Lecture Notes for Machine Learning Theory (CS229M/STATS214)

Instructor: Tengyu Ma

June 26, 2022

Contents

1 Supervised Learning Formulations **6**
1.1 Supervised learning . . . 6
1.2 Empirical risk minimization . . . 7

2 Asymptotic Analysis **9**
2.1 Asymptotics of empirical risk minimization . . . 9
2.1.1 Key ideas of proofs . . . 10
2.1.2 Main proof . . . 10
2.1.3 Well-specified case . . . 12
2.2 Limitations of asymptotic analysis . . . 13

3 Concentration Inequalities **15**
3.1 The big-O notation . . . 15
3.2 Chebyshev's inequality . . . 15
3.3 Hoeffding's inequality . . . 16
3.4 Sub-Gaussian random variables . . . 17
3.4.1 Examples of sub-Gaussian random variables . . . 19
3.5 Concentrations of functions of random variables . . . 20
3.5.1 Bounds for Gaussian random variables . . . 22

4 Generalization Bounds via Uniform Convergence **23**
4.1 Basic concepts . . . 23
4.1.1 Motivation: Uniform convergence implies generalization . . . 23
4.1.2 Deriving uniform convergence bounds . . . 24
4.1.3 Intuitive interpretation of uniform convergence . . . 24
4.2 Finite hypothesis class . . . 25
4.2.1 Comparing Theorem 4.1 with standard concentration inequalities . . . 26
4.2.2 Comparing Theorem 4.1 with asymptotic bounds . . . 27
4.3 Bounds for infinite hypothesis class via discretization . . . 27
4.3.1 Discretization of the parameter space by ϵ-covers . . . 27
4.3.2 Uniform convergence bound for infinite $\mathcal{H}$ . . . 28
4.4 Rademacher complexity . . . 30
4.4.1 Motivation for a new complexity measure . . . 30
4.4.2 Definitions . . . 31
4.4.3 Dependence of Rademacher complexity on P . . . 33
4.5 Empirical Rademacher complexity . . . 34
4.5.1 Rademacher complexity is translation invariant . . . 36
4.6 Covering number upper bounds Rademacher complexity . . . 36
4.6.1 Chaining and Dudley's theorem . . . 40
4.6.2 Translating Covering Number Bounds to Rademacher Complexity . . . 44

4.6.3 Lipschitz composition . . . 45
4.7 VC dimension and its limitations . . . 45

5 Rademacher Complexity Bounds for Concrete Models and Losses 47
5.1 Margin theory for classification problems . . . 47
5.1.1 Intuition . . . 47
5.1.2 Formalizing margin theory . . . 47
5.2 Linear models . . . 50
5.2.1 Linear models with weights bounded in ℓ_2 norm . . . 50
5.2.2 Linear models with weights bounded in ℓ_1 norm . . . 51
5.2.3 Comparing the bounds for different $\mathcal{H}$ . . . 53
5.3 Two-layer neural networks . . . 54
5.3.1 Refined bounds . . . 56
5.4 More implications and discussions on two-layer neural nets . . . 57
5.4.1 Connection to ℓ_2 regularization . . . 58
5.4.2 Generalization bounds that are decreasing in m . . . 58
5.4.3 Equivalence to an ℓ_1-SVM in $m \to \infty$ limit . . . 59
5.5 Deep neural nets (via covering number) . . . 61
5.5.1 Preparation: covering number for linear models . . . 61
5.5.2 Deep neural networks . . . 62

6 Data-dependent Generalization Bounds for Deep Nets 68
6.1 All-layer margin . . . 69

7 Theoretical Mysteries in Deep Learning 75
7.1 Framework for classical machine learning theory . . . 75
7.2 Deep learning theory and its differences . . . 76

8 Nonconvex Optimization 79
8.1 Optimization landscape . . . 79
8.2 Efficient convergence to (approximate) local minima . . . 80
8.2.1 Strict-saddle condition . . . 81
8.3 All local minima are global minima: two examples . . . 81
8.3.1 Principal components analysis (PCA) . . . 81
8.3.2 Matrix Completion [Ge et al., 2016] . . . 83
8.3.3 Other problems where all local minima are global minima . . . 87
8.4 The Neural Tangent Kernel (NTK) Approach . . . 88
8.4.1 Two examples of the NTK regime . . . 91
8.4.2 Optimizing $\hat{L}(g_\theta)$ vs. $\hat{L}(f_\theta)$ . . . 95
8.4.3 Limitations of NTK . . . 99

9 Implicit/Algorithmic Regularization Effect 101
9.1 Implicit regularization effect of zero initialization in overparametrized linear regression . . . 101
9.2 Implicit regularization of small initialization in nonlinear models . . . 103
9.2.1 Main results of algorithmic regularization . . . 103
9.2.2 Ground work for proof and the restricted isometry property . . . 104
9.2.3 Warm-up for analysis: Gradient descent on population loss . . . 105
9.2.4 Proof of main result: gradient descent on empirical loss . . . 107
9.3 From small to large initialization: a precise characterization . . . 111
9.3.1 Preparation: gradient flow . . . 111
9.3.2 Characterizing the implicit bias of initialization . . . 112
9.4 Implicit regularization towards max-margin solutions in classification . . . 117
9.5 Implicit regularization effect of noise in SGD . . . 120

9.5.1 Warmup: SGD on the one dimensional quadratic function 121
9.5.2 SGD on multi-dimensional quadratic functions 122
9.5.3 SGD on non-quadratic functions 123
9.5.4 SGD with label noise 125

10 Unsupervised Learning and Self-supervised Learning 127
10.1 Method of Moments for mixture models 127
10.1.1 Warm-up: mixture of two Gaussians 128
10.1.2 Mixture of Gaussians with more components via tensor decomposition 129
10.2 Graph Decomposition and Spectral Clustering 133
10.2.1 Stochastic block model 133
10.2.2 Clustering the worst-case graph 137
10.2.3 Applying spectral clustering 140
10.3 Self-supervised Learning 141
10.3.1 Pretraining / self-supervised learning / foundation model basics 141
10.3.2 Analysis of contrastive learning 141

11 Online learning 147
11.1 Online learning setup 147
11.1.1 Evaluation of the learner 148
11.1.2 The realizable case 149
11.2 Online (convex) optimization (OCO) 150
11.2.1 Settings and variants of OCO 150
11.3 Reducing online learning to online optimization 151
11.3.1 Example: Online learning regression problem 151
11.3.2 Example: The expert problem 151
11.4 Reducing online learning to batch learning 153
11.5 Follow-the-Leader (FTL) algorithm 154
11.6 Be-the-leader (BTL) algorithm 154
11.7 Follow-the-regularized-leader (FTRL) strategy 156
11.7.1 Regularization and stability 156
11.7.2 Regret of FTRL 157
11.7.3 Applying FTRL to online linear regression 158
11.7.4 Applying FTRL to the expert problem 159
11.8 Convex to linear reduction 161
11.8.1 Online gradient descent 162

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 11 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 be the main 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:

$$L(\hat{\theta}) - \inf_{\theta \in \Theta} L(\theta) \le \frac{c}{n} + o\left(\frac{1}{n}\right). \tag{2.1}$$

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 decreases 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.*[1] *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. $\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{d} \mathcal{N}\left(0, (\nabla^2 L(\theta^*))^{-1} \operatorname{Cov}(\nabla \ell((x,y), \theta^*))(\nabla^2 L(\theta^*))^{-1}\right)$.

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

5. $\lim_{n\to\infty} \mathbb{E}\left[n(L(\hat{\theta}) - L(\theta^*))\right] = \frac{1}{2} \operatorname{tr}\left(\nabla^2 L(\theta^*)^{-1} \operatorname{Cov}(\nabla \ell((x,y), \theta^*))\right)$.

[1] $X_n \xrightarrow{p} X$ implies that for all $\epsilon > 0$, $\mathbb{P}(\|X_n - X\| > \epsilon) \to 0$, while $X_n \xrightarrow{d} X$ implies that $\mathbb{P}(X_n \le t) \to \mathbb{P}(X \le t)$ at all points t for which $\mathbb{P}(X \le t)$ is continuous. These two notions of convergence are known as convergence in probability and convergence in distribution, respectively. These concepts are not essential to this course, but additional information can be found by reading the Wikipedia article on convergence of random variables.

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 $1/n$. While we will not discuss the regularity assumptions in Theorem 2.1 in great detail, we note that the assumption that L is twice differentiable is crucial.

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 $\nabla^2 \widehat{L}(\theta) \xrightarrow{p} \nabla^2 L(\theta)$ 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 $AZ \sim N(0, A\Sigma A^\top)$.*

2. *If $Z \sim N(0, \Sigma^{-1})$ and $Z \in \mathbb{R}^p$, then $Z^\top \Sigma Z \sim \chi^2(p)$, 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,

$$\begin{aligned}\sqrt{n}(\hat{\theta} - \theta^*) &= -(\nabla^2 \widehat{L}(\theta^*))^{-1} \sqrt{n}(\nabla \widehat{L}(\theta^*)) + O(\sqrt{n}||\hat{\theta} - \theta^*||_2^2) & (2.4)\\ &\approx -(\nabla^2 \widehat{L}(\theta^*))^{-1} \sqrt{n}(\nabla \widehat{L}(\theta^*)). & (2.5)\end{aligned}$$

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

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

Note that $\nabla L(\theta^*) = 0$ because θ^* is the minimizer of L, so $\sqrt{n}(\nabla \widehat{L}(\theta^*)) \xrightarrow{d} \mathcal{N}(0, \mathrm{Cov}(\nabla \ell((x, y), \theta^*)))$. 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),

$$\begin{aligned} \sqrt{n}(\hat{\theta} - \theta^*) &\xrightarrow{d} \nabla^2 L(\theta^*)^{-1} \mathcal{N}(0, \mathrm{Cov}(\nabla \ell((x, y), \theta^*))) && (2.7) \\ &\overset{d}{=} \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1} \mathrm{Cov}(\nabla \ell((x, y), \theta^*)) \nabla^2 L(\theta^*)^{-1}\right), && (2.8) \end{aligned}$$

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

$$\begin{aligned} n(L(\hat{\theta}) - L(\theta^*)) &= \frac{n}{2} \langle \hat{\theta} - \theta^*, \nabla^2 L(\theta^*)(\hat{\theta} - \theta^*) \rangle + o(\|\hat{\theta} - \theta^*\|_2^2) && (2.10) \\ &\approx \frac{1}{2} \langle \sqrt{n}(\hat{\theta} - \theta^*), \nabla^2 L(\theta^*) \sqrt{n}(\hat{\theta} - \theta^*) \rangle && (2.11) \\ &= \frac{1}{2} \left\| \nabla^2 L(\theta^*)^{1/2} \sqrt{n}(\hat{\theta} - \theta^*) \right\|_2^2, && (2.12) \end{aligned}$$

where the last equality follows from the fact that for any vector v and positive semi-definite matrix A of appropriate dimensions, the inner product $\langle v, Av \rangle = v^\top A v = \|A^{1/2} v\|_2^2$. 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

$$\begin{aligned} S &\sim \nabla^2 L(\theta^*)^{1/2} \cdot \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1} \mathrm{Cov}(\nabla \ell((x, y), \theta^*)) \nabla^2 L(\theta^*)^{-1}\right) && (2.13) \\ &\overset{d}{=} \mathcal{N}\left(0, \nabla^2 L(\theta^*)^{-1/2} \mathrm{Cov}(\nabla \ell((x, y), \theta^*)) \nabla^2 L(\theta^*)^{-1/2}\right). && (2.14) \end{aligned}$$

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,

$$\begin{aligned} \lim_{n \to \infty} \mathbb{E}\left[n(L(\hat{\theta}) - L(\theta^*))\right] &= \frac{1}{2} \mathbb{E}\left[\|S\|_2^2\right] = \frac{1}{2} \mathbb{E}\left[\mathrm{tr}(S^\top S)\right] && (2.15) \\ &= \frac{1}{2} \mathbb{E}\left[\mathrm{tr}(S S^\top)\right] = \frac{1}{2} \mathrm{tr}\left(\mathbb{E}[S S^\top]\right) && (2.16) \\ &= \frac{1}{2} \mathrm{tr}\left(\mathrm{Cov}(S)\right) && (2.17) \\ &= \frac{1}{2} \mathrm{tr}\left(\nabla^2 L(\theta^*)^{-1} \mathrm{Cov}(\nabla \ell((x, y), \theta^*))\right). && (2.18) \end{aligned}$$

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*

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

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

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

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

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 as it achieves 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 $\theta_* \in \operatorname{argmin}_\theta L(\theta)$. However, the minimizer of $L(\theta)$ is unique because of consistency, so we must have $\operatorname{argmin}_\theta L(\theta) = \theta^*$ 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 $n > p^{50}$ for $\frac{p^{100}}{n^2}$ to be 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, when $O(n_1), \ldots, O(n_k)$ occur in a statement, it means that **there exist** 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 from traditional "big-O" notation is that we do not need to send $n \to \infty$ in order to define "big-O". In nearly all cases, big-O notation is used to define an upper bound; then, the bound is identical if we simply substitute Cx in place of $O(x)$.

Note that the x in our definition of big-O is a surrogate for an arbitrary variable. For instance, later on in this chapter, we will encounter the term $O(\sigma\sqrt{\log n})$. The definition above, applied with $x = \sigma\sqrt{\log n}$, yields the following conclusion: $O(\sigma\sqrt{\log n}) = f(\sigma\sqrt{\log n})$ for some function f and constant C such that $|f(\sigma\sqrt{\log n})| \leq C\sigma\sqrt{\log n}$ for all values that $\sigma\sqrt{\log n}$ can take.

Lastly, 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 Chebyshev's inequality

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

Theorem 3.1 (Chebyshev's inequality)**.** *Let Z be a random variable with finite expectation and variance. Then*

$$\Pr[|Z - \mathbb{E}[Z]| \geq t] \leq \frac{\mathrm{Var}(Z)}{t^2}, \quad \forall t > 0. \tag{3.1}$$

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 (3.1) we see that

$$\Pr\left[|Z - \mathbb{E}[Z]| \leq \frac{\mathrm{sd}(Z)}{\sqrt{\delta}}\right] \geq 1 - \delta. \tag{3.2}$$

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

$$\Pr\left[|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z)\sqrt{2\log(2/\delta)}\right] \geq 1 - \delta. \tag{3.3}$$

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 (3.2) and (3.3) 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 (3.3) implies that

$$|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z) \cdot \sqrt{\log O\left(n^c\right)} = \mathrm{sd}(Z) \cdot O\left(\sqrt{\log n}\right) \quad \textit{w.p. } 1 - \delta, \tag{3.4}$$

while Chebyshev's inequality gives us the weaker result

$$|Z - \mathbb{E}[Z]| \leq \mathrm{sd}(Z) \cdot \sqrt{O(n^c)} = \mathrm{sd}(Z) \cdot O(n^{c/2}) \quad \textit{w.p. } 1 - \delta. \tag{3.5}$$

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. However, in many cases, we will be able to improve the $1/t^2$ rate of tail decay in Chebyshev's inequality to an e^{-t} rate. In the next two sections, we will demonstrate how to construct tail bounds with exponential decay rates.

3.3 Hoeffding's inequality

We next provide a brief overview of Hoeffding's inequality, a concentration inequality for bounded random variables with an exponential tail bound:

Theorem 3.2 (Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be independent 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 $\varepsilon > 0$,*

$$\Pr\left[|\bar{X} - \mu| \leq \varepsilon\right] \geq 1 - 2\exp\left(\frac{-2n^2\varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right). \tag{3.6}$$

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 independence assumption, we can show

$$\mathrm{Var}\left(\bar{X}\right) = \frac{1}{n^2}\sum_{i=1}^n \mathrm{Var}(X_i) \leq \frac{1}{n^2}\sum_{i=1}^n (b_i - a_i)^2. \tag{3.7}$$

Let $\sigma^2 = \frac{1}{n^2}\sum_{i=1}^{n}(b_i - a_i)^2$. If we take $\varepsilon = O(\sigma\sqrt{\log n}) = \sigma\sqrt{c\log n}$ so that ε is 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 (3.6) to reach the following conclusion:

$$\begin{aligned}
\Pr\left[|\bar{X} - \mu| \leq \varepsilon\right] &\geq 1 - 2\exp\left(\frac{-2\varepsilon^2}{\sigma^2}\right) && (3.8)\\
&= 1 - 2\exp(-2c\log n) && (3.9)\\
&= 1 - 2n^{-2c} && (3.10)
\end{aligned}$$

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 more than some sublogarithmic ($\sqrt{c\log n}$) factor of the standard deviation.[1] Thus, we can restate the above claim we reached as follows:

Remark 3.3. For sufficiently large n, $|\bar{X} - \mu| \leq O(\sigma\sqrt{\log n})$ with high probability.

Remark 3.4. 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 3.4 provides a compact form of the Hoeffding bound that we can use when the X_i are bounded almost surely.

So far, we have only shown how to construct exponential tail bounds for bounded random variables. Since requiring boundedness in $[0, 1]$ (or $[a, b]$ more generally) is limiting, 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 establishing this definition, we will see how this bound guarantees the exponential tail decay we desire.

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

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\sigma^2\lambda^2/2}, \quad \forall\lambda \in \mathbb{R}. \tag{3.11}$$

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

Remark 3.6. As it turns out, (3.11) 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:

$$\mathbb{E}[\exp(\lambda X)] = \mathbb{E}\left[\sum_{k=0}^{\infty}\frac{(\lambda X)^k}{k!}\right] = \sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\,\mathbb{E}[X^k]. \tag{3.12}$$

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. Though a proof of this result is beyond the scope of this monograph, Proposition 2.5.2 in [Vershynin, 2018] shows that (3.11) is equivalent to $\mathbb{E}\left[|X|^p\right]^{1/p} \lesssim \sqrt{p}$ for all $p \geq 1$.

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

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

Although (3.11) 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*

$$\Pr[|X-\mu| \geq t] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right), \quad \forall t \in \mathbb{R}. \tag{3.13}$$

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

$$\begin{aligned}
\Pr[X-\mu \geq t] &= \Pr[\exp(\lambda(X-\mu)) \geq \exp(\lambda t)] && (3.14)\\
&\leq \exp(-\lambda t)\,\mathbb{E}[\exp(\lambda(X-\mu))] && \text{(by Markov's inequality)} \quad (3.15)\\
&\leq \exp(-\lambda t)\exp(\sigma^2\lambda^2/2) && \text{(by (3.11))} \quad (3.16)\\
&= \exp(-\lambda t + \sigma^2\lambda^2/2). && (3.17)
\end{aligned}$$

Because the bound (3.17) holds for any choice of $\lambda > 0$ and $\exp(\cdot)$ is monotonically increasing, we can optimize the bound (3.17) 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

$$\Pr[X-\mu \geq t] \leq \exp\left(-\frac{t^2}{2\sigma^2}\right). \tag{3.18}$$

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

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

We can then obtain (3.13) by applying the union bound:

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

□

Remark 3.8 (Tail bound implies sub-Gaussianity). In addition to being a necessary condition for sub-Gaussianity (Theorem 3.7), the tail bound (3.13) 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 (3.13) 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 (3.11) 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 (3.3) 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. This "tensorization" property 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: 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*

$$\Pr[|Z - \mathbb{E}[Z]| \geq t] \leq 2 \exp\left(-\frac{t^2}{2\sum_{i=1}^n \sigma_i^2}\right), \tag{3.21}$$

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

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

$$\begin{aligned}
\mathbb{E}\left[\exp\left\{\lambda(Z - \mathbb{E}[Z])\right\}\right] &= \mathbb{E}\left[\prod_{i=1}^n \exp\left\{\lambda(X_i - \mathbb{E}[X_i])\right\}\right] && (3.22)\\
&= \prod_{i=1}^n \mathbb{E}\left[\exp\left\{\lambda(X_i - \mathbb{E}[X_i])\right\}\right] && (3.23)\\
&\leq \prod_{i=1}^n \exp\left(\frac{\lambda^2\sigma_i^2}{2}\right) && (3.24)\\
&= \exp\left(\frac{\lambda^2\sum_{i=1}^n \sigma_i^2}{2}\right), && (3.25)
\end{aligned}$$

so Z is sub-Gaussian with variance proxy $\sum_{i=1}^n \sigma_i^2$. The tail bound then follows immediately from (3.13). □

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 $\exp(\cdot)$:

$$\begin{aligned}
\mathbb{E}[\exp(\lambda\epsilon)] &= \frac{1}{2}\left\{\exp(-\lambda) + \exp(\lambda)\right\} && && (3.26)\\
&= \frac{1}{2}\left\{\sum_{k=0}^{\infty}\frac{(-\lambda)^k}{k!} + \sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\right\} && && (3.27)\\
&= \sum_{k=0}^{\infty}\frac{\lambda^{2k}}{(2k)!} && \text{(for odd } k\text{, } (-\lambda)^k + \lambda^k = 0) && (3.28)\\
&\leq 1 + \sum_{k=1}^{\infty}\frac{\left(\lambda^2\right)^k}{2^k k!} && (2^k k! \text{ is every other term of } (2k)!) && (3.29)\\
&= \exp(\lambda^2/2), && && (3.30)
\end{aligned}$$

which is exactly the moment generating function bound (3.11) 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 (3.11) 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',*

$$|f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_n)| \leq c_i. \tag{3.31}$$

(Intuitively, (3.31) *states that f is not overly sensitive to arbitrary changes in a single coordinate.) Then, for any independent random variables $X_1, \ldots, X_n$,*

$$\Pr\left[f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1, \ldots, X_n)] \geq t\right] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right). \tag{3.32}$$

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 $a_i \leq x_i \leq b_i$ and

$$f(x_1, \ldots, x_n) = \sum_{i=1}^n x_i. \tag{3.33}$$

Proof. The idea of this proof is to take the quantity $f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1, \ldots, X_n)]$ and break it into manageable components by conditioning on portions of the sample. To this end, we begin by defining:

$$\begin{aligned}
Z_0 &= \mathbb{E}\left[f(X_1, \ldots, X_n)\right] && \text{constant} \\
Z_1 &= \mathbb{E}\left[f(X_1, \ldots, X_n) | X_1\right] && \text{a function of } X_1 \\
&\cdots \\
Z_i &= \mathbb{E}\left[f(X_1, \ldots, X_n) | X_1, \ldots, X_i\right] && \text{a function of } X_1, \ldots, X_i \\
&\cdots \\
Z_n &= f(X_1, \ldots, X_n)
\end{aligned}$$

Using the law of total expectation, we show also that the expectation of Z_i equals Z_0 for all i.

$$\begin{aligned}\mathbb{E}[Z_i] &= \mathbb{E}\left[\mathbb{E}\left[f(X_1,\ldots,X_n)|X_1,\ldots,X_i\right]\right] \\ &= \mathbb{E}[f(X_1,\ldots,X_n)] \\ &= Z_0\end{aligned}$$

The fact that $\mathbb{E}[D_i] = 0$, where $D_i = Z_i - Z_{i-1}$, is an immediate corollary of this result. Next, we observe that we can rewrite the quantity of interest, $Z_n - Z_0$, as a telescoping sum in the increments $Z_i - Z_{i-1}$:

$$\begin{aligned}Z_n - Z_0 &= (Z_n - Z_{n-1}) + (Z_{n-1} - Z_{n-2}) + \cdots + (Z_1 - Z_0) \\ &= \sum_{i=1}^{n} D_i\end{aligned}$$

Next, we show that conditional on $X_1, \ldots, X_{i-1}$, D_i is a bounded random variable. First, observe that:

$$\begin{aligned}A_i &= \inf_x \mathbb{E}\left[f(X_1,\ldots,X_n)|X_1,\ldots,X_{i-1},X_i = x\right] - \mathbb{E}\left[f(X_1,\ldots,X_n)|X_1,\ldots,X_{i-1}\right] \\ B_i &= \sup_x \mathbb{E}\left[f(X_1,\ldots,X_n)|X_1,\ldots,X_{i=1},X_i = x\right] - \mathbb{E}\left[f(X_1,\ldots,X_n)|X_1,\ldots,X_{i-1}\right]\end{aligned}$$

It is clear from their definition that $A_i \leq D_i \leq B_i$. Furthermore, by independence of the X_i's, we have that:

$$\begin{aligned}B_i - A_i &\leq \sup_{x_{1:i-1}} \sup_{x,x'} \int \left(f(x_1,\ldots,x_{i-1},x,x_{i+1},\ldots,x_n) - f(x_1,\ldots,x_{i-1},x',x_{i+1},\ldots,x_n)\right) dP(x_{i+1},\ldots,x_n) \\ &\leq c_i\end{aligned}$$

Using this bound, the properties of conditional expectation, and Example 3.13, we can now prove that that $Z_n - Z_0$ is $O\left(\sqrt{\sum_{i=1}^{n} c_i^2}\right)$-sub-Gaussian.

$$\begin{aligned}\mathbb{E}\left[e^{\lambda(Z_n - Z_0)}\right] &= \mathbb{E}\left[e^{\lambda\sum_{i=1}^{n}(Z_i - Z_{i-1})}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[e^{\lambda(Z_n - Z_{n-1})}\middle| X_1,\ldots,X_{n-1}\right] e^{\lambda\sum_{i=1}^{n-1}(Z_i - Z_{i-1})}\right] \\ &\leq e^{\lambda^2 c_n^2/8}\,\mathbb{E}\left[e^{\lambda\sum_{i=1}^{n-1}(Z_i - Z_{i-1})}\right] \\ &\cdots \\ &\leq e^{\lambda^2(\sum_{i=1}^{n} c_i^2)/8}\end{aligned}$$

The final inequality given in (3.32) follows by Theorem 3.7. □

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 (3.31), 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 $X_1, \dots, X_n$ be independent random variables. Then, for all $t \geq 0$,*

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

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

3.5.1 Bounds for Gaussian random variables

Unfortunately, the bounded difference condition (3.31) 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 $X_1, \dots, X_n$ are independently sampled from $\mathcal{N}(0,1)$, then*

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

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 $X = (X_1, \dots, X_n)$, where $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. 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:

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

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

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

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 $0 \le \ell((x, y); \theta) \le 1$, 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

$$
\begin{aligned}
L(\hat{\theta}) - L(\theta^*) &\leq |L(\hat{\theta}) - \hat{L}(\hat{\theta})| + \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + |\hat{L}(\theta^*) - L(\theta^*)| && (4.4)\\
&\leq |L(\hat{\theta}) - \hat{L}(\hat{\theta})| + 0 + |\hat{L}(\theta^*) - L(\theta^*)| && (4.5)\\
&\leq 2 \sup_{\theta \in \Theta} |L(\theta) - \hat{L}(\theta)|. && (4.6)
\end{aligned}
$$

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

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

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 3.4). 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 $\widehat{L}(\hat{\theta})$, $\ell((x^{(i)}, y^{(i)}); \hat{\theta})$, 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 control 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 4.1a). 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 4.1b). 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 were inspired by [Liang, 2016, Liu and Thomas, 2018].*)

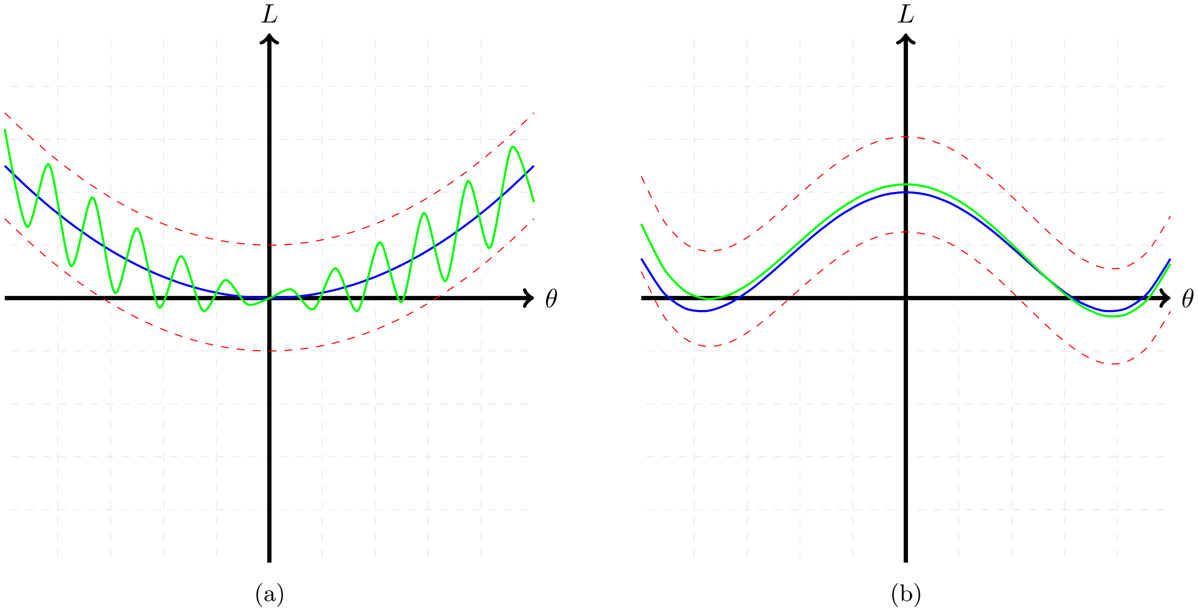


Figure 4.1: These curves demonstrate how we apply uniform convergence to bound the population risk. The blue curves are the unobserved population risk we aim to bound. The green curves denote the empirical risk we observe. Though this curve is often depicted as the fluctuating curve used in Figure 4.1a, it is more often a smooth curve whose shape mimics that of the population risk (Figure 4.1b). Uniform convergence allows us to construct additive error bounds for the excess risk, which are depicted using the red, dashed lines.

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. $0 \le \ell((x,y),h) \le 1$. Then $\forall \delta$ s.t. $0 < \delta < \frac{1}{2}$, with probability at least $1-\delta$, we have*

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

As a corollary, we also have

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

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 $\Pr(E_1 \cup \cdots \cup E_k) \le \sum_{i=1}^k \Pr(E_i)$.)

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

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

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

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

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

$$\epsilon = \sqrt{\frac{2(\ln|\mathcal{H}| + \ln(2/\delta))}{n}}, \tag{4.18}$$

we have that

$$\Pr\left(|L(\hat{h}) - L(h^*)| \geq \epsilon\right) \leq \Pr\left(2\sup_{h \in \mathcal{H}} |\hat{L}(h) - L(h)| \geq \epsilon\right) \tag{4.19}$$

$$\leq 2|\mathcal{H}|\exp\left(-\frac{n\epsilon^2}{2}\right). \tag{4.20}$$

□

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:

$$\forall h \in \mathcal{H}, \quad w.h.p. \quad |\hat{L}(h) - L(h)| \leq \tilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{4.21}$$

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

$$w.h.p., \quad \forall h \in \mathcal{H}, \quad |\hat{L}(h) - L(h)| \leq \tilde{O}\left(\frac{\ln|\mathcal{H}|}{\sqrt{n}}\right), \tag{4.22}$$

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,

$$L(\hat{h}) - L(h^*) \leq \frac{c}{n} + o\left(n^{-1}\right). \tag{4.23}$$

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

$$\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}, \|\theta\|_2 \leq B\}. \tag{4.24}$$

The intuition behind brute-force discretization is as follows: Let $E_\theta = \{|\widehat{L}(\theta) - L(\theta)| \geq \epsilon\}$ be the "bad" events. 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 a 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,

$$S \subseteq \bigcup_{x \in C} \text{Ball}(x, \epsilon, \rho), \quad \text{where} \tag{4.25}$$

$$\text{Ball}(x, \epsilon, \rho) \triangleq \{x' : \rho(x, x') \leq \epsilon\}. \tag{4.26}$$

(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 in this 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 $B, \epsilon > 0$, 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* $\max\left(\left(\frac{3B\sqrt{p}}{\epsilon}\right)^p, 1\right)$ *elements.*

Proof. Note that if $\epsilon > B\sqrt{p}$, then S is trivially contained in the ball centered at the origin with radius ϵ and the ϵ-cover has size 1. Assume $\epsilon \le B\sqrt{p}$. Set

$$C = \left\{ x \in S : x_i = k_i \frac{\epsilon}{\sqrt{p}}, k_i \in \mathbb{Z}, |k_i| \le \frac{B\sqrt{p}}{\epsilon} \right\}, \tag{4.27}$$

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 4.2 for an illustration.

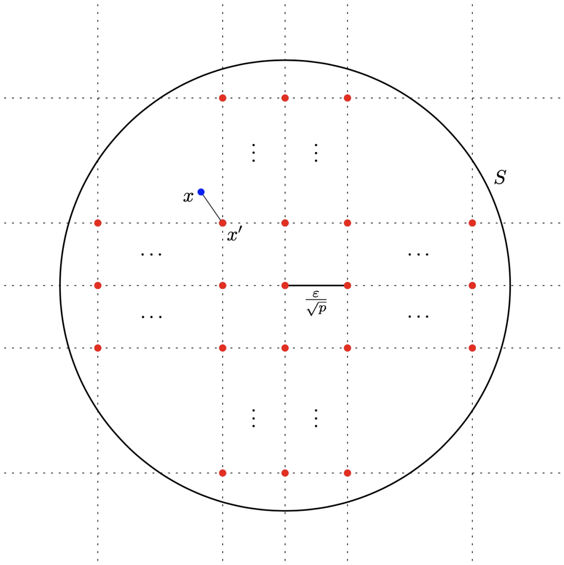


Figure 4.2: The ϵ-cover (shown in red) of S that we construct in the proof of Lemma 4.5. For $x \in S$, we choose the grid point x' such that $\|x - x'\|_2 \le \epsilon$.

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 $|x_i - x'_i| \le \frac{\epsilon}{\sqrt{p}}$ for each i. Therefore,

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

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

$$|C| \le \left(\frac{2B\sqrt{p}}{\epsilon} + 1 \right)^p \le \left(\frac{3B\sqrt{p}}{\epsilon} \right)^p. \tag{4.28}$$

□

Remark 4.6. 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.)

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

Definition 4.7 (κ-Lipschitz functions)**.** Let $\kappa \ge 0$ 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

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

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 4.8. *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*

$$\forall\theta, \quad |\hat{L}(\theta) - L(\theta)| \leq O\left(\sqrt{\frac{p \max(\ln(\kappa B n), 1)}{n}}\right). \tag{4.29}$$

Proof of Theorem 4.8. 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

$$|L(\theta) - L(\theta_0)| \leq \kappa\|\theta - \theta_0\|_2 \leq \kappa\epsilon, \text{ and} \tag{4.30}$$

$$|\hat{L}(\theta) - \hat{L}(\theta_0)| \leq \kappa\|\theta - \theta_0\|_2 \leq \kappa\epsilon. \tag{4.31}$$

Therefore, conditional on E, we have

$$|\hat{L}(\theta) - L(\theta)| \leq |\hat{L}(\theta) - \hat{L}(\theta_0)| + |\hat{L}(\theta_0) - L(\theta_0)| + |L(\theta_0) - L(\theta)| \leq 2\kappa\epsilon + \delta. \tag{4.32}$$

It remains to choose suitable parameters δ and ϵ to get the desired bound in Theorem 4.8 while making the failure probability small. First, set $\epsilon = \delta/(2\kappa)$ so that conditional on E,

$$|\hat{L}(\theta) - L(\theta)| \leq 2\delta. \tag{4.33}$$

To choose the correct δ, we must reason about the probability of E under different choices of the parameter. The event E happens with probability $1 - 2|C| \exp(-2n\delta^2) = 1 - 2\exp(\ln|C| - 2n\delta^2)$. From Remark 4.6, we know that $\ln|C| \leq p \ln(3B/(\delta/2))$. If we ignore the log term and assume $\ln|c| \leq p$, then this would give us the high probability bound we want:

$$2|C| \exp(-2n\delta^2) = 2\exp(\ln|C| - 2n\delta^2) \leq 2\exp(p - 2p) = 2\exp(-p). \tag{4.34}$$

(At the same time, we see from (4.33) that this choice of δ gives $|\hat{L}(\theta) - L(\theta)| \leq 2\sqrt{\frac{p}{n}}$, which is roughly the bound we want.)

Since we cannot actually drop the log term in the inequality $\ln|C| \leq p \ln(3B/(\delta/2))$, we need to make δ a little bit bigger. So, if we set $\delta = \sqrt{\frac{c_0 p \max(1, \ln(\kappa B n))}{n}}$ with $c_0 = 36$, then by Remark 4.6,

$$\begin{aligned}
\ln|C| - 2n\delta^2 &\leq p \ln\left(\frac{6B\kappa}{\delta}\right) - 2n\delta^2 && (4.35)\\
&\leq p \ln\left(\frac{6B\kappa\sqrt{n}}{\sqrt{c_0 p \max(1, \ln(\kappa B n))}}\right) - 2n\frac{c_0 p}{n}\ln(\kappa B n) && (\text{dfn of } \delta) \quad (4.36)\\
&\leq p \ln\left(\frac{B\kappa\sqrt{n}}{\sqrt{p}}\right) - 72p\ln(\kappa B n) && (\max(1, \ln(\kappa B n)) \geq 1, c_0 = 36) \quad (4.37)\\
&\leq p\ln(B\kappa n) - 72p\ln(B\kappa n) && (\sqrt{n/p} \leq n) \quad (4.38)\\
&\leq -p, && (4.39)
\end{aligned}$$

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

$$|\hat{L}(\theta) - L(\theta)| \leq 2\delta = O\left(\sqrt{\frac{p}{n}\max(1, \ln(\kappa B n))}\right). \tag{4.40}$$

□

Remark 4.9. We bounded the generalization error $|\hat{L}(\theta) - L(\theta)|$ by $\delta + 2\epsilon\kappa \leq \sqrt{\frac{\ln |C|}{n}} + 2\epsilon\kappa$. The term $2\epsilon\kappa$ represents the error from our brute-force discretization. It is not a problem 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,
$$L(\hat{h}) - \hat{L}(\hat{h}) \leq \tilde{O}\left(\sqrt{\frac{\log |\mathcal{H}|}{n}}\right). \tag{4.41}$$

2. If the hypothesis class $\mathcal{H}$ is p-dimensional,
$$L(\hat{h}) - \hat{L}(\hat{h}) \leq \tilde{O}\left(\sqrt{\frac{p}{n}}\right). \tag{4.42}$$

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 measures of $\mathcal{H}$.

The bound (4.42) 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:

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

(4.42) 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 to prove a better bound.

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

$$L(\hat{h}) - \hat{L}(\hat{h}) \leq \tilde{O}\left(\sqrt{\frac{\text{Complexity}(\Theta)}{n}}\right). \tag{4.43}$$

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.

$$\mathbb{E}\left[\sup_{h\in H}(L(h)-\hat{L}(h))\right] \leq \text{ upper bound.} \tag{4.44}$$

The expectation is over the randomness in the training data $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$.[1]

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

Definition 4.10 (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

$$R_n(F) \triangleq \mathop{\mathbb{E}}_{z_1,\dots,z_n \overset{\text{iid}}{\sim} P}\left[\mathop{\mathbb{E}}_{\sigma_1,\dots,\sigma_n \overset{\text{iid}}{\sim}\{\pm1\}}\left[\sup_{f\in F}\frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right]\right], \tag{4.45}$$

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

Remark 4.11. For applications to empirical risk minimization, we will take $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$. However, Definition 4.10 holds for abstract input spaces $\mathcal{Z}$ as well.

Remark 4.12. 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 that $R_n(\mathcal{F})$ is the maximal possible correlation between outputs of some $f \in \mathcal{F}$ (on points $f(z_1),\dots,f(z_n)$) and random Rademacher variables $(\sigma_1,\dots,\sigma_n)$. 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 4.13.

$$\mathop{\mathbb{E}}_{z_1,\dots,z_n \overset{\text{iid}}{\sim} P}\left[\sup_{f\in F}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathop{\mathbb{E}}_{z\sim P}[f(z)]\right]\right] \leq 2R_n(\mathcal{F}). \tag{4.46}$$

Remark 4.14. We can think of $\frac{1}{n}\sum_{i=1}^n f(z_i)$ as an empirical average and $\mathbb{E}_{z\sim P}[f(z)]$ as a population average.

Why is Theorem 4.13 useful to us? We can set $\mathcal{F}$ to be the family of loss functions, i.e.

$$\mathcal{F} = \{z = (x,y) \in \mathcal{Z} \mapsto \ell((x,y),h) \in \mathbb{R} : h \in \mathcal{H}\}. \tag{4.47}$$

This is the family of losses induced by the hypothesis functions in $\mathcal{H}$. We also define the function class $-\mathcal{F}$ as $\{-f : f \in \mathcal{F}\}$. It should be obvious from this definition that $R_n(\mathcal{F}) = R_n(-\mathcal{F})$ since $\sigma_i \overset{d}{=} -\sigma_i$ for all i. Then, letting $z_i = (x^{(i)}, y^{(i)})$,

$$\begin{aligned}
\mathbb{E}\left[\sup_{h\in\mathcal{H}}\left(L(h)-\hat{L}(h)\right)\right] &= \mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\}}\left[\sup_{h\in\mathcal{H}}\left[L(h) - \frac{1}{n}\sum_{i=1}^n \ell((x^{(i)},y^{(i)}),h)\right]\right] && (4.48)\\
&= \mathop{\mathbb{E}}_{\{z_i\}}\left[\sup_{f\in\mathcal{F}}\left(\mathbb{E}[f(z)] - \frac{1}{n}\sum_{i=1}^n f(z_i)\right)\right] && (4.49)\\
&= \mathop{\mathbb{E}}_{\{z_i\}}\left[\sup_{f\in-\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}[f(z)]\right)\right] && (4.50)\\
&\leq 2R_n(-\mathcal{F}) = 2R_n(\mathcal{F}) && (4.51)
\end{aligned}$$

[1] Though we might like to pull the sup outside of the $\mathbb{E}$ operator, and bound the expectation of the excess risk (a far simpler quantity to deal with!), in general, the sup and $\mathbb{E}$ operators do not commute. In particular, $\mathbb{E}\left[\sup_{h\in\mathcal{H}}(L(h)-\hat{L}(h))\right] \geq \sup_{h\in\mathcal{H}}\mathbb{E}\left[L(h)-\hat{L}(h)\right]$.

where the last step follows by Theorem 4.13.

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 $\ell((x^{(1)}, y^{(1)}), h), \dots \ell((x^{(n)}, y^{(n)}), h)$ correlates with $\sigma_1, \dots, \sigma_n$.

Example 4.15. Consider the binary classification setting where $y \in \{\pm 1\}$. Let ℓ_{0-1} denote the zero-one loss function. Note that

$$\ell_{0-1}((x, y), h) = \mathbf{1}\{h(x) \neq y\} = \frac{1 - yh(x)}{2}. \tag{4.52}$$

Hence,

$$\begin{aligned}
R_n(\mathcal{F}) &= \mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\},\sigma_i}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell_{0-1}((x^{(i)}, y^{(i)}), h)\sigma_i\right] && \text{(by definition)} && (4.53)\\
&= \mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\},\sigma_i}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \left(\frac{-h(x^{(i)})y^{(i)} + 1}{2}\right)\sigma_i\right] && \text{(by (4.52))} && (4.54)\\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\},\sigma_i}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i + \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} -h(x^{(i)})y^{(i)}\sigma_i\right] && (\sup \text{only over } \mathcal{H}) && (4.55)\\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\},\sigma_i}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} -h(x^{(i)})y^{(i)}\sigma_i\right] && (\mathbb{E}[\sigma_i] = 0) && (4.56)\\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\{(x^{(i)},y^{(i)})\},\sigma_i}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} h(x^{(i)})\sigma_i\right] && (-y_i\sigma_i \overset{d}{=} \sigma_i) && (4.57)\\
&= \frac{1}{2}R_n(\mathcal{H}). && \text{(by definition)} && (4.58)
\end{aligned}$$

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 4.16. Rademacher complexity is invariant to translation. This property manifests in the previous example when 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 4.13.

Proof of Theorem 4.13. We use a technique called *symmetrization*, which is a very important technique in probability theory. We first fix $z_1, \dots, z_n$and draw $z'_1, \dots z'_n \overset{\text{iid}}{\sim} P$. Then we can rewrite the term in the expectation on the LHS of (4.46):

$$\begin{aligned}
\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}[f]\right) &= \sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathop{\mathbb{E}}_{z'_1,\dots,z'_n}\left[\frac{1}{n}\sum_{i=1}^{n} f(z'_i)\right]\right) && (4.59)\\
&= \sup_{f\in\mathcal{F}}\left(\mathop{\mathbb{E}}_{z'_1,\dots,z'_n}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \frac{1}{n}\sum_{i=1}^{n} f(z'_i)\right]\right) && (4.60)\\
&\leq \mathop{\mathbb{E}}_{z'_1,\dots,z'_n}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \frac{1}{n}\sum_{i=1}^{n} f(z'_i)\right)\right]. && (4.61)
\end{aligned}$$

The last inequality is because in general,

$$\sup_u \left(\mathop{\mathbb{E}}_v[g(u, v)]\right) \leq \sup_u \left(\mathop{\mathbb{E}}_v\left[\sup_{u'}(g(u', v))\right]\right) = \mathop{\mathbb{E}}_v\left[\sup_u(g(u, v))\right] \tag{4.62}$$

since the sup over u becomes vacuous after we replace u with u'.

Now, if we take the expectation over $z_1, \dots, z_n$ for both sides of (4.61),

$$\mathop{\mathbb{E}}_{z_1,\dots,z_n}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n f(z_i)-\mathbb{E}[f]\right)\right] \le \mathop{\mathbb{E}}_{z_i}\left[\mathop{\mathbb{E}}_{z_i'}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n (f(z_i)-f(z_i'))\right)\right]\right] \tag{4.63}$$

$$= \mathop{\mathbb{E}}_{z_i,z_i'}\left[\mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n \sigma_i\left(f(z_i)-f(z_i')\right)\right)\right]\right] \tag{4.64}$$

$$\le \mathop{\mathbb{E}}_{z_i,z_i',\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right)+\sup_{f\in\mathcal{F}}\left(\frac{1}{n}\sum_{i=1}^n -\sigma_i f(z_i')\right)\right] \tag{4.65}$$

$$= 2R_n(\mathcal{F}), \tag{4.66}$$

where (4.64) 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 4.13 achieves. Consider the quantities on the LHS and RHS of (4.46):

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

First, we removed $\mathbb{E}[f(z)]$, 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

$$\mathop{\mathbb{E}}_{z_1,\dots,z_n\sim P}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right] = \mathop{\mathbb{E}}_{\sigma_1,\dots,\sigma_n}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}f(z_0)\sum_{i=1}^n\sigma_i\right] \tag{4.67}$$

$$\le \mathop{\mathbb{E}}_{\sigma_1,\dots,\sigma_n}\left[\left|\frac{1}{n}\sum_{i=1}^n\sigma_i\right|\right] \qquad \text{(since } f(z_0)\in[-1,1]\text{)} \tag{4.68}$$

$$\le \mathop{\mathbb{E}}_{\sigma_i}\left[\left(\frac{1}{n}\sum_{i=1}^n\sigma_i\right)^2\right]^{\frac{1}{2}} \qquad \text{(Jensen's Inequality)} \tag{4.69}$$

$$= \frac{1}{n}\left(\mathop{\mathbb{E}}_{\sigma_i,\sigma_j}\left[\sum_{i,j=1}^n\sigma_i\sigma_j\right]\right)^{\frac{1}{2}} \tag{4.70}$$

$$= \frac{1}{n}\left(\mathop{\mathbb{E}}_{\sigma_i}\left[\sum_{i=1}^n\sigma_i^2\right]\right)^{\frac{1}{2}} \tag{4.71}$$

$$= \frac{1}{n}\cdot\sqrt{n} = \frac{1}{\sqrt{n}}. \tag{4.72}$$

This bound does not depend on $\mathcal{F}$ (except on the fact that $f \in \mathcal{F}$ is bounded). This example illustrates that a bound on the Rademacher complexity can sometimes depend only on the (known) distribution of the Rademacher random variables.

