of the entries.) Thus, heuristically the output shrinks by a factor of $\sqrt{2}$ when passing through each layer. To make the output $f(x) \approx \Theta(1)$, we need $\|W_i\|_{\text{op}} \approx \sqrt{2}$, which makes the generalization bound very large.

The deeper cause to this problem is that $\prod_i \|W_i\|_{\text{op}}$ is a worst-case bound on the Lipschitzness of models, since it is data-independent and assumes that the input spans over the entire space. Thus one way to improve the bound is by replacing "worse-case Lipschitzness" with the Lipschitzness at the data points $x^{(1)}, \cdots, x^{(n)}$. This also allows us to estimate the Lipschitzness on the empirical data, and gives us a regularizer roughly in accordance with what SGD prefers.

We want to prove a bound of the form

$$L(w) \le \text{poly}(\text{Lipschitzness of } f_w \text{ on } x^{(1)}, \cdots, x^{(n)}, \text{norms of } W_i\text{'s}). \tag{9.6}$$

The RHS of (9.6) can be used as an explicit regularizer in model training to improve the generalization performance.

9.2 Proving data-dependent generalization bounds

Before we prove a bound in the of form (9.6), we first discuss why classical uniform convergence does not work. Note that the RHS of (9.6) is dependent on random variables $x^{(1)}, \cdots, x^{(n)}$. But typical bound of uniform convergence using Rademacher complexity is in the form

$$\forall f \in \mathcal{F}, \quad L(f) \le \text{comp}(\mathcal{F}, n), \tag{9.7}$$

where comp is some complexity measure, or in the form

$$\forall f \in \mathcal{F}, \quad L(f) \le \text{comp}(f, n). \tag{9.8}$$

The second bound can be achieved by defining $\mathcal{F}_C = \{f : \text{comp}(f, n) \le C\}$ first, applying the first type of bound for the class $\mathcal{F}_C$, then performing a union bound over C. However, this approach does not work for obtaining a bound like the RHS of (9.6) because the the corresponding hypothesis class is

$$\mathcal{F}_C = \left\{f : \text{comp}(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^n, n) \le C\right\}. \tag{9.9}$$

There are random variables in the definition of the hypothesis class, which is not allowed for Rademacher complexity. Hence, we cannot leverage such techniques directly.

To tackle this issue, we introduce a refined version of uniform convergence. Suppose we can decompose the complexity measure into the sum of a property related to each data point and the function we care about:

$$\text{comp}\left(f, \{x^{(i)}, y^{(i)}\}_{i=1}^n, n\right) = \sum_{i=1}^n g((x^{(i)}, y^{(i)}), f). \tag{9.10}$$

We can define the *augmented loss* as

$$\tilde{l}(f) = l(f) \cdot \mathbf{1}(g((x, y), f) \leq C). \tag{9.11}$$

This means that we are changing the loss function to include the data-dependent term. An intuitive example can be found in Figure 9.1, where we have an empirical loss with very bizarre behavior, but only outside the low complexity region that we really care about. The augmented loss notices this and "smooths out" the irregularities outside the low complexity region by ignoring those terms.

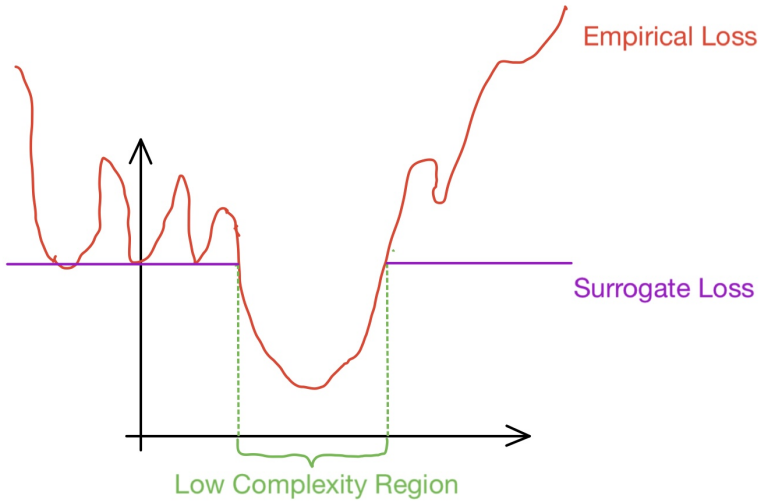

Figure 9.1: The empirical loss has bizarre behavior, but only outside the region of interest. The general idea is to define a surrogate loss and prove uniform convergence over the surrogate loss so as to avoid the bizarre behavior of the empirical loss.

The difficulty with taking this approach is that the low complexity region is random: if it was fixed, we could just zoom into that region and prove something directly by uniform convergence. We deal with this difficulty by defining a *surrogate loss* which it could just be constant outside of the low complexity region. We may then apply uniform convergence over the entire space.

We have talked about the notion of surrogate losses before in this class. For example, the margin loss/ramp loss is a type of surrogate loss. There, we thought of using the surrogate loss to make the zero-one loss more continuous. Here, we use the surrogate loss to avoid dealing with the loss function in "bad" regions. Let us define a generalized version of margin loss:

Definition 9.1 (Generalized margin)**.** Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a classification model. We call $g_f(x, y)$ a

generalized margin if $g_f(x, y)$ satisfies

$$g_f(x,y) = \begin{cases} 0 & \text{if } f(x)\cdot y \le 0 \quad \text{(wrong prediction)}, \\ > 0 & \text{if } f(x)\cdot y > 0. \end{cases} \tag{9.12}$$

Given this definition, we have the following lemma:

Lemma 9.2 (Generalization bound for general margin)**.** *Suppose g_f is a generalized margin. Let $G = \{g_f : f \in \mathcal{F}\}$, and assume we have an ϵ-covering of G under the $\|\cdot\|_\infty$ metric, $N_\infty(\epsilon, G)$, with $|N_\infty(\epsilon, G)| \le \lfloor R^2/\epsilon^2 \rfloor$, where R is the Rademacher complexity of the model.*

Then with probability larger than $1-\delta$ over the draw of training data, $\forall f \in \mathcal{F}$ that correctly predicts the labels on the training data, we have

$$L_{0\text{-}1}(f) \le \widetilde{O}\left(\frac{R}{\min_i g_f(x^{(i)}, y^{(i)})} \cdot \frac{1}{\sqrt{n}}\right) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{9.13}$$

The proof is similar to that for the bound we proved with margin loss earlier in the class, with only a few technical details changed.

9.3 All-layer margin

To use Lemma 9.2, we want to design a generalized margin $g_f(x, y)$ such that $G = \{g_f : f \in \mathcal{F}\}$ has low complexity. We want this margin to capture the Lipschitzness of the model so that the bound will not scale badly in the worst case. If we use the standard margin $g_f(x, y) = yf(x)$, then G depends on $\prod_i \|W_i\|_{\text{op}}$; our goal is to do something better than this. To do so, we want to somehow have the margin depend on Lipschitzness.

The *all-layer margin* [Wei and Ma, 2019] is one such margin. Consider a perturbed model, where $\delta = (\delta_1, ..., \delta_r)$ is the perturbation and the original neural network model is perturbed in the following way:

$$h_1(x, \delta) = \sigma(W_1 \cdot x) + \delta_1 \|x\|_2, \tag{9.14}$$

$$h_2(x, \delta) = \sigma(W_2 \cdot h_1(x, \delta)) + \delta_2 \|h_1(x, \delta)\|_2, \tag{9.15}$$

$$\vdots$$

$$f(x, \delta) = h_r(x, \delta) = \sigma(W_r \cdot h_{r-1}(x, \delta)) + \delta_r \|h_{r-1}(x, \delta)\|_2. \tag{9.16}$$

We can then define the margin of the model as

$$m_f(x, y) \triangleq \min_\delta \sqrt{\sum_{i=1}^r \|\delta_i\|^2} \quad \text{s.t. } f(x, \delta)y \le 0 \quad \text{(incorrect prediction)}. \tag{9.17}$$

(It can be proven that m_f is indeed a generalized margin.) Under this definition, $m_f(x, y)$ is large if $f(x)$ is large (i.e. correct) and f is robust to perturbation of example x. The good property is that under this definition, the margin already captures some Lipschitzness of the model. Applying Lemma 9.2 with m_f gives the the following theorem.

Theorem 9.3 (Generalization bound for all-layer margin)**.** *With probability larger than $1-\delta$ over the draw of training data,*

$$L_{0\text{-}1}(f) \le \widetilde{O}\left(\frac{\sum_{i=1}^r \|W_i\|_{1,1}}{\min_i m_f(x^{(i)}, y^{(i)})} \cdot \frac{1}{\sqrt{n}}\right) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right), \tag{9.18}$$

where $\|W\|_{1,1}$ is the sum of the absolute value of entries of W.

This theorem implies that a larger all-layer-margin implies better generalization. To get a larger all-layer-margin, we should make the network more robust to perturbation, i.e. more Lipschitz.

Proof. We present just the main proof ideas here. To use Lemma 9.2, it suffices to show that

$$N_\infty(\epsilon, G) \leq O\left(\frac{\sum_{i=1}^r \|W_i\|_{1,1}}{\epsilon^2}\right), \tag{9.19}$$

where $G = \{m_f : f \in \mathcal{F}\}$. Let $\mathcal{F}_1, \ldots, \mathcal{F}_r$ be a sequence of hypothesis classes (corresponding to each layer in the network), and let $\mathcal{F} = \{f_r \circ f_{r-1} \circ \cdots \circ f_1 : f_i \in \mathcal{F}_i\}$. [Wei and Ma, 2019] prove the following lemma:

Lemma 9.4 (Decomposition lemma)**.** *Let $m \circ \mathcal{F} = \{m_f : f \in \mathcal{F}\}$ denote the family of all-layer margins of function compositions in $\mathcal{F}$. Then*

$$\log N_\infty\left(\sqrt{\sum_{i=1}^r \epsilon_i^2}, m \circ \mathcal{F}\right) \leq \sum_i \log N_\infty(\epsilon_i, \mathcal{F}_i).$$

This reduces the problem to bounding the covering number for each layer which is much easier, since each layer is basically a linear transformation plus a non-linearity. □

Chapter 10

Online learning

In this chapter, we switch gears and talk about *online learning* and *online convex optimization*. The main idea driving online learning is that we move away from the assumption that the training and test data are both drawn i.i.d from some fixed distribution. In the online setting, training data and test data come to the user in an interwoven manner, and data can be generated *adversarially*. We will describe how online learning can be reduced to online convex optimization, some important algorithms, as well as applications of these algorithms to some illustrative examples.

10.1 Online learning setup

In classical supervised learning, we train the model with the assumption that $(x^{(i)}, y^{(i)}) \overset{i.i.d.}{\sim} P_{\text{train}}$, where P_{train} is the underlying distribution of the training data. In most cases, we assume the test data, i.e., the data we want our model to predict well, comes from the same distribution (or at least one that is close to P_{train}). Reality is often more complicated: data could indeed be generated in sequence, or even in an adversarial manner, so it is often the case that P_{test} differs from P_{train}. The situation where P_{test} and P_{train} are different is known as *domain shift*. There are some theories that tackle the issue of domain shift and generalization properties of transfer learning. However, the field is still largely being developed. (See [Ben-David et al., 2007], for example.)

Online learning is an attempt to deal with domain shift in a way that is agnostic to the relationship between the training and test data distributions (i.e. deal with "worst-case" domain shift). As an example, many recommendation systems today collect users' historical trace of shopping behavior, which are not i.i.d. samples, and makes adaptive recommendations based on users' changing shopping behavior. Hence, one can see that online learning attempts to adapt to the constantly evolving reality on time. Notice that unlike the "offline model" (i.e., classical supervised learning), online learning learns while testing, and hence there is no rigid division in time to differentiate training and testing phase.

Online learning has several distinctive features [Liang, 2016]:

1. The data may be *adversarial*. We cannot assume that sample is drawn independently from some distribution.

2. The data and predictions are *sequential*. At each step, the algorithm makes a prediction after given a single piece of data.

3. The feedback is *limited*. For example, in bandit problems, the algorithm only knows if its right or wrong, but no other feedback is given.

Online learning can be viewed as a game between two parties: (i) the learner/agent/algorithm/player, and (ii) the environment/nature. For simplicity, we will refer to the two parties as "learner" and "environment" in the remainder of this chapter.

The game takes place over T rounds or time steps. At each step $t = 1, \ldots, T$, the learner receives an input $x_t \in \mathcal{X}$ from the environment and makes a prediction $\hat{y} \in \mathcal{Y}$ in response. The learner then receives the label y_t from the environment and suffers some loss. This procedure is outlined in Algorithm 1 and is illustrated in Figure 10.1.

Algorithm 1: General online learning problem

for $t = 1, \ldots T$ **do**
  Learner receives $x_t \in \mathcal{X}$ from environment, which may be chosen adversarially;
  Learner predicts $\hat{y} \in \mathcal{Y}$;
  Learner receives the label y_t, from environment, which may be chosen adversarially; Learner suffers some loss $\ell(y_t, \hat{y}_t)$.

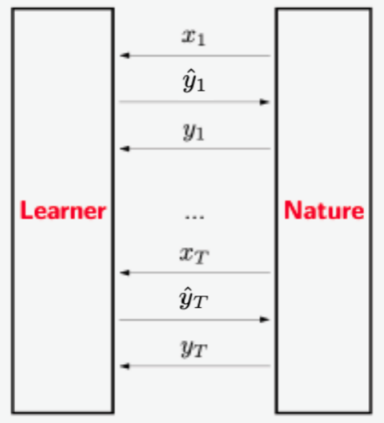

Figure 10.1: A representation of the online learning problem.

Later, we will see that the manner in which nature generates (x_t, y_t) leads to different types of online learning. In the most adversarial setting of online learning, it is possible that the "true label" y_t is not generated at the same time as x_t. The environment could generate the label y_t depending on the prediction $\hat{y}_t$ made by the learner. We can also see that Algorithm 1 is a very general framework as there are very few constraints on how x_t and y_t are generated.