4.5 Empirical Rademacher complexity

In the previous section, we bounded the expectation of $\sup_{f\in F}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}_{z\sim P}[f(z)]\right]$. 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 4.17 (Empirical Rademacher complexity)**.** Given a dataset $S = \{z_1,\dots,z_n\}$, the *empirical Rademacher complexity* is defined as

$$R_S(\mathcal{F}) \triangleq \mathop{\mathbb{E}}_{\sigma_1,\dots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(z_i)\right]. \tag{4.73}$$

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

As the name suggests, the expectation of the empirical Rademacher complexity is the Rademacher complexity:

$$R_n(\mathcal{F}) = \mathop{\mathbb{E}}_{\substack{z_1,\dots,z_n \overset{\text{iid}}{\sim} P\\ S=\{z_1,\dots,z_n\}}}\left[R_S(\mathcal{F})\right]. \tag{4.74}$$

Here is the theorem involving empirical Rademacher complexity:

Theorem 4.18. *Suppose for all $f\in\mathcal{F}$, $0\le f(z)\le 1$. Then, with probability at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}[f(z)]\right] \le 2R_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}. \tag{4.75}$$

Proof. For conciseness, define

$$g(z_1,\dots,z_n) \triangleq \sup_{f\in F}\left[\frac{1}{n}\sum_{i=1}^n f(z_i) - \mathbb{E}[f(z)]\right]. \tag{4.76}$$

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:

$$\begin{aligned}
g(z_1,\dots,z_n) - g(z_1,\dots,z_i',\dots,z_n) &\le \sup_{f\in\mathcal{F}}\left[\frac{1}{n}\sum_{j=1}^n f(z_j)\right] - \sup_{f\in\mathcal{F}}\left[\left(\frac{1}{n}\sum_{j=1,j\ne i}^n f(z_j)\right) + \frac{f(z_i')}{n}\right] && (4.77)\\
&\le \sup_{f\in\mathcal{F}}\left[\frac{1}{n}\sum_{j=1}^n f(z_j) - \left(\frac{1}{n}\sum_{j=1,j\ne i}^n f(z_j)\right) - \frac{f(z_i')}{n}\right] && (4.78)\\
&= \sup_{f\in\mathcal{F}}\left[\frac{1}{n}\left(f(z_i) - f(z_i')\right)\right] && (4.79)\\
&\le \frac{1}{n}. && (4.80)
\end{aligned}$$

(4.78) holds because in general, $\sup_f A(f) - \sup_f B(f) \le \sup_f[A(f) - B(f)]$, and (4.80) holds since f is bounded by $[0,1]$. We can thus apply McDiarmid's Inequality with parameters $c_1 = \dots = c_n = 1/n$:

$$\Pr\left[g(z_1,\dots,z_n) \ge \mathop{\mathbb{E}}_{z_1,\dots,z_n\overset{\text{iid}}{\sim}P}[g] + \epsilon\right] \le \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right) = \exp(-2n\epsilon^2). \tag{4.81}$$

Step 2: We apply Theorem 4.13 to get

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_n \overset{\text{iid}}{\sim} P}[g] \leq 2R_n(\mathcal{F}). \tag{4.82}$$

Step 3: Define

$$\tilde{g}(z_1,\ldots,z_n) = R_S(\mathcal{F}) \triangleq \mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i f(z_i)\right]. \tag{4.83}$$

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

$$\begin{aligned}
&\tilde{g}(z_1,\ldots,z_n) - \tilde{g}(z_1,\ldots,z_i',\ldots,z_n) \\
&\leq \mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{f\in F}\left[\frac{1}{n}\sum_{j=1}^{n}\sigma_j f(z_j)\right] - \sup_{f\in F}\left[\left(\frac{1}{n}\sum_{j=1,j\neq i}^{n}\sigma_j f(z_j)\right) + \frac{1}{n}\sigma_i f(z_i')\right]\right] &(4.84)\\
&\leq \mathop{\mathbb{E}}_{\sigma_i}\left[\sup_{f\in F}\left(\frac{1}{n}\sigma_i(f(z_i) - f(z_i'))\right)\right] &(4.85)\\
&\leq \frac{1}{n}, &(4.86)
\end{aligned}$$

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

$$\Pr[\tilde{g} - \mathbb{E}[\tilde{g}] \geq \epsilon] \leq \exp(-2n\epsilon^2), \quad \text{and} \quad \Pr[\tilde{g} - \mathbb{E}[\tilde{g}] \leq -\epsilon] \leq \exp(-2n\epsilon^2). \tag{4.87}$$

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 $\geq 1-\delta$,

$$\begin{aligned}
\sup_{f\in\mathcal{F}}\left[\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \mathbb{E}[f]\right] = g &\leq \mathbb{E}[g] + \epsilon && \text{(Step 1)} &(4.88)\\
&\leq 2R_n(\mathcal{F}) + \epsilon && \text{(Step 2)} &(4.89)\\
&\leq 2(R_S(\mathcal{F}) + \epsilon) + \epsilon && \text{(Step 3)} &(4.90)\\
&= 2R_S(\mathcal{F}) + 3\epsilon, && &(4.91)
\end{aligned}$$

as required. □

Setting $\mathcal{F}$ to be a family of loss functions bounded by $[0,1]$ in Theorem 4.18 gives the following corollary:

Corollary 4.19. *Let $\mathcal{F}$ 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*

$$\hat{L}(h) - L(h) \leq 2R_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \quad \text{for all } h \in \mathcal{H}. \tag{4.92}$$

Remark 4.20. If we want to bound the generalization gap by the average Rademacher complexity instead, we can replace the RHS of (4.92) with $2R_n(\mathcal{F}) + \sqrt{\frac{\log(2/\delta)}{2n}}$.

Interpretation of Corollary 4.19. 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 4.19 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 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 4.3.)

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.

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

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

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

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

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

4.6 Covering number upper bounds 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 $z_1, \dots, z_n$,

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

we only care about the output of function $f \in \mathcal{F}$, and not the function itself (i.e. it is sufficient for our purposes to know $f(z_1), \dots, f(z_n)$, but not know f). In other words, we can characterize $f \in \mathcal{F}$ by $f(z_1), \dots, f(z_n)$. In the sequel, we will take advantage of this simplification from the (potentially large) space of all functions $\mathcal{F}$ to the *output space*,

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

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 $\mathcal{F}$:

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

In other words, the complexity of $\mathcal{F}$ can be also interpreted as how much the vectors in Q can be correlated with a random vector σ. See Figure 4.3 for an illustration of this idea. One can also view $\mathbb{E}_\sigma\left[\sup_{v\in\mathcal{Q}} \frac{1}{n}\langle\sigma, v\rangle\right]$ as a complexity measure for the set Q. If we replace σ by a Gaussian vector with spherical covariance, then the corresponding quantity (without the $\frac{1}{n}$ scaling), $\mathbb{E}_{g\sim N(0,I)}\left[\sup_{v\in\mathcal{Q}}\langle g, v\rangle\right]$, is often referred to as the Gaussian complexity of the set Q. (It turns out that Gaussian complexity and Rademacher complexity are closely related.)

Another corollary of this is that the empirical Rademacher complexity only depends on the functionality of $\mathcal{F}$ but not 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 $Q_\mathcal{F}$.

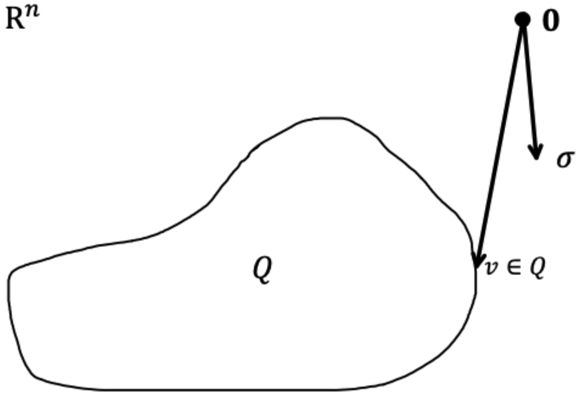


Figure 4.3: We can view empirical Rademacher complexity as the expectation of the maximum inner product between σ and $v \in Q$.

Rademacher complexity of finite hypothesis classes. In practice, we cannot directly evaluate the Rademacher complexity, so we instead bound its value using quantities that are computable. Given finite $|\mathcal{Q}|$, we often rely on the following bound, which is also known as Massart's finite lemma:

Proposition 4.6.1. Let $\mathcal{F}$ be a collection of functions mapping $Z \mapsto \mathbb{R}$ and let $\mathcal{Q}$ be defined as in (4.98). Assume that $\frac{1}{\sqrt{n}}\|v\|_2 \le M < \infty$ for all $v \in \mathcal{Q}$. Then,

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

We prove a (slightly) simplified version of this result in Problem 3(c) of Homework 2, so we omit the proof of Massart's lemma here. Using Massart's lemma, we can also bound the Rademacher complexity in terms of $\mathcal{F}$. Restating the assumption accordingly,

Corollary 4.21. Let $\mathcal{F}$ be a collection of functions mapping $Z \mapsto \mathbb{R}$. If $\sqrt{\frac{1}{n}\sum_{i=1}^n f(z_i)^2} \leq M$ for all $f \in \mathcal{F}$, then

$$R_S(\mathcal{F}) \leq \sqrt{\frac{2M^2 \log |\mathcal{F}|}{n}}. \tag{4.101}$$

Note that Corollary 4.21 yields a looser bound than Massart's lemma since $|\mathcal{Q}| \leq |\mathcal{F}|$.

In practice, we rarely apply Massart's lemma directly since $|\mathcal{Q}|$ is typically infinite. In the sequel, we discuss alternative approaches to bounding the Rademacher complexity that are appropriate for this setting.

Bounding Rademacher complexity using ϵ-covers. When $|\mathcal{Q}|$ is infinite, we can apply the same discretization trick that we used to prove the generalization bound for an infinite-hypothesis space. This time, instead of trying to cover the parameter space, we will cover the output space. To this end, we first recall a few definitions concerning ϵ-covers.

Definition 4.22. $\mathcal{C}$ 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 $\rho(v, v') \leq \epsilon$.

Definition 4.23. The *covering number* is defined as the minimum size of an ϵ-cover, or explicitly:

$$N(\epsilon, \mathcal{Q}, \rho) \triangleq (\text{min size of } \epsilon\text{-cover of } \mathcal{Q} \text{ w.r.t. metric } \rho). \tag{4.102}$$

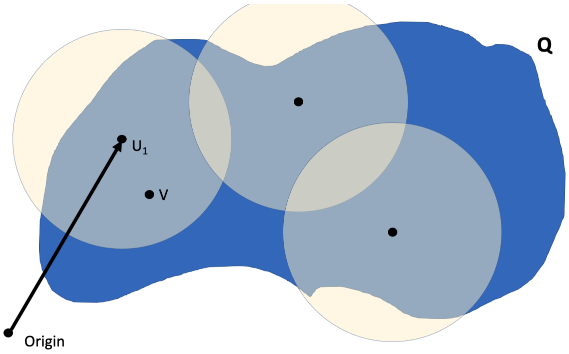


Figure 4.4: We can visualize the ϵ-cover $\mathcal{C}$ by depicting a set of ϵ-balls that cover the output space $\mathcal{Q}$. The yellow circles denote the ϵ-neighborhoods of the covering points $u_i \in \mathcal{C}$.

In subsequent derivations, we will use the metric $\rho(v, v') = \frac{1}{\sqrt{n}}\|v - v'\|_2$.

Remark 4.24. We normalize the ℓ_2 norm in ρ by $\frac{1}{\sqrt{n}}$ to simplify comparisons to the functional analysis view of the Rademacher complexity. In the literature, the ϵ-cover of $\mathcal{Q}$ defined above is also referred to as an ϵ-cover of the function class $\mathcal{F}$ under the $L_2(P_n)$ metric.[2] In particular,

$$L_2(P_n)(f, f') = \sqrt{\frac{1}{n}\sum_{i=1}^n (f(z_i) - f'(z_i))^2}. \tag{4.103}$$

[2] P_n denotes the empirical distribution, i.e. the uniform distribution over the observations $z_1, \ldots, z_n$. More generally the $L_p(Q)$ metric is defined by $\mathbb{E}_Q\left[(f(z) - f'(z))^p\right]^{1/p}$.

Recall we have established the following correspondences between the set of functions $\mathcal{F}$ and the output space $\mathcal{Q}$:

$$f \in \mathcal{F} \iff \begin{pmatrix} f(z_1) \\ \vdots \\ f(z_n) \end{pmatrix} \in \mathcal{Q} \tag{4.104}$$

We can write a trivial correspondence between both the output and function class points of view as follows:

$$N(\epsilon, \mathcal{F}, L_2(P_n)) = N\left(\epsilon, \mathcal{Q}, \frac{1}{\sqrt{n}}||\cdot||_2\right) \tag{4.105}$$

The results below will be stated in the function-space notation, but in the proofs we will shift to the $\mathcal{Q}$-formulation for the sake of clarity. In general, we prefer to reason about covering numbers on $\mathcal{Q}$ as it is more natural to analyze vector spaces compared to function spaces.

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

Theorem 4.25. *Let $\mathcal{F}$ be a family of functions $Z \mapsto [-1, 1]$. Then*

$$R_S(\mathcal{F}) \leq \inf_{\epsilon>0}\left(\epsilon + \sqrt{\frac{2\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}}\right). \tag{4.106}$$

The ϵ term can be thought of as the discretization error, while the second term is the Rademacher complexity of the finite ϵ-cover. The precise form of this complexity bound follows from Proposition 4.6.1.

Proof. Fix any $\epsilon > 0$. Let $\mathcal{C}$ be the minimal ϵ-cover of $\mathcal{Q}$ with respect to the metric $\rho(v, v') = \frac{1}{\sqrt{n}}\|v - v'\|_2$. Note that $|\mathcal{C}| = N(\epsilon, \mathcal{Q}, \frac{1}{\sqrt{n}}\|\cdot\|_2) = N(\epsilon, \mathcal{F}, L_2(P_n))$.

We aim to bound $R_S(\mathcal{F}) = \mathbb{E}_\sigma[\sup_{v\in\mathcal{Q}} \frac{1}{n}\langle v, \sigma\rangle]$ by approximating v with $v' \in \mathcal{C}$. In particular, for every point $v \in \mathcal{Q}$, choose $v' \in \mathcal{C}$ such that $\rho(v, v') \leq \epsilon$ and z is small (specifically, $\frac{1}{\sqrt{n}}\|z\|_2 \leq \epsilon$). This gives

$$\begin{aligned}
\frac{1}{n}\langle v, \sigma\rangle &= \frac{1}{n}\langle v', \sigma\rangle + \frac{1}{n}\langle v - v', \sigma\rangle && (4.107)\\
&\leq \frac{1}{n}\langle v', \sigma\rangle + \frac{1}{n}\|z\|_2\|\sigma\|_2 && (z \triangleq v - v', \text{ Cauchy-Schwarz}) \quad (4.108)\\
&\leq \frac{1}{n}\langle v', \sigma\rangle + \epsilon. && (\text{since } \|z\|_2 \leq \sqrt{n}\epsilon \text{ and } \|\sigma\|_2 \leq \sqrt{n}) \quad (4.109)
\end{aligned}$$

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

$$\begin{aligned}
R_S(\mathcal{F}) &= \mathbb{E}_\sigma\left[\sup_{v\in\mathcal{Q}} \frac{1}{n}\langle v, \sigma\rangle\right] && (4.110)\\
&\leq \mathbb{E}_\sigma\left[\sup_{v'\in\mathcal{C}} \left(\frac{1}{n}\langle v', \sigma\rangle + \epsilon\right)\right] && (4.111)\\
&= \epsilon + \mathbb{E}_\sigma\left[\sup_{v'\in\mathcal{C}} \left(\frac{1}{n}\langle v', \sigma\rangle\right)\right] && (4.112)\\
&\leq \epsilon + \sqrt{\frac{2\log|\mathcal{C}|}{n}} && (\text{Proposition 4.6.1}) \quad (4.113)\\
&= \epsilon + \sqrt{\frac{2\log N(\epsilon, \mathcal{Q}, \rho)}{n}} && (4.114)\\
&= \epsilon + \sqrt{\frac{2\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} && (\text{Remark 4.24}) \quad (4.115)
\end{aligned}$$

Since the argument above holds for any $\epsilon > 0$, we can take the infimum over all ϵ to arrive at Equation (4.106).

□

4.6.1 Chaining and Dudley's theorem

While Theorem 4.25 is useful, the bound in (4.108) 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).

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

Theorem 4.26 (Dudley's Theorem). *If $\mathcal{F}$ is a function class from $Z \mapsto \mathbb{R}$, then*

$$R_S(\mathcal{F}) \leq 12 \int_0^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{4.116}$$

Note that unlike in Theorem 4.25, we do not require $f \in \mathcal{F}$ to be bounded.

It is not obvious how (4.116) improves upon the one-step discretization bound given by (4.106). At a high level, we can interpret this bound as removing the discretization error term by averaging over different scales of ϵ. But before we can explicitly prove this claim, we motivate our approach. In the proof of Theorem 4.25, we approximated v with $v' + z$ where v' is the closest point to v in the minimal ϵ-cover of $\mathcal{Q}$, and z is the vector between v' and v. In particular,

$$\frac{1}{n}\langle v, \sigma\rangle = \frac{1}{n}\langle v', \sigma\rangle + \frac{1}{n}\langle z, \sigma\rangle \tag{4.117}$$

Then, to obtain a bound, we take a sup of both sides, but apply the sup separately to each term on the right hand side. Namely, we show that:

$$\mathbb{E}\left[\sup_v \frac{1}{n}\langle v, \sigma\rangle\right] \leq \mathbb{E}\left[\sup_{v' \in \mathcal{C}} \frac{1}{n}\langle v', \sigma\rangle\right] + \mathbb{E}\left[\sup_{z \in B_{v'}} \frac{1}{n}\langle z, \sigma\rangle\right] \tag{4.118}$$

This bound follows by observing that $\mathbb{E}[\sup(A+B)] \leq \mathbb{E}[\sup A] + \mathbb{E}[\sup B]$ since the sup on the RHS is taken separately over both terms. The difficult term to tightly bound is the last one, $\frac{1}{n}\langle z, \sigma\rangle$. In the previous derivation, we naively upper bounded $\langle z, \sigma\rangle$ using Cauchy-Schwarz,

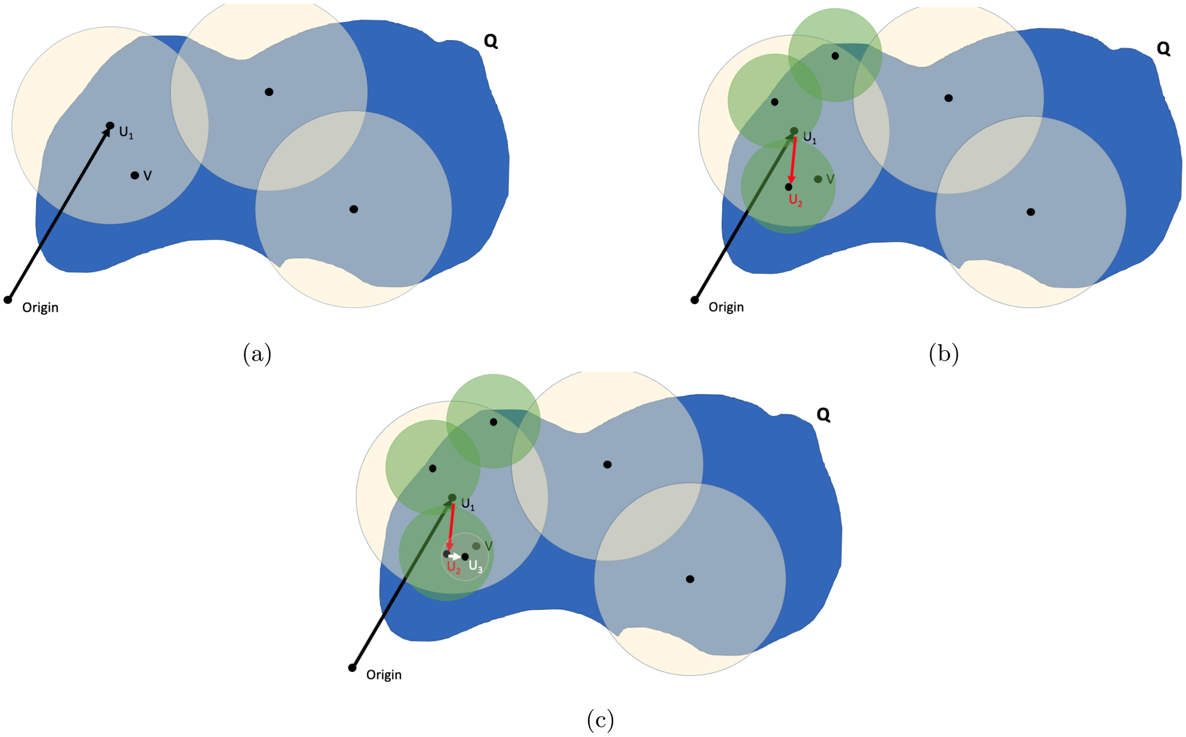
$$\frac{1}{n}\langle z, \sigma\rangle \leq \frac{\|z\|_2 \cdot \|\sigma\|_2}{n}, \tag{4.119}$$

but this bound is only tight if there exists $z \in B_{v'}$ that is perfectly correlated with σ. We claim that such perfect correlation is unlikely. Recall that the output space is defined by possible outputs of $f \in \mathcal{F}$ given n inputs. Unless our function class is extremely expressive, the set of radius ϵ around v' contained in $\mathcal{Q}$ will only be a small subset of the ϵ-ball centered at v'; thus, $\sup_z \frac{1}{n}\langle z, \sigma\rangle \ll \frac{\|z\|_2 \cdot \|\sigma\|_2}{n}$.

To precisely set up our approach, we observe that $\mathbb{E}[\sup_{z \in B_{v'}} \frac{1}{n}\langle z, \sigma\rangle]$ is itself a Rademacher complexity: $R_S(B_{v'} \cap \mathcal{Q})$. To more tightly bound $\mathbb{E}\left[\sup_{z \in B_{v'}} \frac{1}{n}\langle z, \sigma\rangle\right]$, we then repeat the ϵ-covering argument again with a smaller choice of ϵ. Intuitively, this procedure amounts to decomposing $\langle z, \sigma\rangle$ from (4.117) into another pair of terms corresponding to the new ϵ-cover and the discretization error. "Chaining" then repeats this decomposition countably many times. This procedure is illustrated visually by Figure 4.5, and we formalize this argument in the sequel.

Proof. Let $\epsilon_0 = \sup_{f \in \mathcal{F}} \max_i |f(z_i)|$, so that for all $v \in \mathcal{Q}$,

$$\epsilon_0 \geq \sqrt{\frac{1}{n}\sum_{i=1}^n f(z_i)^2} = \sqrt{\frac{1}{n}\|v\|_2^2}. \tag{4.120}$$



(a)

(b)

(c)

Figure 4.5: We depict how the chaining procedure approximates v using a sequence of progressively finer discretizations. Figure 4.5a illustrates how we first approximate v using the nearest covering point u_1, while Figures 4.5b and 4.5c describe how we refine this approximation using two finer covers, whose nearest points are denoted by u_2 and u_3, respectively.

Define $\epsilon_j = 2^{-j}\epsilon_0$ and let $\mathcal{C}_j$ be an ϵ_j-cover of $\mathcal{Q}$. Then, $\mathcal{C}_0$ is the coarsest cover of $\mathcal{Q}$, and as j increases, we obtain progressively more fine-grained covers $\mathcal{C}_j$. We can intuitively think of these covers as nested, but this is not necessary for the proof to hold. We next use this sequence of covers to define a telescoping series that equals v; the terms in this series can then be analyzed using the tools that we have developed in the prequel.

For $v \in \mathcal{Q}$, let u_i denote the nearest neighbor of v in $\mathcal{C}_i$. Note that by definition $\rho(u, v_j) \leq \epsilon_j$. Taking $u_0 = 0$, it follows from our definition of $\mathcal{C}_i$ that as $j \to \infty$, $\epsilon_j \to 0$ and $u_j \to v$. Leveraging these observations, we can express v using the following series:

$$v = u_1 + (u_2 - u_1) + (u_3 - u_2) + \cdots \tag{4.121}$$

$$= (u_1 - u_0) + (u_2 - u_1) + (u_3 - u_2) + \cdots \tag{4.122}$$

$$= \sum_{i=1}^{\infty}(u_i - u_{i-1}). \tag{4.123}$$

Substituting (4.123) in the Rademacher complexity we aim to bound, we obtain

$$\mathbb{E}\left[\sup_{v\in\mathcal{Q}}\frac{1}{n}\langle v,\sigma\rangle\right] = \mathbb{E}\left[\sup_{v\in\mathcal{Q}}\frac{1}{n}\sum_{i=1}^{\infty}\langle u_i - u_{i-1},\sigma\rangle\right] \tag{4.124}$$

$$\leq \mathbb{E}\left[\sum_{i=1}^{\infty}\sup_{u_i\in\mathcal{C}_i,u_{i-1}\in\mathcal{C}_{i-1}}\frac{1}{n}\langle u_i - u_{i-1},\sigma\rangle\right] \tag{4.125}$$

$$= \sum_{i=1}^{\infty}\mathbb{E}\left[\sup_{u_i\in\mathcal{C}_i,u_{i-1}\in\mathcal{C}_{i-1}}\frac{1}{n}\langle u_i - u_{i-1},\sigma\rangle\right]. \tag{4.126}$$

Observe that

$$\mathbb{E}\left[\sup_{u_i\in\mathcal{C}_i,u_{i-1}\in\mathcal{C}_{i-1}}\frac{1}{n}\langle u_i - u_{i-1},\sigma\rangle\right] \tag{4.127}$$

is a Rademacher complexity defined over the *finite* space $\mathcal{C}_i\times\mathcal{C}_{i-1}$, so we can use Proposition 4.6.1 (Massart's lemma) to obtain a tractable upper bound. To do so, we must first compute an upper bound on $\frac{1}{\sqrt{n}}\|u_i - u_{i-1}\|_2$:

$$\frac{1}{\sqrt{n}}\|u_i - u_{i-1}\|_2 = \frac{1}{\sqrt{n}}\|(u_i - v) - (u_{i-1} - v)\|_2 \tag{4.128}$$

$$\leq \frac{1}{\sqrt{n}}\left(\|u_i - v\|_2 - \|u_{i-1} - v\|_2\right) \tag{4.129}$$

$$\leq \epsilon_i + \epsilon_{i-1} \tag{4.130}$$

$$= 3\epsilon_i \qquad (\epsilon_{i-1} \triangleq 2\epsilon_i) \tag{4.131}$$

Now we apply Proposition 4.6.1 with $M = 3\epsilon_i$ and $|\mathcal{Q}| = |\mathcal{C}_i\times\mathcal{C}_{i-1}| \leq |\mathcal{C}_i|\cdot|\mathcal{C}_{i-1}|$.

$$\mathbb{E}\left[\sup_{u_i\in\mathcal{C}_i,u_{i-1}\in\mathcal{C}_{i-1}}\frac{1}{n}\langle u_i - u_{i-1},\sigma\rangle\right] \leq \sqrt{\frac{2(3\epsilon_i)^2\log(|\mathcal{C}_i|\cdot|\mathcal{C}_{i-1}|)}{n}} \tag{4.132}$$

$$= \frac{3\epsilon_i}{\sqrt{n}}\sqrt{2(\log|\mathcal{C}_i| + \log|\mathcal{C}_{i-1}|)} \tag{4.133}$$

$$\leq \frac{6\epsilon_i}{\sqrt{n}}\sqrt{\log|\mathcal{C}_i|} \qquad (|\mathcal{C}_i| \geq |\mathcal{C}_{i-1}|) \tag{4.134}$$

Applying (4.134) to each term in (4.126) and substituting the covering number $N(\epsilon_i,\mathcal{F},L_2(P_n))$ for $|\mathcal{C}_i|$, we obtain the following upper bound on the Rademacher complexity:

$$\mathbb{E}\left[\sup_{v\in\mathcal{Q}}\frac{1}{n}\langle v,\sigma\rangle\right] \leq \sum_{i=1}^{\infty}\frac{6\epsilon_i}{\sqrt{n}}\sqrt{\log N(\epsilon_i,\mathcal{F},L_2(P_n))}. \tag{4.135}$$

Finally, we must relate (4.135) to the target upper bound of $12\int\frac{1}{\sqrt{n}}\sqrt{\log N(\epsilon,\mathcal{F},L_2(P_n))}d\epsilon$. Examining Figure 4.6, we can make two crucial observations. First, for sufficiently large ϵ, $\log N(\epsilon,\mathcal{F},L_2(P_n)) = 0$ since one point is sufficient to construct a cover. Second, we observe that

$$(\epsilon_i - \epsilon_{i+1})\sqrt{\log|\mathcal{C}_i|} \leq \int_{\epsilon_{i+1}}^{\epsilon_i}\sqrt{\log N(\epsilon,\mathcal{F},L_2(P_n))}d\epsilon \tag{4.136}$$

since the LHS of (4.136) is the area of the dotted rectangle illustrated in Figure 4.6 while the RHS is the area under the curve for that interval. Formally, this result is equivalent to observing that the right Riemann sum underestimates the integral for monotone decreasing functions f.

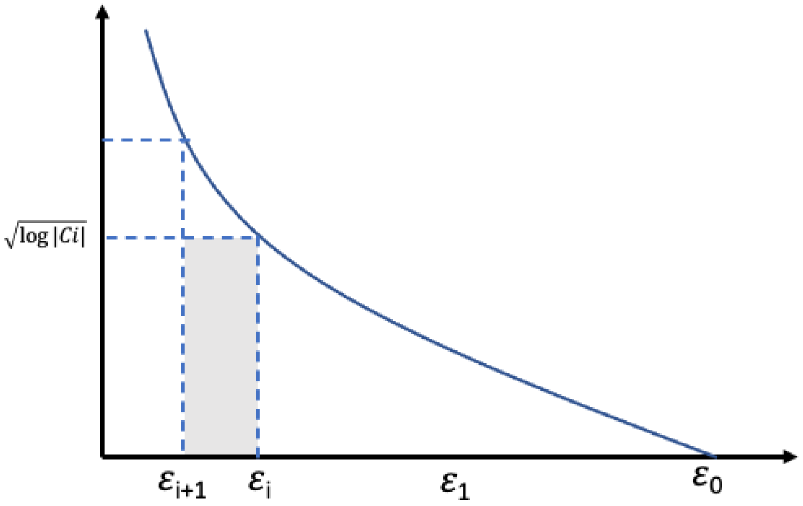


Figure 4.6: We observe that $\log N(\epsilon, \mathcal{F}, L_2(P_n))$ is monotone decreasing in ϵ. The area of the dotted rectangle formed by the vertical lines at ϵ_{i+1} and ϵ_i equals (up to a constant factor) the $i-$th term of the infinite sum derived in our proof of Dudley's theorem (4.135). The figure shows that the area of this rectangle is no larger than the integral of $\log N(\epsilon, \mathcal{F}, L_2(P_n))$ over this same interval.

Recognizing that $\epsilon_i - \epsilon_{i+1} = \frac{\epsilon_i}{2}$, we note that the LHS of (4.136) is equal (up to a constant factor) to the i-th term of (4.135). Thus,

$$\sum_{i=1}^{\infty} \frac{6\epsilon_i}{\sqrt{n}} \sqrt{\log N(\epsilon_i, \mathcal{F}, L_2(P_n))} = \frac{12}{\sqrt{n}} \sum_{i=1}^{\infty} (\epsilon_i - \epsilon_{i+1}) \sqrt{\log N(\epsilon_i, \mathcal{F}, L_2(P_n))} \tag{4.137}$$

$$\leq \frac{12}{\sqrt{n}} \int_{\epsilon_{i+1}}^{\epsilon_i} \sqrt{\log N(\epsilon_i, \mathcal{F}, L_2(P_n))} d\epsilon \tag{4.138}$$

$$= \frac{12}{\sqrt{n}} \int_{0}^{\epsilon_0} \sqrt{\log N(\epsilon, \mathcal{F}, L_2(P_n))} d\epsilon. \tag{4.139}$$

To complete the proof, observe that $\log N(\epsilon, \mathcal{F}, L_2(P_n)) = 0$ for all $\epsilon > \epsilon_0$. This allows us to extend the upper limit of the integral given by (4.139) to ∞ and yields the desired result:

$$\mathbb{E}\left[\sup_{v \in \mathcal{Q}} \frac{1}{n} \langle v, \sigma \rangle\right] \leq \frac{12}{\sqrt{n}} \int_{0}^{\infty} \sqrt{\log N(\epsilon, \mathcal{F}, L_2(P_n))} d\epsilon. \tag{4.140}$$

□

Remark 4.27. 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 (4.116) can be truncated to $[0, 1]$:

$$R_S(\mathcal{F}) \leq 12 \int_{0}^{1} \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon, \tag{4.141}$$

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

4.6.2 Translating Covering Number Bounds to Rademacher Complexity

Of course, the bound in (4.116) 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 after ignoring multiplicative and additive constants,

$$N(\epsilon, \mathcal{F}, L_2(P_n)) \approx (1/\epsilon)^R, \tag{4.142}$$

   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 (4.116) to get

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon = \int_0^1 \sqrt{\frac{R \log(1/\epsilon)}{n}} d\epsilon \approx \sqrt{\frac{R}{n}}. \tag{4.143}$$

2. If after ignoring multiplicative and additive constants, for some a,

$$N(\epsilon, \mathcal{F}, L_2(P_n)) \approx a^{R/\epsilon}, \tag{4.144}$$

   then we have $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx \frac{R}{\epsilon} \log a$. The bound in (4.116) becomes

$$\begin{aligned}
\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon &\approx \int_0^1 \sqrt{\frac{R}{n\epsilon} \log a}\, d\epsilon &(4.145)\\
&= \sqrt{\frac{R}{n} \log a} \int_0^1 \sqrt{\frac{1}{\epsilon}} d\epsilon &(4.146)\\
&= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). &(4.147)
\end{aligned}$$

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:

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon \approx \sqrt{\frac{R}{n} \log a} \underbrace{\int_0^1 \frac{1}{\epsilon} d\epsilon}_{=\infty} = \infty, \tag{4.148}$$

   i.e. the bound in (4.116) 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 (Localized Dudley's Theorem)**.** *If $\mathcal{F}$ is a function class from $Z \mapsto \mathbb{R}$, then for any fixed cutoff $\alpha \geq 0$ we have the bound*

$$R_S(\mathcal{F}) \leq 4\alpha + 12 \int_\alpha^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{4.149}$$

   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 (4.116). If we let $\alpha = 1/\mathsf{poly}(n)$, where $\mathsf{poly}(n)$ denotes some polynomial function of n, the bound in (4.149) becomes

$$R_S(\mathcal{F}) \leq \frac{1}{\mathsf{poly}(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \int_{\alpha}^{1} \frac{1}{\epsilon} d\epsilon \tag{4.150}$$

$$= \frac{1}{\mathsf{poly}(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \log(1/\alpha) \tag{4.151}$$

$$= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). \tag{4.152}$$

The last line follows by observing that $\log(1/\alpha) = \log \mathsf{poly}(n)$.

In summary, we have that $R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{R}{n}}\right)$ for covering numbers of the form $R\log(1/\epsilon)$,$\frac{R}{\epsilon}\log a$, 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 $\alpha^{1-c/2}$; then, for $\alpha = 1/\mathsf{poly}(n)$, the second term in Dudley's integral is no longer $\widetilde{O}\left(\sqrt{\frac{R}{n}}\right)$.

4.6.3 Lipschitz composition

Covering numbers also interact nicely with composition by Lipschitz functions. The following result is the analog of Talagrand's lemma for Rademacher complexity (Lemma 5.3), but its proof is much more elementary as given below. We will use this Lemma in Section 5.5 when bounding the covering number of deep nets.

Lemma 4.29. *Suppose ϕ is κ-Lipschitz, and $\rho = L_2(P_n)$. Then,*

$$\log N(\epsilon, \phi \circ \mathcal{F}, \rho) \leq \log N(\epsilon/\kappa, \mathcal{F}, \rho) \tag{4.153}$$

Proof. Let $\mathcal{C}$ denote an ϵ/κ-cover for $\mathcal{F}$. Then $\phi \circ \mathcal{C}$ is an ϵ-cover of $\phi \circ \mathcal{F}$.

$$\rho(\phi \circ f', \phi \circ f) = \sqrt{\frac{1}{n} \sum (\phi(f'(z_i)) - \phi(f(z_i)))^2} \tag{4.154}$$

$$\leq \sqrt{\frac{1}{n} \cdot \kappa^2 \sum (f'(z_i) - f(z_i))^2} \tag{4.155}$$

$$\leq \kappa \cdot \frac{\epsilon}{\kappa} = \epsilon \tag{4.156}$$

□

4.7 VC dimension and its limitations

In this section, we briefly discuss a classical notion of complexity measure of function class, VC dimension. We will show that VC dimension is an upper bound on the Rademacher complexity. 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. $\ell_{\text{0-1}}((x, y), h) = \mathbb{1}(\text{sgn}(h(x)) \neq y)$. Note that we can re-express the loss function as

$$\ell_{\text{0-1}}((x, y), h) = \frac{1 - \text{sgn}(h(x))y}{2}. \tag{4.157}$$

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

$$\begin{aligned}
R_S(\mathcal{F}) &= \mathop{\mathbb{E}}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \frac{1-\operatorname{sgn}(h(x^{(i)}))y_i}{2}\right] && (4.158)\\
&= \mathop{\mathbb{E}}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \frac{\operatorname{sgn}(h(x^{(i)}))}{2}\right] && (4.159)\\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{v\in Q} \frac{1}{n}\langle \sigma, v\rangle\right]. && (4.160)
\end{aligned}$$

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

$$\Pr\left[\frac{1}{n}\langle \sigma, v\rangle \geq t\right] \leq \exp(-nt^2/2). \tag{4.161}$$

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

$$\Pr\left[\exists v \in Q \text{ such that } \frac{1}{n}\langle \sigma, v\rangle \geq t\right] \leq |Q|\exp(-nt^2/2). \tag{4.162}$$

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

$$R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{d}{n}}\right), \tag{4.163}$$

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

$$R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{\#\text{ of parameters}}{n}}\right). \tag{4.164}$$

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 motivate 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 when we compare it to the outcome (0 or 1), its closeness 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 boundary. In the next section, we will formalize this intuition by proving that the larger this margin is, the smaller the bound on the 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.

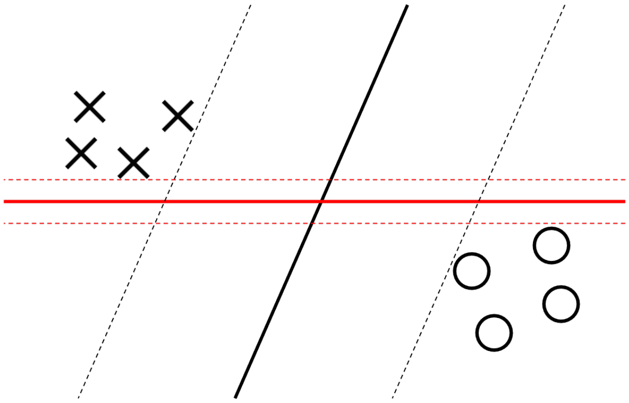


Figure 5.1: The red and black lines are two decision boundaries. The X's are positive examples and the O's are negative examples. The black line has a larger margin than the red line, and is intuitively a better classifier.

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

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

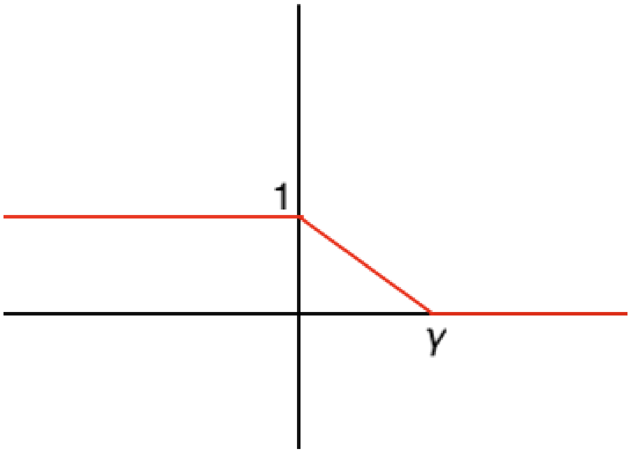


Figure 5.2: Plotted margin loss.

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 $\ell_{\text{0-1}}$ that is more sensitive to the scale of the margin on $[0,\gamma]$. Notice that $\ell_{\text{0-1}}$ is always less than or equal to the ℓ_γ when $\gamma \geq 0$, i.e.

$$\ell_{\text{0-1}}((x,y),h) = \mathbf{1}[yh(x) < 0] \leq \ell_\gamma(yh(x)) = \ell_\gamma((x,y),h) \tag{5.2}$$

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

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

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 versions of the margin loss:

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

By Corollary 4.19, we see that with probability at least $1-\delta$,

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

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

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

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

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

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 use 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.

We sketch out how one might address this issue below. The main idea is to do another union bound over γ. Choose a family $\Gamma = \left\{2^k : k \in [-B, B]\right\}$ for some B. Then, for every fixed $\gamma \in \Gamma$, with probability greater than $1 - \delta$,

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

Taking a union bound over all $\gamma \in \Gamma$, it further holds that for all $\gamma \in (0, B)$,

$$L_{0\text{-}1}(h) \leq \widehat{L}_\gamma(h) + O\left(\frac{R_S(\mathcal{H})}{\gamma}\right) + \widetilde{O}\left(\sqrt{\frac{\log \frac{1}{\delta}}{n}}\right) + \widetilde{O}\left(\sqrt{\frac{\log B}{n}}\right). \tag{5.15}$$

Last, choose the largest $\gamma \in \Gamma$ such that $\gamma \leq \gamma_{\min}$. Then, for this value of γ, our desired bound directly follows from the bound in (5.15). Namely, we have that $\widehat{L}_\gamma(h) = 0$ and $O\left(\frac{R_S(\mathcal{H})}{\gamma}\right) = O\left(\frac{R_S(\mathcal{H})}{\gamma_{\min}}\right)$. The additional term, $\widetilde{O}\left(\sqrt{\frac{\log B}{n}}\right)$, is the price exacted by the uniform convergence argument required to correct the heuristic bound given in (5.13).

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*

$$R_S(\mathcal{H}) \leq \frac{B}{n}\sqrt{\sum_{i=1}^n \left\|x^{(i)}\right\|_2^2}, \tag{5.16}$$

and

$$R_n(\mathcal{H}) \leq \frac{BC}{\sqrt{n}}. \tag{5.17}$$

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 (5.16). By definition,

$$
\begin{aligned}
R_S(\mathcal{H}) &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_2 \le B} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \left\langle w, x^{(i)} \right\rangle\right] && (5.18)\\
&= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_2 \le B} \left\langle w, \sum_{i=1}^{n} \sigma_i x^{(i)} \right\rangle\right] && (5.19)\\
&= \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\left\|\sum_{i=1}^{n} \sigma_i x^{(i)}\right\|_2\right] && (\sup_{\|w\|_2 \le B}\langle w, v\rangle = B\left\|v\right\|_2) \quad (5.20)\\
&\le \frac{B}{n}\sqrt{\mathop{\mathbb{E}}_{\sigma}\left[\left\|\sum_{i=1}^{n} \sigma_i x^{(i)}\right\|_2^2\right]} && (\text{Jensen's ineq. for } \alpha \mapsto \alpha^2) \quad (5.21)\\
&= \frac{B}{n}\sqrt{\mathop{\mathbb{E}}_{\sigma}\left[\sum_{i=1}^{n}\left(\sigma_i^2 \left\|x^{(i)}\right\|_2^2 + \left\langle \sigma_i x^{(i)}, \sum_{j \ne i}^{n} \sigma_j x^{(j)} \right\rangle\right)\right]} && (5.22)\\
&= \frac{B}{n}\sqrt{\sum_{i=1}^{n}\left\|x^{(i)}\right\|_2^2}. && (\sigma_i \text{ indep. and } \mathbb{E}[\sigma_i] = 0) \quad (5.23)
\end{aligned}
$$

This completes the proof of (5.16) for the empirical Rademacher complexity. The bound on the average Rademacher complexity in (5.17) follows from taking the expectation of both sides to get

$$
R_n(\mathcal{H}) = \mathbb{E}\left[R_S(\mathcal{H})\right] = \frac{B}{n}\mathbb{E}\left[\sqrt{\sum_{i=1}^{n}\left\|x^{(i)}\right\|_2^2}\right] \le \frac{B}{n}\sqrt{\sum_{i=1}^{n}\mathbb{E}\left[\left\|x^{(i)}\right\|_2^2\right]} \le \frac{BC}{\sqrt{n}}, \tag{5.24}
$$

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*

$$
R_S(\mathcal{H}) \le BC\sqrt{\frac{2\log(2d)}{n}}. \tag{5.25}
$$

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.,*

$$\mathcal{Q} \subset \{v \in \mathbb{R}^n \mid \|v\|_2 \le M\sqrt{n}\}. \tag{5.26}$$

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

$$\mathop{\mathbb{E}}_{\sigma}\left[\sup_{v \in \mathcal{Q}} \frac{1}{n}\langle \sigma, v\rangle\right] \le M\sqrt{\frac{2\log|\mathcal{Q}|}{n}}. \tag{5.27}$$

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

$$\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} f(z^{(i)})^2 \le M^2, \tag{5.28}$$

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

$$R_S(\mathcal{F}) \le M\sqrt{\frac{2\log|\mathcal{F}|}{n}}, \quad \textit{and} \quad R_n(\mathcal{F}) \le M\sqrt{\frac{2\log|\mathcal{F}|}{n}}. \tag{5.29}$$

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,

$$\begin{aligned} R_S(\mathcal{H}) &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_1 \le B} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \left\langle w, x^{(i)}\right\rangle\right] && (5.30)\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_1 \le B} \left\langle w, \sum_{i=1}^{n} \sigma_i x^{(i)}\right\rangle\right] && (5.31)\\ &= \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\left\|\sum_{i=1}^{n} \sigma_i x^{(i)}\right\|_\infty\right], && (5.32) \end{aligned}$$

where the last equality is because $\sup_{\|w\|_1 \le B}\langle w, v\rangle = B\|v\|_\infty$, 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 $\sup_{\|w\|_1 \le 1}\langle w, v\rangle$ 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

$$\begin{aligned} R_S(\mathcal{H}) &= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|w\|_1 \le B} \left\langle w, \sum_{i=1}^{n} \sigma_i x^{(i)}\right\rangle\right] && (5.33)\\ &= \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\max_{w \in \mathcal{W}} \left\langle w, \sum_{i=1}^{n} \sigma_i x^{(i)}\right\rangle\right] && (5.34)\\ &= BR_S(\bar{\mathcal{H}}). && (5.35) \end{aligned}$$

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

$$\frac{1}{n}\sum_{i=1}^{n}\left\langle w, x^{(i)}\right\rangle^2 \le \frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)}\right\|_\infty^2 \le \frac{1}{n}\sum_{i=1}^{n} C^2 = C^2, \tag{5.36}$$

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

$$R_S(\mathcal{H}) = BR_S(\bar{\mathcal{H}}) \leq BC\sqrt{\frac{2\log|\bar{\mathcal{H}}|}{n}} = BC\sqrt{\frac{2\log(2d)}{n}}, \tag{5.37}$$

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

  $$\|w\|_2 \|x\|_2 \geq \sqrt{d}\|w\|_2 \|x\|_\infty \geq \|w\|_1 \|x\|_\infty . \tag{5.38}$$

- 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

  $$\|w\|_2 \|x\|_2 \leq \frac{1}{\sqrt{d}} \|w\|_1 \cdot \sqrt{d}\|x\|_\infty \leq \|w\|_1 \|x\|_\infty . \tag{5.39}$$

  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

$$\text{generalization loss} \lesssim \frac{\|w\|_2 \|x\|_2}{\sqrt{n}\gamma_{\min}} + \text{low-order term.} \tag{5.40}$$

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*

$$\mathcal{H} = \left\{ f_\theta \mid \|w\|_2 \leq B_w,\ \|u_i\|_2 \leq B_u,\ \forall i \in \{1, 2, \ldots, m\} \right\}, \tag{5.41}$$

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

$$R_n(\mathcal{H}) \leq 2 B_w B_u C \sqrt{\frac{m}{n}}. \tag{5.42}$$

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. We also note that the theorem can be generalized straightforwardly to the setting where the w and U are jointly constrained in the sense that we set $\mathcal{H} = \{f_\theta \mid \|w\|_2 \cdot (\max_i \|u_i\|_2) \leq B\}$ and obtain the generalization bound $R_n(\mathcal{H}) \leq 2BC\sqrt{\frac{m}{n}}$. However, the $\sqrt{m}$ dependency still exists under this formulation of $\mathcal{H}$. Nevertheless, we now derive this bound.

Proof. By definition,

$$
\begin{aligned}
R_S(\mathcal{H}) &= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\theta} \frac{1}{n}\sum_{i=1}^{n}\sigma_i \left\langle w, \phi(Ux^{(i)})\right\rangle\right] && (5.43)\\
&= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{U:\|u_j\|_2\le B_u}\ \sup_{\|w\|_2\le B_w}\left\langle w, \sum_{i=1}^{n}\sigma_i\phi(Ux^{(i)})\right\rangle\right] && (5.44)\\
&= \frac{B_w}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{U:\|u_j\|_2\le B_u}\left\|\sum_{i=1}^{n}\sigma_i\phi(Ux^{(i)})\right\|_2\right] && (\sup_{\|w\|_2\le B}\langle w,v\rangle = B\|v\|_2) \quad (5.45)\\
&\le \frac{B_w\sqrt{m}}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{U:\|u_j\|_2\le B_u}\left\|\sum_{i=1}^{n}\sigma_i\phi(Ux^{(i)})\right\|_\infty\right] && (\|v\|_2 \le \sqrt{m}\,\|v\|_\infty) \quad (5.46)\\
&= \frac{B_w\sqrt{m}}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{U:\|u_j\|_2\le B_u}\ \max_{1\le j\le m}\left|\sum_{i=1}^{n}\sigma_i\phi(u_j^\top x^{(i)})\right|\right] && (5.47)\\
&= \frac{B_w\sqrt{m}}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|u\|_2\le B_u}\left|\sum_{i=1}^{n}\sigma_i\phi(u^\top x^{(i)})\right|\right] && (5.48)\\
&\le \frac{2B_w\sqrt{m}}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|u\|_2\le B_u}\sum_{i=1}^{n}\sigma_i\phi(u^\top x^{(i)})\right] && (\text{by Lemma 5.12}) \quad (5.49)\\
&\le \frac{2B_w\sqrt{m}}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\|u\|_2\le B_u}\sum_{i=1}^{n}\sigma_i u^\top x^{(i)}\right], && (5.50)
\end{aligned}
$$

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 (5.49) is the Rademacher complexity for $\{x \mapsto \phi(u^\top x) \mid \|u\|_2 \le B_u\}$: this is the family that we are applying the contraction lemma to.)