10.1.1 Evaluation of the learner

Given this setup, a natural question to ask is how one can evaluate the performance of the learner. Intuitively, one could simply evaluate the learner's performance by computing the loss between the predicted label and the "true" label sent by the environment $\ell(y_t, \hat{y}_t)$. For the entire sequence of tasks, one can then evaluate in terms of the cumulative loss:

$$\sum_{t=1}^{T} \ell(y_t, \hat{y}_t). \tag{10.1}$$

However, as the environment can be adversarial, the task itself might be inherently hard and even the best possible learner fails to achieve a small loss. Hence, instead instead of using the cumulative loss for a learner by itself, we compare its performance against a suitable baseline, the "best model in hindsight". Assume that our learner comes from a set of hypotheses $\mathcal{H}$. Let us choose the hypothesis $h \in \mathcal{H}$ that

minimizes the cumulative loss, i.e.

$$h^\star = \operatorname*{argmin}_{h\in\mathcal{H}} \sum_{t=1}^{T} \ell(y_t, h(x_t)). \tag{10.2}$$

Note here that in minimizing the cumulative loss, the learner gets to see all the data points (x_t, y_t) at once. The cumulative loss of $h^\star$ is the best we can ever hope to do, and so it would be better to compare the cumulative loss of the learner against it. (This approach is analogous to "excess risk", which tells how far the current model is away from the best we could hope for.) This measurement is denoted as *regret*, and is formally defined as:

$$\text{Regret} \triangleq \left[\sum_{t=1}^{T} \ell(y_t, \hat{y}_t)\right] - \underbrace{\left[\min_{h\in\mathcal{H}} \sum_{t=1}^{T} \ell(y_t, h(x_t))\right]}_{\text{best loss in hindsight}} \tag{10.3}$$

Using this definition, if the best model in hindsight performs well, then the learner has more responsibility to learn to predict well in order to match up the performance of the baseline.

10.1.2 The realizable case

In general, if the environment is too powerful, leading the learner to a large loss, it will also hinder the best model in hindsight from doing well. On the other hand, there are settings where some members of the hypothesis class can actually do well. Such settings/problems are usually referred to as *realizable*:

Definition 10.1 (Realizable problem)**.** An online learning problem is *realizable* (for a family of predictors $\mathcal{H}$) if there exists $h \in \mathcal{H}$ such that for any T, $\sum_{t=1}^{T} \ell(y_t, h(x_t)) = 0$.

Note that even though zero error is possible, this is still an interesting problem to consider because the x_t's are not i.i.d. as they are in classical supervised learning. Hence, standard statistical learning theory does not apply, and there is still research to be done here.

Example 10.2. Consider a classification problem on (x_t, y_t), and for simplicity assume $y_t \in \{0, 1\}$. Suppose there exists $h^\star \in \mathcal{H}$ such that we always have $y_t = \hat{y}_t^\star = h^\star(x_t)$. In this case, the problem is realizable.

In this case, the learner can adopt a "majority algorithm". At each time, the learner maintains a set $V_t \subset \mathcal{H}$ so that $\sum_{t=1}^{T} \ell(y_t, h(x_t)) = 0$ for all $h \in V_t$, and $\hat{y}_t$ is simply the prediction made by the majority of $h \in V_t$. Based on the loss received, learners $h \in V_t$ that fail for time $t+1$ will be eliminated from future V_t's.

With this setup, we can see that for each wrong prediction made by the learner, at least half of the hypotheses $h \in V_t$ will be eliminated. Hence, $1 \le |V_{t+1}| \le |\mathcal{H}|2^{-M}$ where M is the number of mistakes made so far. Thus, one has $M \le \log|\mathcal{H}|$ by taking log on both sides of inequalities and rearrange.

Now, if one puts ℓ as the zero-one loss, the regret for this example will be

$$\text{Regret} = \sum_{t=1}^{T} \ell(y_t, h(x_t)) = M, \tag{10.4}$$

so in this example, one has regret $\le \log|\mathcal{H}|$, which is a non-trivial bound when $\mathcal{H}$ is finite.

As one can see in the example, the realizable case usually indicates that the problem is not too far out of reach. Indeed, for finite hypothesis classes and linear models, the realizable case is considered to be straightforward to solve. This is perhaps why most of the past literature has focused on non-realizable cases. However, the realizable case is still an interesting problem and perhaps a very good starting point when the model class is beyond linear models and when the loss function is no longer convex, because the x_t's are not i.i.d. as they are in classical supervised learning. Hence, standard statistical learning theory does not apply, and there is still research to be done here.

In the rest of the chapter, we will only focus on the convex loss case, where we reduce online learning to online convex optimization.

10.2 Online (convex) optimization (OCO)

Online convex optimization (OCO) is a particularly useful tool to get results for online learning. Many online learning problems (and many other types of problems!) can be reduced to OCO problems, which allow them to be solved and analyzed algorithmically. Algorithm 2 describes the OCO problem, which is more general than the online learning problem. (Note: *Online optimization (OO)* refers to Algorithm 2 except that the f_t's need not be convex. However, due to the difficulty in non-convex function optimization, most research has focused on OCO.)

Algorithm 2: Online (convex) optimization problem

for $t = 1, ..., T$ **do**
- The learner picks some action $w_t \in \Omega$ from the action space Ω;
- The environment picks a (convex) function $f_t : \Omega \to [0, 1]$;
- The learner suffers the loss $f_t(w_t)$ and observes the *entire* loss function $f_t(\cdot)$.

Essentially the learner is trying to minimize the function f_t at each step. As with online learning, one evaluates the performance of learner in online optimization setting using the regret:

$$\text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \underbrace{\min_{w \in \Omega} \sum_{t=1}^{T} f_t(w)}_{\text{best action in hindsight}} . \tag{10.5}$$

At some level, OCO seems like an impossible task, since we are trying to minimize a function f_t that we only get to see *after* we have made our prediction! This is certainly the case for $t = 1$. However, as time goes on, we see more and more functions and, if future functions are somewhat related to past functions, we have more information to make better predictions. (And if the future functions are completely unrelated or contradictory to past functions, then the best action in hindsight would also be bad and therefore our algorithm does not have to do much.)

10.2.1 Settings and variants of OCO

There are multiple settings of the OCO network, which can vary the power of the environment and observations.

- Stochastic setting: $f_1, ..., f_T$ are i.i.d samples from some distribution P. This corresponds to (x_t, y_t) being i.i.d. in online learning. Under this setting, the environment is not adversarial.
- Oblivious setting: $f_1, ..., f_T$ are chosen arbitrarily but before the game starts. This corresponds to (x_t, y_t being chosen before the game starts. In this setting, the environment can be adversarial but cannot be adaptive. The environment can choose these functions based on the learner's algorithm, but not the actual action if the learner's algorithm contains randomness. (This is the setting that we focus on in this course.)
- Non-oblivious/adaptive setting: For all t, f_t can depend on the learner's actions $w_1, ...w_t$. Under this setting, the environment can be adversarial and adaptive. This is the most challenging setting because the environment is powerful enough to know not only the strategy of the learner, but also the exact choice the learner finally made. (Note however that If the learner is deterministic, the environment does not have more power here than in the oblivious setting. The oblivious adversary can simulate the game before the game starts, and chose the most adversarial input accordingly.)

10.3 Reducing online learning to online optimization

There is a natural way to reduce the online learning problem to online optimization, with respect to a specific type of model h_w parametrized by $w \in \Omega$. Recall that in online learning problem, the learner predicts y_t upon receiving x_t. If the learner possesses oracle to solve online optimization problem, the learner can consult the oracle to obtain w_t, the parameter of the model as in online optimization problem, and then predict $\hat{y}_t = h_{w_t}(x_t)$.

In the next two subsections, we give two examples of how an online learning problem can be reduced to an OCO problem.

10.3.1 Example: Online learning regression problem

Consider the regression model $h_w(x) = w^\top x$ parameterized by w in parameter space Ω with squared error loss ℓ. Here is the online learning formulation of the regression problem:

Algorithm 3: Online learning regression problem

for $t = 1, ..., T$ **do**
- The learner receives $x_t \in \mathbb{R}^d$ from the environment;
- The learner predicts $\hat{y}_t$;
- The environment selects y_t and sends it to the learner;
- The learner suffers loss $\ell(y_t, \hat{y}_t) = (y_t - \hat{y}_t)^2$.

This can be reduced to the OCO problem in the following way:

Algorithm 4: OCO formulation of regression problem

for $t = 1, ..., T$ **do**
- The learner receives $x_t \in \mathbb{R}^d$ from the environment;
- The learner gives x_t to the OCO solver and obtains $w_t \in \mathbb{R}^d$;
- The learner predicts $\hat{y}_t = h_{w_t}(x_t) = w_t^\top x_t$;
- The environment selects y_t and sends it to the learner;
- The learner suffers loss $(y_t - h_{w_t}(x_t))^2$;
- With (x_t, y_t) observed, the learner can reconstruct the loss function $f_t(w) = (y_t - h_w(x_t))^2$ and give it to the OCO solver.

In this example, we have the following correspondence:

- f_t in online optimization $\leftrightarrow$ squared error loss functions for (x_t, y_t).
- w_t in online optimization $\leftrightarrow$ parameters of the linear model h_{w_t}.

Since $h_w(\cdot)$ is linear, the corresponding squared error loss function f_t are convex, and so we have effectively reduced the online linear regression problem to an online *convex* optimization problem.

Notice that in the previous example, the loss function f_t actually depends on the label y_t, which demonstrates that the key challenge in online optimization is that the function f_t is unknown to the learner when the prediction $\hat{y}_t$ is made.

10.3.2 Example: The expert problem

Suppose we wish to predict tomorrow's weather and 10 different TV channels provide different forecasts. Which one should we follow? Formally, consider a finite hypothesis class $\mathcal{H}$, where each $h \in \mathcal{H}$ represents an expert, and we wish to choose a h_t wisely at each time step. For simplicity, we assume the prediction is

binary, i.e. $\hat{y} \in \{0, 1\}$, and suppose the loss function is 0-1 loss. (The problem can easily be generalized to more general predictions and losses.) The problem is outlined in Algorithm 5.

Algorithm 5: The expert problem

1 **for** $t = 1, ..., T$ **do**
2 The learner obtains predictions from N experts;
3 The learner chooses to follow prediction of one of the experts $i_t \in [N]$;
4 The environment gives the learner the true value. The learner is thus able to learn the loss of each of the experts: $\ell_t \in \{0, 1\}^N$;
5 The learner suffers the loss of the expert which was chosen: $\ell_t(i_t)$.

We want to design a method that chooses i_t for each step (line 3 in Algorithm 5) to minimize the regret:

$$\text{Regret} \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \underbrace{\min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i)}_{\text{the best expert in hindsight}}\right], \tag{10.6}$$

where the expected value is over i_t, thus covering the case where the i_t's could be random.

To make the expert problem amenable to reduction to OCO, we introduce idea of a *continuous action space*. Instead of choosing i_t from $\Omega = [N]$, the learner chooses a distribution p_t from the N-dimensional simplex $\Delta(N) = \left\{p \in \mathbb{R}^N : \|p\|_1 = 1, p \geq 0\right\}$. The learner then samples $i_t \sim p_t$. With this formulation, instead of selecting particular expert i_t to follow, the learner adjusts the belief p_t, and samples from the distribution to choose which expert to follow. Algorithm 6 outlines this procedure. Note that the loss is the expected loss $\mathbb{E}_{i \sim p_t}[\ell_t(i)]$ instead of the sampled $\ell_t(i_t)$.

Algorithm 6: The expert problem with continuous action

1 **for** $t = 1, ..., T$ **do**
2 The learner obtains predictions from N experts;
3 The learner chooses a distribution $p_t \in \Delta(N)$;
4 The learner samples one expert $i_t \sim p_t$;
5 The environment gives the learner the true value and the loss/error of all experts: $\ell_t \in \{0, 1\}^N$;
6 The learner suffers expected loss $\sum_{i \in [N]} p_t(i)\ell_t(i) = \langle p_t, \ell_t \rangle$;

With the continuous action space, it is easy to reduce the expert problem to an OCO: see Algorithm 7. (The problem is convex since the loss function is convex and the parameter space $\Delta(N)$ is convex.)

Algorithm 7: The expert problem

1 **for** $t = 1, ..., T$ **do**
2 The learner obtains predictions from N experts;
3 The learner invokes the OCO oracle to obtain $p_t \in \Delta(N)$;
4 The learner chooses to follow prediction of one of the experts $i_t \in [N]$;
5 The environment gives the learner the true value. The learner is thus able to learn the loss of each of the experts: $\ell_t \in \{0, 1\}^N$;
6 The learner suffers the loss of the expert which was chosen: $\ell_t(i_t)$. The learner can reconstruct the loss function $f_t(p) = \langle p, \ell_t \rangle$ and give it to the OCO oracle.

In this setting, one can rewrite the regret as:

$$\text{Regret} = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i) \tag{10.7}$$

$$= \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{p \in \Delta(N)} \sum_{t=1}^{T} \langle p, \ell_t \rangle \tag{10.8}$$

$$= \sum_{t=1}^{T} f_t(p_t) - \min_{p \in \Delta(N)} \sum_{t=1}^{T} f_t(p). \tag{10.9}$$

We obtain (10.8) because

$$\sum_{t=1}^{T} \langle p, \ell_t \rangle = \left\langle p, \sum_{t=1}^{T} \ell_t \right\rangle \geq \min_{i \in [N]} \left[\sum_{t=1}^{T} \ell_t(i) \right], \tag{10.10}$$

with equality for the probability distribution $p(i) = 1$ when $i = \text{argmin}_i \left[\sum_{t=1}^{T} \ell_t(i) \right]$ and $p(i) = 0$ otherwise, and (10.9) is by definition of f_t.

10.4 Reducing online learning to batch learning

In this section, we present a reduction from online learning to standard supervised learning problem, also known as the "batch problem" in this literature.

As in the standard supervised learning setting, consider an i.i.d dataset $\{(x_t, y_t)\}_{t=1}^{T}$ and some parameter w. Let $L(w)$ and $\widehat{L}(w)$ be the population loss and empirical loss respectively. For simplicity, assume $|\ell((x_i, y_i), w)| \leq 1$. The theorem below establishes a link between the regret obtained in online learning and the excess risk obtained in the batch setting.

Theorem 10.3 (Relationship between excess risk and regret)**.** *Assume $\ell((x, y), w)$ is convex. Suppose we run an online learning algorithm on the dataset $\{(x_i, y_i)\}_{i=1}^{T}$ and obtain a sequence of models $w_1, \ldots, w_T$, and regret R_T. Let $\overline{w} = \frac{1}{T} \sum_{i=1}^{T} w_i$, then the excess risk of $\overline{w}$ can be bounded above:*

$$L(\overline{w}) - L(w^\star) \leq \frac{R_T}{T} + \widetilde{O}\left(\frac{1}{\sqrt{T}} \right), \tag{10.11}$$

where $w^\star = \text{argmin}_{w \in \Omega} L(w)$.

Here are some intuitive interpretations of the theorem:

- If $R_T = O(T)$, then we have some non-trivial result. Otherwise, the bound in (10.11) is increasing T and does not provide any useful information.
- If the batch problem has a $1/\sqrt{T}$ generalization bound, then the best you can hope for in online learning is $R_T = O(\sqrt{T})$.
- If the batch problem has a $1/T$ generalization bound, you can hope for $O(1)$ regret (or $\widetilde{O}(1)$ regret in some cases).
- We often have $O(\sqrt{T})$ excess risk supervised learning problems; hence it is reasonable to expect $O(\sqrt{T})$ regret in online learning problems.