We now observe that the expectation in (5.50) is the Rademacher complexity of the family of linear models $\{x \mapsto \langle u, x\rangle \mid \|u\|_2 \le B_u\}$. Thus, applying Theorem 5.7 yields

$$R_S(\mathcal{H}) \le \frac{2B_w\sqrt{m}}{n}B_u\sqrt{\sum_{i=1}^{n}\left\|x^{(i)}\right\|_2^2}. \tag{5.51}$$

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

$$
\begin{aligned}
R_n(\mathcal{H}) &= \mathbb{E}\left[R_S(\mathcal{H})\right] && (5.52)\\
&\le \frac{2B_wB_u\sqrt{m}}{n}\mathbb{E}\left[\sqrt{\sum_{i=1}^{n}\left\|x^{(i)}\right\|_2^2}\right] && (5.53)\\
&\le \frac{2B_wB_u\sqrt{m}}{n}C\sqrt{n} && (5.54)\\
&= 2B_wB_uC\sqrt{\frac{m}{n}}, && (5.55)
\end{aligned}
$$

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.

5.3.1 Refined bounds

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.

$$\alpha\phi(x) = \phi(\alpha x) \qquad \forall \alpha > 0. \tag{5.56}$$

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

$$w_i \cdot \phi\left(u_i^\top x^{(i)}\right) = (\lambda_i w_i) \cdot \phi\left(\left(\frac{u_i}{\lambda_i}\right)^\top x^{(i)}\right). \tag{5.57}$$

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 complexity measure used in the hypothesis class (5.41) 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* $B > 0$ *consider the hypothesis class*

$$\mathcal{H} = \{f_\theta \mid C(\theta) \le B\}. \tag{5.58}$$

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

$$R_S(\mathcal{H}) \le \frac{2BC}{\sqrt{n}}. \tag{5.59}$$

Remark 5.11. 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 explicitly grows with the total number of neurons. In fact, Theorem 5.10 is strictly stronger than Theorem 5.9 as elaborated below. Note that

$$\begin{aligned}\sum |w_j|\|u_j\|_2 &\le \left(\sum |w_j|^2\right)^{1/2}\left(\sum \|u_j\|_2^2\right)^{1/2} && \text{(by Cauchy-Schwarz inequality)}\\ &\le \|w\|_2 \cdot \sqrt{m} \cdot \max_j \|u_j\|_2 && (5.60)\end{aligned}$$

Therefore, if we consider $\mathcal{H}^1 = \{f_\theta \mid \sum |w_j|\|u_j\|_2 \le B'\}$ and $\mathcal{H}^2 = \{f_\theta \mid \|w\|_2 \cdot \sqrt{m} \cdot \max_j \|u_j\|_2 \le B'\}$, then either Theorem 5.10 on $\mathcal{H}^1$ or Theorem 5.9 on $\mathcal{H}^2$ gives the same generalization bound $O(B'/\sqrt{n})$, but $\mathcal{H}^1 \supset \mathcal{H}^2$.

Moreover, Theorem 5.10 is stronger as we have more neurons—this is because the hypothesis class $\mathcal{H}$ as defined in (5.58) is bigger as m increases. Because of this, it's possible to obtain a generalization guarantee that decreases as m increases, as shown in Section 5.4.2.

Proof of Theorem 5.10. Due to the positive homogeneity of the ReLU function ϕ, it will be useful to define the ℓ_2-normalized weight vector $\bar{u}_j \triangleq u_j/\|u_j\|_2$ so that $\phi\left(u_j^\top x\right) = \|u_j\|_2 \cdot \phi(\bar{u}_j^\top x)$. The empirical Rademacher complexity satisfies

$$\begin{aligned}R_S(\mathcal{H}) &= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_\theta \sum_{i=1}^n \sigma_i f_\theta\left(x^{(i)}\right)\right] && (5.61)\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_\theta \sum_{i=1}^n \sigma_i\left[\sum_{j=1}^m w_j\phi\left(u_j^T x^{(i)}\right)\right]\right] && \text{(by dfn of } f_\theta) \quad (5.62)\\ &= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_\theta \sum_{i=1}^n \sigma_i\left[\sum_{j=1}^m w_j\|u_j\|_2\phi\left(\bar{u}_j^T x^{(i)}\right)\right]\right] && \text{(by positive homogeneity of } \phi) \quad (5.63)\end{aligned}$$

$$= \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\theta}\sum_{j=1}^{m} w_j\|u_j\|_2\left[\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}_j^T x^{(i)}\right)\right]\right] \tag{5.64}$$

$$\le \frac{1}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\theta}\sum_{j=1}^{m} |w_j|\|u_j\|_2 \max_{k\in[n]}\left|\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}_k^T x^{(i)}\right)\right|\right] \qquad \left(\text{because } \sum_j \alpha_j\beta_j \le \sum_j |\alpha_j| \max_k |\beta_k|\right) \tag{5.65}$$

$$\le \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\theta=(w,U)}\max_{k\in[n]}\left|\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}_k^T x^{(i)}\right)\right|\right] \qquad (\text{because } C(\theta)\le B) \tag{5.66}$$

$$= \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\bar{u}:\|\bar{u}\|_2=1}\left|\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}^T x^{(i)}\right)\right|\right] \tag{5.67}$$

$$\le \frac{B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\bar{u}:\|\bar{u}\|_2\le 1}\left|\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}^T x^{(i)}\right)\right|\right] \tag{5.68}$$

$$\le \frac{2B}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\bar{u}:\|\bar{u}\|_2\le 1}\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}^T x^{(i)}\right)\right] \qquad (\text{see Lemma 5.12}) \tag{5.69}$$

$$= 2BR_S(\mathcal{H}'), \tag{5.70}$$

where $\mathcal{H}' = \left\{x \mapsto \phi(\bar{u}^\top x) : \bar{u}\in\mathbb{R}^d, \|\bar{u}\|_2 \le 1\right\}$. By Talagrand's lemma, since ϕ is 1-Lipschitz, $R_S(\mathcal{H}') \le R_S(\mathcal{H}'')$ where $\mathcal{H}'' = \left\{x \mapsto \bar{u}^\top x : \bar{u}\in\mathbb{R}^d, \|\bar{u}\|_2 \le 1\right\}$ 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 5.12 used in the second-to-last inequality. Notably, the lemma's assumption holds in the current context, since

$$\sup_{\theta}\langle\sigma, f_\theta(x)\rangle = \sup_{\bar{u}:\|\bar{u}\|_2\le 1}\sum_{i=1}^{n}\sigma_i\phi\left(\bar{u}^\top x^{(i)}\right) \ge 0. \tag{5.71}$$

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

Lemma 5.12. *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,*

$$\mathbb{E}_\sigma\left[\sup_{\theta}|\langle\sigma, f_\theta(x)\rangle|\right] \le 2\mathbb{E}_\sigma\left[\sup_{\theta}\langle\sigma, f_\theta(x)\rangle\right]. \tag{5.72}$$

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

$$\sup_{\theta}|\langle\sigma, f_\theta(x)\rangle| = \sup_{\theta}\left[\phi\left(\langle\sigma, f_\theta(x)\rangle\right) + \phi\left(\langle-\sigma, f_\theta(x)\rangle\right)\right] \tag{5.73}$$

$$\le \sup_{\theta}\phi\left(\langle\sigma, f_\theta(x)\rangle\right) + \sup_{\theta}\phi\left(\langle-\sigma, f_\theta(x)\rangle\right) \tag{5.74}$$

$$= \sup_{\theta}\left\langle\sigma, f_\theta(x)\right\rangle + \sup_{\theta}\left\langle-\sigma, f_\theta(x)\right\rangle. \tag{5.75}$$

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

5.4 More implications and discussions on two-layer neural nets

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

5.4.1 Connection to ℓ_2 regularization

Recall that margin theory yields

$$\text{for all } \theta, \quad L_{0\text{-}1}(\theta) \leq \frac{2R_S(\mathcal{H})}{\gamma_{\min}} + \widetilde{O}\left(\sqrt{\frac{\log(2/\delta)}{n}}\right), \tag{5.76}$$

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

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

or equivalently,

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

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:

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

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

$$\begin{aligned} \min_\theta C_{\ell_2}(\theta) &= \min_\theta \min_\lambda \left(\frac{1}{2}\sum_{j=1}^{m} \lambda_j^2 |w_j|^2 + \frac{1}{2}\sum_{j=1}^{m} \lambda_j^{-2} \|u_j\|_2^2\right) &(5.77)\\ &= \min_\theta \sum_{j=1}^{m} |w_j| \cdot \|u_j\|_2 &(5.78)\\ &= \min_\theta C(\theta) &(5.79) \end{aligned}$$

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.

5.4.2 Generalization bounds that are decreasing 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 (5.76) as

$$L(\widehat{\theta}_m) \leq \frac{4C}{\sqrt{n}} \cdot \frac{1}{\gamma^*_m} + (\text{lower-order terms}), \tag{5.80}$$

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

$$(\widetilde{w}_i, \widetilde{u}_i) = \begin{cases} (w_i, u_i) & i \leq m, \\ (0,0) & \text{otherwise.} \end{cases} \tag{5.81}$$

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

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

Since γ^*_m is non-decreasing in m, the quantity

$$\gamma^*_\infty = \lim_{m\to\infty} \gamma^*_m \tag{5.82}$$

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:

$$g_{\mu,\varphi}(x) \triangleq \langle \mu, \varphi(x) \rangle. \tag{5.83}$$

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:

$$\begin{aligned} \text{maximize} \quad & \gamma_{min} \coloneqq \min_{i\in[n]} y^{(i)} \langle \mu, \varphi(x^{(i)}) \rangle \\ \text{subject to} \quad & \|\mu\|_p \leq 1. \end{aligned} \tag{5.84}$$

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,*

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

is an "infinite-dimensional vector" that contains an entry $\phi(u^\top x)$ 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

$$\gamma^*_\infty = \gamma^*_{\ell_1}, \tag{5.85}$$

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 $\{u_i\}$:

$$\widetilde{w}_i \triangleq w_i \cdot \|u_i\|_2, \qquad \overline{u}_i \triangleq \frac{u_i}{\|u_i\|_2}. \tag{5.86}$$

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. $\overline{u}_i \in \mathcal{S}^{d-1}$). 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

$$\mu[u] = \sum_{i \in \mathcal{S}^{d-1}} \widetilde{w}_i \delta(u - \overline{u}_i), \tag{5.87}$$

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

$$\begin{aligned} g_{\mu,\phi_{\text{relu}}}(x) &= \int_{S^{d-1}} \mu[u]\phi_{\text{relu}}(x)[u]du & (5.88) \\ &= \int_{S^{d-1}} \sum_{i \in \mathcal{S}^{d-1}} \widetilde{w}_i \delta(u - \overline{u}_i)\phi\left(\overline{u}^\top x\right) du & (5.89) \\ &= \sum_{i \in \mathcal{S}^{d-1}} \widetilde{w}_i \phi\left(\overline{u}_i^\top x\right), & (5.90) \end{aligned}$$

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

$$\|\mu\|_1 = \sum_{i=1}^{\infty} |\widetilde{w}_i| = \sum_{i=1}^{\infty} |w_i| \cdot \|u_i\|_2 \leq 1, \tag{5.91}$$

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 $m = n+1$:

$$\gamma^*_1 \leq \gamma^*_2 \leq \cdots \leq \gamma^*_{n+1} = \gamma^*_\infty. \tag{5.92}$$

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.

5.5 Deep neural nets (via covering number)

In Section 4.6.2, we discuss how strong our bounds on covering number need to be in order to get a useful result. Here we describe some situations in which we know how to obtain these covering number bounds for concrete models such as linear models and neural networks.

5.5.1 Preparation: covering number for linear models

First, consider the following covering number bound for linear models:

Theorem 5.16 ([Zhang, 2002]). *Suppose $x^{(1)}, \cdots, x^{(n)} \in \mathbb{R}^d$ are n data points, and p, q satisfies $1/p + 1/q = 1$ and $2 \leq p \leq \infty$. Assume that $||x^{(i)}||_p \leq C$ for all i. Let:*

$$\mathcal{F}_q = \{x \mapsto w^\top x : ||w||_q \leq B\} \tag{5.93}$$

and let $\rho = L_2(P_n)$. Then, $\log N(\epsilon, \mathcal{F}_q, \rho) \leq \left[\frac{B^2C^2}{\epsilon^2}\right] \log_2(2d+1)$. When $p = 2, q = 2$, we further obtain that:

$$\log N(\epsilon, \mathcal{F}_2, \rho) \leq \left[\frac{B^2C^2}{\epsilon^2}\right] \log_2(2\min(n,d)+1) \tag{5.94}$$

Remark 5.17. Applying (4.152) to the covering number bound derived above with $R = B^2C^2$, we conclude that the Rademacher complexity of this class of linear models satisfies

$$R_S(\mathcal{F}_q) \leq \widetilde{O}\left(\frac{BC}{\sqrt{n}}\right). \tag{5.95}$$

We also prove this result without relying on Dudley's theorem in Theorem 5.5.

Next, we consider multivariate linear functions as they are building blocks for multi-layer neural networks. Let $M = (M_1, \cdots, M_n) \in \mathbb{R}^{m \times n}$ and $\|M\|_{2,1} = \sum_{i=1}^n \|M_i\|_2$. Then, $\|M^\top\|_{2,1}$ denotes the sum of the ℓ_2 norms of the rows of M.

Theorem 5.18. *Let $\mathcal{F} = \{x \to Wx : W \in \mathbb{R}^{m \times d}, ||W^\top||_{2,1} \leq B\}$ and let $C = \sqrt{\frac{1}{n}\sum_{i=1}^n ||x^{(i)}||_2^2}$. Then,*

$$\log N(\epsilon, \mathcal{F}, L_2(P_n)) \leq \left[\frac{c^2B^2}{\epsilon^2}\right] \ln(2dm). \tag{5.96}$$

Remark 5.19. In some sense, Theorem 5.18 arises from treating each dimension of the multivariate problem independently. We can view the linear layer as applying m different linear functions. Explicitly, if $W = \begin{pmatrix} w_1^\top \\ \vdots \\ w_m^\top \end{pmatrix}$ and $Wx = \begin{pmatrix} w_1^\top x \\ \vdots \\ w_m^\top x \end{pmatrix}$, then as we expect, $\|W^\top\|_{2,1} = \sum \|w_i\|_2$.

5.5.2 Deep neural networks

In this lecture, we discuss a bound on the Rademacher complexity of a dense neural network. We set up notation as follows: W_i denotes the linear weight matrix at the i-th layer of the neural network, we have a total of r layers, and σ is the activation function which is 1-Lipschitz (for example, ReLU, softmax, or sigmoid). If the input is a vector x, the neural network's output can be represented as follows:

$$f_\theta(x) = W_r\sigma(W_{r-1}\sigma(\cdots\sigma(W_1x)\ldots)), \tag{5.97}$$

Using this notation, we establish an upper bound on the Rademacher complexity of a dense neural network.

Theorem 5.20 ([Bartlett et al., 2017])**.** *Suppose that* $\forall i, \|x^{(i)}\|_2 \le c$ *and let*

$$\mathcal{F} = \{f_\theta : \|W_i\|_{\text{op}} \le \kappa_i, \|W_i^\top\|_{2,1} \le b_i\}. \tag{5.98}$$

Then,

$$R_S(\mathcal{F}) \le \frac{c}{\sqrt{n}} \cdot \underbrace{\left(\prod_{i=1}^r \kappa_i\right)}_{\text{(I)}} \cdot \underbrace{\left(\sum_{i=1}^r \frac{b_i^{2/3}}{\kappa_i^{2/3}}\right)^{3/2}}_{\text{(II)}}. \tag{5.99}$$

We use $\|W\|_{\text{op}}$ to denote the operator norm (or spectral norm) of W, and recall that $\|W_i^\top\|_{2,1}$ denotes the sum of the ℓ_2 norms of the rows of W_i. Examining (5.99), we see that (II) is relatively small as it is a sum of matrix norms, and so the bound is dominated by (I), which is a product of matrix norms.

Remark 5.21. We note that $f(x) = Wx$ is Lipschitz with a Lipschitz constant of $\|W\|_{\text{op}}$. This is because

$$\|f(x) - f(y)\|_2 = \|Wx - Wy\|_2 \tag{5.100}$$

$$\le \|W\|_{\text{op}}\|x - y\|_2 \qquad (\|W\|_{\text{op}} = \max_{x:\|x\|_2=1} \|Wx\|_2) \tag{5.101}$$

.

Remark 5.22. As a corollary of the above theorem, we also get a bound on the generalization error for the margin loss of the following form:

$$\text{generalization error} \le \tilde{O}\left(\frac{1}{\gamma_{\min}} \cdot \frac{1}{\sqrt{n}} \cdot \left(\prod_{i=1}^r \|W_i\|_{\text{op}}\right) \cdot \left(\sum_{i=1}^r \frac{\|W_i^\top\|_{2,1}^{2/3}}{\|W_i\|_{\text{op}}^{2/3}}\right)^{3/2}\right), \tag{5.102}$$

where $\gamma_{\min}$ denotes the margin.

First, we motivate the proof by presenting the main idea, and then work through each part of the proof. The main ideas of the proof can be summarized as follows:

- At a high level, we want to show that the covering number $N(\epsilon, \mathcal{F}, \rho)$ for a dense neural network is $\le \frac{R}{\epsilon^2}$. Proving this would enable us to apply Theorem 4.28 to get a bound on the Rademacher Complexity.
- To bound the covering number for a dense neural network, we use ϵ-covers to cover each layer of f_θ separately, and then combine them to prove that there exists an ϵ-cover of the original function f_θ.

- To combine the ϵ-covers of each layer, we use the Lipschitzness of each layer.
- We control and approximate the error propagation that is introduced through discretizing each layer using ϵ_i-coverings in order to get a reasonable final ϵ.

As a prelude to the proof of Theorem 5.20, let us abstractify each layer of $\mathcal{F}$ as $\mathcal{F}_i$ where $\mathcal{F}_i$ corresponds to matrix multiplication by W_i composed with a nonlinear activation function σ. We then denote $\mathcal{F}$ as the composition of each of these (single layer) function spaces as follows:

$$\mathcal{F} = \mathcal{F}_r \circ \mathcal{F}_{r-1} \circ \cdots \circ \mathcal{F}_1 = \{f_r \circ f_{r-1} \circ \cdots f_1 : f_i \in \mathcal{F}_i\} \tag{5.103}$$

We will assume throughout that f_i is κ_i-Lipschitz, i.e.

$$\|f_i(x) - f_i(y)\|_2 \leq \kappa_i \|x - y\|_2 \tag{5.104}$$

Let us also assume, for simplicity, that $f_i(0) = 0$ and $\|x^{(j)}\|_2 \leq c$ for all $j = 1, \ldots, n$. Then, by applying the definition of Lipschitz continuity, we obtain that:

$$\|f_i(f_{i-1}(\cdots(f_1(x^{(j)}))))\|_2 \leq \underbrace{\kappa_i \cdot \kappa_{i-1} \cdots \kappa_1 \cdot c}_{\triangleq c_i} \tag{5.105}$$

We now derive an ϵ-covering of $\mathcal{F}$ in two steps:

1. Given inputs to the i^{th} layer, we construct an ϵ_i-covering of the output space of the function f_i.
2. Using the ϵ_i-covering as inputs to the $(i+1)$-th layer, we show that we can use several single layer coverings to construct an ϵ-covering for a multilayer network.

Formally, the following lemma answers the second step in the above outline. Namely, given a covering number for a single layer, we show how to compute a covering number bound for multiple layers.

Lemma 5.23. *Under the setup given above, if every input to f_i satisfies $\|z^{(j)}\|_2 \leq c_{i-1}$, we assume that*

$$\log N(\epsilon_i, \mathcal{F}_i, L_2(P_n)) \leq g(\epsilon_i, c_{i-1}).^1 \tag{5.106}$$

Then, there exists an ϵ-cover $\mathcal{C}$ of $\mathcal{F}_r \circ \cdots \circ \mathcal{F}_1$ for $\epsilon = \epsilon_r + \kappa_r \epsilon_{r-1} + \cdots + \kappa_r \kappa_{r-1} \ldots \kappa_2 \epsilon_1$ such that

$$\log |\mathcal{C}| \leq \sum_{i=1}^{r} g(\epsilon_i, c_{i-1}) \tag{5.107}$$

Proof. Let $\epsilon_1, \ldots, \epsilon_r$ be the radius for each layer. Let $\mathcal{C}_1$ be an ϵ_1-cover of $\mathcal{F}_1$. Then, for all $f_1' \in \mathcal{C}_1$, we define $\mathcal{C}_{2,f_1'}$ as an ϵ_2-covering of the set

$$\mathcal{F}_2 \circ f_1' = \{f_2(f_1'(X)) : f_2 \in \mathcal{F}_2\}. \tag{5.108}$$

Taking a union of this covering over all $f_1' \in \mathcal{C}_1$ clearly yields an ϵ-covering for $\mathcal{F}_2 \circ \mathcal{F}_2$. In paricular, if

$$\mathcal{C}_2 = \bigcup_{f_1' \in \mathcal{C}_1} \mathcal{C}_{2,f_1'}, \tag{5.109}$$

then $\mathcal{C}_2$ is an ϵ-cover of $\mathcal{F}_2 \circ \mathcal{F}_1$ with $\epsilon = \epsilon_1 \cdot \kappa_2 + \epsilon_2$. We depict this covering procedure in Figure 5.3, and we prove this claim rigorously in the sequel.

[1] If $\mathcal{F}_i$ defines a collection of linear models, then $\log N(\epsilon_i, \mathcal{F}_i, L_2(P_n)) \leq \left\lceil \frac{c_{i-1}^2}{\epsilon_i^2} \right\rceil$.

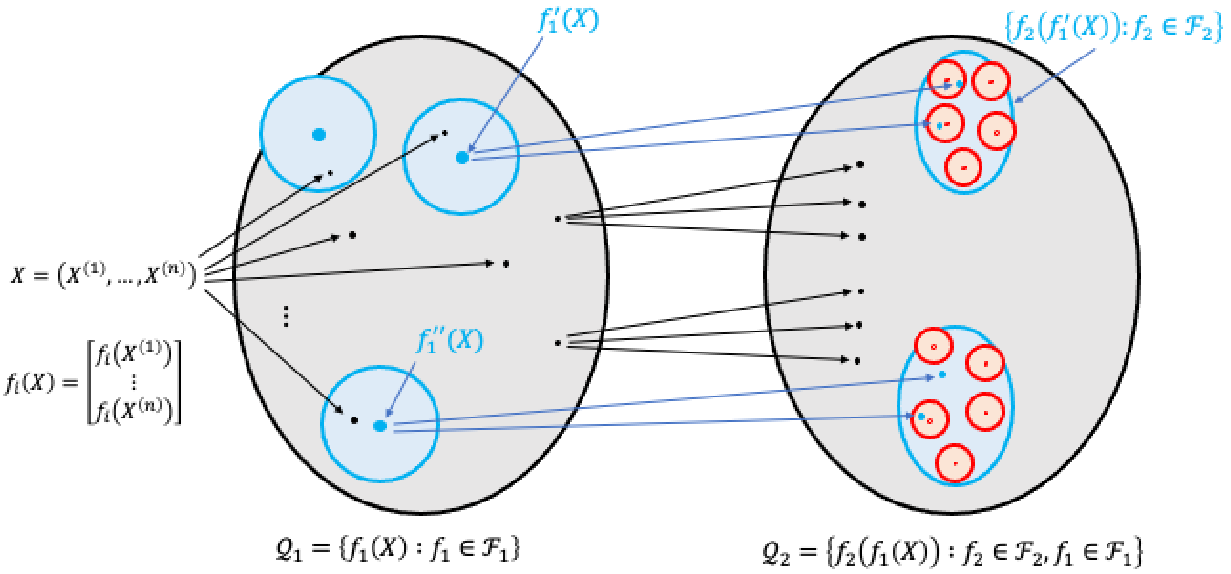

$f_1'(X)$
$\{f_2(f_1'(X)) : f_2 \in \mathcal{F}_2\}$
$X = (X^{(1)}, \dots, X^{(n)})$
$f_i(X) = \begin{bmatrix} f_i(X^{(1)}) \\ \vdots \\ f_i(X^{(n)}) \end{bmatrix}$
$f_1''(X)$
$\mathcal{Q}_1 = \{f_1(X) : f_1 \in \mathcal{F}_1\}$
$\mathcal{Q}_2 = \{f_2(f_1(X)) : f_2 \in \mathcal{F}_2, f_1 \in \mathcal{F}_1\}$

Figure 5.3: We visualize the covering strategy adopted in the proof of Lemma 5.23. The two grey sets depict the output spaces of the first and second layers, namely, $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively. The blue dots in $\mathcal{Q}_1$ are the outputs of three functions in the ϵ_1-cover $\mathcal{C}_1$, while the blue subsets of $\mathcal{Q}_2$ depict $\mathcal{F}_2 \circ f_1'$ and $\mathcal{F}_2 \circ f_1''$. The red circles show how we construct a covering, $\mathcal{C}_2$, of $\mathcal{Q}_2$. In particular, the two collections of red circles depict the $\mathcal{C}_{2,f_1'}$ and $\mathcal{C}_{2,f_1''}$ covers. Taking the union of such covers over all functions in $\mathcal{C}_1$ yields $\mathcal{C}_2$.

Next, we bound the sizes of these covers. Directly applying the assumption given by (5.106), we conclude that

$$\log \left|\mathcal{C}_{2,f_1'}\right| \leq g\left(\epsilon_2, c_1\right). \tag{5.110}$$

Then, because $\mathcal{C}_2 = \bigcup_{f_1' \in \mathcal{C}_1} \mathcal{C}_{2,f_1'}$, it immediately follows that

$$|\mathcal{C}_2| \leq |\mathcal{C}_1| \exp\left(g\left(\epsilon_2, c_1\right)\right) \tag{5.111}$$
$$\log|\mathcal{C}_2| \leq \log|\mathcal{C}_1| + g\left(\epsilon_2, c_1\right) \tag{5.112}$$
$$\leq g\left(\epsilon_1, c_0\right) + g\left(\epsilon_2, c_1\right). \tag{5.113}$$

Similarly, given $\mathcal{C}_k$, for any $f_k' \circ f_{k-1}' \circ \cdots \circ f_1' \in \mathcal{C}_k$, we construct a $\mathcal{C}_{k+1,f_k',\dots,f_1'}$ that is an ϵ_{k+1}-covering of $\mathcal{F}_{k+1} \circ f_k' \circ \cdots \circ f_1'$. We similarly define

$$\mathcal{C}_{k+1} = \bigcup_{\substack{f_i \in \mathcal{C}_i \\ i \leq k}} C_{k+1,f_k',\dots,f_1'}. \tag{5.114}$$

Then, inducting on the argument given in (5.111)-(5.113), we conclude that

$$\log|\mathcal{C}_{k+1}| \leq g\left(\epsilon_{k+1}, c_k\right) + \cdots + g\left(\epsilon_1, c_0\right) \tag{5.115}$$

Next, we show that for the above construction, the radius of the cover for $\mathcal{F}$ is

$$\epsilon = \sum_{i=1}^{r} \left(\epsilon_i \prod_{j=i+1}^{r} \kappa_j\right). \tag{5.116}$$

For any choice of $f_r \circ \cdots \circ f_1 \in \mathcal{F}_r \circ \mathcal{F}_{r-1} \circ \cdots \circ \mathcal{F}_1$, then, by definition of $\mathcal{C}_1$, there exists $f_1' \in \mathcal{C}_1$ such that

$$\rho(f_1, f_1') \le \epsilon_1. \tag{5.117}$$

Similarly, we know there exists $f_2' \circ f_1' \in \mathcal{C}_{2,f_1'}$ such that

$$\rho\left(f_2' \circ f_1', f_2 \circ f_1'\right) \le \epsilon_2. \tag{5.118}$$

We can leverage these two facts and the triangle inequality to now prove that $f_2' \circ f_1'$ is close to $f_2 \circ f_1$. Namely,

$$\begin{align}
\rho\left(f_2' \circ f_1', f_2 \circ f_1\right) &\le \rho\left(f_2' \circ f_1', f_2 \circ f_1'\right) + \rho\left(f_2 \circ f_1', f_2 \circ f_1\right) && \text{(triangle ineq.)} \tag{5.119}\\
&\le \epsilon_2 + \rho\left(f_2 \circ f_1', f_2 \circ f_1\right) && \text{(def. of } \mathcal{C}_{2,f_1'}\text{)} \tag{5.120}\\
&\le \epsilon_2 + \kappa_2 \rho\left(f_1', f_1\right) && \text{(5.104)} \tag{5.121}\\
&\le \epsilon_2 + \kappa_2 \epsilon_1 && \text{(def. of } \mathcal{C}_1\text{)} \tag{5.122}
\end{align}$$

Inducting to prove this argument for arbitrary k, we similarly apply the definition of $\mathcal{C}_{k,f_{k-1}',\ldots,f_1'}$ to conclude that there exists $f_k' \circ f_{k-1}' \circ \cdots \circ f_1' \in \mathcal{C}_k$ such that

$$\rho(f_k' \circ f_{k-1}' \circ \cdots f_1', f_k \circ f_{k-1}' \circ \cdots f_1') \le \epsilon_k \tag{5.123}$$

Then, expanding using the triangle inequality and peeling off terms by applying the definition of our ϵ_i-coverings, we again show that

$$\begin{align}
\rho\left(f_k' \circ f_{k-1}' \circ \cdots \circ f_1', f_k \circ \cdots \circ f_1\right) &\le \rho\left(f_k' \circ f_{k-1}' \circ \cdots \circ f_1', f_k \circ f_{k-1}' \circ \cdots \circ f_1'\right) \tag{5.124}\\
&\quad + \rho\left(f_k \circ f_{k-1}' \circ f_{k-2}' \circ \cdots \circ f_1', f_k \circ f_{k-1} \circ f_{k-2}' \circ \cdots \circ f_1'\right) \notag\\
&\quad + \cdots + \rho\left(f_k \circ f_{k-1} \circ \cdots \circ f_2 \circ f_1', f_k \circ f_{k-1} \circ \cdots \circ f_1\right) \notag\\
&\le \rho\left(f_k' \circ f_{k-1}' \circ \cdots \circ f_1', f_k \circ f_{k-1}' \circ \cdots \circ f_1'\right) \tag{5.125}\\
&\quad + \kappa_k \cdot \rho(f_{k-1}' \circ \cdots \circ f_1', f_{k-1} \circ f_{k-2}' \circ \cdots \circ f_1') \tag{5.126}\\
&\quad + \cdots + \left(\prod_{j=2}^{k} \kappa_j\right) \rho(f_1', f_1) \notag\\
&\le \sum_{i=1}^{k} \left(\epsilon_i \prod_{j=i+1}^{k} \kappa_j\right). \tag{5.127}
\end{align}$$

Note that the first inequality follows by the triangle inequality, the second by the κ_i-Lipschitz continuity of f_i, and the third by applying the definition of each of our ϵ_i-covers. □

Proof of Theorem 5.20. We now apply Lemma 5.23 to dense neural networks. Dense neural networks consist of a composition of layers, where each layer is a linear model composed with a 1-Lipschitz activation. Using Theorem 5.18 along with the property that 1-Lipschitz functions will only contribute a factor of at most 1 (Lemma 4.29), the covering number of each layer can be bounded by:

$$g\left(\epsilon_i, c_{i-1}\right) = \tilde{O}\left(\frac{c_{i-1}^2 b_i^2}{\epsilon_i^2}\right), \tag{5.128}$$

where c_{i-1}^2 is the norm of the inputs, b_i^2 is $\|W_i^\top\|_{2,1}$, and ϵ_i^2 is the radius. From Lemma 5.23, we know that

$$\log N(\epsilon, \mathcal{F}, \rho) = \tilde{O}\left(\sum_{i=1}^{r} \frac{c_{i-1}^2 b_i^2}{\epsilon_i^2}\right) \tag{5.129}$$

for

$$\epsilon = \sum_{i=1}^{r} \left(\epsilon_i \prod_{j=i+1}^{r} \kappa_j \right) \tag{5.130}$$

We now have a bound on $N(\epsilon, \mathcal{F}, \rho)$ that relies on ϵ_i's, but $N(\epsilon, \mathcal{F}, \rho)$ should only be a function of ϵ. Since we already know that $\epsilon = \sum_{i=1}^{r} \left(\epsilon_i \prod_{j=i+1}^{r} \kappa_j \right)$, we keep ϵ constant and optimize the upper bound of $N(\epsilon, \mathcal{F}, \rho)$ over different choices of ϵ_i. To find the optimal ϵ_i, we will first find a lower bound on $N(\epsilon, \mathcal{F}, \rho)$. We then choose ϵ_i so that this lower bound is achieved. Ultimately, our optimized ϵ_i yields a bound on the covering number of the following form: $\log\left(N\left(\epsilon, \mathcal{F}, \rho\right)\right) \leq \frac{R}{\epsilon^2}$, where R is some constant independent of ϵ.

We derive this lower bound using Holder's inequality, which states that

$$\langle a, b \rangle \leq \|a\|_p \|b\|_q \tag{5.131}$$

when $\frac{1}{p} + \frac{1}{q} = 1$. Writing out the vectors a, b, we get that

$$\sum_i a_i b_i \leq \left(\sum a_i^p \right)^{\frac{1}{p}} \left(\sum b_i^q \right)^{\frac{1}{q}} \tag{5.132}$$

Let $\alpha_i^2 = c_{i-1}^2 b_i^2, \beta_i = \prod_{j=i+1}^{r} \kappa_j$. By Holder's inequality, using $p = 3, q = \frac{3}{2}$, we get

$$\left(\sum_{i=1}^{r} \frac{\alpha_i^2}{\epsilon_i^2} \right) \left(\sum_{i=1}^{r} \beta_i \epsilon_i \right)^2 \geq \left(\sum_{i=1}^{r} (\alpha_i \beta_i)^{\frac{2}{3}} \right)^{\frac{3}{2}} \tag{5.133}$$

$$\sum_{i=1}^{r} \frac{\alpha_i^2}{\epsilon_i^2} \geq \frac{R}{\epsilon^2}, \tag{5.134}$$

where $R = \left(\left(\sum_{i=1}^{r} \left(c_{i-1} b_i \prod_{j=i+1}^{r} \kappa_j \right)^{\frac{2}{3}} \right) \right)^{\frac{3}{2}}$. We note that equality holds when

$$\epsilon_i = \left(\frac{c_{i-1}^2 b_i^2}{\prod_{j=i+1}^{r} \kappa_j} \right)^{\frac{1}{3}} \cdot \underbrace{\frac{\epsilon}{\left(\sum_{i=1}^{r} \frac{b_i^{\frac{2}{3}}}{\kappa_i^{\frac{2}{3}}} \right) \prod_{i=1}^{r} \kappa_i^{\frac{2}{3}}}}_{\epsilon'} \tag{5.135}$$

Using this choice of ϵ_i and letting ϵ' equal the second factor in (5.135) for notational convenience, we know that the log covering number is (up to a constant factor):

$$\sum_{i=1}^{r} \frac{c_{i-1}^2 b_i^2}{\epsilon_i^2} = \sum_{i=1}^{r} \frac{c_{i-1}^2 b_i^2 (\kappa_{i+1} \cdots \kappa_r)^{\frac{2}{3}}}{c_{i-1}^{\frac{4}{3}} b_i^{\frac{4}{3}} (\epsilon')^2} \tag{5.136}$$

$$= \sum_{i=1}^{r} (c_{i-1} b_i \kappa_{i+1} \cdots \kappa_r)^{\frac{2}{3}} \frac{1}{(\epsilon')^2} \tag{5.137}$$

$$= c^{\frac{2}{3}} \sum_{i=1}^{r} \left(\frac{b_i}{\kappa_i} \right)^{\frac{2}{3}} \prod_{i=1}^{r} \kappa_i^{\frac{2}{3}} \frac{\left(c^{\frac{2}{3}} \left(\sum_{i=1}^{r} (\frac{b_i}{\kappa_i})^{\frac{2}{3}} \prod_{i=1}^{r} \kappa_i^{\frac{2}{3}} \right) \right)^2}{\epsilon^2} \tag{5.138}$$

$$= \left(c^{\frac{2}{3}} \sum_{i=1}^{r} \left(\frac{b_i}{\kappa_i} \right)^{\frac{2}{3}} \prod_{i=1}^{r} \kappa_i^{\frac{2}{3}} \right)^3 \frac{1}{\epsilon^2} \tag{5.139}$$

$$= c^2 \prod_{i=1}^{r} \kappa_i^2 \left(\sum_{i=1}^{r} \left(\frac{b_i}{\kappa_i} \right)^{\frac{2}{3}} \right)^3 \frac{1}{\epsilon^2}. \tag{5.140}$$

Since this log covering number is of the form R/ϵ^2, we can apply (4.152) and conclude that

$$\mathcal{R}_S(\mathcal{F}) \lesssim \sqrt{\frac{R}{n}} \tag{5.141}$$

Last, plugging in

$$R = c^2 \prod_{i=1}^{r} \kappa_i^2 \left(\sum_{i=1}^{r} \left(\frac{b_i}{\kappa_i} \right)^{\frac{2}{3}} \right)^3 \tag{5.142}$$

we obtain the desired result

$$\mathcal{R}_S(\mathcal{F}) \lesssim \frac{c}{\sqrt{n}} \prod_{i=1}^{r} \kappa_i \left(\sum_{i=1}^{r} \left(\frac{b_i}{\kappa_i} \right)^{\frac{2}{3}} \right)^{\frac{3}{2}} . \tag{5.143}$$

□

Chapter 6

Data-dependent Generalization Bounds for Deep Nets

In Theorem 5.20, we proved the following bound on the Rademacher complexity of deep neural networks:

$$R_S(\mathcal{F}) \leq \prod_{i=1}^{r} \|W_i\|_{\text{op}} \cdot \mathsf{poly}(\|W_1\|, \ldots, \|W_r\|). \tag{6.1}$$

This bound, however, suffers from multiple deficiencies. In particular, it grows exponentially in the depth, r, of the network and $\|W_i\|_{\text{op}}$ measures the worst-case Lipschitz-ness of the network layers over the input space.

In this section, we obtain a tighter generalization bound that depends upon the realized Lipschitz-ness of the model on the training data. To further motivate this approach, we also note that stochastic gradient descent, i.e. the typical optimization method typically used to fit deep neural networks, prefers models that are more Lipschitz (see Chapter (TBD) for further discussion) . This preference must be realized by the model *on empirical data*, however, as no learning algorithm has access to the model's properties over the entire data space.

Ultimately, we aim to prove a tighter bound on the population loss that grows polynomially in the Lipschitz-ness of f on the empirical data. Namely, given that f is parameterized by some θ, we hope to derive a bound on the population loss at θ that is a *polynomial* function of the Lipschitz-ness of f on $x^{(1)}, \ldots, x^{(n)}$ as well as the norm of θ.

Uniform convergence with a data-dependent hypothesis class. So far in this course, given some complexity measure we denote as comp$(\cdot)$, our uniform convergence results always appear in one of the two following forms (which are essentially equivalent). Namely, with high probability,

$$\forall f \in \mathcal{F}, \quad L(f) \leq \frac{\text{comp}(\mathcal{F})}{\sqrt{n}} \qquad \text{(I)} \tag{6.2}$$

$$\forall f, \quad L(f) \leq \frac{\text{comp}(f)}{\sqrt{n}} \qquad \text{(II)} \tag{6.3}$$

Remark 6.1. Most of the results we have obtained so far are of type I, e.g. with $\text{comp}(\mathcal{F})/\sqrt{n} = R_n(\mathcal{F})$. We obtain results of type II by considering a restricted set of functions $\mathcal{F}_C = \{f : \text{comp}(f) \leq C\}$. We then apply a type I bound to $\mathcal{F}_C$ and take a union bound over all C. Therefore, these two type of bounds are essentially equivalent (up to a small additive factor difference due to the additional union bound over the choices of C.)

Note, however, that neither of these approaches produce bounds that depend upon the data. By contrast, in the sequel, we will derive a new *data-dependent* generalization bound. These bounds state that with high probability over the choice of the empirical data and, for all functions $f \in \mathcal{F}$,

$$L(f) \leq \text{comp}\left(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}\right) \tag{6.4}$$

Even though the complexity measure depends on the training data, and is thus a random variable by itself, it can be used as a regularizer which can be added to the original training loss.

Remark 6.2. Although there is no universal consensus on the type of generalization bound we should derive, we can argue that there is no way to leverage more information in a generalization bound beyond the empirical data. For example, one might try to use the input distribution P to define the complexity measure, but if we allowed ourselves access to P, we could just define $\text{comp}(f, P) = \mathbb{E}_P[f(X)]$. In some sense, defining a generalization bound using the true distribution amounts to cheating, and the dependence on the empirical data seems to be proper because the bound can still be used as a regularizer.

In this new paradigm, we can no longer take the previous approach of obtaining type I bounds and then derive a type II bound via a reduction. To see why, suppose that we have the hypothesis class

$$\mathcal{F}_C = \{f : \text{comp}(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}) \leq C)\} \tag{6.5}$$

If our complexity measure depends on the empirical data, then so does our hypothesis class $\mathcal{F}_C$, which makes $\mathcal{F}_C$ itself a random variable. However, our theorems regarding Rademacher complexity require that the hypothesis class be fixed before we ever see the empirical data.

We may hope to get around this by changing the way we think about uniform convergence. Consider the simplified case where our new complexity measure is separable, i.e.

$$\text{comp}(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}) = \sum_{i=1}^{n} h(f, x^{(i)}), \tag{6.6}$$

for some function g. Then we can consider an *augmented loss*:

$$\tilde{\ell}(f) = \ell(f)\mathbf{1}[h(f, x^{(i)} \leq C)] \tag{6.7}$$

Suppose we have a region of low complexity in our existing loss function as depicted in Figure 6.1. Because this region is random, so we cannot selectively apply uniform convergence. However, we can use our new surrogate loss function $\tilde{\ell}$ in that region. By modifying the loss function in this way, we can still fix the hypothesis class ahead of time, allowing us to apply existing tools to $\tilde{\ell}(f)$. The surrogate loss was used in [Wei and Ma, 2019a] to obtain a data-dependent generalization bound, though there are possibly various other ways to define surrogate losses and apply existing uniform convergence guarantees. In the sequel, we introduce a particular surrogate "margin" that allows us to cleanly apply our previous results to a (implicitly) data-dependent hypothesis class [Wei and Ma, 2019a].

6.1 All-layer margin

We next introduce a new surrogate loss called the *all-layer margin* that can also be thought of as a surrogate margin. This loss will essentially zero out high-complexity regions so that we may focus on low-complexity regions for which we can expect small generalization gap. Note that the all-layer margin we analyze will not explicitly zero-out high-complexity regions using an indicator function, but instead implicitly takes into account some data-dependent characteristics of the model. Once we adopt this new loss function, we will be able to apply some of our earlier methods.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a classification model. Recall that the standard margin is defined as $yf(x)$, with y in $\{-1, 1\}$. We will say that $g_f(x, y)$ is a *generalized margin* if it satisfies

$$g_f(x, y) = \begin{cases} 0, & \text{if } f(x)y \leq 0 \text{ (an incorrect classification)} \\ > 0, & \text{if } f(x)y > 0 \text{ (a correct classification)} \end{cases}. \tag{6.8}$$

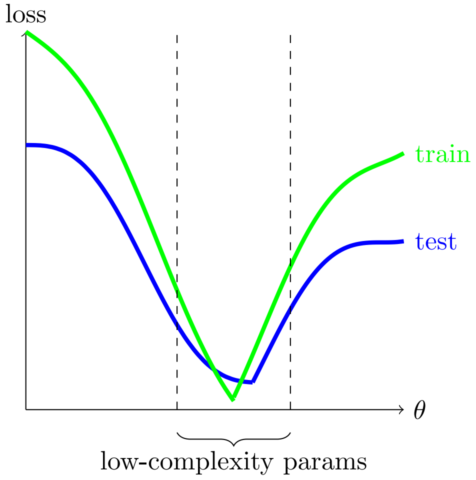


Figure 6.1: These curves depict a "low-complexity" region in parameter space. The blue curve is the unobserved test loss we aim to bound, while the green curve denotes the empirical training loss we observe. Observe that in the region of θ that we identify as being "low-complexity," the gap between the train and test losses is smaller than in the high-complexity regions.

To simplify the exposition of the machinery below, we also introduce the ∞-*covering number* $N_\infty(\epsilon, \mathcal{F})$ as the minimum cover size with respect to the metric ρ defined as the infinity-norm distance on an input domain $\mathcal{X}$:

$$\rho(f, f) \triangleq \sup_{x \in \mathcal{X}} |f(x) - f'(x)| \triangleq \|f - f'\|_\infty.^1 \tag{6.9}$$

Remark 6.3. Notice that $N_\infty(\epsilon, \mathcal{F}) \geq N(\epsilon, \mathcal{F}, L_2(P_n))$. This is because the $\rho = L_\infty(\mathcal{X})$ is a more demanding measure of error: f and f' must be close on *every* input, not just the empirical data. That is,

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - f'(x_i))^2} \leq \sup_{x \in \mathcal{X}} |f(x) - f'(x)|. \tag{6.10}$$

Lemma 6.4. *Suppose g_f is a generalized margin. Let $\mathcal{G} = \{g_f : f \in \mathcal{F}\}$. Suppose that for some R, $\log N_\infty(\epsilon, \mathcal{G}) \leq \lfloor \frac{R^2}{\epsilon^2} \rfloor$ for all $\epsilon > 0$.[2] Then, with high probability over the randomness in the training data, for every f in $\mathcal{F}$ that correctly predicts all the training examples,*

$$L_{01} \leq \widetilde{O}\left(\frac{1}{\sqrt{n}} \cdot \frac{R}{\min_{i \in [n]} g_f(x^{(i)}, y^{(i)})}\right) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{6.11}$$

Proof. The high-level idea of our proof is to replace $\mathcal{F}$ with $\mathcal{G}$ before repeating the standard margin theory argument from Section 5.1.2.

Let ℓ_γ be the ramp loss given in (5.1), which is 1 for negative values, 0 for values greater than γ, and a linear interpolation between 1 and 0 for values between 0 and γ. We define the surrogate loss as $\hat{L}_\gamma(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_\gamma(g_{f_\theta}(x^{(i)}, y^{(i)}))$, and the surrogate population loss as $L_\gamma(\theta) = \mathbb{E}[\ell_\gamma(g_{f_\theta}(x, y))]$. Applying Corollary 4.19, where we used the Rademacher complexity to control the generalization error, we conclude that

$$L_\gamma(\theta) - \hat{L}_\gamma(\theta) \leq R_S(\ell_\gamma \circ \mathcal{G}) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{6.12}$$

[1] If f maps $\mathcal{X}$ to multi-dimensional outputs, we will define $\rho(f, f) \triangleq \sup_{x \in \mathcal{X}} \|f(x) - f'(x)\| \triangleq \|f - f'\|_\infty$ where the norm in $\|f(x) - f'(x)\|$ is a norm in the output space of f (which will be the Euclidean norm in this rest of this section).

[2] Recall that this is the worst dependency on ϵ that we can tolerate when converting covering number bounds to Rademacher complexity.

Next we observe that

$$\begin{aligned}
\log N(\epsilon, \ell_\gamma \circ \mathcal{G}, L_2(P_n)) &\leq \log N(\epsilon\gamma, \mathcal{G}, L_2(P_n)) && \text{(Lemma 4.29)} && (6.13)\\
&\leq \log N_\infty(\epsilon\gamma, \mathcal{G}) && (6.10) && (6.14)\\
&\leq \left\lfloor \frac{R^2}{\epsilon^2\gamma^2} \right\rfloor && \text{(by assumption).} && (6.15)
\end{aligned}$$

Then, using our results relating the log of the covering number to a bound on the Rademacher complexity (recall (4.152) and Theorem 4.28), we conclude that $R_S(\ell_\gamma \circ \mathcal{G}) \leq \widetilde{O}\left(\frac{R}{\gamma\sqrt{n}}\right)$. Take $\gamma = \gamma_{\min} = \min_i g_\gamma(x^{(i)}, y^{(i)})$.[3] Using Corollary 4.19, we conclude that $\hat{L}_{\gamma_{\min}}(\theta) \leq 0 + \widetilde{O}\left(\frac{R}{\sqrt{n}\cdot\gamma_{\min}}\right) + \widetilde{O}\left(\frac{1}{\sqrt{n}}\right)$, as desired. □

For which g_f can we bound the covering number? If we take $g_f(x, y) = yf(x)$, then the covering number depends on the product $\prod_i \|W_i\|_{\text{op}}$, but we originally set out to do better than this. If we have a linear model $w^\top x$, the normalized margin, $\frac{y\cdot w^\top x}{\|w\|}$, governs the generalization performance. But how do we normalize for more general models?

For a deep neural net, a potential normalizer is the product of the Lipschitz constants of the layers. However, we do not want to normalize by a constant that depends only on the function class, so we take a different approach. We interpret the normalized margin as the solution to the following optimization problem:

$$\begin{aligned}
\min_\delta \quad & \|\delta\|_2 \\
\text{s.t.} \quad & w^\top(x+\delta)y \leq 0
\end{aligned} \tag{6.16}$$

In plain English, this problem searches for the minimum perturbation that gets our data point across the boundary.