10.5 Follow-the-Leader (FTL) algorithm

In this section, we analyze an algorithm called "Follow-the-Leader" (FTL) for OCO, which is intuitive but fails to perform well in many cases.

The FTL algorithm behaves as its name suggests: it always selects the action w_t such that it minimizes the historical loss the learner has seen so far, i.e.

$$w_t = \operatorname*{argmin}_{w \in \Omega} \sum_{i=1}^{t-1} f_i(w). \tag{10.12}$$

We now demonstrate how the FTL algorithm can fail for the expert problem. In the expert problem, $f_t(p) = \langle p, \ell_t \rangle$, so

$$p_t = \operatorname*{argmin}_{p \in \Delta(N)} \sum_{i=1}^{t-1} f_i(p) \tag{10.13}$$

$$= \operatorname*{argmin}_{p \in \Delta(N)} \sum_{i=1}^{t-1} \langle \ell_i, p \rangle \tag{10.14}$$

$$= \operatorname*{argmin}_{p \in \Delta(N)} \left\langle \sum_{i=1}^{t-1} \ell_i, p \right\rangle. \tag{10.15}$$

The minimizer $p \in \Delta(N)$ is a point-mass probability, with the point mass at the smallest coordinate of $\sum_{i=1}^{t-1} \ell_i$. This gives regret

$$\text{Regret} = \sum_{i=1}^{t-1} \ell_i(i_t), \quad \text{where } i_t = \operatorname*{argmin}_{j \in [N]} \sum_{i=1}^{t-1} \ell_i(j). \tag{10.16}$$

Now, consider the following example: suppose we have only two experts. Suppose expert 1 makes perfect predictions on even days while expert 2 makes perfect predictions on odd days. Assume also that the FTL algorithm chooses expert 1 to break ties (this is not an important point but makes the exposition simpler.) In this setting, the FTL algorithm always selects the *wrong* expert to follow. A few rounds of simulation of this example is shown in Table 10.1.

Table 10.1: An example where FTL fails

Day	1	2	3	4	...	...
Expert 1's loss	1	0	1	0	...	...
Expert 2's loss	0	1	0	1	...	...
FTL choice i_t	1	2	1	2	1	...

The best expert in hindsight has a loss of $T/2$ (choosing either expert all the time incurs this loss, and so the regret of the FTL algorithm is $T - T/2 = T/2 = \Theta(T)$. The main reason for FTL's failure is that is a deterministic algorithm driven by an extreme update, with no consideration on potential domain shift (it always selects the best expert based on the past with no consideration of the potential next f_t). Knowing its deterministic strategy, the environment can easily play in an adversarial manner. To perform better in a problem like this, we need some randomness to hedge risk.

10.6 Be-the-leader (BTL) algorithm

A better strategy is called *"Be the Leader" (BTL).* At time t, the BTL strategy chooses the action that would have performed best on $f_1, \cdots, f_{t-1}$ *and* f_t. In other words, the BTL action at time t is w_{t+1}, as

defined for the FTL algorithm. Note that this is an "illegal" choice for the action because w_{t+1} depends on f_t: in online convex optimization, the action at time t is required to be chosen *before* seeing the function f_t. Nevertheless, we can still gain some useful insights by analyzing this procedure. In particular, the following lemma shows that the BTL strategy is worth emulating because it achieves very good regret.

Lemma 10.4. *The BTL strategy has non-positive regret. That, is, if w_t is defined as in the FTL algorithm, then*

$$\textit{BTL regret} = \sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq 0, \tag{10.17}$$

for any T and any sequence of functions $f_1, \cdots, f_T$.

Proof. We prove the lemma by induction on T. (10.17) holds trivially for $T = 1$. Suppose that (10.17) holds for all $t \leq T-1$ and any $f_1, \cdots, f_{T-1}$. Now we wish to extend (10.17) to time $t = T$. Let f_T be any function. Since $w_{T+1} = \operatorname{argmin}_w \sum_{t=1}^{T} f_t(w)$, we can write:

$$\sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) = \sum_{t=1}^{T} f_t(w_{t+1}) - \sum_{t=1}^{T} f_t(w_{T+1}) \tag{10.18}$$

$$= \sum_{t=1}^{T-1} f_t(w_{t+1}) - \sum_{t=1}^{T-1} f_t(w_{T+1}) \qquad \text{(final summands cancel)} \tag{10.19}$$

$$\leq \sum_{t=1}^{T-1} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T-1} f_t(w) \tag{10.20}$$

$$\leq 0. \qquad \text{(induction hypothesis)} \tag{10.21}$$

□

A useful consequence of this lemma is a regret bound for the FTL strategy.

Lemma 10.5. (FTL regret bound) *Again, let w_t be as in the FTL algorithm. The FTL strategy has the regret guarantee*

$$\textit{FTL regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})]. \tag{10.22}$$

Proof.

$$\text{FTL regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \tag{10.23}$$

$$= \sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) + \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})] \tag{10.24}$$

$$\leq 0 + \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})], \tag{10.25}$$

where the last inequality is due to (10.17).

□

Lemma 10.5 tells us that if terms $f_t(w_t) - f_t(w_{t+1})$ are small (e.g. w_t does not change much from round to round), then the FTL strategy can have small regret. It suggests that the player should adopt a *stable* policy, i.e. one where the terms $f_t(w_t) - f_t(w_{t+1})$ are small. It turns out that following this intuition will lead to a strategy that improves the regret all the way to $O(\sqrt{T})$ in certain cases.

10.7 Follow-the-regularized-leader (FTRL) strategy

Now, we discuss a OCO strategy aims to improve the stability of FTL by controlling the differences $f_t(w_t) - f_t(w_{t+1})$. To describe the method, we will first need a preliminary definition.

Definition 10.6. We say that a differentiable function $\phi : \Omega \mapsto \mathbb{R}$ is *α-strongly-convex* with respect to the norm $||\cdot||$ on Ω if we have

$$\phi(x) \geq \phi(y) + \langle \nabla f(y), x - y \rangle + \frac{\alpha}{2}\|x - y\|^2 \tag{10.26}$$

for any $x, y \in \Omega$.

Remark 10.7. If ϕ is convex, then we know that $f(x)$ has a linear lower bound $\phi(y) + \langle \nabla f(y), x - y \rangle$. Being α-strong-convex means that $f(x)$ has a quadratic lower bound, the RHS of (10.26). This quadratic lower bound is very useful in proving theorems in optimization.

Remark 10.8. If $\nabla^2 f(y) \succeq \alpha I$ for all y, then f is α-strongly-convex. This follows directly from writing the second-order Taylor expansion of f around y.

Given a 1-strongly-convex function $\phi(\cdot)$, which we call a *regularizer*, we can implement the *"Follow the Regularized Leader" (FTRL)* strategy. At time t, this strategy chooses the action

$$w_t = \operatorname*{argmin}_{w \in \Omega} \left[\sum_{i=1}^{t-1} f_i(w) + \frac{1}{\eta}\phi(w) \right], \tag{10.27}$$

where $\eta > 0$ is a tuning parameter that we will tune later.

10.7.1 Regularization and stability

To understand why we might use the FTRL policy, we first establish that it achieves the intended goal of controlling the differences $f_t(w_t) - f_t(w_{t+1})$. Actually, we will show a more general result that adding a regularizer induces stability for any convex objective.