This perturbation view of the standard margin can be extended naturally to multiple layers. For the math to work, it turns out that we need to perturb all the layers. We define the *all-layer margin* as below. We will consider perturbed models $\delta = (\delta_1, \ldots, \delta_r)$, where each δ_i is a perturbation *vector* associated with the i-th layer (and it has the same dimensionality as the i-th layer activation). We incorporate these perturbations into our model in the following way (so that we can handle the scaling in a clean way):

$$h_1(x, \delta) = W_1 x + \delta_1 \cdot \|x\|_2 \tag{6.17}$$

$$h_2(x, \delta) = \sigma(W_2 h_1(x, \delta)) + \delta_2 \cdot \|h_1(x, \delta)\|_2 \tag{6.18}$$

$$\vdots$$

$$f(x, \delta) = h_r(x, \delta) = \sigma(W_r h_{r-1}(x, \delta)) + \delta_r \cdot \|h_{r-1}(x, \delta)\|_2. \tag{6.19}$$

We can then ask: what was the smallest perturbation that changed our decision? That is, let

$$m_f(x, y) \triangleq \min_\delta \sqrt{\sum_{i=1}^r ||\delta_i||_2^2} \quad \text{s.t.} \quad f(x, \delta)y \leq 0, \tag{6.20}$$

i.e. the smallest perturbation that yields incorrect predictions.

Informally, $m_f(x, y)$ is a measure of how hard it is to perturb the model f. f can be hard to perturb for two reasons: f is Lipschitz (in its intermediate layers) and/or $yf(x)$ is large. In other words, the all-layer margin is a normalized version of the standard margin, normalized by the Lipschitzness of the model at the particular data point (x, y).

We now introduce our main result regarding the all-layer margin.

[3] A caveat: because γ is a random variable, proving this result rigorously requires taking a union bound over a discretized γ. We sketched out this argument more thoroughly in Remark 5.4.

Theorem 6.5. *With high probability, for all f with training error* 0*,*

$$L_{01}(f) \le \widetilde{O}\left(\frac{1}{\sqrt{n}} \cdot \frac{\sum_{i=1}^{r} \|W_i\|_{1,1}}{\min_{i \in [n]} m_f(x^{(i)}, y^{(i)})}\right) + \widetilde{O}\left(\frac{r}{\sqrt{n}}\right), \tag{6.21}$$

where $\|W\|_{1,1}$ is the sum of the absolute values of the entries of W.

In summary, robustness to perturbations in intermediate layers implies good generalization. We will interpret the bound, compare the bounds with previous works, and discuss further extensions in the remarks following the proofs of the theorem. (E.g, in Remark 6.8, we will argue that this bound is strictly better than the one we obtained in Theorem 5.20; in the worst case, we still have that $\frac{1}{m_f(x,y)} \le \frac{\prod \|W_i\|_{\text{op}}}{f(x)}$.)

To prove this theorem, it suffices to bound $N_\infty(\epsilon, \mathcal{G})$ by $O(\frac{\sum \|W_i\|_{1,1}}{\epsilon^2})$ and apply Lemma 6.4. Towards this goal, let $\mathcal{F}_i = \{z \mapsto \sigma(W_i z) : \|W_i\|_{1,1} \le \beta_i\}$. Then, $\mathcal{F} = \mathcal{F}_r \circ \mathcal{F}_{r-1} \circ \cdots \circ \mathcal{F}_1$.

Lemma 6.6 (Decomposition Lemma)**.** *Let $m \circ \mathcal{F}$ denote $\{m_f : f \in \mathcal{F}\}$. Then,*

$$\log N_\infty\left(\sqrt{\sum_{i=1}^{r} \epsilon_i^2}, m \circ \mathcal{F}\right) \le \sum_{i=1}^{r} \log N_\infty(\epsilon_i, \mathcal{F}_i), \tag{6.22}$$

where $N_\infty(\epsilon_i, \mathcal{F}_i)$ is defined with respect to the input domain $\mathcal{X} = \{x : \|x\|_2 \le 1\}$.

That is, we only have to find the covering number for each layer, and then we have the covering number for the (all-layer margin of the) composed function class. Notice that we bounded the covering number of $m \circ \mathcal{F}$ in the above lemma, not $\mathcal{F}$.

Then, the desired result follows directly from the preceding decomposition lemma.

Corollary 6.7. Assume that $\log N_\infty(\epsilon_i, \mathcal{F}_i) \le \left\lfloor \frac{c_i^2}{\epsilon_i^2} \right\rfloor$ for every $\mathcal{F}_i$, i.e. the function class corresponding to the i-th layer of f in Theorem 6.5. Then, by taking $\epsilon_i = \epsilon \cdot \frac{c_i}{\sqrt{\sum_i c_i^2}}$, we have that

$$\log N_\infty(\epsilon, m \circ \mathcal{F}) \le \frac{\sum_i c_i^2}{\epsilon^2}. \tag{6.23}$$

This result gives the complexity of the composed model in terms of the complexity of the layers, with each c_i given by $\|W_i\|_{1,1}$. For linear models, we can show $N_\infty(\epsilon_i, \mathcal{F}_i) \le \widetilde{O}\left(\frac{\beta_i^2}{\epsilon^2}\right)$ (where β_i is a bound on $\|W_i\|_{1,1}$), and this implies Theorem 6.5[4] Finally, we are only left with the proof of Lemma 6.6.

Proof of Lemma 6.6. Now we will prove a limited form of the decomposition lemma for affine models: $\mathcal{F}_i = \{z \mapsto \sigma(W_i z) : \|W_i\|_{1,1} \le \beta_i\}$. There are two crucial steps to this problem. First, we will prove that $m_f(x,y)$ is 1-Lipschitz in f. That is, for all $\mathcal{F} = \mathcal{F}_r \circ \mathcal{F}_{r-1} \circ \cdots \circ \mathcal{F}_1$ and $\mathcal{F}' = \mathcal{F}'_r \circ \mathcal{F}'_{r-1} \circ \cdots \circ \mathcal{F}'_1$,

$$|m_f(x,y) - m_{f'}(x,y)| \le \sqrt{\sum_{i=1}^{r} \max_{\|x\|_2 \le 1} \|f_i(x) - f'_i(x)\|_2^2}. \tag{6.24}$$

Notice that now we are working with a clean sum of differences, with no multipliers!

Second, we construct a cover: Let $U_1, \ldots, U_r$ be $\epsilon_1, \ldots, \epsilon_r$-covers of $\mathcal{F}_1, \ldots, \mathcal{F}_r$, respectively, such that $|U_i| = N_\infty(\epsilon_i, \mathcal{F}_i)$. By definition, for all f_i in $\mathcal{F}_i$, there exists a $u_i \in U_i$ such that $\max_{\|x\| \le 1} \|f_i(x) - u_i(x)\|_2 \le \epsilon_i$. Take $U = U_r \circ U_{r-1} \circ \cdots \circ U_1 = \{u_r \circ u_{r-1} \circ \cdots \circ u_1\}$ as the cover for $m \circ \mathcal{F}$. Suppose we were given

[4] Technically, we also need to union bound over the choices of β_i, which can also be achieved following Remark 5.4.

$f = f_r \circ \cdots \circ f_1 \in \mathcal{F}$. Let $u_r, \ldots, u_1$ be the nearest neighbors of $f_r, \ldots, f_1$. Then

$$|m_f(x,y) - m_u(x,y)| \leq \sqrt{\sum_{i=1}^{r} \max_{||x|| \leq 1} ||f_i(x) - u_i(x)||_2^2} \tag{6.25}$$

$$\leq \sqrt{\sum_{i=1}^{r} \epsilon_i^2} \qquad \text{(by construction).} \tag{6.26}$$

Having established the validity of our cover, we now return to our claim of 1-Lipschitz-ness stated in (6.24). By symmetry, it is sufficient to prove an upper bound for $m_{f'}(x,y) - m_f(x,y)$.

Let $\delta_1^*, \ldots, \delta_r^*$ be the optimal choices of δ in defining $m_f(x,y)$. Our goal is to turn these into a feasible solution of $m_{f'}(x,y)$, which we denote by $\hat{\delta}_1, \ldots, \hat{\delta}_r$. If this solution is feasible, we obtain the bound $m_{f'}(x,y) \leq \sqrt{\sum \|\hat{\delta}_i\|_2^2}$.

Intuitively, we want to define a perturbation for f' that does the same thing as $\delta_1^*, \ldots, \delta_r^*$ for f. In plain English, $(f', \hat{\delta}_1, \ldots, \hat{\delta}_r)$ should do the same thing as $(f_1, \delta_1^*, \ldots, \delta_r^*)$. Recall that f has parameters $W_1, \ldots, W_r$ and f' has parameters $W_1', \ldots, W_r'$. Then, under the optimal perturbation,

$$h_1 = W_1 x + \delta_1^* \|x\|_2 \tag{6.27}$$

$$h_2 = \sigma(W_2 h_1) + \delta_2^* \|h_1\|_2 \tag{6.28}$$

$$\vdots$$

$$h_r = \sigma(W_r h_{r-1}) + \delta_r^* \|h_{r-1}\|_2 \tag{6.29}$$

We want to imitate this by perturbing f' in some way. In particular, let

$$h_1 = W_1' x + \underbrace{\delta_1^* \|x\|_2 + (W_1 - W_1')x}_{\triangleq\, \hat{\delta}_1 \|x\|_2}, \tag{6.30}$$

where the last term serves to compensate for the difference between W_1 and W_1'. Thus, $\hat{\delta}_1 \triangleq \delta_1^* + \frac{(W_1 - W_1')x}{\|x\|_2}$. We repeat this argument for every layer. Using the second layer as an example,

$$h_2 = \sigma(W_2' h_1) + \underbrace{\delta_2^* \|h_1\| + \sigma(W_2 h_1) - \sigma(W_2' h_1)}_{\triangleq\, \hat{\delta}_2 \|h\|_2}. \tag{6.31}$$

So, $\hat{\delta}_2 = \delta_2^* + \frac{\sigma(W_2 h_1) - \sigma(W_2' h_1)}{\|h_1\|_2}$. In general,

$$\hat{\delta}_i \triangleq \delta_i^* + \frac{\sigma(W_i h_{i-1}) - \sigma(W_i' h_{i-1})}{\|h_{i-1}\|_2} \tag{6.32}$$

Then $\hat{\delta}_1, \ldots, \hat{\delta}_r$ on f' are making the same predictions as $\delta_1, \ldots, \delta_r$ on f'. Last, observe that

$$m_{f'}(x,y) \leq \sqrt{\sum ||\hat{\delta}_i||_2^2} \tag{6.33}$$

$$\leq \sqrt{\sum \|\delta_i^*\|_2^2} + \sqrt{\sum_{i=1}^{r} \left(\frac{\sigma(W_i h_{i-1}) - \sigma(W_i' h_{i-1})}{\|h_{i-1}\|_2} \right)^2} \qquad \text{(Minkowski's Ineq.)}^5 \tag{6.34}$$

$$\leq m_f(x,y) + \sqrt{\sum_{i=1}^{r} \max_{\|x\|_2 \leq 1} (\sigma(W_i x) - \sigma(W_i' x))^2} \tag{6.35}$$

$$= m_f(x,y) + \sqrt{\sum_{i=1}^{r} \max_{\|x\|_2 \leq 1} (f_i(x) - f_i'(x))^2} \tag{6.36}$$

Note that in (6.35), constraining $\|x\|_2 \le 1$ is equivalent to dividing by the ℓ_2-norm of x. □

Remark 6.8. We can compare the above with Theorem 5.20 proven in [Bartlett et al., 2017].

$$\begin{aligned} f(x,\delta) - f(x) \le\ & \|\delta_r\|_2 \cdot \|W_{r-1}\|_{\text{op}} \cdots \|W_1\|_{\text{op}} \\ & + \|W_r\|_{\text{op}} \cdot \|\delta_{r-1}\|_2 \cdot \|W_{r-2}\|_{\text{op}} \cdots \|W_1\|_{\text{op}} \\ & + \cdots \\ & + \|W_r\|_{\text{op}} \cdots \|W_2\|_{\text{op}} \cdot \|\delta_1\|_2. \end{aligned} \tag{6.37}$$

Ignoring minor details (e.g. dependency on r), we suppose that $y = 1$. Then, if $f(x) > 0$ and $f(x+\delta) \le 0$, it must be the case that $\|\delta\|_2 \lesssim \frac{|f(x)|}{\prod_{i=1}^r \|W_i\|_{\text{op}}}$. This further implies that

$$\frac{m_f(x,y)}{yf(x)} \gtrsim \frac{1}{\prod_{i=1}^r \|W_i\|_{\text{op}}}. \tag{6.38}$$

Rearranging, we conclude that we have obtained a tighter bound since the inverse margin $\frac{1}{m_f(x,y)} \lesssim \frac{1}{yf(x)} \cdot \prod_{i=1}^r \|W_i\|_{\text{op}}$.

Remark 6.9. Later, we will show that SGD prefers Lipschitz solutions and Lipschitzness on data points. Implicitly, SGD seems to be maximizing the all-layer margin. Since the algorithm is (in a sense) minimizing Lipschitzness on a data point, this likely accounts for the empirically observed gap between the two bounds.

Remark 6.10. The approach we have described here is also similar to other methods in the deep learning literature. Other authors have introduced a method known as SAM (a form of sharpness-aware regularization); this method applies a perturbation to the parameter θ itself rather than on the intermediate hidden parameters h_i. However, these two methods are related! If we consider the (single-example) loss $\frac{\partial \ell}{\partial W_i}$, it equals $\frac{\partial \ell}{\partial h_{i+1}} \cdot h_i^\top$. Note that the norm of the term on the left is bounded by the product of the norms of the two terms of the right; this observation relates the model's Lipschitzness with respect to the parameters to its Lipschitzness with respect to the hidden layer outputs.

Remark 6.11. Finally, we can prove a more general version of this result in which we do not need to study the minimum margin of the entire dataset, and instead consider the average margin. Using this approach, we can show that the test error is bounded above by $\frac{1}{n}\sqrt{\frac{1}{n}\sum_{i=1}^n \frac{1}{m_f(x^{(i)},y^{(i)})^2}}$ times the sum of complexities of each layer, plus a low-order term.

[5] Minkowski's inequality, which states that $\sqrt{\sum \|a_i + b_i\|_2^2} \le \sqrt{\sum \|a_i\|_2^2} + \sqrt{\sum \|b_i\|_2^2}$. In this setting, this inequality can also be proved using Cauchy-Schwarz.

Chapter 7

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.

7.1 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:

$$L(\hat{\theta}) - L(\theta^*) \leq \underbrace{L(\hat{\theta}) - \hat{L}(\hat{\theta})}_{\text{generalization error}} + |L(\theta^*) - \hat{L}(\theta^*)|. \tag{7.1}$$

   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 (parsimonious, or low-complexity) explanations tend to 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

$$\text{Any global minimizer } \hat{\theta}_\lambda \text{ of } \hat{L}_{\text{reg}} \text{ has small excess risk } L(\hat{\theta}_\lambda) - L(\theta^*)\,. \tag{7.2}$$

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), they are conceptually independent, and it is typically possible to study each area individually, then combine all three to get a result.

7.2 Deep learning theory and its differences

The situation is more complex for deep learning theory. Two prominent differences are (a) the models are non-linear and the objective functions are non-convex, and (b) in deep learning, researchers have observed in many cases that more parameters typically help improve the performance, and many state-of-the-art models have much more parameters than the number of training data. (b) is often referred as to "over-parameterization".

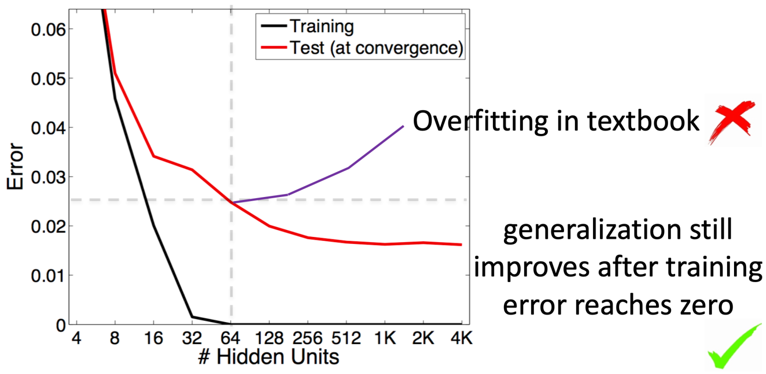


Figure 7.1: The black and red lines denote the training and test error, respectively, of a three layer neural network fit to and evaluated on MNIST [Neyshabur et al., 2015]. While classical generalization theory predicts that beyond some threshold, the test error will increase with complexity (shown by the purple line), the true test error continues to decline with overparameterization. Though not depicted here, Neyshabur et al. observe similar test set error curves for a neural network fit to CIFAR-10.

Let us consider the difference in each of the three components described for classical machine learning theory.

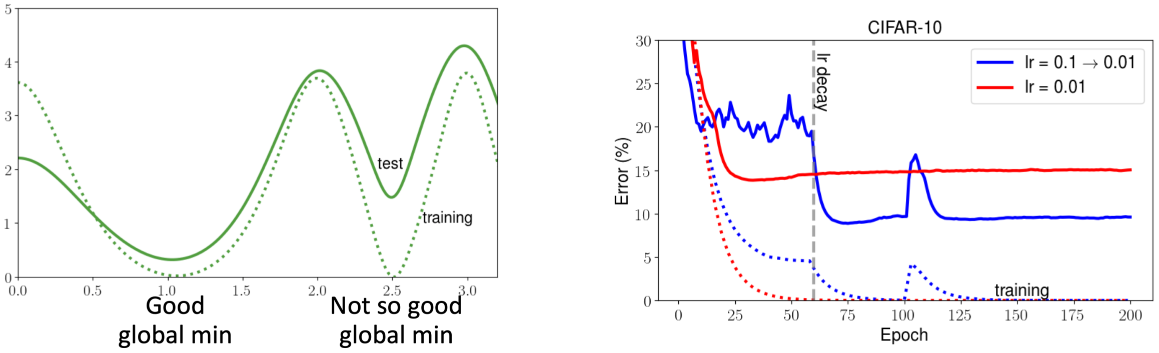
1. **Approximation theory:** Large neural net models are considered to be very expressive. That is, both the population loss $L(\theta)$ and the finite sample loss $\hat{L}(\theta)$ can be made small. In fact, neural networks are *universal approximators*; see for example [Hornik, 1991]. This can be a somewhat misleading statement as the definition of universal approximator allows for the size of the network to be impractically 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 [Arpit et al., 2017].

2. **Statistical generalization:** Relatively weak regularization is used in practice. In many cases only weak ℓ_2 regularization is used, i.e.

$$\widehat{L}_{\text{reg}}(\theta) = \hat{L}(\theta) + \lambda \|\theta\|_2^2. \tag{7.3}$$

   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.



(a) (b)

Figure 7.2: We use dotted and solid lines to depict training and test error, respectively. Figure 7.2a demonstrates how global minimizers for the training loss can have differing performance on test data. In Figure 7.2b, blue and red colors differentiate between the model fit with a decaying learning rate and a small constant learning rate. Though both neural networks shown in this plot achieve 0 training error, the global minimizer obtained by a more sophisticated learning rate schedule appears to generalize better to unseen data.

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 (Figure 7.2a). Take, for example, using stochastic gradient descent (SGD) to learn a model to classify the dataset CIFAR-10. In Figure 7.2b, we show 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 job of optimizers in deep learning is not just to find an arbitrary global minimum: we need to find the right global minimum. This contrasts sharply with (7.2) from the classical setting, where achieving a global minimum leads to good guarantees on generalization error. This means that (7.2) is simply not powerful enough to deal with deep learning, because it cannot distinguish between global minima 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/optimizers have different "implicit biases" or "implicit regularization effect", causing them to converge to different global minimizers. Understanding the implicit regularization effect of optimizers is thus a central goal of deep learning theory. The lack of understanding implicit regularization hinders the development of fast optimizers—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 or implicit regularization effect can be characterized in the form of showing the optimizers prefer $\hat{\theta}$ of certain low complexity among all the global minimizers. The deep learning analog of (7.2) often consists of two statements: (a) the optimizer implicitly prefers low complexity solution according to complexity measure $R(\cdot)$ by converging to a global minimizer $\hat{\theta}$ with low complexity $R(\hat{\theta})$, and (b) low complexity solutions generalize. This means that we end up doing more work on the optimization front—the optimizer needs to ensure both a small training loss and a low complexity solution. On the other hand, proving generalization bounds (statement (b)) works similarly to the classical setting once we understand how our optimizer finds a low-complexity solution.

We summarize some of the results that we will present in the future chapters.

1. **Optimization.** First, we will prove that under certain data distribution assumption, optimizers such as stochastic gradient decent can converge to an approximate global minimum, even though the objective function is non-convex. Results of this form can be shown on matrix factorization problems and linearized neural networks, even without over-parameterization, but so far are limited to these simple models. Second, we will discuss a recent approach, called neural tangent kernels (NTK), which proves that for almost any neural networks, with overparameterization, gradient descent can converge to a global minimum, *under specific hyperparameter settings* (e.g, specific learning rate and initialization). However, it turns out that these specific hyperparaemeter settings *does not* provide sufficient implicit regularization effect for the learned models to generalize. (In other words, the optimizer only returns a global minimizer, but not a global minimizer that generalizes well.)

2. **Implicit regularization effect.** This involves showing that the solution $\hat{\theta}$ obtained by a particular optimizer has low complexity $R(\hat{\theta}) \leq C$ according to some complexity measure $R(\cdot)$ (which depends on the choice of optimizers). It's believed and empirically observed that any changes or tricks in the optimizers (e.g., learning rate schedule, batch size, initialization, batchnorm) could introduce additional implicit regularization effects. We will only demonstrate these on some special cases of models (e.g. logistic regression, matrix factorization) and optimizers (e.g. gradient descent, label noise in SGD, dropout, learning rate). Recently, there are also more general results with label noise SGD [Blanc et al., 2019, Damian et al., 2021].

3. **Generalization bounds.** This part involves showing 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 (e.g., those complexity measures that are similar to the complexity measure in part 2 above that are preferred by the optimizer). Here, many tools we developed in classical machine learning can still apply.

Chapter 8

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 (Section 8.2), discuss how we can make the notion "all local minima are global minima" rigorous, and walk through two examples where this is the case (Section 8.3). Finally, we introduce the neural tangent kernel approach which allows us to characterize of the loss of general neural networks near a specific initialization (or under specific parameterization).

8.1 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 empirically 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 8.1 for an illustration.

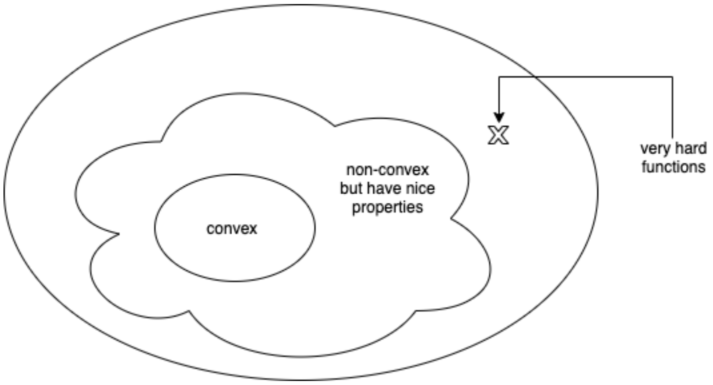


Figure 8.1: Classification of different functions for optimization. The functions we optimize in deep learning seem to fall mostly within the middle cloud.

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. $\theta_0 \triangleq$ initialization

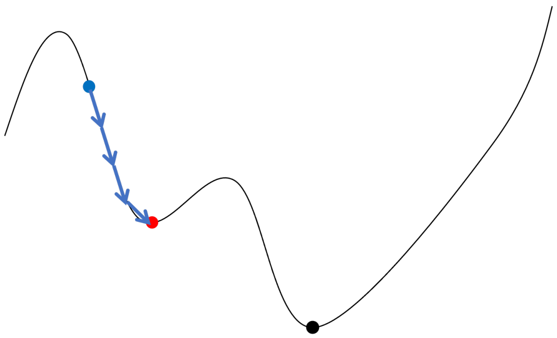


Figure 8.2: Illustration of how gradient descent does not always find the global minimum. In the picture, gradient descent initialized at the blue point only makes it to the local minimum at the red point: it does not find the global minimum at the black point.

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 can find a global minimum for convex functions[1] but cannot always find the global minimum for any general continuous functions (see Figure 8.2 for an illustration).
>
> *Observation 2*: Finding the global minimum of general non-convex functions is NP-hard.
>
> *Observation 3*: The objective function in deep learning is non-convex., but empirically 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.

8.2 Efficient convergence to (approximate) local minima

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

Definition 8.1 (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 8.2. 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.

[1] A more precise version of this claim is that gradient descent can find a point that has function value arbitrary close to the global minimal value.

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 8.3. *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.*

8.2.1 Strict-saddle condition

Theorem 8.3 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 8.4 (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 8.2.

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

Theorem 8.5 (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 8.3 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 8.6. *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 $\left(\frac{1}{\delta}, \frac{1}{\tau_0}, d\right)$.*

8.3 All local minima are global minima: two examples

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.

8.3.1 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:

$$\min_{x \in \mathbb{R}^d} g(x) \triangleq \frac{1}{2}\|M - xx^\top\|_F^2. \tag{8.1}$$

Theorem 8.7. *All local minima of g are global minima (even though g is non-convex).*

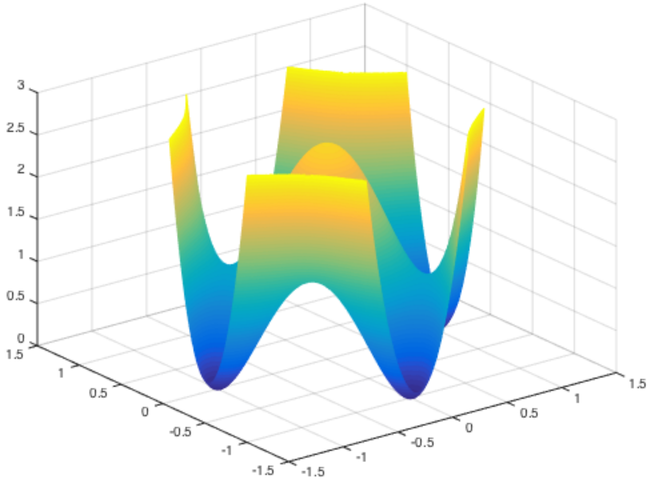


Figure 8.3: A two-dimensional function with the property that all local minima are global minima. It also satisfies the strict-saddle condition because all the saddle points have a strictly negative curvature in some direction.

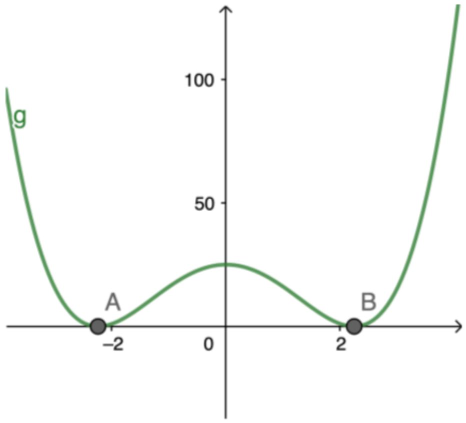


Figure 8.4: Objective function for principal components analysis (PCA) when $d = 1$.

Remark 8.8. For $d = 1$, $g(x) = \frac{1}{2}(m - x^2)^2$ for some constant m. Figure 8.4 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 $\nabla g(x) = -(M - xx^\top)x$, hence

$$\nabla g(x) = 0 \implies Mx = \|x\|_2^2 \cdot x, \tag{8.2}$$

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 $v \in \mathbb{R}^d$,

$$\langle v, \nabla^2 g(x) v\rangle \geq 0. \tag{8.3}$$

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

$$g(x+v) = \frac{1}{2}\|M - (x+v)(x+v)^\top\|_F^2 \tag{8.4}$$

$$\begin{aligned} &= \frac{1}{2}\|M - xx^\top\|_F^2 - \langle M - xx^\top, xv^\top + vx^\top\rangle + \frac{1}{2}\langle xv^\top + vx^\top, xv^\top + vx^\top\rangle \\ &\quad - \langle M - xx^\top, vv^\top\rangle + \text{higher order terms in } v. \end{aligned} \tag{8.5}$$

Hence, we have

$$\frac{1}{2}\langle v, \nabla^2 g(x) v\rangle = \frac{1}{2}\langle xv^\top + vx^\top, xv^\top + vx^\top\rangle - \langle M - xx^\top, vv^\top\rangle \tag{8.6}$$

$$= \langle x, v\rangle^2 + \|x\|_2^2\|v\|_2^2 - v^M v + \langle x, v\rangle^2 \tag{8.7}$$

$$= 2\langle x, v\rangle^2 + \|x\|_2^2\|v\|_2^2 - v^\top M v. \tag{8.8}$$

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

$$\langle v_1, \nabla^2 g(x) v_1\rangle = 2\langle x, v_1\rangle^2 - v_1^\top M v_1 + \|x\|_2^2 \geq 0. \tag{8.9}$$

Note that by (8.2), 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

  $$2\langle x, v_1\rangle^2 - v_1^\top M v_1 + \|x\|_2^2 = 0 - \lambda_1 + \lambda \geq 0, \tag{8.10}$$

  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. □

8.3.2 Matrix Completion [Ge et al., 2016]

We consider rank-1 matrix completion for simplicity. Let $M = zz^\top$ 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 8.9. 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 8.10 (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.

Remark 8.11. 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

$$\min_{x\in\mathbb{R}^d} f(x) \triangleq \frac{1}{2}\sum_{(i,j)\in\Omega}(M_{ij} - x_i x_j)^2 \tag{8.11}$$

$$= \frac{1}{2}\|P_\Omega(M - xx^\top)\|_F^2. \tag{8.12}$$

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

Definition 8.12 (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 $\|z\|_\infty \le \frac{\mu}{\sqrt{d}}$, where μ is considered to be a constant or log in dimension d.

Remark 8.13. A nice counterexample to think about why such condition is necessary is when $z = e_1$ and $M = e_1 e_1^\top$. 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 8.14. *Assume* $p = \dfrac{poly(\mu, \log d)}{d\epsilon^2}$ *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 8.15. $f(x)$ can be viewed as a sampled version of the PCA loss function $g(x) = \frac{1}{2}\|M - xx^\top\|_F^2 = \frac{1}{2}\sum_{(i,j)\in[d]\times[d]}(M_{ij} - x_i x_j)^2$, 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 8.5 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 8.16. Key idea: concentration for scalars is easy. We can approximate a sum of scalars via a sample:

$$\sum_{(i,j)\in\Omega} T_{ij} \approx p \sum_{(i,j)\in[d]\times[d]} T_{ij}, \tag{8.13}$$

where we use $\approx$ to mean that

$$\Big|\sum_{(i,j)\in\Omega} T_{ij} - p \sum_{(i,j)\in[d]\times[d]} T_{ij}\Big| < \epsilon \tag{8.14}$$

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,

$$\langle A, P_\Omega(B)\rangle = \sum_{(i,j)\in\Omega} A_{ij}B_{ij} \approx p\langle A, B\rangle. \tag{8.15}$$

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:

$$\langle v, \nabla f(x)\rangle = \langle v, P_\Omega(M - xx^\top)x\rangle = \langle vx^\top, P_\Omega(M - xx^\top)\rangle, \tag{8.16}$$

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

$$\langle v, \nabla^2 f(x)v\rangle = \|P_\Omega(vx^\top + xv^\top)\|_F^2 - 2\langle P_\Omega(M - xx^\top), vv^\top\rangle \tag{8.17}$$

$$= \langle P_\Omega(vx^\top + xv^\top), vx^\top + xv^\top\rangle - 2\langle P_\Omega(M - xx^\top), vv^\top\rangle, \tag{8.18}$$

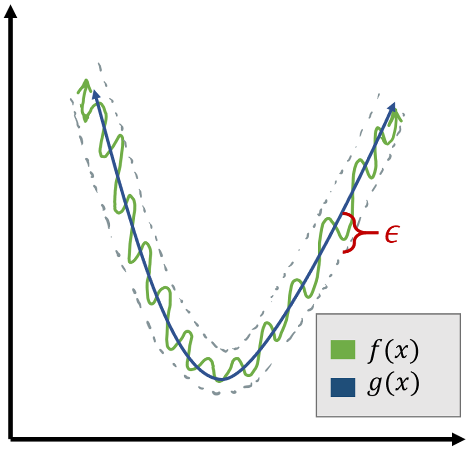


Figure 8.5: Even if $f(x)$ and $g(x)$ are no more than ϵ apart at any given x, without any additional knowledge, the local minima of f may possibly look dramatically different from the local minima of g. However, the proofs in this section show that the landscape of f (the matrix completion objective) and g (the PCA objective) are have similar properties by proving more advanced concentration inequalities.

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 8.17. *Let $\epsilon > 0$, $p = \frac{poly(\mu, \log d)}{d\epsilon^2}$. Given that $A = uu^\top, B = vv^\top$ for some u, v satisfying $\|u\|_2 \leq 1$, $\|v\|_2 \leq 1$, $\|u\|_\infty \leq \mu/\sqrt{d}$, $\|v\|_\infty \leq \mu/\sqrt{d}$, we have $|\langle P_\Omega(A), B\rangle/p - \langle A, B\rangle| \leq \epsilon$ 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 8.17 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 8.14.

Lemma 8.18 (Connecting inner product and norm for g)**.** *If x satisfies $\nabla g(x) = 0$, then $\langle x, z\rangle^2 = \|x\|_2^4$.*

Proof.

$$\begin{aligned}
\nabla g(x) = 0 &\implies \langle x, \nabla g(x)\rangle = 0 && (8.19)\\
&\implies \langle x, (zz^\top - xx^\top)x\rangle = 0 \qquad \text{(because } \nabla g(x) = (M - xx^\top)x\text{)} && (8.20)\\
&\implies \langle x, z\rangle^2 = \|x\|_2^4. && (8.21)
\end{aligned}$$

□

Lemma 8.19 (Connecting inner product and norm for f)**.** *Suppose $\|x\|_\infty \leq 2\mu/\sqrt{d}$. If x satisfies $\nabla f(x) = 0$, then $\langle x, z\rangle^2 \geq \|x\|_2^4 - \epsilon$ with high probability.*

Proof.

$$\begin{aligned}
\nabla f(x) = 0 &\implies \langle x, \nabla f(x)\rangle = 0 && (8.22)\\
&\implies \langle x, \nabla g(x)\rangle \approx \langle x, \nabla f(x)\rangle/p \pm \epsilon \qquad \text{(by Lemma 8.17)} && (8.23)\\
&\implies |\langle x, (zz^\top - xx^\top)x\rangle| \leq \epsilon \qquad \text{w.h.p.} && (8.24)\\
&\implies \langle x, z\rangle^2 \geq \|x\|_2^4 - \epsilon \qquad \text{w.h.p.} && (8.25)
\end{aligned}$$

□

Lemma 8.20 (Bound norm for g)**.** *If $\nabla^2 g(x) \succeq 0$, then $\|x\|_2^2 \geq 1/3$.*

Proof.

$$\begin{aligned}
\nabla^2 g(x) \succeq 0 &\implies \langle z, \nabla^2 g(x) z\rangle \geq 0 && (8.26)\\
&\implies \|zx^\top + xz^\top\|_F^2 - 2z^\top(zz^\top - xx^\top)z \geq 0 && (8.27)\\
&\implies 2\|x\|_2^2 + 2\langle x, z\rangle^2 - 2 + 2\langle x, z\rangle^2 \geq 0 \qquad \text{(cyclic trace prop.)} && (8.28)\\
&\implies 3\|x\|_2^2 = \|x\|_2^2 + 2\|x\|_2^2 \geq \|x\|_2^2 + 2\langle x, z\rangle^2 \geq 1 \qquad \text{(by Cauchy-Schwarz)} && (8.29)\\
&\implies \|x\|_2^2 \geq 1/3. && (8.30)
\end{aligned}$$

□

Lemma 8.21 (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.

$$\begin{aligned}
\nabla^2 f(x) \succeq 0 &\implies \langle z, \nabla^2 f(x) z\rangle \geq 0 && (8.31)\\
&\implies \langle z, \nabla^2 g(x) z\rangle \geq -\epsilon \qquad \text{w.h.p. (by Lemma 8.17)} && (8.32)\\
&\implies 3\|x\|_2^2 \geq 1 - \epsilon \qquad \text{w.h.p.} && (8.33)\\
&\implies \|x\|_2^2 \geq 1/3 - \epsilon/3 \qquad \text{w.h.p.} && (8.34)
\end{aligned}$$

□

Lemma 8.22 (g has no bad local minimum)**.** *All local minima of g are global minima.*

Proof.

$$\begin{aligned}
\nabla g(x) = 0 &\implies \langle z, \nabla g(x)\rangle = 0 && (8.35)\\
&\implies \langle z, (zz^\top - xx^\top)x\rangle = 0 && (8.36)\\
&\implies \langle x, z\rangle(1 - \|x\|_2^2) = 0. && (8.37)
\end{aligned}$$

Since $|\langle x, z\rangle| \geq 1/3 \neq 0$ (by Lemma 8.20), we must have $\|x\|_2^2 = 1$. But then Lemma 8.18 implies $\langle x, z\rangle^2 = \|x\|_2^4 = 1$, so $x = \pm z$ by Cauchy-Schwarz. □

We now prove Theorem 8.14, restated for convenience:

Theorem 8.23 (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 $\epsilon \leq 1/16$,

$$\begin{aligned}
\langle x, z\rangle^2 &\geq \|x\|_2^4 - \epsilon \qquad \text{(by Lemma 8.19)} && (8.38)\\
&\geq (1/3 - \epsilon/3)^2 - \epsilon \qquad \text{(by Lemma 8.21)} && (8.39)\\
&\geq 1/32. && (8.40)
\end{aligned}$$

With this, we can get a bound on $\|x\|_2^2$:

$$\begin{aligned}
\nabla f(x) = 0 &\implies \langle x, \nabla f(x)\rangle = 0 && (8.41)\\
&\implies |\langle z, \nabla g(x)\rangle| \le \epsilon && \text{(by Lemma 8.17)} \quad (8.42)\\
&\implies |\langle x, z\rangle| \cdot |1 - \|x\|_2^2| \le \epsilon && \text{(by dfn of } g\text{)} \quad (8.43)\\
&\implies |1 - \|x\|_2^2| \le 32\epsilon = O(\epsilon) && \text{(by (8.40))} \quad (8.44)\\
&\implies \|x\|_2^2 = 1 \pm O(\epsilon). && (8.45)
\end{aligned}$$

Next, we bound $\langle x, z\rangle$:

$$\begin{aligned}
\langle x, z\rangle^2 &\ge \|x\|_2^4 - \epsilon && \text{(by Lemma 8.19)} \quad (8.46)\\
&\ge (1 - O(\epsilon))^2 - \epsilon && \text{(by (8.45))} \quad (8.47)\\
&= 1 - O(\epsilon). && (8.48)
\end{aligned}$$

Finally, we put these quantities together to bound $\|x - z\|_2^2$. We have two cases:
Case 1: $\langle x, z\rangle \ge 1 - O(\epsilon)$. Then

$$\begin{aligned}
\|x - z\|_2^2 &= \|x\|_2^2 + \|z\|_2^2 - 2\langle x, z\rangle && (8.49)\\
&\le \|x\|_2^2 + 1 - 2\langle x, z\rangle && (8.50)\\
&\le 1 + O(\epsilon) + 1 - 2(1 - O(\epsilon)) && (8.51)\\
&\le O(\epsilon). && (8.52)
\end{aligned}$$

Hence we conclude x is $O(\sqrt{\epsilon})$-close to z.
Case 2: $\langle x, z\rangle \le -(1 - O(\epsilon))$. 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 8.17).

8.3.3 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:

$$\text{maximize} \quad \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d}\sum_{l=1}^{d} T_{ijkl} x_i x_j x_k x_l \quad \text{such that} \quad \|x\|_2 = 1. \qquad (8.53)$$

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].

8.4 The Neural Tangent Kernel (NTK) Approach

In the previous sections, we studied non-convex optimization problems in which all local minima are global. Selecting the parameters of a deep neural network is another commonly encountered non-convex optimization problem, but it is unrealistic to expect that all local minima will also be global minima in this setting. Here we consider a particular objective for which we can identify particular regions of the input space in which all local minima are also global minima. We can show that this objective corresponds to certain types of deep neural networks, but this analysis remains limited. For further reading about this approach to studying neural network optimization, see [Liang et al., 2018] and [Du and Hu, 2019].

To be more formal, we take an appropriate parameter initialization θ^0 such that in a neighborhood around it, which we denote by $B(\theta^0)$, the loss function is convex and its global minimum is attained. Figure 8.6 depicts a function and region for which this condition holds.

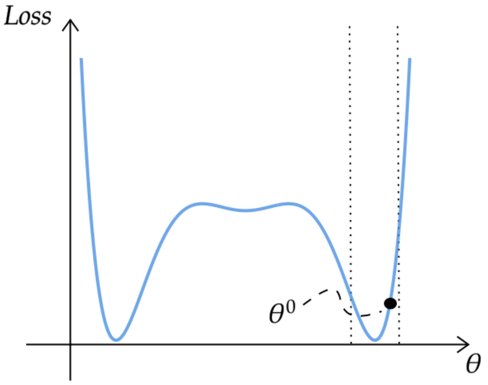


Figure 8.6: Training loss around an initialized θ^0. The dotted lines indicate $B(\theta^0)$, a region where the loss is convex, and where a global minimum exists.

Given a nonlinear $f_\theta(x)$, we examine the Taylor expansion at θ^0:

$$f_\theta(x) = \underbrace{f_{\theta^0}(x) + \langle \nabla_\theta f_{\theta^0}(x), \theta - \theta^0 \rangle}_{\triangleq g_\theta(x)} + \text{higher order terms} \tag{8.54}$$

Note that $g_\theta(x)$ is an affine function in θ, as $f_{\theta^0}(x)$ is a constant for fixed x, θ^0. Similarly, defining $\Delta\theta = \theta - \theta^0$, we can say that $g_\theta(x)$ is linear in $\Delta\theta$. For convenience, we will sometimes choose θ^0 such that $f_{\theta^0}(x) = 0$ for all x. It is easy to see why such an initialization exists. Consider splitting a two-layer neural network $f_\theta(x)$ with width $2m$ into two halves, each with m neurons; the outputs of these two networks are then given by $\sum_{i=1}^m a_i \sigma(w_i^\top x)$ and $\sum_{i=1}^m -a_i \sigma(w_i^\top x)$, respectively. Here, w_i can be randomly chosen so long as W_i is the same in both halves, and a_i can be randomly chosen as long as the other half is initialized with $-a_i$. Summing these two networks together yields $f_{\theta^0}(x) \equiv 0$ for all x.

When $f_{\theta^0}(x) \equiv 0$, we have that

$$g_\theta(x) = \langle \nabla_\theta f_{\theta^0}(x), \Delta\theta \rangle , \tag{8.55}$$

we observe that $\Delta\theta$ depends upon the parameter we evaluate the network at, while $\nabla_\theta f_{\theta^0}(x)$ can be thought of as a feature map since it is a fixed function of x (given the architecture and θ^0) that does not depend on θ whatsoever. We thus let $\phi(x) \triangleq \nabla_\theta f_{\theta^0}(x)$, which motivates the following definition:

Definition 8.24 (Neural Tangent Kernel)**.** For simplicity, we assume $f_{\theta^0}(x) = 0$ so that $y = y'$. The *neural tangent kernel* K is given by

$$\begin{aligned} K(x, x') &= \langle \phi(x), \phi(x') \rangle && (8.56)\\ &= \langle \nabla_\theta f_{\theta^0}(x), \nabla_\theta f_{\theta^0}(x') \rangle . && (8.57) \end{aligned}$$

Here, the feature $\nabla_\theta f_{\theta^0}(x)$ is precisely the gradient of the neural network. This is where the "tangent" in Neural Tangent Kernel comes from.

Instead of $f_\theta(x)$, suppose we use the approximation $g_\theta(x)$, which we recall is linear in θ. The kernel method gives a linear model on top of features. When $\theta \approx \theta^0$, given a convex loss function ℓ, we have

$$\underbrace{\ell(f_\theta(x), y)}_{\substack{\text{not}\\\text{necessarily}\\\text{convex}}} \approx \underbrace{\ell(g_\theta(x), y)}_{\text{convex}} . \tag{8.58}$$

Convexity of the RHS follows from the fact that a convex function, ℓ, composed with a linear function, g_θ, is still convex.

A natural question to ask is: how valid is this approximation? We devote the rest of this chapter to answering this question. First, we define the empirical loss:

$$\hat{L}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_\theta(x^{(i)}), y^{(i)}\right) \tag{8.59}$$

$$\hat{L}(g_\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(g_\theta(x^{(i)}), y^{(i)}\right) . \tag{8.60}$$

The key idea is that the Taylor approximation works for certain cases. We defer a more complete enumeration of these cases to a later section of this monograph. Here we outline the high-level approach we take to validate and use this Taylor expansion. Namely, we will show that there exists a neighborhood around θ^0 called $B(\theta^0)$, such that we have the following:

1. Accurate approximation: $f_\theta(x) \approx g_\theta(x)$, and $\hat{L}(f_\theta) \approx \hat{L}(g_\theta)$ for all $\theta \in B(\theta^0)$.

2. It suffices to optimize in $B(\theta^0)$: There exists an approximate global minimum $\hat{\theta} \in B(\theta^0)$, so $\hat{L}(g_{\hat{\theta}}) \approx 0$. This is the lowest possible loss (because the loss is nonnegative), which implies we are close to the global minimum. Because of 1, this implies that $\hat{L}(f_{\hat{\theta}}) \approx 0$ as well. See Figure 8.7 for an illustration.

3. Optimizing $\hat{L}(f_\theta)$ is similar to optimizing $\hat{L}(g_\theta)$ and does not leave $B(\theta^0)$, i.e. everything is confined to this region. Intuitively, this last point to some extent is "implied" by (1) and (2), but this claim still requires a formal proof.

Note (1), (2), and (3) can all be true in various settings. In particular, to attain all three, we will require:

(a) Overparametrization and/or a particular scaling of the initialized θ^0.

(b) Small (or even zero) stochasticity, so θ never leaves $B(\theta^0)$. This condition is guaranteed by a small learning rate or full-batch gradient descent.

Despite the limitations of the requirements of (a) and (b), the existence of such a region is still surprising. Given the loss landscape which could potentially be highly non-convex, it is striking to find a neighborhood where the loss function is convex (e.g. quadratic) with a global minimum. This suggests there is some flexibility in the loss landscape.

To begin our formal discussion, we start by providing tools for proving (1) and (2). Let

$$\phi^{(i)} = \phi(x^{(i)}) = \nabla_\theta f_{\theta^0}(x^{(i)}) \in \mathbb{R}^p \tag{8.61}$$

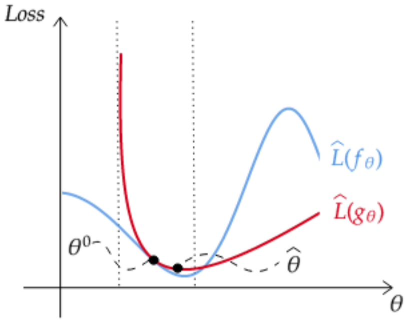


Figure 8.7: Here, $\hat{L}(g_\theta)$ and $\hat{L}(f_\theta)$ are both plotted. At $\hat{\theta}$, we have reached the approximate global minimum where $\hat{L}(g_{\hat{\theta}}) \approx 0$, in turn implying also that $\hat{L}(f_{\hat{\theta}}) \approx 0$.

and

$$\Phi = \begin{bmatrix} {\phi^{(1)}}^\top \\ \vdots \\ {\phi^{(n)}}^\top \end{bmatrix} \in \mathbb{R}^{n \times p} \tag{8.62}$$

where p is the number of parameters. Taking the quadratic loss, we have

$$\hat{L}(g_\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \phi\left(x^{(i)}\right)^\top \Delta\theta \right)^2 = \frac{1}{n} \|\vec{y} - \Phi \cdot \Delta\theta\|_2^2 \tag{8.63}$$

where $\vec{y} = \left[y^{(1)}, \cdots, y^{(n)}\right]^\top \in \mathbb{R}^n$. Note that this looks a lot like linear regression, where Φ and $\Delta\theta$ are the analogues of the design matrix and parameter, respectively. We further assume that $y^{(i)} = O(1)$ and $\|y\|_2 = O(\sqrt{n})$. Now, we can prove a lemma that addresses the second of the three conditions we described above, i.e. that it is sufficient to optimize in some small ball around θ^0.

Lemma 8.25 (for (2))**.** *Suppose we are in the setting where $p \geq n$, $\mathrm{rank}(\Phi) = n$, and $\sigma_{\min}(\Phi) = \sigma > 0$. Then, letting $\Delta\hat{\theta}$ denote the minimum norm solution, i.e. the nearest global minimum, of $\vec{y} = \Phi\Delta\theta$, we have*

$$\|\Delta\hat{\theta}\|_2 \leq O(\sqrt{n}/\sigma) \tag{8.64}$$

Remark 8.26. The meaning of the bound on $\Delta\hat{\theta}$ becomes clear if we consider the ball given by

$$B_{\theta^0} = \{\theta = \theta^0 + \Delta\theta : \|\Delta\theta\|_2 \leq O(\sqrt{n}/\sigma)\}. \tag{8.65}$$

In particular, notice that B_{θ^0} contains a global minimum, so this lemma characterizes how large the ball must be to contain a global minimum.

Remark 8.27. We also note that the condition $\mathrm{rank}(\Phi) = n$ and $\sigma > 0$ can be thought of as a "finite-sample expressivity" condition, saying that the features Φ are expressive enough so that there exists a linear model on top of these features that perfectly fit the data. The condition $\mathrm{rank}(\Phi) = n$ requires $p \geq n$—so we need some amount of over-parameterization to apply these analysis.

Proof. Letting Φ^+ denote the Moore-Penrose pseudoinverse of Φ, note that $\Delta\hat{\theta} = \Phi^+\boldsymbol{y}$, and $\|\Phi^+\|_{\mathrm{op}} = \frac{1}{\sigma_{\min}(\Phi)} = \frac{1}{\sigma}$. A simple argument shows

$$\|\Delta\hat{\theta}\|_2 \leq \|\Phi^+\|_{\mathrm{op}} \cdot \|\vec{y}\|_2 \tag{8.66}$$

$$\leq O\left(\frac{1}{\sigma} \cdot \sqrt{n}\right), \tag{8.67}$$

where the last inequality follows from the assumption that $\|\vec{y}\|_2 \leq O(\sqrt{n})$. □

Next, we prove a lemma that addresses the first of the three steps we described above.

Lemma 8.28 (for (1))**.** *Suppose $\nabla_\theta f_\theta(x)$ is β-Lipschitz in θ, i.e. for every x, and θ, θ', we have*

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2 \leq \beta \cdot \|\theta - \theta'\|_2. \tag{8.68}$$

Then,

$$|f_\theta(x) - g_\theta(x)| \leq O\left(\beta \|\Delta\theta\|_2^2\right). \tag{8.69}$$

If we further restrict our choice of θ using B_{θ^0} as defined in Remark 8.26, we obtain that

$$|f_\theta(x) - g_\theta(x)| \leq O\left(\frac{\beta n}{\sigma^2}\right), \quad \forall \theta \in B_{\theta^0}. \tag{8.70}$$

Proof. The proof comes from the following fact: if $h(\theta)$ is such that $\nabla h(\theta)$ is β-Lipschitz (which if differentiable is equivalent to $\|\nabla^2 h(\theta)\|_{\text{op}} \leq \beta$), then

$$\left| \underbrace{h(\theta)}_{f_\theta(x)} \underbrace{- h(\theta^0) - \left\langle \nabla h(\theta^0), \theta - \theta^0 \right\rangle}_{-g_\theta(x)} \right| \leq O\left(\beta \|\theta - \theta^0\|_2^2\right). \tag{8.71}$$

As shown above, the proof is as simple as plugging in $f_\theta(x) = h(\theta)$ and $g_\theta(x) = h(\theta^0) + \left\langle \nabla h(\theta^0), \Delta\theta \right\rangle$. □

Remark 8.29*.* The lemma above bounds the approximation error. Intuitively, as you move farther away from θ^0, the Taylor approximation gets worse; the approximation error is bounded above by a second order $\Delta\theta$ term.

Remark 8.30*.* Note that if f_θ involves a relu function, then ∇f_θ is not continuous everywhere. This requires a technical fix outside the scope of our discussion.[2]

8.4.1 Two examples of the NTK regime

By (8.70), we have now established a bound on our approximation error, but we have yet to analyze how good it is, as $\beta n/\sigma^2$ is neither obviously either big nor small. An important fact to notice is that β/σ^2 is not scaling invariant, so we can play with the scaling in order to drive this term to 0. In particular, there are two notable cases (with specific parameterization, initialization, etc) where $\beta/\sigma^2 \to 0$. In the literature, such situation is often referred to as the NTK regime or the lazy training regime [Chizat and Bach, 2018].

1. **Reparameterize with a scalar** [Chizat and Bach, 2018]. Let $f_\theta(x) = \alpha \cdot \bar{f}_\theta(x)$ where $\bar{f}_\theta(x)$ is an arbitrary neural net with fixed width and depth. We only vary α, i.e. the scaling, and we see how the crucial quantity β/σ^2 changes accordingly. Fix an initial θ^0, and let

$$\bar{\sigma} = \sigma_{\min}\left(\begin{bmatrix} \nabla_\theta \bar{f}_{\theta^0}\left(x^{(1)}\right)^\top \\ \vdots \\ \nabla_\theta \bar{f}_{\theta^0}\left(x^{(n)}\right)^\top \end{bmatrix}\right). \tag{8.72}$$

   Furthermore, let $\bar{\beta}$ be the Lipschitz parameter of $\nabla_\theta \bar{f}_\theta(x)$ in θ. A simple chain-rule gradient argument shows that scaling $\bar{f}_\theta$ by α also scales σ and β accordingly, i.e. $\sigma = \alpha\bar{\sigma}$, and $\beta = \alpha\bar{\beta}$. Some straightforward algebra yields

$$\frac{\beta}{\sigma^2} = \frac{\bar{\beta}}{\bar{\sigma}^2} \cdot \frac{1}{\alpha} \to 0 \quad \text{as} \quad \alpha \to \infty. \tag{8.73}$$

   Once α becomes big enough, then by Lemma 8.28, the approximation $|f_\theta(x) - g_\theta(x)| \leq O\left(\beta n/\sigma^2\right)$ becomes very good.

[2] A relu function is continuous almost everywhere, so we can make some minor fixes and still use some modified notion of Lipschitzness to derive an upper bound.

Remark 8.31. A priori, such a phenomenon may appear to be too good to be true. To understand it better, we first note that this re-parameterizaton does not change the scale of the loss, but rather change the shape of the loss function. Intuitively, as α becomes larger, the function f_θ becomes sharper and more non-smooth (leading to higher approximation error). However, on the other hand, we note that we only need to travel a little bit away from θ^0 to find a global minimum given that there is a global minimum within radius $O(\sqrt{n}/\sigma)$. It turns out that the radius needed shrinks faster than the smoothness grows.

To visualize this effect, we can consider the following example with only 1 data point with 1-dimensional input $(x, y) = (1, 1)$ and the quadratic model $\bar{f}_\theta(x) = x(\theta + \beta\theta^2) = \theta + \beta\theta^2$. Using the squared loss, we have

$$\widehat{L}(\bar{f}_\theta) = (1 - (\theta + \beta\theta^2))^2 \tag{8.74}$$

Let $\theta^0 = 0$. Taylor expanding at θ^0 gives the linear approximation $\bar{g}_\theta(x) = \theta x = \theta$, and the resulting loss function that is quadratic

$$\widehat{L}(\bar{g}_\theta) = (1 - \theta)^2 \tag{8.75}$$

In this case, $\nabla f_{\theta^0}(x) = 2\beta\theta x = 2\beta\theta$ is 2β-Lipschitz, and $\sigma = 1$.

Now we vary α and get

$$\widehat{L}(\alpha\bar{f}_\theta) = (1 - \alpha(\theta + \beta\theta^2))^2 \tag{8.76}$$

and

$$\widehat{L}(\alpha\bar{g}_\theta) = (1 - \alpha\theta)^2 \tag{8.77}$$

Note that the minimizer of $\widehat{L}(\alpha\bar{g}_\theta)$ is $1/\alpha$, which is closer to θ^0 as $\alpha \to \infty$. We zoom into the region $[0, 1/\alpha]$ and find out the difference between $\alpha\bar{f}_\theta$ and $\alpha\bar{g}_\theta$ is $\alpha\beta\theta^2 \leq \beta/\alpha$, which is much smaller than the value of $\alpha\bar{g}_\theta \approx O(1)$.

We visualize the these functions in Figure 8.8. We observe that $\widehat{L}(\alpha\bar{g}_\theta)$ becomes a better approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ in the region $[0, 1/\alpha]$ as $\alpha \to \infty$ (though $\widehat{L}(\alpha\bar{g}_\theta)$ is a worse approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ globally.)

2. **Overparametrization (with specific initialization)**. Early papers on the NTK take this approach (e.g., [Li and Liang, 2018, Du and Hu, 2019]). Consider a two-layer network with m neurons.

$$\hat{y} = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} a_i \sigma(w_i^\top x) \tag{8.78}$$

The scaling $1/\sqrt{m}$ is to ensure that a random initialization with constant scale will have output on the right order, as we see momentarily. We make the following assumptions regarding the network and its inputs.

$$W = \begin{bmatrix} w_1^\top \\ \vdots \\ w_m^\top \end{bmatrix} \in \mathbb{R}^{m \times d} \tag{8.79}$$

$$\sigma \text{ is 1-Lipschitz and twice-differentiable} \tag{8.80}$$

$$a_i \sim \{\pm 1\} \qquad \text{(not optimized)} \tag{8.81}$$

$$w_i^0 \sim \mathcal{N}(0, I_d) \tag{8.82}$$

$$\|x\|_2 = \Theta(1) \tag{8.83}$$

$$\theta = \text{vec}(W) \in \mathbb{R}^{dm} \qquad \text{(vectorized } W\text{)} \tag{8.84}$$

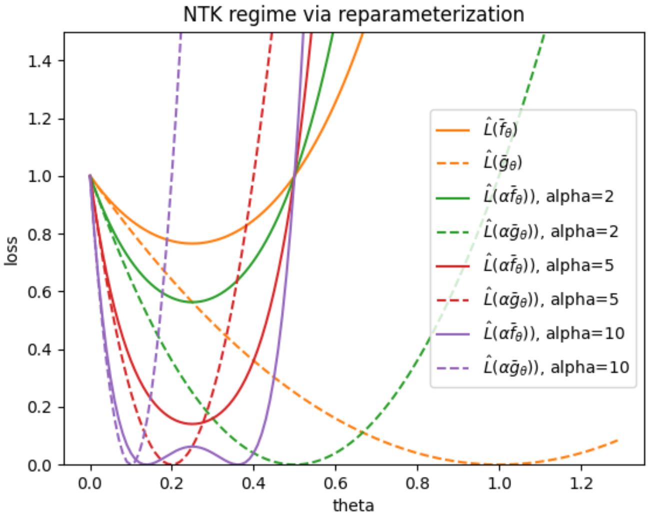


Figure 8.8: The approximation $\widehat{L}(\alpha\bar{g}_\theta)$ becomes a better approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ in the region $[0, 1/\alpha]$ as $\alpha \to \infty$ (though $\widehat{L}(\alpha\bar{g}_\theta)$ is a worse approximation of $\widehat{L}(\alpha\bar{f}_\theta)$ globally).

We will assume $m \to \infty$ polynomially in n and d. In particular, for fixed n, d, we have $m = \mathsf{poly}(n, d)$. Why do we use the $1/\sqrt{m}$ scaling? Note that $\sigma\left(w_i^0{}^\top x\right) \approx 1$ because $\|x\|_2 = \Theta(1)$ and w_i^0 is drawn from a spherical Gaussian. Thus, as some a_i are positive and others are negative, $\left|\sum_{i=1}^m a_i \sigma\left(w_i^0{}^\top x\right)\right| = \Theta\left(\sqrt{m}\right)$, and finally $f_{\theta^0}(x) = \Theta(1)$.

Now we analyze σ and β. We let

$$\sigma = \sigma_{\min}(\Phi) = \sqrt{\sigma_{\min}\left(\Phi\Phi^\top\right)} \tag{8.85}$$

where

$$\left(\Phi\Phi^\top\right)_{ij} = \left\langle \nabla_\theta f_{\theta^0}\left(x^{(i)}\right), \nabla_\theta f_{\theta^0}\left(x^{(j)}\right)\right\rangle \tag{8.86}$$

Note that the gradient with respect to w_i is given by

$$\frac{\partial f_\theta(x)}{\partial w_i} = \frac{1}{\sqrt{m}} \sigma'(w_i^\top x) \cdot x \tag{8.87}$$

Now observe that

$$\begin{aligned}
\|\nabla f_\theta(x)\|_2^2 &= \frac{1}{m}\sum_{i=1}^m \|\sigma'\left(w_i{}^\top x\right) \cdot x\|_2^2 && (8.88)\\
&= \frac{1}{m}\|x\|_2^2 \cdot \sum_{i=1}^m \left(\sigma'\left(w_i{}^\top x\right)\right)^2 && (8.89)\\
&\to \mathop{\mathbb{E}}_{w \sim \mathcal{N}(0, I_d)}\left[\sigma'\left(w^\top x\right)^2\right] \cdot \|x\|_2^2 \quad \text{as} \quad m \to \infty && (8.90)\\
&= O(1) \qquad\qquad \text{(not depending on } m\text{)} && (8.91)
\end{aligned}$$

where the penultimate line follows from the law of large numbers, as $\frac{1}{m}\sum_{i=1}^m \left(\sigma'(w_i^\top x)\right)^2$ can be interpreted as a mean.

Note that the scale of $\|\nabla_\theta f_{\theta^0}(x)\|_2$ does not depend on m, so the inner product in (8.86) also does not depend on m either. As above, we can show

$$\langle \nabla_\theta f_{\theta^0}(x), \nabla_\theta f_{\theta^0}(x')\rangle = \frac{1}{m}\langle x, x'\rangle \sum_{i=1}^{m} \sigma'(w^\top x)\sigma'(w^\top x') \tag{8.92}$$

$$\to \mathop{\mathbb{E}}_{w\sim\mathcal{N}(0,I_d)}\left[\sigma'(w^\top x)\sigma'(w^\top x')\right]\langle x, x'\rangle \tag{8.93}$$

(8.93) implies that as $m \to \infty$, $\Phi\Phi^\top$ converges to a constant matrix denoted by

$$K^\infty = \lim_{m\to\infty} \Phi\Phi^\top \tag{8.94}$$

This is precisely the NTK with $m = \infty$. Though we omit the proof of this claim, it can be shown that K^∞ is full rank. Then, let

$$\sigma_{\min} \triangleq \sigma_{\min}(K^\infty) > 0. \tag{8.95}$$

We can show that

$$\sigma = \sigma_{\min}\left(\Phi\Phi^\top\right) > \frac{1}{2}\sigma_{\min} \tag{8.96}$$

Intuitively, $\Phi\Phi^\top \to K^\infty$, so the spectrum of the matrix should also converge. Thus, in some sense, we have shown that σ is constant in the limit.

Now what about β? If we can show $\beta \to 0$ as $m \to \infty$, we are done. We begin by analyzing this key expression:

$$\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x) = \left[\frac{1}{\sqrt{m}}\left(\sigma'\big(w_i^\top x\big) - \sigma'\big({w'_i}^\top x\big)\right)\cdot x\right]_{i=1}^{m} \tag{8.97}$$

Note that (8.97) above consists of matrices, as θ is a vectorized matrix. Then,

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2^2 = \frac{1}{m}\sum_{i=1}^{m} \|x\|_2^2\left(\sigma'\big(w_i^\top x\big) - \sigma'\big({w_i}'^\top x\big)\right)^2 \tag{8.98}$$

$$\le O\left(\frac{1}{m}\sum_{i=1}^{m}\|x\|_2^2\big(w_i^\top x - {w'_i}^\top x\big)^2\right) \tag{8.99}$$

$$= O\left(\frac{1}{m}\sum_{i=1}^{m}\|w_i - w'_i\|_2^2\right) \tag{8.100}$$

$$= O\left(\frac{1}{m}\|\theta - \theta'\|_2^2\right) \tag{8.101}$$

The first line follows from the fact that $\frac{1}{\sqrt{m}}\left(\sigma'\big(w_i^\top x\big) - \sigma'\big({w'_i}^\top x\big)\right)$ is a scalar. The second line uses the assumption that σ' is $O(1)$-Lipschitz. The third line uses Cauchy-Schwarz and the fact that $\|x\|_2^2 \approx 1$. Taking the square root, we have that

$$\|\nabla_\theta f_\theta(x) - \nabla_\theta f_{\theta'}(x)\|_2 \lesssim \frac{1}{\sqrt{m}}\|\theta - \theta'\|_2 \tag{8.102}$$

Thus, the Lipschitz parameter is $\beta = O(1/\sqrt{m})$. Thus, our key quantity β/σ^2 goes to 0 as m grows. Namely,

$$\frac{\beta}{\sigma^2} \approx \frac{1}{\sqrt{m}}\cdot\frac{1}{\sigma_{\min}^2} \to 0 \quad \text{as} \quad m \to \infty. \tag{8.103}$$

Recall here that $\sigma_{\min}$ does not depend on m. Concretely, this result tells us that our function becomes more smooth (the gradient has a smaller Lipschitz constant) as we add more neurons.

8.4.2 Optimizing $\hat{L}(g_\theta)$ vs. $\hat{L}(f_\theta)$

We now discuss how to establish the last of the three conditions under which we claimed a Taylor approximation is reasonable. We need to show that optimizing $\hat{L}(f_\theta)$ is similar to optimizing $\hat{L}(g_\theta)$. To do so, we require two steps:

(A) Analyze optimization of $\hat{L}(g_\theta)$.

(B) Analyze optimization of $\hat{L}(f_\theta)$ by re-using or modifying the proofs in (A).

There are two approaches in the literature for (A), which implies that there exist two approaches for (B) as well.

(i) We leverage the strong convexity of $\hat{L}(g_\theta)$, and then show an exponential convergence rate.[3]

(ii) Instead of strong convexity, we rely on the smoothness of f_θ (i.e. bounded second derivative).

We will only discuss the first of these two methods in the sequel.

Remark 8.32. In both either approach (i) or (ii), we will implicitly or explicitly use the following simple fact. Suppose at any θ^t, we take the Taylor expansion of f_θ at θ^t:

$$g_\theta^t(x) = f_{\theta^t}(x) + \left\langle \nabla f_{\theta^t}(x), \theta - \theta^t \right\rangle \tag{8.105}$$

Consider the gradient we are interested in taking: $\nabla \hat{L}(f_{\theta^t})$. Notice that:

$$\nabla \hat{L}(f_{\theta^t}) = \nabla \hat{L}(g_{\theta^t}^t) \tag{8.106}$$

This is really saying that f_θ and g_θ^t agree up to first-order at θ^t. This implies that $L(f_\theta)$ and $L(g_\theta^t)$ also agree to first-order at θ^t. This also means that T steps of gradient descent on $\hat{L}(f_\theta)$ is the same as performing online gradient descent[4] on a sequence of changing objectives $L(g_\theta^0), \dots, L(g_\theta^T)$, and this online learning perspective is useful in the approach (ii).

We will now show that under the strong convexity regime, optimizing a neural network f_θ is equivalent to optimizing a linear model g_θ. We will also observe that this regime is not particularly practically relevant, but this analysis is nevertheless of interest to us for two reasons. First, the approach used in the subsequent exposition is of technical interest and second, it remains quite interesting that optimizing f_θ and optimization g_θ yields the same results under *any* regime.

Optimizing g_θ

We relate the optimization of g_θ to performing linear regression. Recall that we can think of $\nabla f_{\theta^0}(x)$ as a feature map. Then, the problem of choosing $\Delta\theta$ to get $g_\theta(x)$ to be close to $\vec{y}$ is a linear regression. In particular, we use gradient descent to minimize

$$\|\vec{y} - \Phi \Delta\theta\|_2^2, \tag{8.107}$$

where

$$\Phi = \begin{bmatrix} \nabla f_{\theta^0}(x^{(1)})^\top \\ \vdots \\ \nabla f_{\theta^0}(x^{(n)})^\top \end{bmatrix} \in \mathbb{R}^{n \times p}. \qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} \in \mathbb{R}^n \tag{8.108}$$

[3] Recall that a differentiable function f is μ-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|_2^2 \tag{8.104}$$

for some $\mu > 0$ and all x, y.

[4] Online gradient descent is the algorithm that takes one gradient descent step upon receiving a new objective function. See Chapter 11 for more discussions about online learning.

For learning rate η, the gradient descent update rule is

$$\Delta\theta^{t+1} = \Delta\theta^t - \eta\Phi^\top(\Phi\Delta\theta^t - \vec{y}). \tag{8.109}$$

This analysis considers changes in the output space. Define $\hat{y}^t = \Phi\Delta\theta^t$. Then, we're interested in changes in

$$\begin{aligned}
\hat{y}^{t+1} - \vec{y} &= \Phi\Delta\theta^{t+1} - \vec{y} && (8.110)\\
&= \Phi\left(\Delta\theta^t - \eta\Phi^\top(\Phi\Delta\theta^t - \vec{y})\right) - \vec{y} \qquad \text{(by (8.109))} && (8.111)\\
&= \left(\Phi - \eta\Phi\Phi^\top\Phi\right)\Delta\theta^t - (I - \eta\Phi\Phi^\top)\vec{y} && (8.112)\\
&= (I - \eta\Phi\Phi^\top)\Phi\Delta\theta^t - (I - \eta\Phi\Phi^\top)\vec{y} && (8.113)\\
&= (I - \eta\Phi\Phi^\top)(\Phi\Delta\theta^t - \vec{y}) && (8.114)\\
&= (I - \eta\Phi\Phi^\top)(\hat{y}^t - \vec{y}). && (8.115)
\end{aligned}$$

From this decomposition, we see that the residuals, $\hat{y}^t - \vec{y}$, are monotonically shrinking since $\eta\Phi\Phi^\top$, i.e. the term we are subtracting from I in (8.115), is positive semidefinite. Next, we quantify how quickly we are shrinking the residuals. Define

$$\tau^2 = \sigma_{\max}(\Phi\Phi^\top) \tag{8.116}$$

$$\sigma = \sigma_{\min}(\Phi) = \sqrt{\sigma_{\min}(\Phi\Phi^\top)}. \tag{8.117}$$

Then, we claim that when $\eta \le \frac{1}{\tau^2}$,

$$\|I - \eta\Phi\Phi^\top\|_{\text{op}} \le 1 - \eta\sigma^2. \tag{8.118}$$

Why? Let the eigenvalues of $\Phi\Phi^\top$ be (in descending order) $\tau_1^2, \dots, \tau_n^2$. By definition, $\tau_1^2 = \tau^2$ and $\tau_n^2 = \sigma^2$. Now, given the singular value decomposition, $\Phi = U\Sigma V^\top$, we obtain the eigendecomposition:

$$\begin{aligned}
I - \eta\Phi\Phi^\top &= I - \eta U\Sigma^2U^\top && (8.119)\\
&= UU^\top - \eta U\Sigma^2U^\top && (8.120)\\
&= U(I - \eta\Sigma^2)U^\top. && (8.121)
\end{aligned}$$

(8.121) is the eigendecomposition of $I - \eta\Phi\Phi^\top$, so $I - \eta\Phi\Phi^\top$ has eigenvalues $1 - \eta\tau_1^2, \dots, 1 - \eta\tau_n^2$. Note that assuming $\eta \le \frac{1}{\tau^2}$ ensures that all eigenvalues of $I - \eta\Phi\Phi^\top$ are non-negative. Thus,

$$\begin{aligned}
\|I - \eta\Phi\Phi^\top\|_{\text{op}} &\le \max_j |1 - \eta\tau_j^2| && (8.122)\\
&= 1 - \eta\tau_n^2 && (8.123)\\
&= 1 - \eta\sigma^2, && (8.124)
\end{aligned}$$

where the non-negativity of $1 - \eta\tau_j^2$ for all j implies (8.123).

Using this result, we obtain our desired result. Namely, assuming $\eta \le \frac{1}{\tau^2}$,

$$\begin{aligned}
\|\hat{y}^{t+1} - \vec{y}\|_2 &= \|I - \eta\Phi\Phi^\top\|_{\text{op}} \cdot \|\hat{y}^t - \vec{y}\|_2 && (8.125)\\
&\le (1 - \eta\sigma^2)\|\hat{y}^t - \vec{y}\|_2 && (8.126)\\
&\le (1 - \eta\sigma^2)^{t+1}\|\hat{y}^0 - \vec{y}\|_2. && (8.127)
\end{aligned}$$

This yields the desired exponential decay in the error. Thus, after $T = O\left(\frac{\log 1/\epsilon}{\eta\sigma^2}\right)$ iterations,

$$\|\hat{y}^T - \vec{y}\|_2 \le \epsilon\|\hat{y}^0 - \vec{y}\|_2. \tag{8.128}$$

Optimizing f_θ

We now transition to an analysis of the optimization of f_θ. Our key result is Theorem 8.33. If we compare it against what we have in (8.128), we see the claimed similarity between f_θ and g_θ in error decay under optimization.

Theorem 8.33. *There exists a constant $c_0 \in (0,1)$ such that for $\frac{\beta}{\sigma^2} \leq \frac{c_0}{n}$ and sufficiently small η (which could depend on β, σ, or p), $\hat{L}\left(f_{\theta^T}\right) \leq \epsilon$ after $T = O\left(\frac{\log 1/\epsilon}{\eta\sigma^2}\right)$ steps.*

Proof. (This is actually a proof sketch that elides a few technical details for the sake of a simpler exposition.) Our approach is to follow the preceding analysis of g_θ, making changes where necessary.

Let

$$\Phi^t = \begin{bmatrix} \nabla f_{\theta^t}(x^{(1)})^\top \\ \vdots \\ \nabla f_{\theta^t}(x^{(n)})^\top \end{bmatrix} \in \mathbb{R}^{n \times p}. \tag{8.129}$$

To obtain our gradient descent update rule, we find, using the chain rule,

$$\begin{aligned} \nabla \hat{L}\left(f_{\theta^t}\right) &= \sum_{i=1}^n \left(f_{\theta^t}\left(x^{(i)}\right) - y^{(i)}\right) \nabla f_{\theta^t}\left(x^{(i)}\right) && (8.130) \\ &= \sum_{i=1}^n \left(\hat{y}^{(i),t} - y^{(i)}\right) \nabla f_{\theta^t}\left(x^{(i)}\right) && (8.131) \\ &= (\Phi^t)^\top \left(\hat{y}^t - \vec{y}\right). && (8.132) \end{aligned}$$

This results in the policy

$$\begin{aligned} \theta^{t+1} &= \theta^t - \eta \nabla \hat{L}\left(f_{\theta^t}\right) && (8.133) \\ &= \theta^t - \eta (\Phi^t)^\top \left(\hat{y}^t - \vec{y}\right) && (8.134) \\ &= \theta^t - \eta b^t, && (8.135) \end{aligned}$$

where we have let $b^t = (\Phi^t)^\top \left(\hat{y}^t - \vec{y}\right)$. Following our treatment of g_θ, we want to express $\hat{y}^{t+1}$ as a function of $\hat{y}^t$. The challenge now is that f is nonlinear. To deal with this, we Taylor expand f_θ at θ_t:

$$\begin{aligned} f_{\theta^{t+1}}(x^{(i)}) &= f_{\theta^t}(x^{(i)}) + \left\langle \nabla f_{\theta^t}(x^{(i)}), \theta^{t+1} - \theta^t \right\rangle + \text{high order terms} && (8.136) \\ &= f_{\theta^t}(x^{(i)}) + \left\langle \nabla f_{\theta^t}(x^{(i)}), -\eta b^t \right\rangle + O\left(\|\theta^{t+1} - \theta^t\|_2^2\right). && (8.137) \end{aligned}$$

Since $O\left(\|\theta^{t+1} - \theta^t\|_2^2\right)$ is $O\left(\eta^2\right)$, we can ignore this term as $\eta \to 0$. Vectorizing (8.137) without $O\left(\|\theta^{t+1} - \theta^t\|_2^2\right)$,

$$\begin{aligned} \hat{y}^{t+1} &= \hat{y}^t - \eta \Phi^t b^t && (8.138) \\ &= \hat{y}^t + \eta \Phi^t \left(\Phi^t\right)^\top (\vec{y} - \hat{y}^t). && (8.139) \end{aligned}$$

Subtracting $\vec{y}$ and re-arranging,

$$\begin{aligned} \hat{y}^{t+1} - \vec{y} &= \hat{y}^t - \vec{y} + \eta \Phi^t \left(\Phi^t\right)^\top (\vec{y} - \hat{y}^t) && (8.140) \\ &= \left(I - \eta \Phi^t \left(\Phi^t\right)^\top\right) \left(\hat{y}^t - \vec{y}\right). && (8.141) \end{aligned}$$

Comparing (8.141) with (8.115), we see one difference: in (8.141), our convergence depends on $\eta\Phi^t\left(\Phi^t\right)^\top$, which is a matrix that changes as we iterate, whereas in (8.115), convergence is controlled by a matrix that is fixed as we iterate.

To understand the convergence implications of (8.141), we examine the eigenvalues of $I - \eta\Phi^t\left(\Phi^t\right)^\top$. For now, suppose

$$\|\theta^t - \theta^0\|_2 \leq \sigma/(4\sqrt{n}\beta) \tag{8.142}$$

at time t. This implies that $\|\Phi^t - \Phi\|_F \leq \frac{\sigma}{4}$ by the Lipschitzness of $\nabla f_\theta(x)$ in θ. Then, we claim that

$$\sigma_{\min}(\Phi^t) \geq 3\sigma/4. \tag{8.143}$$

Why does (8.143) hold? Observe that

$$\begin{aligned}
\sigma_{\min}(\Phi^t) &= \min_{\|x\|_2=1} x^\top \Phi^t x &(8.144)\\
&\geq \min_{\|x\|_2=1} x^\top(\Phi^t - \Phi)x + \min_{\|x\|_2=1} x^\top \Phi x. &(8.145)
\end{aligned}$$

We can lower bound the first term of (8.145) as follows:

$$\begin{aligned}
x^\top(\Phi^t - \Phi)x &\geq -|\left\langle x, (\Phi^t - \Phi)x\right\rangle| && &(8.146)\\
&\geq -\|x\|_2\|(\Phi^t - \Phi)x\|_2 && \text{(Cauchy-Schwarz)} &(8.147)\\
&\geq -\|\Phi^t - \Phi\|_2 && (\|x\|_2 = 1) &(8.148)\\
&\geq -\sigma/4 && \text{(Lipschitzness of } \Phi\text{)}. &(8.149)
\end{aligned}$$

Next, we note that the second term of (8.145) is lower bounded by σ by simplifying and applying the definition of σ given in (8.117). Combining this observation with (8.149), we conclude that (8.143) must hold.

Applying this lower bound on the eigenvalues of Φ^t, we can use the same argument we used to establish (8.118) to conclude that

$$\|I - \eta\Phi^t\left(\Phi^t\right)^\top\|_{\text{op}} \leq 1 - 3\eta\sigma/4, \tag{8.150}$$

and

$$\|\hat{y}^{t+1} - \vec{y}\|_2 \leq (1 - 3\eta\sigma/4)^{t+1}\|\hat{y}^0 - \vec{y}\|_2. \tag{8.151}$$

So, as desired, we see exponential decay in the error at each iteration and after $T = O\left(\frac{\log 1/\epsilon}{n\sigma^2}\right)$ iterations,

$$\hat{L}(f_{\theta^T}) \leq \epsilon. \tag{8.152}$$

To complete our proof, observe that this argument is predicated upon the assumption that $\|\theta^t - \theta^0\|_2 \leq \sigma/(4\sqrt{n}\beta)$. This assumption is reasonable, however, given what we have already proven. Recall that in Lemma 8.25, we proved that

$$\|\Delta\hat{\theta}\|_2 = \|\hat{\theta} - \theta^0\|_2 \lesssim \sqrt{n}/\sigma. \tag{8.153}$$

Thus, when $\beta/\sigma^2 \to 0$, eventually, $\sqrt{n}/\sigma \ll \sigma/(4\sqrt{n}\beta)$. To extend this to $\|\hat{\theta} - \theta^t\|_2$ for arbitrary t, we heuristically argue that since the empirical minimizer is within $\sigma/(4\sqrt{n}\beta)$ of θ^0, we would not expect to have traveled more than $\sigma/(4\sqrt{n}\beta)$ from θ^0 at *any* iteration.

More formally, we claim that for all $t \in \mathbb{N}$,

$$\|\hat{y}^t - \vec{y}\|_2 \leq \mathcal{O}(\sqrt{n}). \tag{8.154}$$

We proceed via induction. For $t = 0$, because each element of $\hat{y}$ is of order 1, we know that:

$$\frac{1}{\sqrt{n}}\|\hat{y}^0 - \vec{y}\|_2 \leq O(1). \tag{8.155}$$

Now, suppose that (8.154) holds for some t. Then, because the errors are monotonically decreasing, (cf. (8.141) and (8.150)),

$$\frac{1}{\sqrt{n}}\|\hat{y}^{t+1} - \vec{y}\|_2 \leq \frac{1}{\sqrt{n}}\|\hat{y}^t - \vec{y}\|_2 \leq O(1). \tag{8.156}$$

Thus, (8.154) holds for all $t \in \mathbb{N}$.

Next, applying Lemma 8.28 with $\theta = \theta^t$ and our assumption that $\frac{\beta}{\sigma^2} \lesssim \frac{1}{n}$, we conclude that:

$$\frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t\|_2 \leq O(1) \tag{8.157}$$

Using this result and (8.154), we can show that $\frac{1}{\sqrt{n}}\|\Phi(\theta^t - \hat{\theta})\|_2$ is $O(1)$.

$$\begin{aligned}
\frac{1}{\sqrt{n}}\|\Phi(\theta^t - \hat{\theta})\|_2 &= \frac{1}{\sqrt{n}}\|\Phi\theta^t - \vec{y}\|_2 && (\vec{y} = \Phi\hat{\theta}) && (8.158)\\
&= \frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t + \hat{y}^t - \vec{y}\|_2 && && (8.159)\\
&\leq \frac{1}{\sqrt{n}}\|\Phi\theta^t - \hat{y}^t\|_2 + \frac{1}{\sqrt{n}}\|\hat{y}^t - \vec{y}\|_2 && \text{(triangle ineq.)} && (8.160)\\
&\leq O(1). && && (8.161)
\end{aligned}$$

Then, leveraging the definition of σ given in (8.117) and rearranging, we obtain (nearly) the desired result:

$$\|\theta^t - \hat{\theta}\|_2 \leq \frac{1}{\sigma}\|\Phi(\theta^t - \hat{\theta})\|_2 \leq O(\sqrt{n}/\sigma). \tag{8.162}$$

Recall that in Lemma 8.25, we proved that

$$\|\hat{\theta} - \theta^0\|_2 \leq O(\sqrt{n}/\sigma). \tag{8.163}$$

If $\beta/\sigma^2 \ll 1/n$, we conclude that

$$\begin{aligned}
\|\theta^t - \theta^0\|_2 &\leq \|\hat{\theta} - \theta^0\|_2 + \|\theta^t - \hat{\theta}\|_2 && \text{(triangle ineq.)} && (8.164)\\
&\leq O\left(\frac{\sqrt{n}}{\sigma}\right) \leq \frac{\sigma}{4\sqrt{n}\beta}. && && (8.165)
\end{aligned}$$

□

8.4.3 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 8.34. 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):

$$\begin{aligned} \hat{y} &= \frac{1}{2}\left[\phi_{\text{relu}}(x_1 + x_2) + \phi_{\text{relu}}(-x_1 - x_2) - \phi_{\text{relu}}(x_1 - x_2) - \phi_{\text{relu}}(x_2 - x_1)\right] && (8.166) \\ &= \frac{1}{2}\left(|x_1 + x_2| - |x_1 - x_2|\right) && (8.167) \\ &= x_1 x_2. && (8.168) \end{aligned}$$

(8.167) follows from the fact that $\phi_{\text{relu}}(t) + \phi_{\text{relu}}(-t) = |t|$ for all t, while (8.168) 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 8.35. *The kernel method with NTK requires $n = \Omega(d^2)$ samples to learn Example 8.34 well. In contrast, the neural network regularized by $\sum_{j=1}^{m} |u_j| \|w_j\|_2$ only requires $n = O(d)$ samples.*

Chapter 9

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.

9.1 Implicit regularization effect of zero initialization 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 $X \triangleq \left[x^{(1)}, ..., x^{(n)}\right]^\top \in \mathbb{R}^{n\times d}$ denote our data matrix and $\vec{y} \triangleq \left[y^{(1)}, ..., y^{(n)}\right]^\top \in \mathbb{R}^n$ 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 $\widehat{L}(\beta) \triangleq \frac{1}{2}\|\vec{y} - X\beta\|_2^2$.

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 9.1. *Let X^+ denote the pseudoinverse*[1] *of X. Then β is a global minimizer if and only if $\beta = X^+\vec{y} + \zeta$ for some ζ such that $\zeta \perp x_1, ..., x_n$.*

Proof. For any $\beta \in \mathbb{R}^d$, we can decompose it as $\beta = X^+ + \zeta$ for some $\zeta \in \mathbb{R}^d$. Since

$$X\beta = X(X^+\vec{y} + \zeta) = \vec{y} + X\zeta, \tag{9.1}$$

β is a global minimizer if and only if $X\zeta = 0$, which happens if and only if $\zeta \perp x_1, ..., x_n$.

□

[1] Since X is full rank, $XX^\top$ is invertible and so we have $X^+ = X^\top(XX^\top)^{-1}$. Note that $XX^+X = X$.

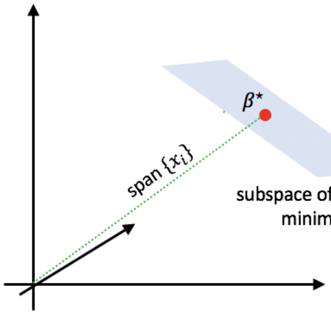


Figure 9.1: Visualization of proof intuition for Theorem 9.3. The solution $\beta^\star$ is the projection of the origin onto the subspace of global minima.

From Lemma 9.1, we can derive an explicit formula for the minimum norm interpolant $\beta^\star \triangleq \operatorname{argmin}_{\beta:\widehat{L}(\beta)=0} \|\beta\|_2$.

Corollary 9.2. $\beta^\star = X^+\vec{y}$.

Proof. Take any β such that $\widehat{L}(\beta) = 0$, and write $\beta = X^+\vec{y} + \zeta$. Then from the definition of X^+ and the fact that $X\zeta = 0$ (see the proof of Lemma 9.1), we have

$$\begin{aligned}
||\beta||_2^2 &= ||X^+\vec{y}||_2^2 + ||\zeta||_2^2 + 2\langle X^+\vec{y}, \zeta\rangle && (9.2)\\
&= ||X^+\vec{y}||_2^2 + ||\zeta||_2^2 + 2\langle X^\top(XX^\top)^{-1}\vec{y}, \zeta\rangle && (9.3)\\
&= ||X^+\vec{y}||_2^2 + ||\zeta||_2^2 + 2\langle (XX^\top)^{-1}y, X\zeta\rangle && (9.4)\\
&= ||X^+\vec{y}||_2^2 + ||\zeta||_2^2 \qquad \text{(because } X\zeta = 0\text{)} && (9.5)\\
&\geq ||X^+\vec{y}||_2^2, && (9.6)
\end{aligned}$$

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 9.3. *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 9.1 for an illustration).

Proof. We first show via induction that $\beta^t \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$ for all t. For the induction base case, note that $\beta^0 = 0 \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. Now suppose $\beta^{t-1} \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. 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 $\eta\nabla\widehat{L}(\beta^{t-1}) = \eta X^\top(X\beta^{t-1} - \vec{y}) \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$, and so $\beta^t = \beta^{t-1} - \eta\nabla\widehat{L}(\beta^{t-1}) \in \operatorname{span}\left\{x^{(1)}, \ldots, x^{(n)}\right\}$. 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 $0 = X\hat{\beta} - \vec{y} = XX^\top v - \vec{y}$. This implies $v = (XX^\top)^{-1}y$, and so $\hat{\beta} = X^\top v = X^\top(XX^\top)^{-1}\vec{y} = X^+\vec{y} = \hat{\beta}^\star$. □

9.2 Implicit regularization of small initialization in nonlinear models

We give another example of implicit regularization effect of small initialization in a non-convex version of the overparametrized linear regression task considered in the previous section. The results in this subsection are largely simplifications of the paper Li et al. [2017] which studies over-parameterized compressed sensing and two-layer neural nets with quadratic activation.

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 $\beta^\star_i = \mathbf{1}\{i \in S\}$ for some $S \subset [d]$ such that $|S| = r$. We again analyze the overparametrized setting, where this time $n \ll d$ but also $n \geq \widetilde{\Omega}(r^2)$.

Our goal is to find a weight vector that minimizes our empirical loss function

$$\widehat{L}(\beta) \triangleq \frac{1}{4n}\sum_{i=1}^{n}\left(y^{(i)} - f_\beta(x^{(i)})\right)^2, \tag{9.7}$$

where $f_\beta(x) \triangleq \langle \beta \odot \beta, x\rangle$. The operation $\odot$ denotes the Hadamard product: for $u, v \in \mathbb{R}^d$, $u \odot v \in \mathbb{R}^d$ is defined by $(u \odot v)_i \triangleq u_i v_i$ for $i = 1, \ldots, d$.

9.2.1 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

$$\sum_{i=1}^{n}\left(\langle \theta, x^{(i)}\rangle - y^{(i)}\right)^2 + \lambda\|\theta\|_1 \tag{9.8}$$

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 9.4. *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*

$$\left\|\beta^\top \odot \beta^\top - \beta^\star \odot \beta^\star\right\|_2^2 \leq O\left(\alpha\sqrt{d}\right). \tag{9.9}$$

(Here, T indexes the gradient descent steps.)

We make several remarks about Theorem 9.4 before presenting the proof.

Remark 9.5. 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 9.6. Theorem 9.4 is a simplified version of Theorem 1.1 in [Li et al., 2018].

Remark 9.7. $\widehat{L}(\beta)$ 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 9.8. $\beta^\star$ is the min-norm solution in this case. That is,

$$\beta^\star = \operatorname{argmin} \|\beta\|_2^2 \qquad \text{s.t. } \widehat{L}(\beta) = 0. \tag{9.10}$$

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 (9.10) becomes

$$\theta^\star = \operatorname{argmin} \|\theta\|_1 \qquad \text{s.t. } \widehat{L}(\theta) = 0, \tag{9.11}$$

which is a lasso regression, whose solution is sparse.

Remark 9.9. 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).

9.2.2 Ground work for proof and the restricted isometry property

In this section we prepare the ground work for the proof of Theorem 9.4.

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

$$\mathbb{E}\left[\langle x, v\rangle^2\right] = \mathbb{E}\left[v^\top x x^\top v\right] = v^\top \mathbb{E}\left[x x^\top\right] v = \|v\|_2^2. \tag{9.12}$$

It follows that

$$\begin{aligned}
L(\beta) &= \frac{1}{4} \mathop{\mathbb{E}}_{x \sim \mathcal{N}(0,I)} \left[(y - \langle \beta \odot \beta, x\rangle^2\right] && (9.13)\\
&= \frac{1}{4} \mathop{\mathbb{E}}_{x \sim \mathcal{N}(0,I)} \left[\langle \beta^\star \odot \beta^\star - \beta \odot \beta, x\rangle^2\right] && \text{(by definition of } y) \quad (9.14)\\
&= \frac{1}{4} \left\|\beta^\star \odot \beta^\star - \beta \odot \beta\right\|_2^2. && \text{(by (9.12))} \quad (9.15)
\end{aligned}$$

Note that (9.15) is the metric that we use to characterize how close β is to the ground-truch parameter $\beta^\star$ (see (9.9)).

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 9.10. *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*

$$(1-\delta)\|v\|_2^2 \leq \frac{1}{n}\sum_{i=1}^n \langle v, x^{(i)}\rangle^2 \leq (1+\delta)\|v\|_2^2. \tag{9.16}$$

Lemma 9.10 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 (9.16) holds.

By algebraic manipulation, (9.16) is equivalent to

$$(1-\delta)\|v\|_2^2 \le v^\top \left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) v \le (1+\delta)\|v\|_2^2. \tag{9.17}$$

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 9.4.

Lemma 9.11. *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*

$$\left|\frac{1}{n}\sum_{i=1}^n \langle x^{(i)}, v\rangle\langle x^{(i)}, w\rangle - \langle v, w\rangle\right| = \left|v^T\left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) w - \langle v, w\rangle\right| \tag{9.18}$$

$$\le 4\delta \left\|v\right\|_2 \cdot \left\|w\right\|_2. \tag{9.19}$$

Corollary 9.12. Taking $w = e_1, \dots, e_d$ in Lemma 9.11, we can conclude that

$$\left\|\frac{1}{n}\sum_{i=1}^n \langle x^{(i)}, v\rangle x^{(i)} - v\right\|_\infty = \left\|\left(\frac{1}{n}\sum_{i=1}^n x^{(i)}(x^{(i)})^\top\right) v - v\right\|_\infty \tag{9.20}$$

$$\le 4\delta \left\|v\right\|_2. \tag{9.21}$$

9.2.3 Warm-up for analysis: Gradient descent on population loss

The main intuition for proving Theorem 9.4 is to leverage the uniform convergence when β belongs to the set S_r (see Figure 9.2). 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 9.13. *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

$$\nabla L(\beta) = (\beta \odot \beta - \beta^\star \odot \beta^\star) \odot \beta, \tag{9.22}$$

the gradient descent step is

$$\beta^{t+1} = \beta^t - \eta(\beta^t \odot \beta^t - \beta^\star \odot \beta^\star) \odot \beta^t. \tag{9.23}$$

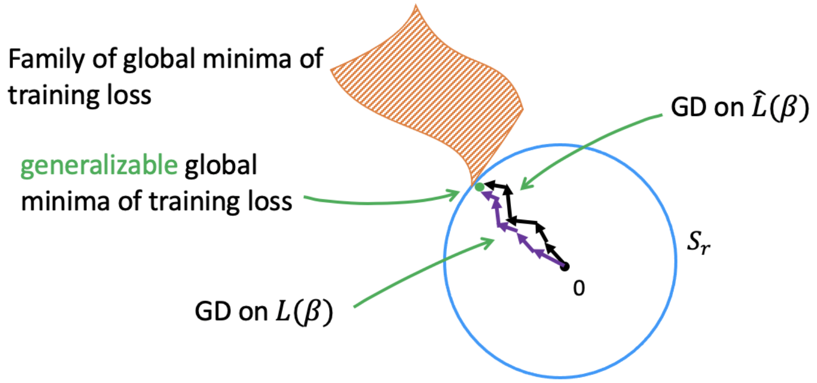

Family of global minima of training loss
GD on $\hat{L}(\beta)$
generalizable global minima of training loss
S_r
0
GD on $L(\beta)$

Figure 9.2: Visualization of proof intuition for Theorem 9.4.

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

$$\begin{aligned}\beta_i^{t+1} &= \beta_i^t - \eta(\beta_i^t \cdot \beta_i^t - 1 \cdot 1) \cdot \beta_i^t && (9.24)\\ &= \beta_i^t - \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t. && (9.25)\end{aligned}$$

Consider the following two cases:

- If $\beta_i^t \le 1/2$, we have

$$\begin{aligned}\beta_i^{t+1} &= \beta_i^t\left[1 + \eta\left(1 - \left(\beta_i^t\right)^2\right)\right] && (9.26)\\ &\ge \beta_i^t\left(1 + \frac{3}{4}\eta\right). && (9.27)\end{aligned}$$

  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

$$\begin{aligned}1 - \beta_i^{t+1} &= 1 - \beta_i^t + \eta\left[\left(\beta_i^t\right)^2 - 1\right]\beta_i^t && (9.28)\\ &= 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\left(1 + \beta_i^t\right)\beta_i^t && (9.29)\\ &\le 1 - \beta_i^t - \eta\left(1 - \beta_i^t\right)\beta_i^t && \text{(because } 1 + \beta_i^t \ge 1\text{)} \quad (9.30)\\ &= \left(1 - \beta_i^t\right)\left(1 - \eta\beta_i^t\right) && (9.31)\\ &\le \left(1 - \beta_i^t\right)\left(1 - \eta/2\right). && \text{(because } \beta_i^t \ge 1/2\text{)} \quad (9.32)\end{aligned}$$

  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

$$\beta_i^{t+1} = \left[\beta^t - \eta(\beta^t \odot \beta^t - \beta^\star \odot \beta^\star) \odot \beta^t\right]_i \tag{9.33}$$
$$= \beta_i^t - \eta(\beta_i^t \cdot \beta_i^t) \cdot \beta_i^t \qquad \text{(since } \beta_i^\star = 0\ \forall i \notin S\text{)} \tag{9.34}$$
$$= \beta_i^t - \eta(\beta_i^t)^3. \tag{9.35}$$

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 $s \le t$,

$$\beta_i^{s+1} = (1 - \eta(\beta_i^s)^2)\beta_i^s \tag{9.36}$$
$$\le (1 + \eta(\beta_i^s)^2)\beta_i^s \tag{9.37}$$
$$\le (1 + 4\eta\alpha^2)\beta_i^s. \qquad \text{(since } \beta_i^s \le 2\alpha\text{)} \tag{9.38}$$

With strong induction, we can repeatedly apply this gradient update starting from $t = 0$ to obtain

$$\beta_i^{t+1} \le \beta_0 \cdot (1 + 4\eta\alpha^2)^t \tag{9.39}$$
$$\le \beta_0 (1 + 4\eta\alpha^2)^{\frac{1}{10\eta\alpha^2}} \tag{9.40}$$
$$\le \beta_0 \exp\left(\frac{4\eta\alpha^2}{10\eta\alpha^2}\right) \tag{9.41}$$
$$= \beta_0 \cdot e^{2/5} \tag{9.42}$$
$$\le 2\alpha, \tag{9.43}$$

which completes the inductive proof of the claim.

□

9.2.4 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 9.14. *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*

$$\left\|\beta^t \odot \beta^t - \beta^\star \odot \beta^\star\right\|_2^2 \le \widetilde{O}\left(\frac{1}{\sqrt{n}}\right). \tag{9.44}$$

Remark 9.15. Note that Theorem 9.14 is a slightly weaker version of Theorem 9.4 for $r = 1$, since the bound on the RHS depends on the number of examples and not the initialization α. In Theorem 9.4, we could take α as small as we like to drive the bound to zero; we cannot do this for Theorem 9.14.