Lemma 10.9. (Regularizers induce stability) *Let F and f be functions taking Ω into $\mathbb{R}$, and assume that F is α-strongly-convex with respect to the norm $\|\cdot\|$ and that f is convex. Let $w = \operatorname{argmin}_{z \in \Omega} F(z)$ and $w' = \operatorname{argmin}_{z \in \Omega}[f(z) + F(z)]$. Then*

$$0 \leq f(w) - f(w') \leq \frac{1}{\alpha}\|\nabla f(w)\|_*^2, \tag{10.28}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

Proof. By strong convexity,

$$F(w') - F(w) \geq \langle \nabla F(w), w' - w \rangle + \frac{\alpha}{2}\|w - w'\|^2 \tag{10.29}$$

$$\geq \frac{\alpha}{2}\|w - w'\|^2, \tag{10.30}$$

where in the second step we used the fact that the KKT optimality conditions for w imply $\langle \nabla F(w), w' - w \rangle \geq 0$. (Informally, if $\Omega = \mathbb{R}^d$, then $\nabla F(w) = 0$ as w minimizes F. If Ω is a convex subset of $\mathbb{R}^d$, then the gradient $\nabla F(w)$ must be perpendicular to the tangent to Ω at w; otherwise, we could move in the direction of the negative gradient and project back to the set Ω to lower the value of F.) Since $F + f$ is also α-strongly convex, exactly the same argument implies:

$$[F(w) + f(w)] - [F(w') + f(w')] \geq \frac{\alpha}{2}\|w - w'\|^2. \tag{10.31}$$

Adding these two inequalities gives

$$f(w) - f(w') \geq \alpha \|w - w'\|^2. \tag{10.32}$$

Since this lower bound is clearly positive, this shows $0 \leq f(w) - f(w')$.

Next, we prove the upper bound on $f(w) - f(w')$. Rearranging the inequality (10.32), we obtain

$$\|w - w'\| \leq \sqrt{\frac{1}{\alpha}[f(w) - f(w')]}. \tag{10.33}$$

Since f is convex, we have $f(w') \geq f(w) + \langle \nabla f(w), w' - w \rangle$. Rearranging this gives

$$\begin{aligned} f(w) - f(w') &\leq \langle \nabla f(w), w - w' \rangle \\ &\leq \|\nabla f(w)\|_* \cdot \|w - w'\| && \text{(by Cauchy-Schwarz)} \\ &\leq \|\nabla f(w)\|_* \sqrt{\frac{1}{\alpha}[f(w) - f(w')]}. && \text{(by (10.33))} \end{aligned}$$

Since $f(w) - f(w') \geq 0$, we can square both sides of this inequality to conclude that

$$[f(w) - f(w')]^2 \leq ||\nabla f(w)||_*^2 \frac{1}{\alpha}[f(w) - f(w')]. \tag{10.34}$$

Dividing both sides of this expression by $f(w) - f(w')$ gives the desired upper bound. □

Remark 10.10. Consider the special case where $\nabla f(w) = 0$. In this situation, w is the minimizer of both F and f, and hence is the minimizer of $F + f$. This implies that $w = w'$, and the inequalities in (10.28) become equalities.

10.7.2 Regret of FTRL

We are now ready to prove a regret bound for the FTRL procedure, based on the idea that strongly convex regularizers induce stability.

Theorem 10.11. (Regret of FTRL) *Let ϕ be a 1-strongly-convex regularizer with respect to the norm $\|\cdot\|$ on Ω. Then the FTRL algorithm (10.27) satisfies the regret guarantee*

$$\textit{FTRL regret} = \sum_{t=1}^{T} f_t(w_t) - \operatorname*{argmin}_{w \in \Omega} \sum_{t=1}^{T} f_t(w) \leq \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2, \tag{10.35}$$

where $D = \max_{w \in \Omega} \phi(w) - \min_{w \in \Omega} \phi(w)$.

Remark 10.12. Suppose that for all t and w, we have the uniform bound $||\nabla f_t(w)||_* \leq G$. Then Theorem 10.11 implies that the regret is upper bounded by $D/\eta + \eta G T$. Optimizing this upper bound over η by taking $\eta = \sqrt{\frac{D}{TG^2}}$ gives the guarantee

$$\text{FTRL regret} \leq 2\sqrt{DG} \times \sqrt{T}. \tag{10.36}$$

In other words, optimally-tuned FTRL can achieve $O(\sqrt{T})$ regret in many cases.

Proof. For convenience, define $f_0(w) = \phi(w)/\eta$. Then the FTRL policy can be written as

$$w_t = \operatorname*{argmin}_{w \in \Omega} \sum_{i=0}^{t-1} f_i(w), \tag{10.37}$$

i.e. FTRL is just FTL with an additional "round" of play at time zero. Thus, by Lemma 10.5 with time starting from $t = 0$, we have

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \le \sum_{t=0}^{T} [f_t(w_t) - f_t(w_{t+1})]. \tag{10.38}$$

For any $t \ge 1$, applying Lemma 10.9 with $F(w) = \sum_{i=0}^{t-1} f_i(w)$ (which is $1/\eta$-strongly-convex) and $f(w) = f_t(w)$ gives the bound $f_t(w_t) - f_t(w_{t+1}) \le \eta ||\nabla f_t(w_t)||_*^2$. Plugging this into the preceding display gives the upper bound:

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \le f_0(w_0) - f_0(w_1) + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2. \tag{10.39}$$

Next, we need to relate the LHS of the above display (which starts at time $t = 0$) to the actual regret of FTRL (which starts at time $t = 1$). To do this, define $w^* = \operatorname{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w)$. Then,

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \ge \sum_{t=0}^{T} f_t(w_t) - \sum_{t=0}^{T} f_t(w^*) \tag{10.40}$$

$$= f_0(w_0) - f_0(w^*) + \underbrace{\left(\sum_{t=1}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \right)}_{\text{Regret of FTRL}}. \tag{10.41}$$

Combining this inequality with (10.39) gives

$$\text{Regret of FTRL} \le f_0(w_0) - f_0(w_1) + f_0(w^*) - f_0(w_0) + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2 \tag{10.42}$$

$$= \frac{\phi(w^*) - \phi(w_1)}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2 \tag{10.43}$$

$$\le \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2. \tag{10.44}$$

This concludes the proof of the theorem. □

10.7.3 Applying FTRL to online linear regression

We apply the FTRL algorithm to a concrete machine learning problem. Let $\Omega = \{\omega : \|w\|_2 \le 1\}$, and let $f_t(\omega) = \frac{1}{2}(y_t - \omega^\top x_t)^2$ for some observation pair (x_t, y_t) satisfying $\|x_t\|_2 \le 1$ and $|y_t| \le 1$. This corresponds to a problem where we are trying to make accurate predictions using a linear model, but we do not assume any structure on the observation sequence (x_t, y_t) beyond boundedness.