We proceed to prove Theorem 9.14 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

$$\beta^t = r_t \cdot e_1 + \zeta_t, \tag{9.45}$$

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 $t \le \widetilde{O}\big(1/\eta\big)$. 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 $\beta^t = r_t e_1 + \zeta_t = r_t\beta^\star + \zeta_t$,

$$\begin{aligned}
\nabla\widehat{L}(\beta^t) &= \frac{1}{n}\sum_{i=1}^{n}(\langle \beta^t \odot \beta^t, x^{(i)}\rangle - y^{(i)})x^{(i)} \odot \beta^t && (9.46)\\
&= \frac{1}{n}\sum_{i=1}^{n}(\langle \beta^t \odot \beta^t - \beta^\star \odot \beta^\star, x^{(i)}\rangle)x^{(i)} \odot \beta^t && (9.47)\\
&= \frac{1}{n}\sum_{i=1}^{n}\langle r_t^2\beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t - \beta^\star \odot \beta^\star, x^{(i)}\rangle x^{(i)} \odot \beta^t && (9.48)\\
&= \underbrace{\frac{1}{n}\sum_{i=1}^{n}\Big\langle \big(r_t^2 - 1\big)\beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t, x^{(i)}\Big\rangle x^{(i)}}_{m_t} \odot \beta^t. && (9.49)
\end{aligned}$$

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,

$$\begin{aligned}
m_t &= \frac{1}{n}\sum_{i=1}^{n}\Big\langle \big(r_t^2 - 1\big)\beta^\star \odot \beta^\star + \zeta_t \odot \zeta_t,\ x^{(i)}\Big\rangle\, x^{(i)} && (9.50)\\
&= \left(\frac{1}{n}\sum_{i=1}^{n} x^{(i)}\big(x^{(i)}\big)^\top\right)\big(r_t^2 - 1\big)\cdot(\beta^\star \odot \beta^\star) + \left(\frac{1}{n}\sum_{i=1}^{n} x^{(i)}(x^{(i)})^\top\right)(\zeta_t \odot \zeta_t) && (9.51)\\
&= \underbrace{X\big(r_t^2 - 1\big)\cdot\big(\beta^\star \odot \beta^\star\big)}_{\text{part of } u_t} + \underbrace{X\big(\zeta_t \odot \zeta_t\big)}_{v_t}. && (9.52)
\end{aligned}$$

Now, define $u_t \triangleq (r_t^2 - 1)(\beta^\star \odot \beta^\star) - X(r_t^2 - 1)(\beta_* \odot \beta_*)$ and $v_t \triangleq X\big(\beta_t \odot \beta_t\big)$. Then we can rewrite the gradient as

$$\nabla\widehat{L}(\beta^t) = m_t \odot \beta^t = [(r_t^2 - 1)\beta^\star \odot \beta^\star - u_t + v_t] \odot \beta_t. \tag{9.53}$$

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 9.12, we can bound $\|u_t\|_\infty$ as

$$\|u_t\|_\infty \le 4\delta\,\big\|(r_t^2 - 1)\beta^\star \odot \beta^\star\big\|_2 \le 4\delta||\beta^\star \odot \beta^\star||_2 \le 4\delta. \tag{9.54}$$

(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 9.12 implies $\|v_t - \zeta_t \odot \zeta_t\|_\infty \le 4\delta \|\zeta_t \odot \zeta_t\|_2$,

$$\begin{aligned}
\|v_t\|_\infty &\le \|\zeta_t \odot \zeta_t\|_\infty + 4\delta \|\zeta_t \odot \zeta_t\|_2 && \text{(by the triangle inequality)} && (9.55)\\
&\le \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t \odot \zeta_t\|_1 && \text{(since } \zeta_t \text{ very small)} && (9.56)\\
&= \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t\|_2^2 . && && (9.57)
\end{aligned}$$

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 9.16. *For all $t \le 1/(c\eta\delta)$ with sufficiently large constant c, we have*

$$\|\zeta_t\|_\infty \le 2\alpha, \qquad \|\zeta_t\|_2^2 \le \frac{1}{2}, \qquad \textit{and} \quad \|\zeta_{t+1}\|_\infty \le \big(1 + O(\eta\delta)\big) \|\zeta_t\|_\infty . \tag{9.58}$$

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 (9.53) 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

$$\begin{aligned}
\beta^{t+1} &= \beta^t - \eta \left[\left(r_t^2 - 1\right) \beta^\star \odot \beta^\star - u_t + v_t\right] \odot \beta^t && (9.59)\\
&= \underbrace{\beta^t - \eta \left(r_t^2 - 1\right) \beta^\star \odot \beta^\star \odot \beta^t}_{\text{GD update for population loss}} - \eta \left(-u_t + v_t\right) \odot \beta^t. && (9.60)
\end{aligned}$$

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,

$$\begin{aligned}
\zeta_{t+1} &= \left(I - e_1 e_1^\top\right) \beta^{t+1} && && (9.61)\\
&= \left(I - e_1 e_1^\top\right) \cdot \beta^t - \eta \left(I - e_1 e_1^\top\right) (v_t - u_t) \odot \beta^t && \text{(by (9.60), second term} = 0) && (9.62)\\
&= \zeta_t - \eta \left[\left(I - e_1 e_1^T\right)(v_t - u_t) \odot \left(I - e_1 e_1^T\right) \beta^t\right] && \text{(by distribution law for } \odot) && (9.63)\\
&= \zeta_t - \eta \underbrace{\left[\left(I - e_1 e_1^T\right)(v_t - u_t)\right]}_{\rho_t} \odot \zeta_t. && && (9.64)
\end{aligned}$$

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

$$\|\zeta_{t+1}\|_\infty \le (1 + \eta \|\rho_t\|_\infty) \|\zeta_t\|_\infty . \tag{9.65}$$

Now, we will prove the lemma by using strong induction on t. Suppose that the first two pieces of (9.58) hold for all iterations up to t. We can show that

$$\begin{aligned}
\|\rho_t\|_\infty &\le \|u_t\|_\infty + \|v_t\|_\infty && && (9.66)\\
&\le 4\delta + \|\zeta_t\|_\infty^2 + 4\delta \|\zeta_t\|_2^2 && \text{(by (9.54) and (9.57))} && (9.67)\\
&\le 4\delta + (2\alpha)^2 + 4\delta \cdot \frac{1}{2} && \text{(by the inductive hypothesis)} && (9.68)\\
&\le 8\delta, && && (9.69)
\end{aligned}$$

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 (9.65), we have

$$\|\zeta_{t+1}\|_\infty \le (1 + 8\eta\delta) \|\zeta_t\|_\infty = \big(1 + O(\eta\delta)\big) \|\zeta_t\|_\infty , \tag{9.70}$$

which proves the third piece of the lemma. Using this piece, we can show that

$$\|\zeta_{t+1}\|_\infty \leq (1+8\eta\delta)^{t+1} \|\zeta_0\|_\infty \leq \big(1+8\eta\delta\big)^{1/(c\eta\delta)} \cdot \alpha \leq 2\alpha \tag{9.71}$$

for a sufficiently large constant c, which proves the second piece. Finally, we show that

$$\|\zeta_{t+1}\|_2^2 \leq \big(1+8\eta\delta\big)^{t+1} \|\zeta_0\|_2^2 \leq \big(1+8\eta\delta\big)^{1/(c\eta\delta)} \cdot \alpha\sqrt{d} \leq \frac{1}{2}, \tag{9.72}$$

if $\alpha \leq \frac{1}{n^{O(1)}}$, 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 9.17. *For all $t \leq 1/(c\eta\delta)$ 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 (9.53) is

$$\beta^{t+1} = \beta^t - \eta\big[\big(r_t^2-1\big)\beta^\star \odot \beta^\star - u_t + v_t\big] \odot \beta_t. \tag{9.73}$$

We have that

$$\begin{aligned} r_{t+1} &= \big\langle \beta^{t+1}, e_1\big\rangle && (9.74)\\ &= \big\langle \beta^t, e_1\big\rangle - \eta\big(r_t^2-1\big)\big\langle \beta^t, e_1\big\rangle - \eta\big\langle v_t - u_t, e_1\big\rangle\big\langle \beta^t, e_1\big\rangle && (9.75)\\ &= r_t - \eta\big(r_t^2-1\big)r_t - \eta\langle v_t-u_t, e_1\rangle r_t && (9.76)\\ &= \Big(1+\eta\big(1-r_t^2\big)\Big)r_t + \eta\big\langle u_t - v_t, e_1\big\rangle r_t && (9.77)\end{aligned}$$

so all we need to do is bound the second term as follows:

$$\begin{aligned} |\eta\langle v_t-u_t, e_1\rangle r_t| &\leq \eta \cdot r_t \|v_t - u_t\|_\infty && (9.78)\\ &\leq \eta \cdot r_t \cdot 8\delta \qquad\qquad \text{(by (9.69))} && (9.79)\\ &= O(\eta\delta)\cdot r_t. && (9.80)\end{aligned}$$

□

Putting it all together Finally, we return to the proof of Theorem 9.14. By Lemma 9.17, we know that as long as $r_t \leq 1/2$ it will grow exponentially fast, since

$$r_{t+1} \geq \Big(1+\eta\big(1-r_t^2\big) - O(\eta\delta)\Big)\cdot r_t \geq \Big(1+\frac{\eta}{2}\Big)\cdot r_t. \tag{9.81}$$

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

$$1 - r_{t+1} \leq 1 - r_t - \eta\big(1 - r_t^2\big)r_t + O(\eta\delta) \cdot r_t \tag{9.82}$$

$$\leq 1 - r_t - \frac{\eta\big(1 - r_t^2\big)}{2} + O(\eta\delta) \tag{9.83}$$

$$\leq 1 - r_t - \frac{\eta\big(1 - r_t\big)}{2} + O(\eta\delta) \tag{9.84}$$

$$= \left(1 - \frac{\eta}{2}\right)(1 - r_t) + O(\eta\delta). \tag{9.85}$$

  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

$$r_{t+1} - 1 \leq (1 - \eta)(r_t - 1) + O(\eta\delta) \leq O(\delta), \tag{9.86}$$

  implying that r_t remains very close to 1 after the same order of iterations.

This completes the proof of Theorem 9.14, bounding the number of iterations needed for gradient descent on the empirical loss to converge to β^*. □

9.3 From small to large initialization: a precise characterization

We have previously discussed how certain initializations of gradient descent converge to minimum-norm solutions. In the sequel, we characterize the effect of initialization more precisely—we will show that in a variant of the settings in Section 9.2, we can precisely compute the corresponding regularizer induced by any initialization. We will see that when the initialization is small, we obtain the bias towards minimum norm solution (in the parameter space used in optimization), whereas when the initialization is large, we are in the NTK regime (Section 8.4) where the implicit bias is towards the min norm solution under the NTK kernel. The materials in this subsection are simplifications of results in the recent paper Woodworth et al. [2020].

9.3.1 Preparation: gradient flow

To simplify the analysis, we will consider the concept of gradient flow, i.e. gradient descent with an infinitesimal learning rate. This allows us omit the second order effect from the learning rate and simplify the analysis.

We begin by recalling the gradient descent update formula. In our previous description of gradient descent, we indexed the updated parameters by $t = 1, 2, \dots$. Anticipating our generalization to infinitesimal steps, we will index the updated parameters using parentheses instead of subscripts. In particular, the standard gradient descent update given a loss function $L(w)$ is

$$w(t+1) = w(t) - \eta\nabla L(w(t)). \tag{9.87}$$

If we scale the time by η so that each update by gradient descent corresponds to a time step of size η (rather than size 1), the update becomes

$$w(t+\eta) = w(t) - \eta\nabla L(w(t)). \tag{9.88}$$

Taking $\eta \to 0$ yields a differential equation, which can be thought of as a continuous process rather than discrete updates:

$$w(t+dt) = w(t) - dt \cdot \nabla L(w(t)). \tag{9.89}$$

This can also be written as:

$$\dot{w}(t) = -\nabla L(w(t) \quad \text{with} \quad \dot{w}(t) = \frac{\partial w(t)}{\partial t} \tag{9.90}$$

This allows us to ignore the η^2 term (alternatively the (dt^2) term), which will simplify some of the technical details that follow.

9.3.2 Characterizing the implicit bias of initialization

The results in this section are slight simplification of the recent paper by Woodworth et al. [2020]. The model is a variant of the one we introduced in (9.7). Recalling that $x^{\odot 2} = x \odot x$, let

$$f_w(x) = \left(w_+^{\odot 2} - w_-^{\odot 2}\right)^\top x. \tag{9.91}$$

where $w_+, w_- \in \mathbb{R}^d$. Let w denote the concatenation of the two parameter vectors, i.e. $= (w_+, w_-)$. In (9.7), we defined $f_\beta(x) = (\beta \odot \beta)^\top x$; this model can only represent positive linear combinations of x. By contrast, $f_w(x)$ can represent any linear model. Moreover, if we choose our initialization for w such that $w_+(0) = w_-(0)$, we obtain $f_{w(0)}(x) \equiv 0$ for all x. Similar to our analysis of the NTK, this initialization will simplify the subsequent derivations.

Next, we define the following loss function,

$$\widehat{L}(w) = \frac{1}{2} \sum_{i=1}^{n} \left(y^{(i)} - f_w(x^{(i)})\right)^2, \tag{9.92}$$

and consider the initialization

$$w_+(0) = w_-(0) = \alpha \cdot \vec{\mathbf{1}} \tag{9.93}$$

where $\vec{\mathbf{1}}$ denotes the all-ones vector. The analysis technique still applies to any general initializations as long as all the dimension are initialized to be non-zero, but the the initialization scale is the most important factor, and therefore we chose this simplification for the ease of exposition.

Note that every $w = (w_+, w_-)$ corresponds to a de facto linear function of x. We denote the resulting linear model as θ_w:

$$\theta_w = w_+^{\odot 2} - w_-^{\odot 2}. \tag{9.94}$$

Note that $\theta_w^\top x = f_w(x)$.

Let $w(\infty)$ denote the limit of the gradient flow, i.e.

$$w(\infty) = \lim_{t \to \infty} w(t). \tag{9.95}$$

Then, the converged linear model in the θ space is defined by $\theta_\alpha(\infty) = \theta_{w(\infty)}$—we are interested in understanding its properties. For simplicity, we will omit the ∞ index and refer to this quantity as θ_α. We assume throughout that the limit exists and all corresponding regularity conditions are met.

Let

$$X = \begin{bmatrix} {x^{(1)}}^\top \\ \vdots \\ {x^{(n)}}^\top \end{bmatrix} \in \mathbb{R}^{n \times d} \quad \text{and} \quad \vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}. \tag{9.96}$$

In the sequel, we formally state our result relating the complexity of the solution discovered by gradient flow to the size of the initialization.

Theorem 9.18 (Theorem 1 in Woodworth et al. [2020])**.** *For any $0 < \alpha < \infty$, assume that gradient flow with initialization $w_+(0) = w_-(0) = \alpha \cdot \vec{\mathbf{1}}$ converges to a solution that fits the data exactly: $X\theta_\alpha = \vec{y}$.*[3] *Then, the solution satisfies the following notion of minimum complexity:*

$$\theta_\alpha = \operatorname*{argmin}_\theta Q_\alpha(\theta) \tag{9.97}$$

$$\text{s.t.} \quad X\theta = \vec{y} \tag{9.98}$$

where

$$Q_\alpha(\theta) = \alpha^2 \cdot \sum_{i=1}^{n} q\left(\frac{\theta_i}{\alpha^2}\right) \tag{9.99}$$

and

$$q(z) = 2 - \sqrt{4 + z^2} + z \cdot \operatorname{arcsinh}\left(\frac{z}{2}\right) \tag{9.100}$$

In words, Theorem 9.18 claims that θ_α is the minimum complexity solution for the complexity measure Q_α.

Remark 9.19. In particular, when $\alpha \to \infty$ we have that

$$q(\theta_i/\alpha^2) \asymp \theta_i^2/\alpha^4 \tag{9.101}$$

and so

$$Q_\alpha(\theta) \asymp \frac{1}{\alpha^2} \|\theta\|_2^2 . \tag{9.102}$$

This means that if $\alpha \to \infty$ than the complexity measure Q_α is the ℓ_2-norm, $||\theta||_2$. If $\alpha \to 0$, then the complexity measure becomes

$$q\left(\frac{\theta_i}{\alpha^2}\right) \asymp \frac{|\theta_i|}{\alpha^2} \log\left(\frac{1}{\alpha^2}\right) \quad \text{(by Taylor expansion)} \tag{9.103}$$

and so,

$$Q_\alpha(\theta) \asymp \frac{\|\theta\|_1}{\alpha^2} \log\left(\frac{1}{\alpha^2}\right) \tag{9.104}$$

To summarize, for $\alpha \to \infty$, the constrained minimization problem we solve in (9.98) yields the minimum ℓ_2-norm solution of θ (i.e. the ℓ_4-norm for w). When $\alpha \to 0$, solving (9.98) yields the minimum ℓ_1-norm θ (which is the ℓ_2-norm for w). For $0 < \alpha < \infty$, we obtain some interpolation of ℓ_1 and ℓ_2 regularization of the optimum.

Remark 9.20. Note that when $\alpha \to 0$, the intuition is similar to what we had observed in previous analyses; in particular, the solution is the global minimum closest to the initialization. Note however, that when $\alpha \neq 0$, the solution discovered by gradient descent will not *exactly* correspond to the solution closest to the initialization.

Remark 9.21. When $\alpha \to \infty$, we claim that the model optimization is in the neural tangent kernel (NTK) regime. Recall that we had two parameters, (σ, β), that determined if we could treat the optimization problem as a kernel regression. Further recall that σ denotes the minimum singular value of Φ and β is the Lipschitzness of the gradient. Let us now compute σ and β for large α initializations of our model.

[3] This assumption can likely be proved to be true and thus not required. Here we still include the condition because the original paper Woodworth et al. [2020] assumed it.

For $w_-(0) = w_+(0) = \alpha\vec{\mathbf{1}}$,

$$\nabla f_{w(0)}(x) = 2 \begin{bmatrix} w_+(0) \cdot x \\ -w_-(0) \odot x \end{bmatrix} = 2\alpha \begin{bmatrix} x \\ -x \end{bmatrix} \tag{9.105}$$

by the chain rule. It is clear then that both σ and β linearly depend on α. This implies that

$$\frac{\beta}{\sigma^2} \to 0 \quad \text{as } \alpha \to \infty \tag{9.106}$$

since the denominator is $O(\alpha^2)$, while the numerator is $O(\alpha)$. In particular, the features used in this kernel method are:

$$\phi(x) = \nabla f_{w(0)}(x) = 2\alpha \begin{bmatrix} x \\ -x \end{bmatrix} \tag{9.107}$$

The neural tangent kernel perspective then gives an alternative proof of this complexity minimization result for $\alpha \to \infty$. In the NTK regime, the solution (to our convex problem) is always the minimum ℓ_2-norm solution for the feature matrix, which in this case equals $\begin{bmatrix} X \\ -X \end{bmatrix}$.

Note that practice tends not to follow the assumptions made here. Often, people either do not use large initializations or do not use infinitesimally small step sizes. But this is a good thing because we do not want to be in the NTK regime; being in the NTK regime implies that we are doing no different or better than just using a kernel method.

We can now prove Theorem 9.18, which is similar to the overparametrized linear regression proof of Theorem 9.3.

This proof follows in two steps:

1. We find an invariance maintained by the optimizer. In the overparametrized linear regression proof of Theorem 9.3, we required $\theta \in \text{span}\{x^{(i)}\}$. For this proof, we will use a slightly more complicated invariance.

2. We characterize the solution using this invariance. The invariance, which depends on α, will tell us which zero error solution the optimization converges to.

Note also that all of these conditions only depend upon the empirically observed samples. The invariance and minimum is not defined with respect to any population quantities.

Proof. Let

$$\tilde{X} = \begin{bmatrix} X & -X \end{bmatrix} \in \mathbb{R}^{n \times 2d} \quad \text{and} \quad w(t) = \begin{bmatrix} w_+(t) \\ w_-(t) \end{bmatrix} \in \mathbb{R}^{2d}. \tag{9.108}$$

Then, the model output on n data points can be described in matrix notation as follows:

$$\tilde{X}w(t)^{\odot 2} = \begin{bmatrix} X & -X \end{bmatrix} \begin{bmatrix} w_+(t)^{\odot 2} \\ w_-(t)^{\odot 2} \end{bmatrix} = \begin{bmatrix} f_{w(t)}(x^{(1)}) \\ \vdots \\ f_{w(t)}(x^{(n)}) \end{bmatrix} \in \mathbb{R}^n. \tag{9.109}$$

Given the loss function,

$$L(w(t)) = \frac{1}{2} \left\| \tilde{X}w(t)^{\odot 2} - \vec{y} \right\|_2^2, \tag{9.110}$$

the gradient of $w(t)$ can be computed as

$$\begin{aligned}\dot{w}(t) &= -\nabla L(w(t)) && (9.111)\\ &= -\nabla\left(\left\|\tilde{X}w(t)^{\odot 2} - \vec{y}\right\|_2^2\right) && (9.112)\\ &= \left(\tilde{X}^\top r(t)\right) \odot w(t) \qquad \text{(chain rule)} && (9.113)\end{aligned}$$

where $r(t) = \tilde{X}w(t)^{\odot 2} - \vec{y}$ denotes the residual vector. We see that the $\tilde{X}^\top r(t)$ term in (9.113) is reminiscent of linear regression for which it would correspond to the gradient, although the $\odot w(t)$ reminds us that this problem is indeed quadratic.

We cannot directly solve this differential equation, but we claim that

$$w(t) = w(0) \odot \exp\left(-2\tilde{X}^\top \int_0^\top r(s)ds\right) \quad \text{(exp is applied entry-wise)} \tag{9.114}$$

which is not quite a closed form solution of equation 9.113 since $r(s)$ is still a function of $w(t)$. To understand how we obtained this “solution,” we consider a more abstract setting. Suppose that

$$\dot{u}(t) = v(t)\dot{u}(t) \tag{9.115}$$

We can then “solve” this differential equation as follows. Rearranging, we observe that

$$\begin{aligned}\frac{\dot{u}(t)}{u(t)} &= v(t) && (9.116)\\ \frac{d \log u(t)}{dt} &= v(t) \quad \text{(chain rule)} && (9.117)\\ \log u(t) - \log u(0) &= \int_0^t v(s)ds \quad \text{(integration)} && (9.118)\\ \frac{u(t)}{u(0)} &= \exp\left(\int_0^t v(s)ds\right) && (9.119)\end{aligned}$$

In our problem, $u \leftrightarrow w_i$ and $v \leftrightarrow (\tilde{X}^\top r(t))_i$.

We have characterized w, but we want to transform this to a characterization that involves θ. Recall that $w_+(0) = \alpha\vec{\mathbf{1}}$ and $w_-(0) = \alpha\vec{\mathbf{1}}$ so that $w(0) = \alpha\vec{\mathbf{1}} \in \mathbb{R}^{2d}$. Additionally, we have that $\theta(t) = w_+(t)^{\odot 2} - w_-(t)^{\odot 2}$. We can now apply (9.114) to expand $w(t)$ and simplify.

Note that if we have $\tilde{X}^\top = \begin{bmatrix} X^\top \\ -X^\top \end{bmatrix} \in \mathbb{R}^{2n \times d}$, then for some vector v,

$$\begin{aligned}\left(\exp(-2\tilde{x}^\top v)\right)^{\odot 2} &= \begin{bmatrix}\exp(-2X^\top v)\\ \exp(2X^\top v)\end{bmatrix}^{\odot 2} && (9.120)\\ &= \begin{bmatrix}\exp(-4X^\top v)\\ \exp(4X^\top v)\end{bmatrix}. && (9.121)\end{aligned}$$

Applying this result for $v = \int_0^T r(s)ds$, we obtain that:

$$\begin{aligned}\theta(t) &= w_+(t)^{\odot 2} - w_-(t)^{\odot 2} && (9.122)\\ &= \alpha^2\left[\exp\left(-4X^\top \int_0^t r(s)ds\right) - \exp\left(4X^\top \int_0^t r(s)ds\right)\right] && (9.123)\\ &= 2\alpha^2 \sinh\left(-4X^\top \int_0^t r(s)ds\right). && (9.124)\end{aligned}$$

Letting $t \to \infty$, we have that

$$\theta_\alpha = 2\alpha^2 \sinh\left(-4X^\top \int_0^\infty r(s)ds\right). \tag{9.125}$$

Lastly, we also know

$$X\theta_\alpha = \vec{y} \tag{9.126}$$

since this is the assumption by the theorem (which should can be proven because the optimization should converge to a zero-error solution). We next show that (9.125) and (9.126) are also sufficient conditions for a solution to the constrained optimization problem given by (9.98). In particular, (9.125) and (9.126) correspond to the Karush-Kuhn-Tucker (or KKT) conditions of (9.98).

A KKT condition is an optimality condition for constrained optimization problems. While these conditions can have a variety of formulations and typically one can invoke some off-the-shelf theorems to use them, we can motivate the conditions we encountered by considering the following general optimization program:

$$\text{argmin} \quad Q(\theta) \tag{9.127}$$

$$\text{s.t.} \quad X\theta = \vec{y}. \tag{9.128}$$

We say that θ satisfies the (first order) KKT conditions if

$$\nabla Q(\theta) = X^\top \nu \text{ for some } \nu \in \mathbb{R}^n \tag{9.129}$$

$$X\theta = \vec{y} \tag{9.130}$$

More intuitively, we know that optimality implies that there are no first order local improvements that satisfy the constraint (up to first order). Then, consider a perturbation $\Delta\theta$. In order to satisfy the constraint, we must enforce the following:

$$\Delta\theta \perp \text{row-span}\{X\} \quad \text{so} \quad X\Delta\theta = 0 \tag{9.131}$$

So, if we look at $\theta + \Delta\theta$ satisfying the constraint, we can use a Taylor expansion to show that

$$Q(\theta + \Delta\theta) = Q(\theta) + \langle \Delta\theta, \nabla Q(\theta) \rangle \leq Q(\theta) \tag{9.132}$$

because if $\langle \Delta\theta, \nabla Q(\theta) \rangle$ is positive it violates the optimality assumption. In fact, it is very easy to make the sign flip for $\langle \Delta\theta, \nabla Q(\theta) \rangle$ because you can flip $\Delta\theta$ to be the opposite direction. This means that

$$\forall\, \Delta\theta \perp \text{row-span}\{X\}, \quad \langle \Delta\theta, \nabla Q(\theta) \rangle = 0 \tag{9.133}$$

because if it is negative, you can equivalently flip it to be positive which violates optimality. This means that $Q(\theta) \subseteq \text{row-span}\{X\}$, or $Q(\theta) = X^\top \nu$ for some ν.

Returning to our problem, the KKT condition gives

$$\nabla Q(\theta) = X^\top \nu \tag{9.134}$$

and the invariance gives us

$$\theta_\alpha = 2\alpha^2 \sinh\left(-4X^\top \int_0^\infty r(s)ds\right) \tag{9.135}$$

$$= 2\alpha^2 \sinh\left(-4X^\top v'\right) \tag{9.136}$$

where we let $v' = \int_0^\infty r(s)ds$ for simplicity. Taking the gradient of Q gives

$$\nabla Q_\alpha(\theta) = \text{arcsinh}\left(\frac{1}{2\alpha^2}\theta\right) \tag{9.137}$$

Plugging in θ_α, we get

$$\nabla Q(\theta_\alpha) = \text{arcsinh}\left(\frac{1}{2\alpha^2}\theta_\alpha\right) = -4X^\top v' \tag{9.138}$$

Thus, θ_α satisfies both KKT conditions. Even further, since our optimization problem (9.98) is convex (we do not formally argue this), we conclude that θ_α is a global minimum. □

9.4 Implicit regularization towards max-margin solutions in classification

We now switch our focus to classification problems. We consider linear models (though these results also apply to nonlinear models with a weaker version of the conclusion). We assume that our data is separable and will prove that gradient descent converges to the max-margin solution. This result holds for any initialization and does not require any additional regularization; we only require the use of gradient descent and the standard logistic loss function. The results in this subsection are originally given by Soudry et al. [2018], and our exposition heavily depends on those in [Ji and Telgarsky, 2018, Telgarsky, 2021].

Assume we have data $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$, where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{\pm 1\}$. We consider the linear model $h_w(x) = w^\top x$ and the cross entropy loss function $\widehat{L}(w) = \sum_{i=1}^n \ell\left(y^{(i)}, h_w\left(x^{(i)}\right)\right)$, where $\ell(t) = \log(1 + \exp(-t))$ is the logistic loss.

As we have separable data, there can be multiple global minima, as you can trivially take an infinite number of separators. More formally, there are an infinite number of unit vectors $\bar{w}$ such that $\bar{w}^\top x^{(i)} y^{(i)} > 0$ for all i as one can perturb any strict separator while still maintaining a separation of classes. Then, we can scale the separator to make the loss arbitrarily small—we have that $\widehat{L}(\alpha\bar{w}) \to 0$ as $\alpha \to \infty$. Thus, informally, for any unit vector $\bar{w}$ that separate the data, $\infty \cdot \bar{w}$ is a global minimum.

We would like to understand which global minimum gradient descent converges to. We will now show that it finds the max-margin solution. Before we can do so, we recall/introduce the following definitions.

Definition 9.22 (Margin)**.** Let $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ be given data. Assuming $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ is linearly separable, the *margin* is defined as

$$\min_{i\in[n]} y^{(i)} w^\top x^{(i)} \tag{9.139}$$

Definition 9.23 (Normalized Margin)**.** Let $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ be given data. Assuming $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ is linearly separable, the *normalized margin* is defined as

$$\gamma(w) = \frac{\min_{i\in[n]} y^{(i)} w^\top x^{(i)}}{\|w\|_2} \tag{9.140}$$

Definition 9.24 (Max-Margin Solution)**.** Using the normalized margin γ defined in Definition 9.23, we define a *max-margin solution* as

$$\bar{\gamma} = \max_w \gamma(w) \tag{9.141}$$

and let w^* be the unit-norm maximizer. [4]

Using these definitions, we claim the following result.

Theorem 9.25. *Gradient flow converges to the direction of max-margin solution in the sense that*

$$\gamma(w(t)) \to \bar{\gamma} \text{ as } t \to \infty \tag{9.142}$$

where $w(t)$ is the iterate at time t.

The following observations provide some intuition for Theorem 9.25.

1. $\widehat{L}(w(t)) \to 0$ by a standard optimization argument. Namely, if the objective is monotone decreasing at each iteration, $\widehat{L}(w(t)) \approx 0$ for large enough t.
2. Using a Taylor expansion, we can show that $\ell(z) = \log(1 + \exp(-z)) \approx \exp(-z)$ for large z. Thus, logistic loss is close to exponential loss when z is very large.

[4] The normalized margin $\bar{\gamma}$ is scale-invariant. For $c \neq 0$, $\gamma(cw) = \min_{i\in[n]} \frac{y^{(i)} c w^\top x^{(i)}}{\|cw\|_2} = \min_{i\in[n]} \frac{y^{(i)} w^\top x^{(i)}}{\|w\|_2} = \gamma(w)$.

3. Using observation 1, we see that $\|w(t)\|_2 \to \infty$ because if $\|w(t)\|_2$ were instead bounded, then the loss $\widehat{L}(w(t))$ will be bounded below by a constant that is strictly greater than zero, contradicting observation 1. Formally, if $\|w(t)\|_2 \le B$, then

$$|y^{(i)} w^t x^{(i)}| \le B\|x^{(i)}\|, \tag{9.143}$$

and therefore we get

$$\widehat{L}(w(t)) \ge \sum_{i=1}^{n} \exp\left(-B\|x^{(i)}\|_2\right) > 0. \tag{9.144}$$

4. Suppose we have w such that $\|w\|_2 = q$ is very big. Then, using observation 2, we see that

$$\begin{aligned}
\widehat{L}(w) &= \sum_{i=1}^{n} \ell(y^{(i)} w^\top x^{(i)}) && (9.145)\\
&\approx \sum_{i=1}^{n} \exp\left(-y^{(i)} w^\top x^{(i)}\right) && (9.146)\\
\log \widehat{L}(w) &\approx \log \sum_{i=1}^{n} \exp\left(-y^{(i)} w^\top x^{(i)}\right) && (9.147)\\
&= \log \sum_{i=1}^{n} \exp\left(-q y^{(i)} \bar{w}^\top x^{(i)}\right) && (9.148)\\
&\approx \max_{i\in[n]} -q y^{(i)} \bar{w}^\top x^{(i)} && (9.149)
\end{aligned}$$

where $\bar{w} = \frac{w}{\|w\|_2}$ and the last step holds because the log of a sum of exponentials (*log-sum-exp*) is a smooth approximation to the maximum function. To motivate this claim, observe that:

$$\begin{aligned}
\log \sum_{i=1}^{n} \exp(a u_i) &\ge q \max_i u_i && (9.150)\\
\log \sum_{i=1}^{n} \exp(a u_i) &\le \log\left(n \exp(q \max_i u_i)\right) && (9.151)\\
&= \log n + q \max_i u_i && (9.152)\\
&\approx q \max_{i\in[n]} u_i + o(q) \text{ as } q \to \infty && (9.153)
\end{aligned}$$

Thus, minimizing the loss is the same as

$$\min_w \max_{i\in[n]} -q y^{(i)} \bar{w}^\top x^{(i)} \tag{9.154}$$

which can be reformulated as

$$\max_w \min_{i\in[n]} q y^{(i)} \bar{w}^\top x^{(i)} \tag{9.155}$$

The above observations heuristically demonstrate that minimizing the logistic loss with gradient descent is equivalent (in the limit) to maximizing the margin. Below, we prove Theorem 9.25 rigorously for the exponential loss function $\ell(t) = \exp(-t)$, which is nearly the same as the logistic loss.

Proof of Theorem 9.25. We begin by defining the smooth margin as

$$\tilde{\gamma}(w) \triangleq \frac{-\log \hat{L}(w)}{\|w\|_2} \tag{9.156}$$

$$= \frac{-\log\left(\sum_{i=1}^n \exp(-y^{(i)} w^\top x^{(i)})\right)}{\|w\|_2}. \tag{9.157}$$

Note that $\tilde{\gamma}(w)$ approximates $\gamma(w)$ by the log-sum-exp approximation. To make this precise, recall that $\gamma(w) \geq \tilde{\gamma}(w)$ because $y^{(i)} w^\top x^{(i)} \geq \gamma(w)\|w\|_2$ for all i.

Then, since $\gamma(w) \leq \bar{\gamma}$ by definition, it suffices to show that

$$\lim_{t\to\infty} \tilde{\gamma}(w(t)) = \bar{\gamma}. \tag{9.158}$$

Let $\dot{w}(t) = -\nabla \hat{L}(w(t))$. Then,

$$\frac{\partial}{\partial t}\left(-\log \hat{L}(w(t))\right) = \left\langle \nabla\left(-\log \hat{L}(w(t))\right), \dot{w}(t)\right\rangle \tag{9.159}$$

$$= \left\langle -\frac{\nabla \hat{L}(w(t))}{\hat{L}(w(t))}, \dot{w}(t)\right\rangle \tag{9.160}$$

$$= \frac{\|\nabla \hat{L}(w(t))\|_2^2}{\hat{L}(w(t))} \tag{9.161}$$

$$= \frac{\|\dot{w}(t)\|_2^2}{\hat{L}(w(t))} \geq 0 \tag{9.162}$$

This result tells us that the log loss is decreasing at each infinitesimal step of the gradient flow. By integrating (9.162), we can also evaluate the log loss at time T:

$$-\log \hat{L}(w(T)) = -\log \hat{L}(w(0)) + \int_0^T \frac{\partial}{\partial t} \log \hat{L}(w(t))dt \tag{9.163}$$

$$= -\log \hat{L}(w(0)) + \int_0^T \frac{\|\dot{w}(t)\|_2^2}{\hat{L}(w(t))}dt. \tag{9.164}$$

While the derivation above tells us how the numerator of (9.156) is changing, we have yet to relate this to the denominator, i.e. the norm of w. Recall that w^* is the direction of the max-margin solution. Then, we have

$$\|\dot{w}(t)\|_2 \geq \langle \dot{w}(t), w^*\rangle \qquad \text{(Cauchy-Schwarz)} \tag{9.165}$$

$$= \left\langle -\nabla \hat{L}(w(t)), w^*\right\rangle \tag{9.166}$$

$$= \left\langle \sum_{i=1}^n y^{(i)} \exp(-y^{(i)} w^\top x^{(i)}) \cdot x^{(i)}, w^*\right\rangle \tag{9.167}$$

$$= \sum_{i=1}^n y^{(i)} \exp(-y^{(i)} w^\top x^{(i)}) \cdot \left\langle w^*, x^{(i)}\right\rangle \tag{9.168}$$

$$\geq \bar{\gamma} \sum_{i=1}^n \exp(-y^{(i)} w^\top x^{(i)}) \tag{9.169}$$

$$= \bar{\gamma} \cdot \hat{L}(w(t)). \tag{9.170}$$

This shows that $\dot{w}(t)$ is correlated to w^*, and that this correlation depends on $\bar{\gamma}$ and the loss. In addition, $\dot{w}(t)$ is not too small compared to the loss.

Next, we substitute (9.165) into the second term of the right-hand-side of (9.163):

$$\int_0^T \frac{\|\dot{w}(t)\|_2^2}{\hat{L}(w(t))} dt \geq \bar{\gamma} \cdot \int_0^T \|\dot{w}(t)\|_2 dt \tag{9.171}$$

$$\geq \bar{\gamma} \cdot \left\| \int_0^T \dot{w}(t) dt \right\|_2 \tag{9.172}$$

$$= \bar{\gamma} \|w(T)\|_2. \tag{9.173}$$

Applying this bound to the RHS of (9.163), we obtain

$$-\log \hat{L}(w(T)) \geq -\log \hat{L}(w(0)) + \bar{\gamma}\|w(T)\|_2. \tag{9.174}$$

Dividing both sides by $\|w(T)\|_2$,

$$-\frac{\log \hat{L}(w(T))}{\|w(T)\|_2} \geq -\frac{\log \hat{L}(w(0))}{\|w(T)\|_2} + \bar{\gamma}. \tag{9.175}$$

Since $\lim_{T\to\infty} \|w(T)\|_2 = \infty$, we know that the first term on the RHS of (9.175) goes to 0 in the limit. Thus,

$$\lim_{T\to\infty} -\frac{\log \hat{L}(w(T))}{\|w(T)\|_2} \geq \bar{\gamma}. \tag{9.176}$$

Recognizing the LHS as the definition of the smooth margin, i.e. (9.156), we conclude that

$$\lim_{T\to\infty} \tilde{\gamma}(w(T)) \geq \bar{\gamma}. \tag{9.177}$$

Meanwhile, since we know that

$$\bar{\gamma} \geq \gamma(w(T)) \geq \tilde{\gamma}(w(T)), \tag{9.178}$$

we conclude by the squeeze theorem that

$$\lim_{T\to\infty} \gamma(w(T)) = \lim_{T\to\infty} \tilde{\gamma}(w(T)) = \bar{\gamma}. \tag{9.179}$$

□

9.5 Implicit regularization effect of noise in SGD

In the previous section, we discussed implicit regularization via initialization and the implicit regularization of gradient descent for logistic loss-minimizing classifiers. In the sequel, we will move forward to the implicit regularization effect of SGD noise. Starting from the quadratic case, we analyze how the SGD noise will affect the optimization solution, and present (heuristically) a result for non-quadratic loss functions. In particular, the main (heuristic) results are:

1. On the one dimensional quadratic function, the iterate can be disentangled into a contraction part and a stochastic part, the latter of which is characterized by the Ornstein–Uhlenbeck (OU) process. The noise makes the iterate bounce around the global minimum.

2. On the multi-dimensional quadratic function, the iterate can be disentangled into multiple separate 1-D OU processes. The noise makes the iterate bounce around the global minimum, while the fluctuation is closely related to the shape of the noise.

3. On non-quadratic functions, SGD with *label noise* on empirical loss $\hat{L}(\theta)$ converges to a stationary point of the regularized loss $\hat{L}(\theta) + \lambda \mathrm{tr}(\nabla^2 \hat{L}(\theta))$, which is mainly due to the accumulation of a third order effect.

Given the score of the lectures, we will only be able to discuss some of these results informally and heuristically. For example, we refer to the paper Damian et al. [2021] for the a concrete, formal version result for the third bullet.

For the remainder of this section, let $g(x)$ denote the general loss function. Then, the formulation of SGD is: for t in $[0, T]$,

$$\theta_{t+1} = x_t - \eta(\nabla g(x_t) + \xi_t), \tag{9.180}$$

where $\eta > 0$ is the learning rate, ξ_t denotes the SGD noise, and $\mathbb{E}[\xi_t] = 0$. Note that in the most general case, ξ_t can depend on x_t.

9.5.1 Warmup: SGD on the one dimensional quadratic function

In this section, we consider the one dimensional function $g(x) = \frac{1}{2}x^2$. Suppose the noise ξ_t are independent Gaussians, i.e. $\xi_t \sim \mathcal{N}(0, 1)$,

$$\begin{align} x_{t+1} &= x_t - \eta(\nabla g(x_t) + \sigma\xi_t) \tag{9.181}\\ &= x_t - \eta(x_t + \sigma\xi_t) \tag{9.182}\\ &= \underbrace{(1-\eta)x_t}_{\text{contraction}} - \underbrace{\eta\sigma\xi_t}_{\text{stochastic}} . \tag{9.183} \end{align}$$

$(1-\eta)x_t$ is called the contraction because $\eta > 0$, which means that this term will shrink after each iteration. The random noise term $\eta\sigma\xi_t$ will accumulate over time, and the scale of $\eta\sigma\xi_t$ remains unchanged. When x_t is large, the contraction term will dominate. When x_t is small, the noise term will dominate. Without the noise term, as x_t continues its contraction, we approach the global minimum $x = 0$. However, with the presence of the noise $\sigma\xi_t$, x_t will not stay at 0, but instead bounce around it.

To characterize this intuition more precisely, we have

$$\begin{align} x_{t+1} &= (1-\eta)x_t - \eta\sigma\xi_t \tag{9.184}\\ &= (1-\eta)((1-\eta)x_{t-1} - \eta\sigma\xi_{t-1}) - \eta\sigma\xi_t \tag{9.185}\\ &= (1-\eta)^2 x_{t-1} - (1-\eta)\eta\sigma\xi_{t-1} - \eta\sigma\xi_t \tag{9.186}\\ &= (1-\eta)^3 x_{t-2} - (1-\eta)^2\eta\sigma\xi_{t-2} - (1-\eta)\eta\sigma\xi_{t-1} - \eta\sigma\xi_t \tag{9.187}\\ &\vdots \tag{9.188}\\ &= (1-\eta)^{t+1}x_0 - \eta\sigma\sum_{k=0}^{t}\xi_{t-k}(1-\eta)^k. \tag{9.189} \end{align}$$

The first term in (9.189) becomes negligible when $\eta t \gg 1$ (since $(1-\eta)^t \approx e^{-\eta t}$). The second term in (9.189) is the accumulation of noise, which is the sum of Gaussians. Leveraging the properties of Gaussian distributions, we know that its variance equals $\eta^2\sigma^2\sum_{k=0}^{t}(1-\eta)^{2k}$.

From the analysis above, we know that as $t \to \infty$, $\mathrm{Var}(x_t) \approx \eta^2\sigma^2\sum_{k=0}^{\infty}(1-\eta)^{2k} = \frac{\eta^2\sigma^2}{2\eta-\eta^2} = \Theta(\eta\sigma^2)$. Therefore, as $t \to \infty$, $x_t \sim \mathcal{N}(0, \Theta(\eta\sigma^2))$.

Interpretation. In the one dimensional case, the noise only makes it harder to converge to the global minimum. Classical convex optimization tells us: (1) noisy GD leads to a less accurate solution and (2) noisy GD is faster than GD. What we do in practice is achieve a balance between (1) and (2). This does *not* lead to implicit regularization since $\mathbb{E}[x_t] \to 0$ as $t \to \infty$. However, this case is important for further analysis because (9.183) corresponds to the Ornstein–Uhlenbeck (OU) process which we use more extensively in the multi-dimensional cases.

9.5.2 SGD on multi-dimensional quadratic functions

Consider a PSD matrix $A \in \mathbb{R}^{d\times d}$. In this section, $g(x) = \frac{1}{2}x^\top Ax$. Suppose $\xi_t \sim \mathcal{N}(0, \Sigma)$. For ease of presentation, assume that A and Σ are simultaneously diagonizable (they have the same set of eigenvectors). We use K to denote the span of the eigenvectors of A/Σ. Then, consider the following SGD iterate:

$$\begin{aligned} x_{t+1} &= x_t - \eta(\nabla g(x_t) + \xi_t) && (9.190)\\ &= x_t - \eta(Ax_t + \xi_t) && (9.191)\\ &= (I - \eta A)x_t - \eta\xi_t && (9.192)\\ &= \underbrace{(I-\eta A)^{t+1}x_0}_{\text{contraction}} - \underbrace{\eta\sum_{k=0}^{t}(I-\eta A)^k\xi_{t-k}}_{\text{noise accumulation}}. && (9.193)\end{aligned}$$

Similar to the analysis in the 1-D case above, we have $x_t \sim \mathcal{N}(0, \eta^2\sum_{k=0}^{\infty}(I-\eta A)^k\Sigma(I-\eta A)^k)$ as $t \to \infty$. [5]

Since A and Σ are simultaneously diagonizable, we can easily disentangle the iterates into d separate OU process in the eigencoordinate system. Concretely, by eigendecomposition, suppose that $A = U^\top \text{diag}(d_i)U$ and $\Sigma = U^\top\text{diag}(\sigma_i^2)U$, where U is the orthogonal matrix consisting of the eigenvectors of A and Σ. We can express the covariance of the stationary distribution as

$$\begin{aligned}\eta^2\sum_{k=0}^{\infty}(I-\eta A)^k\Sigma(I-\eta A)^k &= \eta^2 U\text{diag}\left(\sum_{k=0}^{\infty}\sigma_i^2(1-\eta d_i)^{2k}\right)U^\top && (9.194)\\ &= \eta U\text{diag}\left(\frac{\sigma_i^2}{d_i}\right)U^\top. && (9.195)\end{aligned}$$

Interpretation. Intuitively, $\frac{\sigma_i^2}{d_i}$ here is the iterate fluctuation in the direction of the i-th eigenvector. This results tell us that the fluctuation of the iterates depends on the shape of Σ and A. If Σ is not full rank, the fluctuations will be limited to the subspace K. Also note that $\mathbb{E}[\|x_t\|_2] = \Theta(\sqrt{\eta})$. This reflects the noise accumulation since the scale of noise in each step is $\Theta(\eta)$. However, we still do not have any implicit regularization effect. This is because the Hessian of the quadratic objective is unchanged. When we have the change in Hessian, SGD noise will exert an implicit bias on the iterate. See Figure 9.3 for an example.

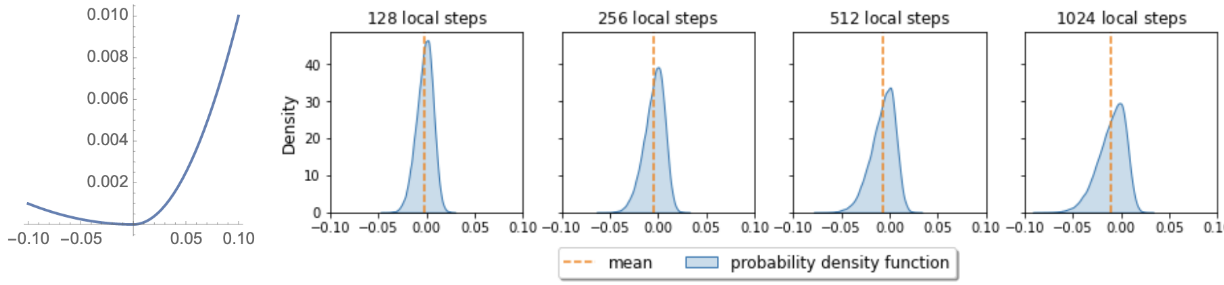


Figure 9.3: The effect of SGD noise with the change in Hessian when $x = 0$. Consider the objective $F(x) = x^2$ when $x \le 0$ and $F(x) = \frac{1}{10}x^2$ when $x > 0$. Suppose we initialize SGD at $x = 0$ and run 1024 steps of SGD with step size 0.01. We plot the probability density of the iterate after various steps of SGD. Note that the density function and the mean gradually move to the left.

In the sequel, we separately analyze the second order and third order effects of SGD on a general non-quadratic function. The second order effect exactly corresponds to this section's analysis when A equals the Hessian of the general non-quadratic function.

[5] For random variable $\xi \in \mathbb{R}^d$, $\mathbb{E}[(W\xi)(W\xi)^\top] = W\,\mathbb{E}[\xi\xi^\top]W^\top$

9.5.3 SGD on non-quadratic functions

In this section, we analyze SGD on non-quadratic functions based on [Damian et al., 2021]. Due to the complexity of the analysis, we provide heuristic derivations to convey the main insights.

Without loss of generality, suppose a global minimum of $g(x)$ is $x = 0$. Therefore, $\nabla_x g(0) = 0$ and $\nabla_x^2 g(0)$ is PSD. We also assume the iterates x_t are close to 0, so we can Taylor expand around 0.

$$\begin{aligned} x_{t+1} &= x_t - \eta(\nabla g(x_t) + \xi_t) && (9.196) \\ &= x_t - \eta(\nabla g(0) + \nabla^2 g(0)(x_t - 0) + \nabla^3 g(0)[x_t, x_t] + \text{higher order terms} + \xi_t). && (9.197) \end{aligned}$$

Let $H = \nabla_x^2 g(0)$ and $T = \nabla_x^3 g(0)$. Since T is a tensor, we first clarify our notation. First, for $T \in \mathbb{R}^{d\times d\times d}$, $x, y \in \mathbb{R}^d$, $T[x, y] \in \mathbb{R}^d$, and

$$T[x,y]_i \triangleq \sum_{j,k\in[d]} T_{ijk} x_j y_k. \tag{9.198}$$

For $S \in \mathbb{R}^{d\times d}$, $T(S) \in \mathbb{R}^d$, and

$$T(S)_i \triangleq \sum_{j,k\in[d]} T_{ijk} S_{jk} \tag{9.199}$$

Now returning to (9.197), after dropping the higher order terms, we obtain the following third-order Taylor expansion:

$$\begin{aligned} x_{t+1} &\approx x_t - \eta H x_t - \eta \xi_t - \eta T[x_t, x_t] && (9.200) \\ &= (I - \eta H)x_t - \eta\xi_t - \eta T[x_t, x_t]. && (9.201) \end{aligned}$$

If we don't consider the third order term $\eta T[x_t, x_t]$, the update reduces to the one we studied in the previous subsection. Next, recall that $\|x_t\|_2 \approx \sqrt{\eta}$. Therefore, $\eta T[x_t, x_t] \approx \eta^2$. This quantity is dominated by both $\eta\xi_t$ and $\eta H x_t \approx \eta^{1.5}$.

So, when H is positive definite, the third order term can be negligible. However, in overparametrized models, H is typically low-dimensional. For instance, if the NTK matrix is full rank, then the manifold of interpolators has dimension $d - n$. Then, in the direction orthogonal to the span of H, the contraction term disappears. Letting Π_A denote projections onto the subspace A, we see that $\eta H \Pi_{K^\perp}(x_t) = 0$ and $T[x_t, x_t] \approx \eta^2$ will dominate the update in that direction.

Consider the case in which both H and Σ are not full rank. When the loss is quadratic as in the previous section, we know that the iterate x_t bounces in the subspace K and remains stable in the subspace $K^\perp$. What happens when the loss is not quadratic, i.e. $T[x_t, x_t]$ affects the gradient update?

To answer this question, we decompose the effect of the update in (9.201) between the two subspaces of interest, K and $K^\perp$. First, observe that $(I - \eta H)x_t - \eta\xi_t$ is working in K, and $-\eta T[x_t, x_t]$ is only working in $K^\perp$ because in K the effect of $\eta T[x_t, x_t]$ is dominated by $(I - \eta H)x_t - \eta\xi_t$. In previous section, we already well-characterized the effect of optimization without a third order effect. To refine our analysis of the gradient update, we define an iterate $u_{t+1} = (I - \eta H)y_t - \eta\xi_t$ in which we do not have the third order effect.[6] Then, to analyze what the implicit regularization effect is, we study $r_t = x_t - u_t$.

$$\begin{aligned} r_{t+1} &= x_{t+1} - u_{t+1} \\ &= (I - \eta H)(x_t - u_t) - \eta T[x_t, x_t] \\ &= (I - \eta H) r_t - \eta T[x_t, x_t] \\ &\approx (I - \eta H) r_t - \eta T[u_t, u_t]. \end{aligned}$$

Note that we only have the contraction and the bias terms for the r_t iterate. The stochasticity term $\eta\xi_t$ is canceled out.

[6] Note that ξ_t is the same for each u_t and x_t.

In the subspace $K = \text{span}(H)$, the effect of $\eta T[x_t, x_t]$ is again dominated by $(I - \eta H)x_t - \eta \xi_t$, so no meaningful regularization occurs. But letting Π_A denote the projection onto the subspace A, we have that in $K^\perp$,

$$\Pi_{K^\perp} r_{t+1} = \Pi_{K^\perp} r_t - \eta \Pi_{K^\perp} T[u_t, u_t] \tag{9.202}$$

$$= \Pi_{K^\perp} r_0 - \eta \sum_{k=0}^{t} \Pi_{K^\perp} T[u_k, u_k]. \tag{9.203}$$

Namely, the effect of $T[u_k, u_k]$ is slowly accumulating in $K^\perp$. In Figure 9.4, an illustration of this phenomenon is provided.

Note that the OU process is a Markov chain and a Gaussian process. Here we assume that H is constructed such that u_t converges to its stationary distribution. Suppose the Markov chain u_t mixes as $t \to \infty$. Then, $\sum_{k=0}^{t} \Pi_{K^\perp} T[u_k, u_k] \approx t\, \mathbb{E}[T[u_\infty, u_\infty]]$. By equation (9.198) and equation (9.199),

$$\mathbb{E}[T[u, u]]_i = \mathbb{E}[\sum_{j,k} T_{ijk} u_i u_j] \tag{9.204}$$

$$= \sum_{j,k} T_{ijk}\, \mathbb{E}[uu^\top]_{jk} \tag{9.205}$$

$$= T(\mathbb{E}[uu^\top])_i. \tag{9.206}$$

Therefore $\sum_{k=0}^{t} \Pi_{K^\perp} T[u_k, u_k] \approx tT(S)$ where $S \triangleq \mathbb{E}[u_\infty u_\infty^\top]$ is the covariance of the stationary distribution.

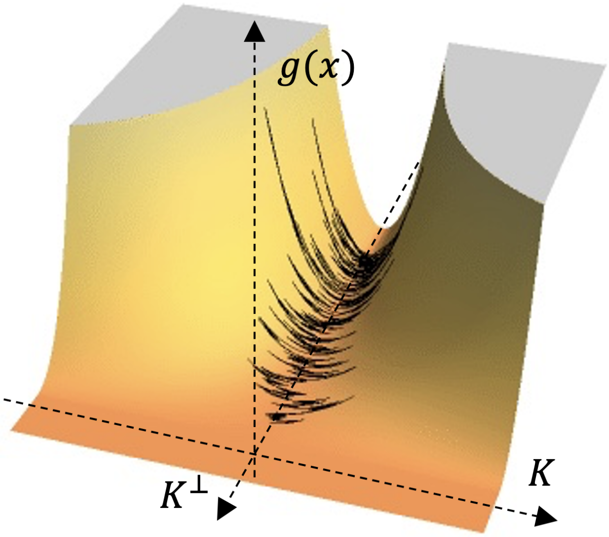


Figure 9.4: The effect of SGD noise on non-quadratic functions. K is the span of the noise covariance Σ. In the quadratic case, the iterates will fluctuate in K, but remains unchanged in $K^\perp$. When the function is non-quadratic, the third order effect slowly accumulates in $K^\perp$, resulting in implicit regularization.

Interpretation. Intuitively, the direction of the implicit regularization is $T(S) = \nabla_x \left(\langle \nabla_x^2 g(0), S \rangle \right)$. In other words, the implicit bias $-T(S)$ is trying to make $\langle \nabla_x^2 g(0), S \rangle$ small. [Damian et al., 2021] further prove that SGD with label noise on loss $\hat{L}(\theta)$ converges to a stationary point of the regularized loss $\hat{L}(\theta) + \lambda \text{tr}(\nabla_\theta^2 \hat{L}(\theta))$. In the next subsection, we will heuristically explain why this regularization term is useful.

9.5.4 SGD with label noise

We previously claimed that SGD with label noise minimizes the regularized loss

$$\hat{L}(\theta) + \lambda \text{tr}(\nabla_\theta^2 \hat{L}(\theta)). \tag{9.207}$$

But why is $\text{tr}(\nabla_\theta^2 \hat{L}(\theta))$ a useful regularizer? This question has been the subject of recent study in the implicit regularization literature. [Wei and Ma, 2019b] show that the complexity of neural networks can be controlled by its Lipschitzness. Indeed, we will see that $\text{tr}(\nabla^2 \hat{L}(\theta))$ is intimately related to the Lipschitzness of the networks. [Foret et al., 2020] also discover empirically that regularizing the sharpness of the local curvature leads to better generalization performance on a wide range of tasks. In the sequel, we will unpack some of these arguments to justify regularizing by $R(\theta) \triangleq \text{tr}\left(\nabla^2 \hat{L}(\theta)\right)$.

We first consider the case of one data point, i.e. $\hat{L}(\theta) = \ell(f_\theta(x), y)$. For notational simplicity, let $f \triangleq f_\theta(x)$ denote the model output, p be the number of parameters, and $\ell(f, y)$ be the loss function. Then,

$$\nabla^2 \hat{L}(\theta) = \nabla_\theta \left(\frac{\partial \ell}{\partial f} \cdot \frac{\partial f}{\partial \theta} \right) \tag{9.208}$$

$$= \nabla_\theta \left(\frac{\partial \ell}{\partial f} \cdot \nabla_\theta f_\theta(x) \right) \tag{9.209}$$

$$= \frac{\partial^2 \ell}{\partial f^2} \cdot \nabla_\theta f_\theta(x) \nabla_\theta f_\theta(x)^\top + \frac{\partial \ell}{\partial f} \underbrace{\nabla_\theta^2 f_\theta(x)}_{\in \mathbb{R}^{p \times p}}. \tag{9.210}$$

Suppose the loss function is $\ell(f, y) = \frac{1}{2}(f - y)^2$. Then, observing that ℓ is simply a quadratic function of f, we have

$$\nabla^2 \hat{L}(\theta) = 1 \cdot \nabla_\theta f(x) \nabla_\theta f_\theta(x)^\top + (f - y) \cdot \nabla_\theta^2 f_\theta(x), \tag{9.211}$$

Note that the first term of (9.211) is positive semi-definite (PSD), while the second term is not necessarily PSD. In general, (9.211) is referred to as the Gauss-Newton decomposition. Note also that for convex losses ℓ,

$$\frac{\partial^2 \ell}{\partial f^2} \geq 0, \tag{9.212}$$

which further implies that

$$\frac{\partial^2 \ell}{\partial f^2} \nabla f_\theta(x) \nabla f_\theta(x)^\top \succcurlyeq 0. \tag{9.213}$$

Empirically, we observe that the second term $(f - y)\nabla^2 f_\theta(x)$ is generally smaller. This is especially evident when θ is at a global minimum for which $\ell(f_\theta, y) = 0$. In this case, $(f - y)\nabla^2 f_\theta(x) = 0$ because $f_\theta(x) = y$. These two observations suggest that we can ignore the second term. In that case,

$$\text{tr}\left(\nabla^2 \hat{L}(\theta)\right) \approx \frac{\partial^2 \ell}{\partial f^2} \cdot \text{tr}\left(\nabla f(x) \nabla f(x)^\top\right) \tag{9.214}$$

$$= \frac{\partial^2 \ell}{\partial f^2} \cdot \|\nabla f_\theta(x)\|_2^2 \tag{9.215}$$

Thus, minimizing $\text{tr}\left(\nabla^2 \hat{L}(\theta)\right)$ is approximately equivalent to minimizing the Lipschitzness of the model output with respect to θ, which is approximately equivalent to minimizing the Lipschitzness of the model output with respect to hidden variables.

For example, let $\theta = (w_1, \dots, w_r)$, then we have

$$\frac{\partial f}{\partial w_i} = \frac{\partial f}{\partial h'_{i+1}} \cdot h_i^\top, \tag{9.216}$$

where $h'_{i+1} = w_i h_i$, and h_i denotes the hidden variables of the i-th layer and h'_{i+1} is the pre-activation of the $(i+1)$-th layer. Then,

$$\left\|\frac{\partial f}{\partial w_i}\right\|_F = \left\|\frac{\partial f}{\partial h_{i+1}}\right\|_2 \cdot \|h_i\|_2. \tag{9.217}$$

This validates our claim that minimizing the Lipschitzness of the model output with respect to the parameters is (approximately) equivalent to minimizing the Lipschitzness of the model output with respect to the hidden variables. We have previously connected the latter concept to generalization of deep neural networks. See Section 6.1 for a discussion of the all-layer margin, a measure of Lipschitzness of the model with respect to hidden layer variables that can be directly used to bound generalization error of a deep net.

Chapter 10

Unsupervised Learning and Self-supervised Learning

We venture into unsupervised learning by first studying classical (and analytically tractable) approaches to unsupervised learning. Classical unsupervised learning usually consists of specifying a latent variable model and fitting using the expectation-maximization (EM) algorithm. However, so far we do not have a comprehensive theoretical analysis for the convergence of EM algorithms because fundamentally analyzing EM algorithms involves understanding non-convex optimization. Most analysis of EM only applies to special cases (e.g., see Xu et al. [2016], Daskalakis et al. [2016]) and it is not clear whether any of the results can be extended to more realistic, complex setups, without a fundamentally new technique for understanding nonconvex optimization. Instead, we will analyze a family of algorithms which are broadly referred to as spectral methods or tensor methods, which are a particular application of the method of moments [Pearson, 1894] with the algorithmic technique of tensor decomposition [Anandkumar et al., 2015]. While the spectral method appears to be not as empirically sample-efficient as EM, it has provable guarantees and arguably is more reliable than EM given the provable guarantees.

After discussing the basics of classical unsupervised learning, we will move on to modern applications of deep learning. In particular, we'll focus on theoretical interpretations of contrastive learning, which is a class of successful self-supervised learning algorithms in computer vision.

10.1 Method of Moments for mixture models

We begin by formally describing the unsupervised learning problem. First, assume that we are studying a family of distributions P_θ parameterized by $\theta \in \Theta$, where P_θ can be described by a latent variable model. Then, given data $x^{(i)}, ..., x^{(n)}$ that is sampled i.i.d. from some distribution in $\{P_\theta\}_{\theta\in\Theta}$, our goal is to recover the true θ.

Perhaps the most well-studied latent variable model in machine learning is the mixture of Gaussians. We consider the following model for the mixture of k d-dimensional Gaussians. Let

$$\theta = ((\mu_1, \cdots, \mu_k), (p_1, \cdots, p_k)), \tag{10.1}$$

where $\mu_i \in \mathbb{R}^d$ is the mean of the i-th component and p is a vector of probabilities belonging to the k-simplex, which represents the mixture coefficient for clusters. Formally, for $\Delta(k) \triangleq \{p : \|p\|_1 = 1, p \geq 0, p \in \mathbb{R}^k\}$,

$$p = (p_1, \cdots, p_k) \in \Delta(k). \tag{10.2}$$

We then sample $x \sim P_\theta$ in a two-step approach:

$$\begin{aligned} &i \sim \text{categorical}(p), \\ &x \sim \mathcal{N}(\mu_i, I). \end{aligned} \tag{10.3}$$

Here i is called the latent variable since we only observe x. Here we assume the covariances of the Gaussians to be identity, but they can also be parameters that are to be learned.

There are many other latent variables that could be defined via a similar generative process, such as Hidden Markov Models, Independent Component Analysis, which we will discuss later.

10.1.1 Warm-up: mixture of two Gaussians

We first study a simple case: the mixture of two Gaussians. In this case, $k = 2$, and we assume $p_1 = p_2 = \frac{1}{2}$. For simplicity, we also assume $\mu_1 = -\mu_2$, that is, the means of the two Gaussians are symmetric around the origin. To simplify our notation, let $\mu_1 = \mu$ and $\mu_2 = -\mu$. These assumptions yield the following model for x:

$$x \sim \frac{1}{2}\mathcal{N}(\mu, I) + \frac{1}{2}\mathcal{N}(-\mu, I). \tag{10.4}$$

To implement the moment method, we need to complete the following two tasks:

1. Estimate the moment(s) of x using empirical samples.
2. Recover parameters from the moment(s) of x.

The first moment of x is

$$\begin{aligned} M_1 &\triangleq \mathbb{E}[x] && (10.5)\\ &= \frac{1}{2}\,\mathbb{E}[x|i=1] + \frac{1}{2}\,\mathbb{E}[x|i=2] && (10.6)\\ &= \frac{1}{2}\mu + \frac{1}{2}(-\mu) && (10.7)\\ &= 0. && (10.8) \end{aligned}$$

Therefore, the first moment provides no information about μ. We compute the second moment as

$$\begin{aligned} M_2 &\triangleq \mathbb{E}[xx^\top] && (10.9)\\ &= \frac{1}{2}\,\mathbb{E}[xx^\top|i=1] + \frac{1}{2}\,\mathbb{E}[xx^\top|i=2] && (10.10) \end{aligned}$$

To compute these expectations, consider an arbitrary $Z \sim \mathcal{N}(\mu, I)$. Then,

$$\begin{aligned} \mathbb{E}[ZZ^\top] &= \mathbb{E}[Z]\,\mathbb{E}[Z]^\top + \mathrm{Cov}(Z) && (10.11)\\ &= \mu\mu^\top + I && (10.12) \end{aligned}$$

Recognizing that this second moment calculation is the same for both Gaussians in our mixture, we obtain:

$$\begin{aligned} M_2 &= \frac{1}{2}(\mu\mu^\top + I) + \frac{1}{2}(\mu\mu^\top + I) && (10.13)\\ &= \mu\mu^\top + I && (10.14) \end{aligned}$$

Since the second moment provides information about μ, we can complete the two tasks required for the moment method using the second moment.

If we had access to infinite data, then we can compute the exact second moment $M_2 = \mu\mu^\top + I$. Then, we can recover μ by evaluating the top eigenvector and eigenvalue of M_2.[1] The top eigenvector and eigenvalue of M_2 is $\bar{\mu} \triangleq \frac{\mu}{\|\mu\|_2}$ and $\|\mu\|_2^2 + 1$, respectively.

In practice, however, we do not have infinite data. In that case, we need to estimate the second moment by an empirical average.

$$\widehat{M_2} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} {x^{(i)}}^\top \tag{10.15}$$

[1] This approach is known as the spectral method.

We can then recover μ by evaluating the top egivenvector and eigenvalue of $\widehat{M_2}$. However, we need this algorithm to be robust to errors, i.e., similar estimates, $\widehat{M_2}$, of the second moment should yield similar estimates of μ. Fortunately, most algorithms we might use for obtaining the top eigenvector and eigenvalue are robust, so we can limit our attention to the infinite data case. Having outlined the moment method approach to the mixture of two Gaussians problem, we study a generalization of this problem in the sequel.

10.1.2 Mixture of Gaussians with more components via tensor decomposition

The general moment method for solving latent variable models is summarized by the following steps.

1. Compute $M_1 = \mathbb{E}[x]$, $M_2 = \mathbb{E}[xx^\top]$, $M_3 = \mathbb{E}[x \otimes x \otimes x]$, $M_4 = \cdots$. Note that $x \otimes x \otimes x$ is in $\mathbb{R}^{d \times d \times d}$ and $(x \otimes x \otimes x)_{ijk} = x_i \cdot x_j \cdot x_k$. For example, $M_{3,ijk} = \mathbb{E}[x_i x_j x_k]$.
2. Design as algorithm $A(M_1, M_2, M_3, \dots)$ that outputs θ.
3. Show that A is robust to errors in our moment estimates, i.e., we apply A to $(\widehat{M_1}, \widehat{M_2}, \widehat{M_3}, ...)$ in reality.

In the sequel, we instantiate this paradigm for mixtures of k Gaussians ($k \geq 3$).

For the simplicity of demonstrating the idea, we assume $p_1 = \cdots = p_k = \frac{1}{k}$, i.e. $i \overset{\text{unif}}{\sim} [k]$, and $x \sim \mathcal{N}(\mu_i, I)$. Equivalently,

$$x \sim \frac{1}{k} \sum_{i=1}^{k} \mathcal{N}(\mu_i, I). \tag{10.16}$$

In this example, we only describe steps (1) and (2) in the general algorithm described above.

We first evaluate the first and second moments. The first moment follows from

$$\begin{aligned} M_1 &= \mathbb{E}[x] && (10.17)\\ &= \sum_{i=1}^{k} \frac{1}{k} \mathbb{E}[x|i] && (10.18)\\ &= \frac{1}{k} \sum_{i=1}^{k} \mu_i, && (10.19) \end{aligned}$$

and the second moment is computed as

$$\begin{aligned} M_2 &= \mathbb{E}[xx^\top] && (10.20)\\ &= \sum_{i=1}^{k} \frac{1}{k} \mathbb{E}[xx^\top|i] && (10.21)\\ &= \sum_{i=1}^{k} \frac{1}{k} (\mu_i \mu_i^\top + I) && (10.22)\\ &= \frac{1}{k} \sum_{i=1}^{k} \mu_i \mu_i^\top + I. && (10.23) \end{aligned}$$

Second moments do not suffice

Can we recover $\mu = (\mu_1, ..., \mu_k)$ from $\frac{1}{k}\sum_{i=1}^{k} \mu_i$ and $\frac{1}{k}\sum_{i=1}^{k} \mu_i \mu_i^\top$? Unfortunately, in most of the cases when $k \geq 3$, the first and second moments are not sufficent to recover μ.

One reason is the so-called "missing rotation information" problem. Let

$$U = \begin{bmatrix} | & & | \\ \mu_1 & \cdots & \mu_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{d \times k} \tag{10.24}$$

denote the matrix we aim to recover. Then, consider some rotation matrix $R \in \mathbb{R}^{k\times k}$. We consider U versus UR:

$$\frac{1}{k}\sum_{i=1}^{k}\mu_i\mu_i^\top = \frac{1}{k}UU^\top \tag{10.25}$$

$$= \frac{1}{k}(UR)(UR)^\top \qquad (RR^\top = I) \tag{10.26}$$

This result proves that the second moment is invariant to rotations. To prove a similar claim for the first moment, we also constrain our choice of R such that

$$R \cdot \vec{1} = \vec{1} \tag{10.27}$$

Then,

$$\frac{1}{k}\sum_{i=1}^{k}\mu_i = \frac{1}{k}U \cdot \vec{1} \tag{10.28}$$

$$= \frac{1}{k}UR \cdot \vec{1} \tag{10.29}$$

Therefore, the first and second moments of U and UR are indistinguishable, and we must consider the third moment in order to identify U.

Computing the third moment

The third moment is the tensor $\mathbb{E}[x \otimes x \otimes x] \in \mathbb{R}^{d\times d\times d}$. To express this expectation in terms of tractable quantities, we can condition on the Gaussian observed and average:

$$\mathbb{E}[x \otimes x \otimes x] = \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}[x \otimes x \otimes x \mid i] \tag{10.30}$$

Each term in the sum now corresponds to the third moment for some multivariate Gaussian. Fortunately, Lemma 10.1 suggests a formula for estimating its value.

Lemma 10.1. *Suppose $z \in \mathcal{N}(v, I)$. Then,*

$$\mathbb{E}[z \otimes z \otimes z] = v \otimes v \otimes v + \sum_{l=1}^{d}\mathbb{E}[z] \otimes e_l \otimes e_l + \sum_{l=1}^{d} e_l \otimes \mathbb{E}[z] \otimes e_l + \sum_{l=1}^{d} e_l \otimes e_l \otimes \mathbb{E}[z] \tag{10.31}$$

where $e_1, \ldots, e_d$ denote the canonical basis vectors.

This lemma suggests that we can compute $v \otimes v \otimes v$ from a linear combination of $\mathbb{E}[z \otimes z \otimes z]$ and $\mathbb{E}[z]$. Also note that $\mathbb{E}[z] = v$. .

Proof. We compute the third moment element-wise. That is,

$$(\mathbb{E}[z \otimes z \otimes z])_{ijk} = \mathbb{E}[z_i z_j z_k] \tag{10.32}$$

$$= \mathbb{E}[(v_i + \xi_i)\cdot(v_j + \xi_j)\cdot(v_k + \xi_k)] \qquad (z = v + \xi, \xi \sim \mathcal{N}(0, I)) \tag{10.33}$$

$$= v_i v_j v_k + \underbrace{\mathbb{E}[v_i v_j \xi_k] + \mathbb{E}[v_i \xi_j v_k] + \mathbb{E}[\xi_i v_j v_k]}_{=0}$$

$$+ \mathbb{E}[v_i \xi_j \xi_k] + \mathbb{E}[v_j \xi_i \xi_k] + \mathbb{E}[v_k \xi_i \xi_j] + \mathbb{E}[\xi_i \xi_j \xi_k] \tag{10.34}$$

To explicitly compute the last four terms in (10.34), we note that:

$$\mathbb{E}[\xi_i \xi_k] = \begin{cases} 0 & \text{if } i \neq k \\ \mathbb{E}[\xi_i^2] = 1 & \text{if } i = k \end{cases} \tag{10.35}$$

and that for any choice of i, j, and k,

$$\mathbb{E}[\xi_i \xi_j \xi_k] = 0. \tag{10.36}$$

Therefore,

$$(\mathbb{E}[z \otimes z \otimes z])_{ijk} = v_i v_j v_k + v_i \mathbf{1}[j = k] + v_j \mathbf{1}[i = k] + v_k \mathbf{1}[i = j] \tag{10.37}$$

Since $(\sum_{l=1}^d v \otimes e_l \otimes e_l)_{ijk} = \sum_{l=1}^d v_i e_{lj} e_{lk} = v_i \mathbb{I}(j = k)$, we have proven that:

$$\mathbb{E}[z \otimes z \otimes z] = v \otimes v \otimes v + \sum_{l=1}^d v \otimes e_l \otimes e_l + \sum_{l=1}^d e_l \otimes v \otimes e_l + \sum_{l=1}^d e_l \otimes e_l \otimes v. \tag{10.38}$$

□

We can now apply Lemma 10.1 to compute the third moment of the mixture of k Gaussians. In particular,

$$\begin{aligned}
\mathbb{E}[x \otimes x \otimes x] &= \frac{1}{k} \sum_{i=1}^k \mathbb{E}[x \otimes x \otimes x \mid i] && (10.39) \\
&= \frac{1}{k} \sum_{i=1}^k \left(\mu_i \otimes \mu_i \otimes \mu_i + \sum_{l=1}^d \mu_i \otimes e_l \otimes e_l + \sum_{l=1}^d e_l \otimes \mu_i \otimes e_l + \sum_{l=1}^d e_l \otimes e_l \otimes \mu_i \right) && (10.40) \\
&= \frac{1}{k} \sum_{i=1}^k \mu_i \otimes \mu_i \otimes \mu_i + \sum_{l=1}^d \frac{1}{k} \left(\sum_{i=1}^k \mu_i \right) \otimes e_l \otimes e_l + \sum_{l=1}^d e_l \otimes \frac{1}{k} \left(\sum_{i=1}^k \mu_i \right) \otimes e_l \\
&\quad + \sum_{l=1}^d e_l \otimes e_l \otimes \frac{1}{k} \left(\sum_{i=1}^k \mu_i \right) && (10.41) \\
&= \frac{1}{k} \sum_{i=1}^k \mu_i \otimes \mu_i \otimes \mu_i + \sum_{l=1}^d \mathbb{E}[x] \otimes e_l \otimes e_l + \sum_{l=1}^d e_l \otimes \mathbb{E}[x] \otimes e_l + \sum_{l=1}^d e_l \otimes e_l \otimes \mathbb{E}[x] && (10.42) \\
& && (10.43)
\end{aligned}$$

For notational convenience, let

$$a^{\otimes 3} \triangleq a \otimes a \otimes a. \tag{10.44}$$

So far, we have shown how to compute $\frac{1}{k} \sum_{i=1}^k \mu_i^{\otimes 3}$ from $\mathbb{E}[x^{\otimes 3}]$ and $\mathbb{E}[x]$. In the sequel, we will formalize the remaining problem, recovering $\{\mu_i\}_{i=1}^k$ from $\frac{1}{k} \sum_{i=1}^k \mu_i^{\otimes 3}$, as the tensor decomposition problem, and discuss efficient algorithms for it.

Tensor decomposition Recovering the Gaussian means, $\{\mu_i\}_{i=1}^k$, from the third mixture moment, $\frac{1}{k} \sum_{i=1}^k \mu_i^{\otimes 3}$, is a special case of the general tensor decomposition problem. That problem is set up as follows: Assume that $a_1, \cdots a_k \in \mathbb{R}^d$ are unknown. Then, given $\sum_{i=1}^k a_i^{\otimes 3}$, our goal is to reconstruct the a_i vectors.

Before we present some standard results on tensor decomposition, we first describe some basic facts about tensors. Much like matrices, tensors have an associated rank. For example, $a \otimes b \otimes c$ is a rank-1 tensor. In general, the rank of a tensor T is the minimum k such that T can be decomposed as

$$T = \sum_{i=1}^k a_i \otimes b_i \otimes c_i. \tag{10.45}$$

for some $\{a_i\}_{i=1}^k, \{b_i\}_{i=1}^k, \{c_i\}_{i=1}^k$. Another difference between tensors and matrices is that the former objects do not have the typical rotational invariance. In particular, consider applying a right rotation $R \in \mathbb{R}^{k\times k}$ to the matrix

$$A = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{d\times k} \tag{10.46}$$

and get

$$\widetilde{A} = AR = \begin{bmatrix} | & & | \\ \tilde{a}_1 & \cdots & \tilde{a}_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{d\times k} \tag{10.47}$$

Then,

$$\sum_{i=1}^k a_i a_i^\top = AA^\top = (AR)\cdot(AR)^\top = \sum_{i=1}^k \tilde{a}_i \tilde{a}_i^\top \tag{10.48}$$

However, there is no tensor analogue to the rotation invariance result above. But tensors do maintain an interesting (and useful) permutation invariance; that is, $T = \sum_{i=1}^k a_i^{\otimes 3}$ is invariant to permutations of the indices of a_i. Or in other words, let $P \in \mathbb{R}^{k\times k}$ be a permutation matrix suppose, and let

$$\widetilde{A} = AP = \begin{bmatrix} | & & | \\ \tilde{a}_1 & \cdots & \tilde{a}_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{d\times k} \tag{10.49}$$

Then,

$$\sum_{i=1}^k a_i^{\otimes 3} = \sum_{i=1}^k \tilde{a}_i^{\otimes 3} \tag{10.50}$$

The lack of rotation invariance in the sense above and the existence of permutation invariance make tensor decomposition computationally challenging as well as powerful.

We now summarize some standard results regarding tensor decomposition for $T = \sum_{i=1}^k a_i^{\otimes 3}$. The results for decomposing the asymmetric version $T = \sum_{i=1}^k a_i \otimes b_i \otimes$ are largely analogous. We will not prove these results here.

0. In the most general case, recovering the a_i's from T is computationally hard. Any procedure will either fail to find a unique a_i or it fails to find a_i *efficiently.*

1. In the orthogonal case, i.e. $a_1, \dots, a_k$ are orthogonal vectors, each a_i is a global maximizer of

$$\max_{\|x\|_2=1} T(x,x,x) = \sum_{i,j,k} T_{ijk} x_i x_j x_k \tag{10.51}$$

   We can heuristically think of a_i as eigenvectors of T and there exists an algorithm to compute a_i in poly-time.

2. In the independent case, i.e. $a_1, \dots, a_k$ are linearly independent. Jennrich's algorithm can be used to efficiently recover $\{a_i\}_{i=1}^k$.

Results 1 and 2 above both involve the so-called "under-complete" case ($k \le d$), e.g., when the number of Gaussians in the mixture is smaller than the dimension of the data. Next, we describe certain overcomplete cases for which efficient tensor decomposition is possible.

3. Suppose $a_1^{\otimes 2}, \dots, a_k^{\otimes 2}$ are independent for $k \leq d^2$. Then, applying Result 2, we can recover a_i from $\sum_{i=1}^k (a_i^{\otimes 2})^{\otimes 3} = \sum_{i=1}^k (a_i^{\otimes 6}) \in \mathbb{R}^{d^6}$.
4. Excluding an algebraic set of measure 0, we can use the FOOBI algorithm to recover a_i from the fourth-order tensor $\sum_{i=1}^k a_i^{\otimes 4}$ when $k \leq d^2$. A robust version of the FOOBI algorithm is described in Ma et al. [2016].
5. Assume a_i are *randomly* generated unit vectors. Then, for the third order tensor, k can be large as $d^{1.5}$ [Ma et al., 2016, Schramm and Steurer, 2017].

In summary, the moment method is a recipe in which we first compute high order moments (i.e. tensors), assume that these estimates are noiseless, and decompose these tensors to recover the latent variables. Though we do not discuss these results here, there is an extensive literature analyzing the robustness of the moment method to error in the moment estimates. Last, we note that even though we only explicitly analyze the mixture of Gaussians model here, latent variable models amenable to analysis by the moment method include ICA, Hidden Markov Models, topic models, etc.

10.2 Graph Decomposition and Spectral Clustering

Introduced by Shi and Malik [2000] and Ng et al. [2001], spectral clustering learns a model for the data points using a *pairwise* notion of similarity. Formally, assume that we are given n data points $x^{(1)}, \dots, x^{(n)}$ as well as a similarity matrix $G \in \mathbb{R}^{n \times n}$ such that

$$G_{ij} = \rho(x^{(i)}, x^{(j)}) \tag{10.52}$$

where ρ is some measure of similarity that assigns larger values to more similar pairs of points.

For example, $x^{(i)}$ could denote images for which $\rho(x^{(i)}, x^{(j)})$ measures the semantic similarity. Alternatively, $x^{(i)}$ might be users of a social network and $\rho(x^{(i)}, x^{(j)}) = 1$ if $x^{(i)}$ and $x^{(j)}$ are friends (hence usually share similar interests / jobs / $\cdots$).

We note that in moment methods, $\mathbb{E}[xx^\top]$ captures pairwise information between coordinates / dimensions, whereas matrix G here captures pairwise information between datapoints.

Our goal is to cluster the data points by viewing G as a graph. For instance, in the social network example, we can naturally view G as the adjacency matrix of an *unweighted* graph, where $G_{ij} \in \{0, 1\}$ defines the edges. Then, the clustering problem is to partition the graph into distinct neighborhoods, i.e., components that are as separate from each other as possible. As we will see repeatedly in the sequel, the eigendecomposition of G is closely related to this graph paritioning / clustering problem.

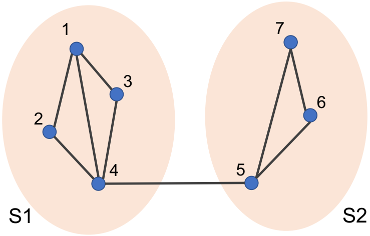


Figure 10.1: A demonstration of graph partitioning. Sets S_1 and S_2 form a good partition of the graph since there's only one edge between them.

10.2.1 Stochastic block model

In the stochastic block model (SBM), G is assumed to be generated randomly from two hidden communities. Formally,

$$\{1, \cdots n\} = S \cup \bar{S}, \tag{10.53}$$

where S and $\bar{S}$ partition $[n]$. Assume $|S| = \frac{n}{2}$. We then assume the following generative model for G. If $i, j \in S$ or $i, j \in \bar{S}$, then

$$G_{ij} = \begin{cases} 1 & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}. \tag{10.54}$$

Otherwise, for i and j in distinct components,

$$G_{ij} = \begin{cases} 1 & \text{w.p. } q \\ 0 & \text{w.p. } 1-q \end{cases} \tag{10.55}$$

for $p > q$ (i.e., more likely to be connected if from the same group). For instance, S and $\bar{S}$ could mean two companies, and $i \in [n]$ is a user of a social network. Two users i, j are more likely to know each other if they are in the same company.

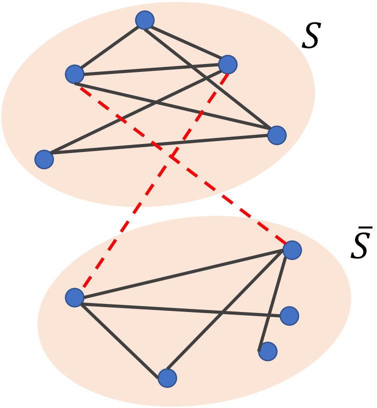


Figure 10.2: A graph generated by the stochastic block model with $p = \frac{2}{3}$ and $q = \frac{1}{5}$.

Our goal is then to recover S and $\bar{S}$ from G; the primary tool we use towards this goal is the eigendecomposition of G.

In some trivial cases, it is not necessary to eigendecompose G to recover the two hidden communities. Suppose, for instance, that $p = 0.5$ and $q = 0$. Then, the graph represented by G will contain two connected components that correspond to S and $\bar{S}$.

As a warm-up to motivate our approach, we eigendecompose $\bar{G} = \mathbb{E}[G]$. Observe that

$$\bar{G}_{ij} = \begin{cases} p & \text{if } i, j \text{ from the same community} \\ q & \text{o.w.} \end{cases}. \tag{10.56}$$

It is then easy to see that $\bar{G}$ is a matrix of rank 2:

$$\bar{G} = \left[\begin{array}{c|c} p \cdots p & q \cdots q \\ \vdots & \vdots \\ p \cdots p & q \cdots q \\ \hline q \cdots q & p \cdots p \\ \vdots & \vdots \\ q \cdots q & p \cdots p \end{array}\right]. \tag{10.57}$$

Lemma 10.2. *Let $\bar{G} = \mathbb{E}[G]$ for the stochastic block model. Then, letting $u_i(A)$ denote the i-th eigenvector of the matrix A,*

$$u_1(\bar{G}) = \vec{1} \tag{10.58}$$

$$u_2(\bar{G}) = [\underbrace{1, \dots, 1}_{|S|}, \underbrace{-1, \dots, -1}_{|\bar{S}|}]^\top \tag{10.59}$$

where $u_2(\bar{G})$ has $|S|$ entries of 1 and $|\bar{S}|$ entries of -1.

Proof. We begin by directly proving (10.58).

$$\bar{G} \cdot \vec{1} = \begin{bmatrix} \frac{pn}{2} + \frac{qn}{2} \\ \vdots \\ \frac{pn}{2} + \frac{qn}{2} \end{bmatrix} \tag{10.60}$$

$$= \frac{p+q}{2} \cdot n \cdot \vec{1}. \tag{10.61}$$

More generally, $\vec{1}$ is the top eigenvector for any matrix with fixed row sum or any graph with uniform degree (i.e., regular graph).

Next, we prove (10.59). Let

$$G' = \left[\begin{array}{c|c} \begin{matrix} r \cdots r \\ \vdots \\ r \cdots r \end{matrix} & \text{\Large 0} \\ \hline \text{\Large 0} & \begin{matrix} r \cdots r \\ \vdots \\ r \cdots r \end{matrix} \end{array}\right] \tag{10.62}$$

for $r = p - q$. To precisely define G', we note that G' is block diagonal with two blocks of size $|S|$ and $|\bar{S}|$, respectively. Then,

$$\bar{G} = \vec{1}\vec{1}^\top q + G'. \tag{10.63}$$

Thus,

$$G' \cdot u = \left[\begin{array}{c|c} \begin{matrix} r \cdots r \\ \vdots \\ r \cdots r \end{matrix} & \text{\Large 0} \\ \hline \text{\Large 0} & \begin{matrix} r \cdots r \\ \vdots \\ r \cdots r \end{matrix} \end{array}\right] \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} = r \cdot \frac{n}{2} \cdot u. \tag{10.64}$$

Then, because $u \perp \vec{1}$, we can combine (10.63) and (10.64) to obtain

$$\bar{G} \cdot u = G' \cdot u = r \cdot \frac{n}{2} \cdot u = \frac{p-q}{2} \cdot n \cdot u \tag{10.65}$$

Thus, u has eigenvalue $\frac{p-q}{2} \cdot n$. □

Remark 10.3. Lemma 10.2 shows that

$$\bar{G} = \frac{p+q}{2} \cdot \vec{1}\vec{1}^\top + \frac{p-q}{2} \cdot uu^\top. \tag{10.66}$$

More generally, when we have more than two clusters in the graph, G' is block diagonal with more than two blocks. In this setting, the eigenvectors will still align with the blocks. We illustrate this below for a generic block diagonal matrix. Let

$$A = \left[\begin{array}{c|c|c} \begin{matrix}1\cdots1\\ \vdots \\ 1\cdots1\end{matrix} & 0 & 0 \\ \hline 0 & \begin{matrix}1\cdots1\\ \vdots \\ 1\cdots1\end{matrix} & 0 \\ \hline 0 & 0 & \begin{matrix}1\cdots1\\ \vdots \\ 1\cdots1\end{matrix} \end{array}\right] \tag{10.67}$$

Then, the three eigenvectors of A are

$$\begin{bmatrix}1\\ \vdots\\ 1\\ 0\\ \vdots\\ 0\\ 0\\ \vdots\\ 0\end{bmatrix}, \begin{bmatrix}0\\ \vdots\\ 0\\ 1\\ \vdots\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix}, \begin{bmatrix}0\\ \vdots\\ 0\\ 0\\ \vdots\\ 0\\ 1\\ \vdots\\ 1\end{bmatrix} \tag{10.68}$$

Furthermore, the rows of the matrix formed by the three eigenvectors given by (10.68) clearly give the cluster IDs of the vertices in G. Note also that permutations of A will result in equivalent permutations in the coordinates of each of the three eigenvectors.

Next, we relate this observation to the result in Lemma 10.2. While there are no negative values in the eigenvectors given in (10.68), we observe that any linear combination of eigenvectors is also an eigenvector, so recovering a solution that look more like (10.59) is straightforward. Indeed, taking linear combinations of the eigenvectors defined above shows that there is an alternative eigenbasis that includes the all-ones vector, $\vec{1}$. How for this choice of A, the all-ones vector is not the unique top eigenvector. For that to be the case, we require background noise in $\bar{G}$.

In reality, we only observe G. In the sequel, we will show that in terms of the spectrum, $G \approx \mathbb{E}[G]$. Formally, we will leverage earlier concentration results to prove that $\|G - \mathbb{E}[G]\|_{\text{op}}$ is small. Concretely, then,

$$\begin{aligned} G &= (G - \mathbb{E}[G]) + \mathbb{E}[G] && (10.69)\\ &= (G - \mathbb{E}[G]) + \frac{p+q}{2} \cdot \vec{1}\vec{1}^\top + \frac{p-q}{2} \cdot uu^\top && (10.70)\end{aligned}$$

Rearranging, we obtain that:

$$G - \frac{p+q}{2} \cdot \vec{1}\vec{1}^\top = (G - \mathbb{E}[G]) + \frac{p-q}{2} \cdot uu^\top \tag{10.71}$$

We then hope that $G - \mathbb{E}[G]$ is a small perturbation, so that the top eigenvector of $G - \frac{p+q}{2} \cdot \vec{1}\vec{1}^\top$ is close to u. Namely, it suffices to show that

$$\|G - \mathbb{E}[G]\|_{\text{op}} \ll \left\| \frac{p-q}{2} \cdot uu^\top \right\|_{\text{op}}. \tag{10.72}$$

We will start by proving the following lemma.

Lemma 10.4. *With high probability,*

$$\|G - \mathbb{E}[G]\|_{\mathrm{op}} \leq O(\sqrt{n \log n}) \ . \tag{10.73}$$

Note that this concentration inequality is different from the ones we have seen in the course so far because both G and $\mathbb{E}[G]$ are matrices, not scalars. Our goal will be to turn the quantity on the LHS into something that we are familiar with.

Proof. The key idea is that we can use uniform convergence, after noting that

$$\begin{aligned}
\|G - \mathbb{E}[G]\|_{\mathrm{op}} &= \max_{\|v\|_2=1} \left|v^\top (G - \mathbb{E}[G])v\right| && (10.74)\\
&= \max_{\|v\|_2=1} \left|v^\top G v - v^\top \mathbb{E}[G] v\right| && (10.75)\\
&= \max_{\|v\|_2=1} \left|\sum_{i,j\in[n]} v_i v_j G_{ij} - \mathbb{E}\left[\sum_{i,j\in[n]} v_i v_j G_{ij}\right]\right| \ . && (10.76)
\end{aligned}$$

Now, the quantity inside the max is the difference between the sum of independent random variables and their expectation, which we are familiar with. We can use brute force discretization to deal with the max. First, note that for a fixed v with $\|v\|_2 = 1$, we can use Hoeffding's inequality to find that

$$\Pr\left(\left|\sum_{i,j\in[n]} v_i v_j G_{ij} - \mathbb{E}\left[\sum_{i,j\in[n]} v_i v_j G_{ij}\right]\right| \geq \epsilon\right) \leq \exp(-\frac{\epsilon^2}{2}) \ . \tag{10.77}$$

Then, we choose $\epsilon = O(\sqrt{n \log n})$, take a discretization of the unit ball with granularity $1/n^{O(1)}$ (which yields a cover of cardinality $\exp(n \log n)$), and take a union bound over this discretized set to achieve the desired result. □

Remark 10.5. Comparing this bound to $\frac{p-q}{2} \cdot n$, we can deduce that if $p - q \gg \frac{\sqrt{\log n}}{\sqrt{n}}$, then we can recover the vector u approximately. Via a post-processing step that we do not discuss here, u can actually be recovered exactly.

10.2.2 Clustering the worst-case graph

Given a graph $G(V, E)$ where V denotes the set of vertices and E the set of edges, we define the *conductance* of a component S as

$$\phi(S) \triangleq \frac{|E(S, \bar{S})|}{\mathrm{Vol}(S)} \tag{10.78}$$

where $E(S, \bar{S})$ is the total number of edges between S and $\bar{S}$, and $\mathrm{Vol}(S)$ is the total number of edges connecting to S. To be precise, let A be the adjacency matrix of G,

$$\begin{aligned}
E(S, \bar{S}) &= \sum_{i\in S, j\in\bar{S}} A_{ij} && (10.79)\\
\mathrm{Vol}(S) &= \sum_{i\in S, j\in[n]} A_{ij} \ . && (10.80)
\end{aligned}$$

Intuitively, conductance captures how separated S and $\bar{S}$ are.

Since $\mathrm{Vol}(S) \geq E(S, \bar{S})$, it follows that $\phi(S) \leq 1$. Next, observe that $\mathrm{Vol}(S) + \mathrm{Vol}(\bar{S}) = \mathrm{Vol}(V)$. Then, if $\mathrm{Vol}(S) \leq \mathrm{Vol}(V)/2$, it must be the case that $\mathrm{Vol}(S) \leq \mathrm{Vol}(\bar{S})$ and therefore $\phi(S) \geq \phi(\bar{S})$. In some sense, thus suggests that the conductance of a set $\bar{S}$ with volume larger than $\mathrm{Vol}(V)/2$ could be misleading, because the conductance of the other part could be larger. Therefore, typically people only consider the conductance of a smaller part of the partition.

Next, we define $\phi(G)$ to be the *sparsest cut* of G:

$$\phi(G) = \min_{S:\mathrm{Vol}(S) \le \mathrm{Vol}(V)/2} \phi(S)\ . \tag{10.81}$$

One may wonder why we need to normalize by $\mathrm{Vol}(S)$ in the definition of conductance. The reason is that $E(S, \bar{S})$ itself is typically minimized when S is small. Thus, without this normalization, the sparsest cut would not be very meaningful. For instance, suppose the graph G contains N nodes and can be divided into two halves each containing $N/2$ nodes, and every node is connected to all the other nodes in the same half, but is connected to only 2 nodes in the other half (as shown in Figure 10.3). Then, we can consider two subsets S_1 and S_2, where S_1 contains half the nodes, and S_2 contains two nodes in the same half. It's easy to see that $E(S_1, \bar{S}_1) = \frac{N}{2} \cdot 2 > E(S_2, \bar{S}_2) = \frac{N}{2}$. However, the conductance of S_1 is $\phi(S_1) = \frac{E(S_1,\bar{S}_1)}{\mathrm{Vol}(S_1)} = \frac{\frac{N}{2}\cdot 2}{\frac{N}{2}\cdot(\frac{N}{2}-1)+\frac{N}{2}\cdot 2} \approx \frac{4}{N}$, whereas the conductance of S_2 is $\phi(S_2) = \frac{E(S_2,\bar{S}_2)}{\mathrm{Vol}(S_2)} = \frac{\frac{N}{2}}{N+2} \approx \frac{1}{2}$. Thus, when n is large, S_1 is a sparser cut than S_2 under $\phi(\cdot)$.

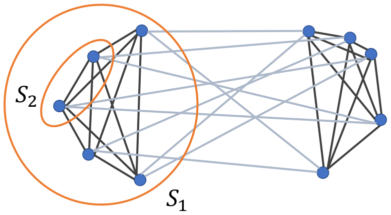


Figure 10.3: A demonstration of the sparsest cut. S_1 is a sparser cut than S_2.

Our goal is to find an approximate sparsest cut $\hat{S}$ such that $\phi(\hat{S}) \approx \phi(G)$.[2] Our main technique is eigendecomposition or spectral clustering, though in the literature much more advanced and better algorithms have been proposed, e.g., the famous ARV algorithm [Arora et al., 2009]. Let $d_i = \mathrm{Vol}(\{i\})$ be the degree of node i, and let $D = \mathrm{diag}(\{d_i\})$ be the diagonal matrix that contains the degrees d_i as entries. Furthermore, let

$$\bar{A} = D^{-\frac{1}{2}} A D^{\frac{1}{2}} \tag{10.82}$$

be the normalized adjacency matrix. This is equivalent to normalizing each element A_{ij} of the adjacency matrix by $\frac{1}{\sqrt{d_i d_j}}$ (i.e., $\bar{A}_{ij} = \frac{A_{ij}}{\sqrt{d_i d_j}}$). In most cases, it suffices to starting with considering G as a regular graph (whose degrees are all the same), because the proof for regular graph can oftentimes extend to general graph easily. Assuming G is a κ-regular graph, i.e. $d_i = \kappa$; then, this normalization simply scales A by $\frac{1}{\kappa}$.

Let $L = I - \bar{A}$ be the Laplacian matrix. Note that any eigenvector of L is also an eigenvector of $\bar{A}$. Suppose L has eigenvalues $\lambda_1 \le \ldots \le \lambda_n$ with corresponding eigenvectors $u_1 \ldots u_n$, then $\bar{A}$ has eigenvalues $1 - \lambda_1 \ge \ldots \ge 1 - \lambda_n$ with the same eigenvectors.

The following famous Cheeger's inequality relates the eigendecompositions to the graph partitioning.