Consider using FTRL in this problem with a ridge regularizer, $\phi(\omega) = \frac{1}{2}\|w\|_2^2$. One can check that ϕ is 1-strongly-convex with respect to the ℓ_2-norm, and also that $D = \max_{\omega\in\Omega} \phi(\omega) - \min_{\omega\in\Omega} \phi(\omega) = \frac{1}{2}$. Moreover, for all t and w we have

$$\nabla f_t(w) = -(y_t - w^\top x_t) x_t, \tag{10.45}$$

$$\|\nabla f_t(w)\|_2 \le |y_t - w^\top x_t| \cdot \|x_t\|_2 \tag{10.46}$$

$$\le 2 \cdot 1 = 2. \tag{10.47}$$

Therefore, by choosing $\eta = \sqrt{1/(8T)}$ and applying the FTRL regret theorem (Theorem 10.11), we can obtain the regret guarantee

$$\sum_{t=1}^{T}(y_t - w_t^\top x_t)^2 - \min_{||w||_2 \le 1} \sum_{t=1}^{T}(y_t - w^\top x_t)^2 \le 4\sqrt{T}. \tag{10.48}$$

10.7.4 Applying FTRL to the expert problem

For the expert problem, recall that the action space is $\Delta(N)$ and $f_t = \langle \ell_t, p \rangle$, where $\ell_t \in [0,1]^N$. As a first attempt at applying FTRL to this problem, we set $\phi(p) = \frac{1}{2}\|p\|_2^2$. With this choice,

$$D = \max_{p \in \Delta(N)} \phi(p) - \min_{p \in \Delta(N)} \phi(p) \tag{10.49}$$

$$\le \max_{p \in \Delta(N)} \frac{1}{2}\|p\|_2^2 \tag{10.50}$$

$$\le \max_{p \in \Delta(N)} \frac{1}{2}\|p\|_1^2 \tag{10.51}$$

$$= \frac{1}{2}. \tag{10.52}$$

Also,

$$\|\nabla f_t\|_2 = \|\ell_t\|_2 \le \sqrt{N}. \tag{10.53}$$

Thus, the regret bound is $O(G\sqrt{DT}) = O(\sqrt{NT})$. This is optimal dependency on T, but not good dependency on N.

Next, we show that if we change our regularization, we can get a better regret guarantee which is logarithmic in N, i.e., the regret is $O(\sqrt{(logN) \cdot T})$. The new regularizer we choose is the *(negative) entropy regularizer*:

$$\phi(p) = -H(p) = \sum_{j=1}^{N} p(j) \log p(j), \tag{10.54}$$

where $p \in \Delta(N)$ is in the set of distributions over $[N]$. We first introduce the following nice property of this regularizer:

Lemma 10.13. *$\phi(p)$ defined above is 1-strongly convex with respective to the ℓ_1 norm $\|\cdot\|_1$.*

Proof. By definition of strong convexity, we need to show that for all $p, q \in \Delta(N)$,

$$\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q \rangle \ge \frac{1}{2}\|p - q\|_1^2. \tag{10.55}$$

From direct computation, we know the gradient of $\phi(q)$ is

$$\nabla\phi(q) = \begin{bmatrix} 1 + \log q(1) \\ \cdots \\ 1 + \log q(N) \end{bmatrix}. \tag{10.56}$$

Plugging this into the LHS of (10.55), we get

$$\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle \tag{10.57}$$

$$= \sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} q(j)\log q(j) - \sum_{j=1}^{N} (1 + \log q(j))\,(p(j) - q(j)) \tag{10.58}$$

$$= \sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} p(j)\log q(j) - \sum_{j=1}^{N} (p(j) - q(j)) \tag{10.59}$$

$$= \sum_{j=1}^{N} p(j)\log \frac{p(j)}{q(j)} \tag{10.60}$$

$$= KL(p||q), \tag{10.61}$$

where $KL(p||q)$ is the KL-divergence between p and q. (We used the fact that $\sum_{j=1}^{N} p(j) = \sum_{j=1}^{N} q(j) = 1$ to get (10.60).) Finally, we finish the proof by applying Pinsker's inequality: $KL(p||q) \geq \frac{1}{2}\|p - q\|_1^2$.

□

Hence, ϕ is a satisfies the condition on the regularizer for our FTRL regret guarantee. To obtain the regret bound (10.36), we also need to bound $D = \sup\phi(p) - \inf\phi(p)$ and $G = \sup\|\nabla f_t(w)\|_\infty$ (since $\|\cdot\|_\infty$ is the dual norm of $\|\cdot\|_1$). Since negative entropy is always non-positive and (positive) entropy is always bounded above by $\log N$, we bound D with

$$D = \sup\phi(p) - \inf\phi(p) \leq -\inf\phi(p) = -\inf(-H(p)) = \sup(H(p)) \leq \log N, \tag{10.62}$$

and we bound G with

$$G = \|\nabla f_t(w)\|_\infty = \|l_t\|_\infty \leq 1. \tag{10.63}$$

Plugging these two into the regret bound (10.36) we get bound $O(\sqrt{(\log N)\cdot T})$.

Thus far, we have looked at FTRL and the expert problem abstractly: at each time t we choose action p_t based on the update

$$p_t = \operatorname*{argmin}_{p\in\Delta(N)} \sum_{i=1}^{t+1} f_t(p) - \frac{1}{\eta}H(p). \tag{10.64}$$

Can we get an exact analytical solution for p_t? Since we are minimizing a convex function, we can call some off-the-shelf convex optimization algorithm to solve this at each step. Another way is to write down the KKT conditions and solve that set of equations. Instead, we will show that there exists much simpler ways to solve this update. In particular, we will be using the *Gibbs variational principle*, which is essentially the KKT conditions under the hood.

Lemma 10.14 (Gibbs variational principle)**.** *Let ν, μ be probability distributions on $[N]$. Then*

$$\sup_{\nu} \left(\mathbb{E}_\nu[f] - KL(\nu||\mu)\right) = \log \mathbb{E}_\mu\left[e^f\right], \tag{10.65}$$

where $\mathbb{E}_\nu[f] = \mathbb{E}_{x\sim\nu}[f(x)] = \langle v, f\rangle$ and $\mathbb{E}_\mu\left[e^f\right] = \mathbb{E}_{x\sim\mu}\left[e^{f(x)}\right]$. Moreover, the optimal solution is attained at

$$\nu(x) \propto \mu(x)\cdot e^{f(x)}. \tag{10.66}$$

Intuitively, Lemma 10.14 says that taking the supremum over distributions μ of a linear function plus the KL divergence as the regularizer will give us the same distribution as exponentiating f.

If we take μ to be the uniform distribution on $[N]$ and replace f with $-f$ in Lemma 10.14, we get the following corollary:

Corollary 10.15. Let ν be a probability distribution. Then, $\mathbb{E}_\nu[f] - H(\nu)$ is minimized at $\nu(x) \propto e^{-f(x)}$.

Proof. When μ is uniform distribution, we have

$$KL(\nu||\mu) = \sum_x \nu(x) \log \frac{\nu(x)}{\mu(x)} \tag{10.67}$$

$$= \log N - \sum_x \nu(x) \log \frac{1}{\nu(x)} \tag{10.68}$$

$$= \log N - H(\nu). \tag{10.69}$$

So $\sup_\nu \left(\mathbb{E}_\nu[-f] - KL(\nu||\mu)\right) = -\inf_\nu \left(\mathbb{E}_\nu[f] - H(\nu) + \log N\right)$. This means that the value of ν that attains the infimum of $\mathbb{E}_\nu[f] - H(\nu)$ is the same ν attaining the supremum of $\mathbb{E}_\nu[-f] - KL(\nu||\mu)$, which by Lemma 10.14 is proportional to $e^{-f(x)}$. □

We now apply the Gibbs variational principle to the expert problem. Notice that our FTRL update for the expert problem at time t can be written as

$$\operatorname*{argmin}_{p_t \in \Delta(N)} \left\langle \sum_{i=1}^{t-1} l_i, p_t \right\rangle - \frac{1}{\eta} H(p_t) = \operatorname*{argmin}_{p_t \in \Delta(N)} \left\langle \eta \sum_{i=1}^{t-1} l_i, p_t \right\rangle - H(p_t), \tag{10.70}$$

where l_i is the vector of expert losses at time i. Letting $f = \eta \sum_{i=1}^{t-1} l_i$, we know from Corollary 10.15 that the minimizer is attained at $p_t \propto \exp\left(-\eta \sum_{i=1}^{t-1} l_i\right)$, or equivalently,

$$p_t(j) = \frac{\exp(-\eta L_t(j))}{\sum_{k=1}^N \exp(-\eta L_t(k))}, \tag{10.71}$$

where $L_t = \sum_{i=1}^{t-1} l_i$ is the cumulative loss vector. Basically, solving the expert problem is to look a the historical loss of each expert and take softmax to find the probability distribution of how much to trust each expert.

This algorithm is also called the "Multiplicative Weights Update", which has been studied before online learning framework became popular [Arora et al., 2005, Freund and Schapire, 1997, Littlestone and Warmuth, 1994]. One way of doing multiplicative weights update is the following: Let $\tilde{p}_t$ be the unnormalized distribution that we keep track of. At each time step t, for each expert j, we look at $l_{t-1}(j)$. if $l_{t-1}(j) = 1$, i.e. the expert made a mistake at the previous time step, we update $\tilde{p}_t(j) = \tilde{p}_{t-1}(j) \cdot \exp(-\eta)$; otherwise we make no change. We then get a distribution by normalizing $\tilde{p}_t$:

$$p_t = \frac{\tilde{p}_t}{\|\tilde{p}_t\|_1}. \tag{10.72}$$

10.8 Convex to linear reduction

In the previous section we considered the expert problem, where the loss function is a *linear* function of the parameters. At first glance we may think this is a very restrictive constraint for online convex optimization. However, as we will see in this section, we can always assume f_t to be linear in online convex optimization without loss of generality. That means that for online learning, the linear case is the hardest one.

More concretely, assume we have an algorithm $\mathcal{A}$ that solves the linear case. Given any online convex optimization, we will build an algorithm $\mathcal{A}'$ which invokes algorithm $\mathcal{A}$ in the following fashion: for $t = 1, \ldots, T$,

1. The learner invoke $\mathcal{A}$ to get output action $w_t \in \Omega$.
2. The environment gives the learner the loss function $f_t(\cdot)$.

3. The learner construct a linear function $g_t(w) = \langle \nabla f_t(w_t), w \rangle$, which is the local linear approximation of f at w. (Technically the local linear approximation of f and w is $\langle \nabla f_t(w_t), w - w_t \rangle$, but we drop the w_t shift for convenience.)

4. The learner feeds $g_t(\cdot)$ to algorithm $\mathcal{A}$ as the loss function.

We have the following informal claim[1]:

Proposition 10.8.1 (Informal)**.** If a deterministic algorithm $\mathcal{A}$ has regret no more than $\gamma(T)$ for linear cases for some function $\gamma(\cdot)$, then $\mathcal{A}'$ stated above has regret no more than $\gamma(T)$ for convex functions.

Proof. For all $w \in \Omega$, the regret guarantee on $\mathcal{A}$ tells us that

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) \leq \gamma(T). \tag{10.73}$$

Since f_t is convex, we also know that

$$g_t(w_t) - g_t(w) = \langle \nabla f_t(w_t), w_t - w \rangle \geq f_t(w_t) - f_t(w). \tag{10.74}$$

Therefore, for all $w \in \Omega$,

$$\sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w) \leq \sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) \tag{10.75}$$

$$\leq \gamma(T). \tag{10.76}$$

Hence, the regret for $\mathcal{A}'$ is upper bounded by $\gamma(T)$ as well. □

10.8.1 Online gradient descent

In this section we combine the FTRL framework with ℓ_2-regularization and the online-to-linear reduction. The resulting algorithm is *online gradient descent*.

Concretely, given any convex online optimization problem, we first do the online-to-linear reduction, then we use FTRL with ℓ_2 regularization ($\phi(w) = \frac{1}{2}\|w\|_2^2$) to solve the resulting linear case. This gives us the following update:

$$w_t = \operatorname{argmin} \sum_{i=1}^{t-1} g_i(w) + \frac{1}{\eta}\|w\|_2^2 \tag{10.77}$$

$$= \operatorname*{argmin}_{w \in \Omega} \sum_{i=1}^{t-1} \langle \nabla f_i(w_i), w \rangle + \frac{1}{\eta}\|w\|_2^2 \tag{10.78}$$

$$= \Pi_\Omega \left(-\eta \cdot \sum_{i=1}^{t-1} \nabla f_i(w_i) \right), \tag{10.79}$$

where $\Pi_\Omega(\cdot)$ is the projection operator onto the set Ω.The last equality is because for any vector a, we have

$$\operatorname*{argmin}_{w \in \Omega} \langle a, w \rangle + \frac{1}{\eta}\|w\|_2^2 = \operatorname*{argmin}_{w \in \Omega} \frac{1}{2\eta}\|w + \eta a\|_2^2 - \eta\|a\|_2^2 \tag{10.80}$$

$$= \operatorname*{argmin}_{w \in \Omega} \|w + \eta a\|_2^2 \tag{10.81}$$

$$= \operatorname*{argmin}_{w \in \Omega} \|w - (-\eta a)\|_2^2 \tag{10.82}$$

$$= \Pi_\Omega(-\eta a). \tag{10.83}$$

[1]For rigorous proof, we need additional assumptions and restrictions on f_t, g_t.

Intuitively, we can think of this algorithm as gradient descent with "lazy" projection:

$$u_t = u_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{10.84}$$
$$w_t = \Pi_\Omega(u_t). \tag{10.85}$$

Similarly, we can define gradient descent with "eager" projection (which can get similar regret bounds):

$$u_t = w_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{10.86}$$
$$w_t = \Pi_\Omega(u_t). \tag{10.87}$$

This concludes our discussion of online learning in this course.

Bibliography

Sanjeev Arora, Elad Hazan, and Satyen Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 339–348. IEEE, 2005.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Peter Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *NeurIPS*, 2017.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences (PNAS)*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pages 483–513. PMLR, 2020.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — online stochastic gradient for tensor decomposition. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842, Paris, France, 03–06 Jul 2015. PMLR.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/7fb8ceb3bd59c7956b1df66729296a4c-Paper.pdf`.

Jeff Z HaoChen, Colin Wei, Jason D Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. *arXiv preprint arXiv:2006.08680*, 2020.

Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM*, 60(6), 2013.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1246–1257, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.

Percy Liang. Cs229t/stat231: Statistical learning theory (winter 2016), April 2016.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Pengda Liu and Garrett Thomas. Cs229t/stat231: Statistical learning theory (fall 2018), October 2018.

Haipeng Luo. Introduction to online learning, 2017. URL `https://haipeng-luo.net/courses/CSCI699/`.

Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for non-convex losses, 2017.

Katta G. Murty and Santosh N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

John A. Rice. *Mathematical Statistics and Data Analysis.* Belmont, CA: Duxbury Press., third edition, 2006.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Ramon van Handel. Probability in high dimension: Apc 550 lecture notes, December 2016.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Colin Wei and Tengyu Ma. Improved sample complexities for deep neural networks and robust classification via an all-layer margin. In *International Conference on Learning Representations*, 2019.

Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel, 2020.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.