Theorem 10.6 (Cheeger's inequality)**.** *The second eigenvalue of L, namely λ_2, is related to the sparsest cut $\phi(G)$ as follows:*

$$\frac{\lambda_2}{2} \le \phi(G) \le \sqrt{2\lambda_2}\ . \tag{10.83}$$

Moreover, we can find $\hat{S}$ such that $\phi(\hat{S}) \le \sqrt{2\lambda_2} \le 2\sqrt{\phi(G)}$ efficiently by rounding the second eigenvector. Suppose $u_2 = [\beta_1 \cdots \beta_n]^\top \in \mathbb{R}^n$ is the second eigenvector of L. Then we can choose a threshold $\tau = \beta_i$ and consider $\hat{S}_i = \{j \in [n] : \beta_j \le \tau\}$. At least one such $\hat{S}_i$ satisfies $\phi(\hat{S}_i) \le 2\sqrt{\phi(G)}$.

[2] Finding the exact sparsest cut is a NP-hard problem.

Note that this can be viewed as a general but weaker version of the theorem that we proved for stochastic block model. There is no randomized assumption; the conclusion is weaker than those for SBM; also the rounding algorithm to recover the cluster is also more complicated—one has to try multiple thresholding instead of using threshold 1/2.

We will refer the readers to Chung [2007] for the proof of the theorem. Here below we give a few basic lemmas that help build up intuitions on why eigendecompositions relate to graph decomposition.

First, one might wonder why this algorithm uses the second eigenvector of $\bar{A}$, but not the first eigenvector. As we have seen in the SBM case, the first eigenvector captures the background in some sense. Here for general graph, we see the same phenomenon. The top eigenvector is generally not that interesting as it only captures the "background density" of the graph. For instance, when A is κ-regular, $\vec{1}$ is the top eigenvector of A and is thus also the top eigenvector of $\bar{A} = \frac{1}{\kappa} \cdot A$. More generally, we have the following lemma:

Lemma 10.7. *The top eigenvector of $\bar{A}$ (respectively, the smallest eigenvector of L) is $u_1 = [\sqrt{d_1} \cdots \sqrt{d_n}]^\top$.*

Proof.

$$\begin{aligned}(\bar{A} \cdot u_1)_i &= \sum_j \bar{A}_{ij} \sqrt{d}_j && (10.84)\\ &= \sum_j \frac{A_{ij}}{\sqrt{d_i}\sqrt{d_j}} \sqrt{d}_j && (10.85)\\ &= \frac{1}{\sqrt{d_i}} \sum_j A_{ij} && (10.86)\\ &= \frac{d_i}{\sqrt{d_i}} = \sqrt{d}_i. && (10.87)\end{aligned}$$

□

To understand why the eigenvectors of the Laplacian are related to the sparsest cut, we examine the quadratic form the Laplacian. In particular, we have the following lemma:

Lemma 10.8. *Let $v \in \mathbb{R}^N$ be a vector, L is the graph Laplacian. Then,*

$$v^\top L v = \frac{1}{2} \sum_{(i,j) \in E} \left(\frac{v_i}{\sqrt{d_i}} - \frac{v_j}{\sqrt{d_j}} \right)^2. \tag{10.88}$$

Proof.

$$\begin{aligned}v^\top L v &= v^\top I v - v^\top \bar{A} v && (10.89)\\ &= \sum_{i=1}^n v_i^2 - \sum_{i,j=1}^n v_i v_j \bar{A}_{ij} && (10.90)\\ &= \sum_{i=1}^n v_i^2 - \sum_{i,j=1}^n v_i v_j \frac{A_{ij}}{\sqrt{d_i d_j}} && (10.91)\\ &= \frac{1}{2} \cdot \left(2 \sum_{i=1}^n v_i^2 - 2 \sum_{(i,j) \in E} \frac{v_i}{\sqrt{d_i}} \cdot \frac{v_j}{\sqrt{d_j}} \right) && (10.92)\\ &= \frac{1}{2} \sum_{(i,j) \in E} \left(\frac{v_i}{\sqrt{d_i}} - \frac{v_j}{\sqrt{d_j}} \right)^2. && (10.93)\end{aligned}$$

□

If G is κ-regular, then this becomes $v^\top Lv = \frac{1}{2\kappa}\sum_{(i,j)\in E}(v_i - v_j)^2$. Furthermore, suppose $v \in \{0,1\}$ is a binary vector with support $S = \text{supp}(v)$. Then,

$$\frac{1}{2\kappa}\sum_{(i,j)\in E}(v_i - v_j)^2 = \frac{1}{\kappa}E(S, \bar{S}) \tag{10.94}$$

$$= \frac{1}{\kappa}E(\text{supp}(v), \overline{\text{supp}}(v)) \; . \tag{10.95}$$

If $|\text{supp}(v)| \leq n/2$, implying $\text{Vol}(S) \leq \text{Vol}(V)/2$, then

$$\frac{v^\top Lv}{\|v\|_2^2} = \frac{\frac{1}{\kappa}E(S,\bar{S})}{\frac{1}{\kappa}\text{Vol}(S)} = \phi(S) \; . \tag{10.96}$$

The term $\frac{v^\top Lv}{\|v\|_2^2}$ is also called the *Rayleigh quotient*. This result nicely connects the abstract linear algebraic approach to the sparsest cut approach. Note that we only achieve an approximate sparsest cut because when we compute eigenvectors, we minimize the Rayleigh quotient *without any constraints on v*. By contrast, the sparsest cut minimizes the Rayleigh quotient subject to the constraint that $v \in \{0,1\}^n$. Proving Cheeger inequality essentially involves controlling the difference caused by real v vs binary v. We omit the proof of Cheeger's inequality because it's beyond the scope of this notes.

10.2.3 Applying spectral clustering

How do the ideas from the previous sections connect to our previous discussion of spectral clustering? Suppose that we have some raw data $x^{(1)} \cdots x^{(n)}$ that we'd like to group into k clusters. Ng et al. [2001] propose to define a weighted graph G such that each element

$$G_{ij} = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|_2^2}{2\sigma^2}\right) \tag{10.97}$$

represents a distance between two data points. Then, we compute the first k eigenvectors of the Laplacian L and arrange them into the columns of a matrix:

$$\begin{bmatrix} | & & | \\ u_1 & \cdots & u_k \\ | & & | \end{bmatrix} \in \mathbb{R}^{n\times k}. \tag{10.98}$$

The i-th row of this matrix (which we denote by v_i) is then a k-dimensional embedding of the i-th example. Note that this is usually a much lower-dimensional representation than the raw data. Finally, we can use k-means to cluster the embeddings $\{v_1, \ldots, v_n\}$.

In high dimensions, the issue with Ng et al. [2001]'s approach is that the training data points are all far away from each other. The Euclidean distance between points becomes meaningless, and so our definition of G does not make sense.

How do we solve this issue? HaoChen et al. [2021] propose a solution. They consider an infinite weighted graph $G(V, w)$, where w are the edge weights, and $V = \mathcal{X} \subseteq \mathbb{R}^n$ is the set of all possible data points. We define $w(x, x')$ to be large only if x and x' are very close in ℓ_2 distance. Now, the graph is more meaningful, because only data points that are small perturbations of each other have high connection weights. However, we do not have explicit access to this graph, and its eigenvectors are infinite-dimensional.

Now, suppose we have some eigenvector $u = (u_x)_{x\in\mathcal{X}}$. Rather than explicitly represent u_x, we represent u_x by $f_\theta(x)$ where f_θ is some parameterized model. Now, the challenge is to find θ such that $[f_\theta(x)]_{x\in\mathcal{X}}$ is the second smallest eigenvector of Laplacian of G. It turns out solving this problem gives a form of contrastive learning, which we will discuss in Section 10.3.2.

10.3 Self-supervised Learning

10.3.1 Pretraining / self-supervised learning / foundation model basics

Self-supervised learning is widely used for pretraining modern models. These large pretrained models are also called foundation models [Bommasani et al., 2021]. One simplified setup / workflow contains the following two stages:

Pretraining on unlabeled, massive data. Let $\{x^{(1)}, \cdots, x^{(n)}\}$ be a dataset where $x^{(i)} \in \mathbb{R}^d$ is sampled from some pretraining data distribution $x^{(i)} \sim P_{\text{pre}}$. The goal is to learn a pretrained model $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$, where k is the dimension of features / representations / embeddings, and θ is the model parameter. This model can be learned by minimizing certain pretrained loss function: $\hat{L}_{\text{pre}}(\theta) = \hat{L}_{\text{pre}}(x^{(1)}, \cdots, x^{(n)}; \theta)$, which sometimes is of the form $\hat{L}_{\text{pre}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell_{\text{pre}}(x^{(i)}; \theta)$. We use $\hat{\theta} = \text{argmin}_\theta \hat{L}_{\text{pre}}(\theta)$ to denote the parameter learned during pretraining.

Adaptation. During adaptation, we have access to a set of labeled downstream task examples $\{(x_{\text{task}}^{(1)}, y_{\text{task}}^{(1)}), \cdots, (x_{\text{task}}^{(n_{\text{task}})}, y_{\text{task}}^{(n_{\text{task}})})\}$, where usually $n_{\text{task}} \ll n$. One popular adapataion method is *linear probe*, which uses $f_{\hat{\theta}}(x)$ as features / representations / embeddings, and train a linear classifier on downstream tasks. Concretely, the prediction on data x is $w^\top f_{\hat{\theta}}(x)$, where w is the linear head learned from $\min_w \hat{L}_{\text{task}}(w) = \frac{1}{n_{\text{task}}}\sum_{i=1}^{n_{\text{task}}} \ell_{\text{task}}(y_{\text{task}}^{(i)}, w^\top f_{\hat{\theta}}(x_{\text{task}}^{(i)}))$. Another popular adaptation method is *finetuning*, which also updates the parameter θ. Concretely, one initializes from $\theta = \hat{\theta}$, and solve $\min_{\theta,w} \hat{L}_{\text{task}}(w, \theta) = \frac{1}{n_{\text{task}}}\sum_{i=1}^{n_{\text{task}}} \ell_{\text{task}}(y_{\text{task}}^{(i)}, w^\top f_\theta(x_{\text{task}}^{(i)}))$.

Why does pretraining on unlabeled data with an unsupervised (self-supervised) loss help a wide range of downstream prediction tasks? There are many open questions to be answered in this field. For instance, we may ask: (1) how pretraining helps label efficiency of downstream tasks, (2) why pretraining can give "universal" representations, and (3) why does pretraining provide robustness to distribution shift.

10.3.2 Analysis of contrastive learning

Let $\bar{X}$ be the set of all natural images in the image domain, $\bar{P}_{\bar{X}}$ be the distribution over $\bar{X}$. Contrastive learning learns f_θ such that representations of augmentations of the same image are close, whereas augmentations of random images are far away (as visualized in Figure 10.4).

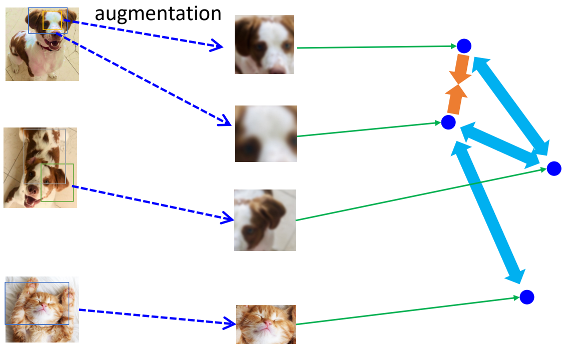


Figure 10.4: A demonstration of contrastive learning. Representations of augmentations of the same image are pulled close, whereas augmentations of random images are pushed far away.

Given a natural image $\bar{x} \in \bar{X}$, one can generate augmentations by random cropping, flipping, adding Gaussian noise or color transformation. We use $x \sim \mathcal{A}(\cdot|\bar{x})$ to denote the conditional distribution of augmentations given the natural image. For simplicity, we consider the case where Gaussian blurring is the augmentation, we have

$$x \sim \mathcal{A}(\cdot|\bar{x}) \Leftrightarrow x = \bar{x} + \xi \qquad\qquad \xi \sim \mathcal{N}(0, \sigma^2 \mathcal{I}), \tag{10.99}$$

where $\|\xi\|_2 \approx \sigma\sqrt{d}$ should be $\ll \|\bar{x}\|$.

We say (x, x^+) is a *positive pair* if they are generated as follows: first sample $\bar{x} \sim \bar{P}_{\bar{X}}$, then sample $x \sim \mathcal{A}(\cdot|\bar{x})$ and $x^+ \sim \mathcal{A}(\cdot|\bar{x})$ independently. In other words, (x, x^+) are augmentations of the same natural image.

We say (x, x') is a *random pair / negative pair* if they are sampled as: first sample two natural images $\bar{x} \sim \bar{P}_{\bar{X}}$ and $\bar{x}' \sim \bar{P}_{\bar{X}}$, then sample augmentations $x \sim \mathcal{A}(\cdot|\bar{x})$ and $x' \sim \mathcal{A}(\cdot|\bar{x}')$.

The design principle for contrastive learning is to find θ such that $f_\theta(x)$, $f_\theta(x^+)$ are close, while $f_\theta(x), f_\theta(x')$ are far away [Chen et al., 2020, Zbontar et al., 2021, He et al., 2020].

One example of contrastive learning is SimCLR [Chen et al., 2020]. Given B positive pairs $(x^{(1)}, x^{(1)+}), \cdots, (x^{(B)}, x^{(B)+})$, note that $(x^{(i)}, x^{(j)+})$ is a random pair if $i \neq j$, SimCLR defines the loss on the i-th pair as

$$\text{loss}_i = -\log \frac{\exp(f_\theta(x^{(i)})^\top f_\theta(x^{(i)+}))}{\exp(f_\theta(x^{(i)})^\top f_\theta(x^{(i)+})) + \sum_{j \neq i} \exp(f_\theta(x^{(i)})^\top f_\theta(x^{(j)+}))}. \tag{10.100}$$

Notice that $-\log \frac{A}{A+B}$ is decreasing in A but increasing in B, the loss above encourages $f_\theta(x^{(i)})^\top f_\theta(x^{(i)+})$ to be large, while $f_\theta(x^{(i)})^\top f_\theta(x^{(j)+})$ to be small.

In the rest of this section, we consider a variant of contrastive loss, proposed in [HaoChen et al., 2021]:

$$L(\theta) = -2 \mathop{\mathbb{E}}_{(x,x^+)\sim\text{positive}} f_\theta(x)^\top f_\theta(x^+) + \mathop{\mathbb{E}}_{(x,x')\sim\text{random}} \left(f_\theta(x)^\top f_\theta(x')\right)^2. \tag{10.101}$$

This contrastive loss follows the same design principle as other contrastive losses in the literature: suppose all representations have the same norm, then the first term encourages the representations of a positive pair to be closer while the second term encourages the representations of a random pair to be orthogonal to each other (hence far away). [HaoChen et al., 2021] show that the loss function, though inspired by theoretical derivations, still perform competitively, nearly matching the SOTA methods.

We can also define the empirical loss on a set of tuples $(x^{(i)}, x^{+(i)}, x'^{(i)})$ sampled i.i.d. as follows: $\bar{x} \sim \bar{P}_{\bar{X}}, x^{(i)} \sim \mathcal{A}(\cdot|\bar{x}^{(i)}), x^{+(i)} \sim \mathcal{A}(\cdot|\bar{x}^{(i)})$, $\bar{x}' \sim \bar{P}_{\bar{X}}$, $x'^{(i)} \sim \mathcal{A}(\cdot|\bar{x}'^{(i)})$. The empirical loss is defined as

$$\hat{L}(\theta) = \sum_{i=1}^{n} \left[-2 f_\theta(x^{(i)})^\top f_\theta(x^{+(i)}) + \left(f_\theta(x^{(i)})^\top f_\theta(x'^{(i)})\right)^2\right]. \tag{10.102}$$

Then the empirically learned parameter is $\hat{\theta} = \min_\theta \hat{L}(\theta)$.

Suppose the downatream task is binary classification with label set $\{1, -1\}$. We define the downstream loss as

$$\hat{L}_{\text{task}}(w, \theta) = \frac{1}{n_{\text{task}}} \sum_{i=1}^{n_{\text{task}}} \frac{1}{2} \left(y_{\text{task}}^{(i)} - w^\top f_\theta(x_{\text{task}}^{(i)})\right)^2. \tag{10.103}$$

We learn the linear head $\hat{w} = \operatorname{argmin}_w \hat{L}_{\text{task}}(w, \hat{\theta})$, and the evaluate its performance on downstream population data:

$$L_{\text{task}}(\hat{w}, \hat{\theta}) = \mathbb{E}\left[\frac{1}{2}\left(y_{\text{task}} - \hat{w}^\top f_\theta(x_{\text{task}})\right)^2\right]. \tag{10.104}$$

Analysis pipeline. We give a summary of our analysis pipeline below. The key takeaway is that we only have to focus on the population distribution case (step 3).

0. Assume expressivity, i.e., assuming $\exists \theta^*$ such that $L(\theta^*)$ is sufficiently small (the details will be quantified later).

1. For large enough n (e.g., $n > \text{Comp}(\mathcal{F})/\epsilon^2$ where $\mathcal{F} = \{f_\theta\}$ is the function class, $\text{Comp}(\cdot)$ is some measure of complexity, ϵ is the target error), show that $\hat{L}(\theta) = L(\theta) \pm \epsilon$.

2. Let $\hat{\theta}$ be the parameter learned on empirical data. Since $\hat{L}(\hat{\theta}) = \min_\theta \hat{L}(\theta) \leq \hat{L}(\theta^*) \leq L(\theta^*) + \epsilon$, we have
$$\hat{L}(\hat{\theta}) \leq \epsilon \Rightarrow L(\hat{\theta}) \leq 2\epsilon \tag{10.105}$$

3. **Key step:** (infinite data case) We will prove a theorem (Theorem 10.12 below as a simplified version, or Theorem 3.8 of HaoChen et al. [2021]) that shows if $L(\hat{\theta}) \leq 2\epsilon$, then there exists w such that $L_{\text{task}}(\theta, w) \leq \delta$, where δ is a function of ϵ and data distribution $\bar{P}$.

4. When we have enough downstream data $n_{\text{task}} \geq \text{poly}(k, \frac{1}{\epsilon'})$, for any θ, with high probability we have (via uniform convergence) that for any w, $\hat{L}_{\text{task}}(w, \theta) \approx L_{\text{task}}(w, \theta) \pm \epsilon'$.

5. Using the results in step 3 and step 4, we have $\hat{L}_{\text{task}}(\hat{w}, \hat{\theta}) = \min_w \hat{L}_{\text{task}}(w, \hat{\theta}) \leq \min_w L_{\text{task}}(w, \hat{\theta}) + \epsilon' \leq \delta + \epsilon'$. Thus, the final evaluation loss on the downstream task is $L_{\text{task}}(\hat{w}, \hat{\theta}) \leq \hat{L}_{\text{task}}(\hat{w}, \hat{\theta}) + \epsilon' \leq \delta + 2\epsilon'$.

Key step: the case with population pretraining and downstream data. We will now dive into the analysis of step 3, as all the other steps are from standard concentration inequalities. Recall that
$$L(\theta) = -2 \mathop{\mathbb{E}}_{(x,x^+)} f_\theta(x)^\top f_\theta(x^+) + \mathop{\mathbb{E}}_{(x,x')} \left(f_\theta(x)^\top f_\theta(x')\right)^2. \tag{10.106}$$

As expected, the analysis requires structural assumptions on the data. In particular, we will use the graph-theoretic language to describe the assumptions on population data. Let X be the set of all augmented data, P be the distribution of augmented data $x \sim \mathcal{A}(\cdot|\bar{x})$ where $\bar{x} \sim \bar{P}_{\bar{X}}$. Let $p(x, x^+)$ be the probability density of positive pair (x, x^+). We define a graph $G(V, w)$ where vertex set $V = X$ and edge weights $w(x, z) = p(x, z)$ for any $(x, z) \in X \times X$. In general, this graph may be infinitely large. To simplify math and avoid integrals, we assume $|X| = N$ where N is the number of all possible augmented images (which can be infinite or exponential in dimension).

The degree of node x is $p(x) = \sum_{z \in X} p(x, z)$. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix of this graph defined as $A_{x,z} = p(x, z)$, and let $\bar{A}$ ber the normalized adjacency matrix such that $\bar{A}_{x,z} = \frac{p(x,z)}{\sqrt{p(x)p(z)}}$.

The following lemma shows that contrastive learning is closely related to the eigendecomposition of $\bar{A}$.

Lemma 10.9. *Let $L(f) = -2\,\mathbb{E}_{(x,x^+)}\, f(x)^\top f(x^+) + \mathbb{E}_{(x,x')} \left(f(x)^\top f(x')\right)^2$. Suppose $X = \{x_1, \cdots, x_N\}$, let matrix*
$$F = \begin{bmatrix} p(x_1)^{\frac{1}{2}} f(x_1)^\top \\ \vdots \\ p(x_N)^{\frac{1}{2}} f(x_N)^\top \end{bmatrix}. \tag{10.107}$$

Then,
$$L(f) = \|\bar{A} - FF^\top\|_F^2 + \textit{const.} \tag{10.108}$$

Hence, minimizing $L(f)$ w.r.t the variable f is equivalent to eigendecomposition of $\bar{A}$.

Proof. Directly expanding the Frobenius norm $\|\bar{A} - FF^\top\|_F^2$ as a sum over entries, we have
$$\|\bar{A} - FF^\top\|_F^2 = \sum_{x,z \in X} \left(\frac{p(x,z)}{\sqrt{p(x)}\sqrt{p(z)}} - f(x)^\top f(z) \sqrt{p(x)}\sqrt{p(z)} \right)^2 \tag{10.109}$$
$$= \text{const} - 2 \sum_{x,z \in X} p(x,z) f(x)^\top f(z) + \sum_{x,z \in X} p(x)p(z) \left(f(x)^\top f(z)\right)^2 \tag{10.110}$$
$$= \text{const} - 2 \mathop{\mathbb{E}}_{(x,x^+) \sim \text{positive}} f(x)^\top f(x^+) + \mathop{\mathbb{E}}_{(x,x') \sim \text{random}} \left(f(x)^\top f(x')\right)^2, \tag{10.111}$$

where the last equation uses the fact that $p(x,z)$ and $p(x)p(z)$ are the probability densities of (x,z) being a positive pair and a random pair, respectively. □

Standard matrix decomposition results tell us that the minimizer of $\|\bar{A} - FF^\top\|_F^2$ satisfies $F = U \cdot \mathrm{diag}(\gamma_i^{\frac{1}{2}})$, where γ_i's are the eigenvalues of $\bar{A}$ and $U \in \mathbb{R}^{N \times k}$ contains the top k eigenvectors of $\bar{A}$ as its columns. Suppose we use $v_1, \cdots, v_N$ to represent the rows of U, i.e.,

$$U = \begin{bmatrix} v_1^\top \\ \vdots \\ v_N^\top \end{bmatrix}. \tag{10.112}$$

Then we know $f(x_j) = p(x_j)^{-\frac{1}{2}} \cdot \mathrm{diag}(\gamma_i^{\frac{1}{2}}) \cdot v_j$ is the minimizer of the contrastive loss.

One interesting thing is that $f(x_i)$ has the same separability as v_i. This is because for any vector $b \in \mathbb{R}^k$, we have $\mathbb{1}\left[b^\top v_i > 0\right] = \mathbb{1}\left[b^\top \mathrm{diag}(\gamma_i^{-\frac{1}{2}}) f(x) > 0\right]$, suggesting linear head $\mathrm{diag}(\gamma_i^{-\frac{1}{2}})b$ applied on feature $f(x_i)$ would achieve the same classification accuracy as v applied on v_i. Thus, it suffices to analyze v_i's downstream accuracy under linear head.

Since v_i is exactly the feature used by the classic spectral clustering algorithm, we may ask when spectral clustering produces good features. As discussed in Section 10.2, spectral clustering is good at graph partitioning in stochastic block models. In this section, we aim to find more general settings where spectral clustering produces good features. For simplicity, let's consider a regular graph where $w(x) = \sum_{x' \in V} w(x, x') = \kappa$.[3]

The following lemma shows that suppose the graph roughly contains two clusters, then the spectral clustering features can be used to accurately predict which cluster a node belongs to.

Lemma 10.10. *Suppose the graph G can be partitioned into 2 clusters S_1, S_2 with size $|S_1| = |S_2| = \frac{N}{2}$, such that $E(S_1, S_2) = \sum_{x \in S_1, z \in S_2} w(x,z) \le \alpha\kappa N$. Furthermore, suppose G cannot be partitioned well into 3 clusters in the sense that for all partition T_1, T_2, T_3, we have $\max\{\phi(T_1), \phi(T_2), \phi(T_3)\} \ge \rho$. (Figure 10.5 gives a demonstration of these assumptions.) Then, let $g = \mathbb{1}(S_1) \in \mathbb{R}^N$ (i.e., $g_i = 1$ if $i \in S_1$), and $k \ge 6$,*

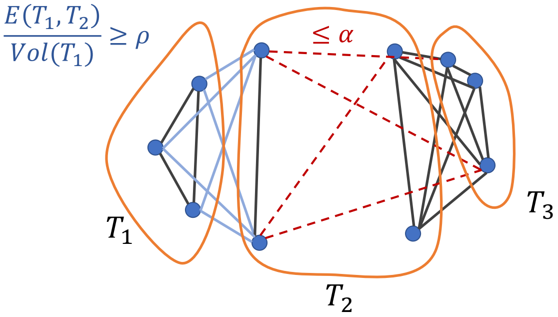


Figure 10.5: A demonstration of the assumptions in Lemma 10.10. The left half and right half of the graph can be chosen as S_1 and S_2, since there's at most α proportion of edges between them. Sets T_1, T_2, T_3 form a 3-way partition where $\phi(T_1) \ge \rho$.

there exists linear classifier b such that

$$\|Ub - g\|_2^2 \lesssim \frac{N\alpha}{\rho^2}, \tag{10.113}$$

where U contains the top k eigenvectors of $\bar{A}$ as its columns.

[3] It turns out that most, if not all, spectral graph theory tools on regular graph can extend to general graph settings. Therefore, it oftentimes suffices to consider a regular graph.

The above lemma essentially says that $\langle v_x, b\rangle \approx g_x$ for all data $x \in X$, where v_x is the x-th row of U.

Before proving the above lemma, we first introduce the following higher-order Cheeger inequality, which shows that when the graph cannot be partitioned well into 3 clusters, the 6-th smalled eigenvalue of the Laplacian cannot be too small.

Lemma 10.11 (Proposition 1.2 in [Louis and Makarychev, 2014])**.** *Let $G = (V, w)$ be a weight graph. Suppose the graph cannot be partitioned into 3 sets S_1, S_2, S_3 such that $\max\{\phi(S_1), \phi(S_2), \phi(S_3)\} \le \rho$. Then, we have*

$$\lambda_6 \gtrsim \rho^2.$$

Now we give a proof of Lemma 10.10.

Proof of Lemma 10.10. By Lemma 10.8 we know that

$$\frac{2}{N} g^\top L g = \frac{1}{N\kappa}\sum_{x,z}(g_x - g_z)^2 w(x,z) \tag{10.114}$$

$$= \frac{1}{N\kappa}\left(\sum_{x\in S_1, z\in S_2} w(x,z) + \sum_{x\in S_2, z\in S_1} w(x,z)\right) \tag{10.115}$$

$$= \frac{2}{N\kappa} E(S_1, S_2) \tag{10.116}$$

$$\le \alpha. \tag{10.117}$$

Thus, g has to be mostly in the smaller eigenspace of L. Suppose L has eigenvalue $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N$, with corresponding eigenvectors $u_1, u_2, \cdots u_N$. Define matrix $U = [u_1, \cdots, u_k] \in \mathbb{R}^{N\times k}$. Suppose $\sqrt{\frac{2}{N}} g = \sum_{i=1}^N \beta_i u_i$. Since $\|\sqrt{\frac{2}{N}} g\| = 1$, we know $\sum_{i=1}^N \beta_i^2 = 1$.

Since we know $g^\top L g = \sum_{i=1}^N \beta_i^2 \lambda_i \le \frac{N\alpha}{2}$, we can conclude $\sum_{i>k}\beta_i^2\lambda_i \le \frac{N\alpha}{2}$, which implies that $\sum_{i>k}\beta_i^2 \le \frac{N\alpha}{2\lambda_{k+1}} \lesssim \frac{N\alpha}{\rho^2}$. Here we used the fact $\lambda_6 \gtrsim \rho^2$ by higher-order Cheeger inequality (Lemma 10.11). Thus, we have $\|g - \sum_{i=1}^k \beta_i u_i\|_2^2 = \|\sum_{i>k}\beta_i u_i\|_2^2 \lesssim \frac{N\alpha}{\rho^2}$ which finishes the proof. □

We can combine Lemma 10.9 and Lemma 10.10 to prove the following theorem, which shows that when the graph roughly contains 2 clusters, the feature learned from contrastive learning can be used to predict the cluster membership accurately.

Theorem 10.12. *Let $L(f) = -2\,\mathbb{E}_{(x,x^+)} f(x)^\top f(x^+) + \mathbb{E}_{(x,x')}\left(f(x)^\top f(x')\right)^2$, and $f^* : X \to \mathbb{R}^k$ is a minimizer of $L(f)$ for $k \ge 6$. Suppose the graph G can be partitioned into 2 clusters S_1, S_2 with size $|S_1| = |S_2| = \frac{N}{2}$, such that $E(S_1, S_2) = \sum_{x\in S_1, z\in S_2} w(x,z) \le \alpha\kappa N$. Furthermore, suppose G cannot be partitioned well into 3 clusters in the sense that for all partition T_1, T_2, T_3, we have $\max\{\phi(T_1), \phi(T_2), \phi(T_3)\} \ge \rho$. Let $y(x_i) = \mathbb{1}(x_i \in S_1)$ (i.e., $y(x_i) = 1$ if $x_i \in S_1$, otherwise $y(x_i) = 0$). Then, there exists linear classifier $b \in \mathbb{R}^k$ such that*

$$\frac{1}{N}\sum_{i\in[N]}\left(f(x_i)^\top b - y(x_i)\right)^2 \lesssim \frac{\alpha}{\rho^2}. \tag{10.118}$$

Proof. Let $U \in \mathbb{R}^{N\times k}$ contains the top k eigenvectors of $\bar{A}$ as its columns. By Lemma 10.10, we know there exists some $\hat{b} \in \mathbb{R}^k$ such that $\|U\hat{b} - g\|_2^2 \lesssim \frac{N\alpha}{\rho^2}$, where $g \in \mathbb{R}^N$ such that $g_i = y(x_i)$. Let $v_1, \cdots, v_N$ be the rows of U. According to Lemma 10.9, we know that $f(x_i) = p(x_i)^{-\frac{1}{2}} \cdot \mathrm{diag}(\gamma_j^{\frac{1}{2}}) \cdot v_i = \kappa^{-\frac{1}{2}} \cdot \mathrm{diag}(\gamma_j^{\frac{1}{2}}) \cdot v_i$, where γ_j is the j-th largest eigenvalue of $\bar{A}$, and $\mathrm{diag}(\gamma_j^{\frac{1}{2}})$ is a diagonal matrix containing $\gamma_1^{\frac{1}{2}}, \gamma_2^{\frac{1}{2}}, \cdots, \gamma_k^{\frac{1}{2}}$ as its entries. Thus, if we let $b = \sqrt{\kappa} \cdot \mathrm{diag}(\gamma_j^{-\frac{1}{2}}) \cdot \hat{b}$, we would have

$$\sum_{i \in [N]} (f(x_i)^\top b - y(x_i))^2 = \sum_{i \in [N]} (v_i^\top \hat{b} - g_i)^2 = \|U\hat{b} - g\|_2^2 \lesssim \frac{N\alpha}{\rho^2}. \tag{10.119}$$

□

Chapter 11

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.

11.1 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 11.1.

Algorithm 1: General online learning problem

```
for $t = 1, ...T$ 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 $\ell(y_t, \hat{y}_t)$.
```

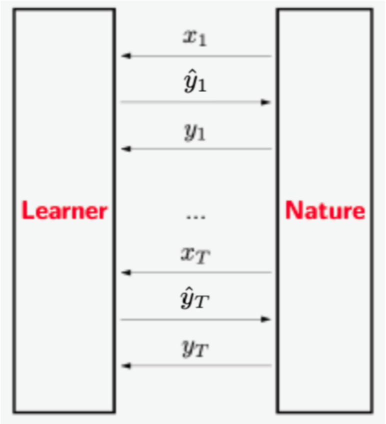


Figure 11.1: A representation of the online learning problem.

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.

11.1.1 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:

$$\sum_{t=1}^{T} \ell(y_t, \hat{y}_t). \tag{11.1}$$

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.

$$h^{\star} = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(y_t, h(x_t)). \tag{11.2}$$

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:

$$\text{Regret} \triangleq \left[\sum_{t=1}^{T} \ell(y_t, \hat{y}_t)\right] - \underbrace{\left[\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(y_t, h(x_t))\right]}_{\text{best loss in hindsight}} \tag{11.3}$$

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.

11.1.2 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 11.1 (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 11.2. 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, $1 \leq |V_{t+1}| \leq |\mathcal{H}|2^{-M}$ where M is the number of mistakes made so far. Thus, one has $M \leq \log|\mathcal{H}|$ 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

$$\text{Regret} = \sum_{t=1}^{T} \ell(y_t, h(x_t)) = M, \tag{11.4}$$

so in this example, one has regret $\leq \log|\mathcal{H}|$, 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.

11.2 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 $\Omega$;
    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:

$$\text{Regret} = \sum_{t=1}^{T} f_t(w_t) - \underbrace{\min_{w \in \Omega} \sum_{t=1}^{T} f_t(w)}_{\text{best action in hindsight}} . \tag{11.5}$$

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.)

11.2.1 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.)

11.3 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.

11.3.1 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 ↔ squared error loss functions for (x_t, y_t).
- w_t in online optimization ↔ 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.

11.3.2 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:

$$\text{Regret} \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \underbrace{\min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i)}_{\text{the best expert in hindsight}}\right], \tag{11.6}$$

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

```
for $t = 1, ..., T$ do
    The learner obtains predictions from $N$ experts;
    The learner chooses a distribution $p_t \in \Delta(N)$;
    The learner samples one expert $i_t \sim p_t$;
    The environment gives the learner the true value and the loss/error of all experts: $\ell_t \in \{0, 1\}^N$;
    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

```
for $t = 1, ..., T$ do
    The learner obtains predictions from $N$ experts;
    The learner invokes the OCO oracle to obtain $p_t \in \Delta(N)$;
    The learner chooses to follow prediction of one of the experts $i_t \in [N]$;
    The environment gives the learner the true value. The learner is thus able to learn the loss of
      each of the experts: $\ell_t \in \{0, 1\}^N$;
    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:

$$\text{Regret} = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i) \tag{11.7}$$

$$= \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{p\in\Delta(N)} \sum_{t=1}^{T} \langle p, \ell_t \rangle \tag{11.8}$$

$$= \sum_{t=1}^{T} f_t(p_t) - \min_{p\in\Delta(N)} \sum_{t=1}^{T} f_t(p). \tag{11.9}$$

We obtain (11.8) because

$$\sum_{t=1}^{T} \langle p, \ell_t \rangle = \left\langle p, \sum_{t=1}^{T} \ell_t \right\rangle \geq \min_{i\in[N]} \left[\sum_{t=1}^{T} \ell_t(i) \right], \tag{11.10}$$

with equality for the probability distribution $p(i) = 1$ when $i = \operatorname{argmin}_i \left[\sum_{t=1}^{T} \ell_t(i) \right]$ and $p(i) = 0$ otherwise, and (11.9) is by definition of f_t.

11.4 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 11.3 (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:*

$$L(\overline{w}) - L(w^\star) \leq \frac{R_T}{T} + \widetilde{O}\left(\frac{1}{\sqrt{T}}\right), \tag{11.11}$$

where $w^\star = \operatorname{argmin}_{w\in\Omega} L(w)$.

Here are some intuitive interpretations of the theorem:

- If $R_T = O(T)$, then we have some non-trivial result. Otherwise, the bound in (11.11) 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.

11.5 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.

$$w_t = \operatorname*{argmin}_{w \in \Omega} \sum_{i=1}^{t-1} f_i(w). \tag{11.12}$$

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

$$\begin{aligned} p_t &= \operatorname*{argmin}_{p \in \Delta(N)} \sum_{i=1}^{t-1} f_i(p) && (11.13) \\ &= \operatorname*{argmin}_{p \in \Delta(N)} \sum_{i=1}^{t-1} \langle \ell_i, p \rangle && (11.14) \\ &= \operatorname*{argmin}_{p \in \Delta(N)} \left\langle \sum_{i=1}^{t-1} \ell_i, p \right\rangle. && (11.15) \end{aligned}$$

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

$$\text{Regret} = \sum_{i=1}^{t-1} \ell_i(i_t), \quad \text{where } i_t = \operatorname*{argmin}_{j \in [N]} \sum_{i=1}^{t-1} \ell_i(j). \tag{11.16}$$

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 11.1.

Table 11.1: 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.

11.6 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 11.4. *The BTL strategy has non-positive regret. That, is, if w_t is defined as in the FTL algorithm, then*

$$\textit{BTL regret} = \sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq 0, \tag{11.17}$$

for any T and any sequence of functions $f_1, \cdots, f_T$.

Proof. We prove the lemma by induction on T. (11.17) holds trivially for $T = 1$. Suppose that (11.17) holds for all $t \leq T-1$ and any $f_1, \cdots, f_{T-1}$. Now we wish to extend (11.17) 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:

$$\begin{aligned}
\sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) &= \sum_{t=1}^{T} f_t(w_{t+1}) - \sum_{t=1}^{T} f_t(w_{T+1}) && (11.18)\\
&= \sum_{t=1}^{T-1} f_t(w_{t+1}) - \sum_{t=1}^{T-1} f_t(w_{T+1}) \qquad \text{(final summands cancel)} && (11.19)\\
&\leq \sum_{t=1}^{T-1} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T-1} f_t(w) && (11.20)\\
&\leq 0. \qquad \text{(induction hypothesis)} && (11.21)
\end{aligned}$$

□

A useful consequence of this lemma is a regret bound for the FTL strategy.

Lemma 11.5. (FTL regret bound) *Again, let w_t be as in the FTL algorithm. The FTL strategy has the regret guarantee*

$$\textit{FTL regret} = \sum_{t=1}^{T} f_t(w_t) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})]. \tag{11.22}$$

Proof.

$$\begin{aligned}
\text{FTL regret} &= \sum_{t=1}^{T} f_t(w_t) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) && (11.23)\\
&= \sum_{t=1}^{T} f_t(w_{t+1}) - \min_{w\in\Omega} \sum_{t=1}^{T} f_t(w) + \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})] && (11.24)\\
&\leq 0 + \sum_{t=1}^{T} [f_t(w_t) - f_t(w_{t+1})], && (11.25)
\end{aligned}$$

where the last inequality is due to (11.17).

□

Lemma 11.5 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.

11.7 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 11.6. 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

$$\phi(x) \geq \phi(y) + \langle \nabla f(y), x - y \rangle + \frac{\alpha}{2}\|x - y\|^2 \tag{11.26}$$

for any $x, y \in \Omega$.

Remark 11.7. 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 (11.26). This quadratic lower bound is very useful in proving theorems in optimization.

Remark 11.8. 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

$$w_t = \operatorname*{argmin}_{w \in \Omega} \left[\sum_{i=1}^{t-1} f_i(w) + \frac{1}{\eta}\phi(w) \right], \tag{11.27}$$

where $\eta > 0$ is a tuning parameter that we will tune later.

11.7.1 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 11.9. (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*

$$0 \leq f(w) - f(w') \leq \frac{1}{\alpha}\|\nabla f(w)\|_*^2, \tag{11.28}$$

where $\| \cdot \|_*$ is the dual norm of $\| \cdot \|$.

Proof. By strong convexity,

$$\begin{aligned} F(w') - F(w) &\geq \langle \nabla F(w), w' - w \rangle + \frac{\alpha}{2}\|w - w'\|^2 && (11.29) \\ &\geq \frac{\alpha}{2}\|w - w'\|^2, && (11.30) \end{aligned}$$

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:

$$[F(w) + f(w)] - [F(w') + f(w')] \geq \frac{\alpha}{2}\|w - w'\|^2. \tag{11.31}$$

Adding these two inequalities gives

$$f(w) - f(w') \geq \alpha\|w - w'\|^2. \tag{11.32}$$

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 (11.32), we obtain

$$\|w - w'\| \leq \sqrt{\frac{1}{\alpha}[f(w) - f(w')]}. \tag{11.33}$$

Since f is convex, we have $f(w') \geq f(w) + \langle \nabla f(w), w' - w\rangle$. Rearranging this gives

$$\begin{aligned} f(w) - f(w') &\leq \langle \nabla f(w), w - w'\rangle \\ &\leq \|\nabla f(w)\|_* \cdot \|w - w'\| && \text{(by Cauchy-Schwarz)} \\ &\leq \|\nabla f(w)\|_* \sqrt{\frac{1}{\alpha}[f(w) - f(w')]}. && \text{(by (11.33))} \end{aligned}$$

Since $f(w) - f(w') \geq 0$, we can square both sides of this inequality to conclude that

$$[f(w) - f(w')]^2 \leq ||\nabla f(w)||_*^2 \frac{1}{\alpha}[f(w) - f(w')]. \tag{11.34}$$

Dividing both sides of this expression by $f(w) - f(w')$ gives the desired upper bound. □

Remark 11.10. 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 (11.28) become equalities.

11.7.2 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 11.11. (Regret of FTRL) *Let ϕ be a 1-strongly-convex regularizer with respect to the norm $\|\cdot\|$ on Ω. Then the FTRL algorithm (11.27) satisfies the regret guarantee*

$$\textit{FTRL regret} = \sum_{t=1}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \leq \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2, \tag{11.35}$$

where $D = \max_{w\in\Omega} \phi(w) - \min_{w\in\Omega} \phi(w)$.

Remark 11.12. Suppose that for all t and w, we have the uniform bound $||\nabla f_t(w)||_* \leq G$. Then Theorem 11.11 implies that the regret is upper bounded by $D/\eta + \eta GT$. Optimizing this upper bound over η by taking $\eta = \sqrt{\frac{D}{TG^2}}$ gives the guarantee

$$\text{FTRL regret} \leq 2\sqrt{DG} \times \sqrt{T}. \tag{11.36}$$

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

$$w_t = \operatorname*{argmin}_{w\in\Omega} \sum_{i=0}^{t-1} f_i(w), \tag{11.37}$$

i.e. FTRL is just FTL with an additional "round" of play at time zero. Thus, by Lemma 11.5 with time starting from $t = 0$, we have

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \le \sum_{t=0}^{T} [f_t(w_t) - f_t(w_{t+1})]. \tag{11.38}$$

For any $t \ge 1$, applying Lemma 11.9 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:

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \le f_0(w_0) - f_0(w_1) + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2. \tag{11.39}$$

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,

$$\sum_{t=0}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=0}^{T} f_t(w) \ge \sum_{t=0}^{T} f_t(w_t) - \sum_{t=0}^{T} f_t(w^*) \tag{11.40}$$

$$= f_0(w_0) - f_0(w^*) + \underbrace{\left(\sum_{t=1}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \right)}_{\text{Regret of FTRL}}. \tag{11.41}$$

Combining this inequality with (11.39) gives

$$\text{Regret of FTRL} \le f_0(w_0) - f_0(w_1) + f_0(w^*) - f_0(w_0) + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2 \tag{11.42}$$

$$= \frac{\phi(w^*) - \phi(w_1)}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2 \tag{11.43}$$

$$\le \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2. \tag{11.44}$$

This concludes the proof of the theorem. □

11.7.3 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

$$\nabla f_t(w) = -(y_t - w^\top x_t)x_t, \tag{11.45}$$

$$\|\nabla f_t(w)\|_2 \le |y_t - w^\top x_t| \cdot \|x_t\|_2 \tag{11.46}$$

$$\le 2 \cdot 1 = 2. \tag{11.47}$$

Therefore, by choosing $\eta = \sqrt{1/(8T)}$ and applying the FTRL regret theorem (Theorem 11.11), we can obtain the regret guarantee

$$\sum_{t=1}^{T}(y_t - w_t^\top x_t)^2 - \min_{||w||_2 \le 1} \sum_{t=1}^{T}(y_t - w^\top x_t)^2 \le 4\sqrt{T}. \tag{11.48}$$

11.7.4 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,

$$\begin{aligned} D &= \max_{p\in\Delta(N)} \phi(p) - \min_{p\in\Delta(N)} \phi(p) && (11.49)\\ &\le \max_{p\in\Delta(N)} \frac{1}{2}\|p\|_2^2 && (11.50)\\ &\le \max_{p\in\Delta(N)} \frac{1}{2}\|p\|_1^2 && (11.51)\\ &= \frac{1}{2}. && (11.52)\end{aligned}$$

Also,

$$\|\nabla f_t\|_2 = \|\ell_t\|_2 \le \sqrt{N}. \tag{11.53}$$

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*:

$$\phi(p) = -H(p) = \sum_{j=1}^{N} p(j) \log p(j), \tag{11.54}$$

where $p \in \Delta(N)$ is in the set of distributions over $[N]$. We first introduce the following nice property of this regularizer:

Lemma 11.13. *$\phi(p)$ 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 $p, q \in \Delta(N)$,

$$\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle \ge \frac{1}{2}\|p - q\|_1^2. \tag{11.55}$$

From direct computation, we know the gradient of $\phi(q)$ is

$$\nabla\phi(q) = \begin{bmatrix} 1 + \log q(1) \\ \cdots \\ 1 + \log q(N) \end{bmatrix}. \tag{11.56}$$

Plugging this into the LHS of (11.55), we get

$$\begin{aligned}
&\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle && (11.57)\\
=&\sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} q(j)\log q(j) - \sum_{j=1}^{N} (1 + \log q(j))\,(p(j) - q(j)) && (11.58)\\
=&\sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} p(j)\log q(j) - \sum_{j=1}^{N} (p(j) - q(j)) && (11.59)\\
=&\sum_{j=1}^{N} p(j)\log\frac{p(j)}{q(j)} && (11.60)\\
=&KL(p||q), && (11.61)
\end{aligned}$$

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 (11.60).) 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 (11.36), 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

$$D = \sup\phi(p) - \inf\phi(p) \leq -\inf\phi(p) = -\inf(-H(p)) = \sup(H(p)) \leq \log N, \tag{11.62}$$

and we bound G with

$$G = \|\nabla f_t(w)\|_\infty = \|l_t\|_\infty \leq 1. \tag{11.63}$$

Plugging these two into the regret bound (11.36) 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

$$p_t = \operatorname*{argmin}_{p\in\Delta(N)} \sum_{i=1}^{t+1} f_t(p) - \frac{1}{\eta}H(p). \tag{11.64}$$

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 11.14 (Gibbs variational principle)**.** *Let ν, μ be probability distributions on $[N]$. Then*

$$\sup_{\nu}\Big(\mathop{\mathbb{E}}_{\nu}[f] - KL(\nu||\mu)\Big) = \log\mathop{\mathbb{E}}_{\mu}\left[e^f\right], \tag{11.65}$$

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

$$\nu(x) \propto \mu(x)\cdot e^{f(x)}. \tag{11.66}$$

Intuitively, Lemma 11.14 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 11.14, we get the following corollary:

Corollary 11.15. 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

$$KL(\nu||\mu) = \sum_x \nu(x) \log \frac{\nu(x)}{\mu(x)} \tag{11.67}$$

$$= \log N - \sum_x \nu(x) \log \frac{1}{\nu(x)} \tag{11.68}$$

$$= \log N - H(\nu). \tag{11.69}$$

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 11.14 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

$$\underset{p_t \in \Delta(N)}{\operatorname{argmin}} \left\langle \sum_{i=1}^{t-1} l_i, p_t \right\rangle - \frac{1}{\eta} H(p_t) = \underset{p_t \in \Delta(N)}{\operatorname{argmin}} \left\langle \eta \sum_{i=1}^{t-1} l_i, p_t \right\rangle - H(p_t), \tag{11.70}$$

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 11.15 that the minimizer is attained at $p_t \propto \exp\left(-\eta \sum_{i=1}^{t-1} l_i\right)$, or equivalently,

$$p_t(j) = \frac{\exp(-\eta L_t(j))}{\sum_{k=1}^N \exp(-\eta L_t(k))}, \tag{11.71}$$

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 $\tilde{p}_t$:

$$p_t = \frac{\tilde{p}_t}{\|\tilde{p}_t\|_1}. \tag{11.72}$$

11.8 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 11.8.1 (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

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) \le \gamma(T). \tag{11.73}$$

Since f_t is convex, we also know that

$$g_t(w_t) - g_t(w) = \langle \nabla f_t(w_t), w_t - w\rangle \ge f_t(w_t) - f_t(w). \tag{11.74}$$

Therefore, for all $w \in \Omega$,

$$\begin{aligned}\sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w) &\le \sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) &(11.75)\\ &\le \gamma(T). &(11.76)\end{aligned}$$

Hence, the regret for $\mathcal{A}'$ is upper bounded by $\gamma(T)$ as well. □

11.8.1 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:

$$\begin{aligned}w_t &= \operatorname{argmin} \sum_{i=1}^{t-1} g_i(w) + \frac{1}{\eta}\|w\|_2^2 &(11.77)\\ &= \operatorname*{argmin}_{w\in\Omega} \sum_{i=1}^{t-1} \langle \nabla f_i(w_i), w\rangle + \frac{1}{\eta}\|w\|_2^2 &(11.78)\\ &= \Pi_\Omega\left(-\eta \cdot \sum_{i=1}^{t-1} \nabla f_i(w_i)\right), &(11.79)\end{aligned}$$

where $\Pi_\Omega(\cdot)$ is the projection operator onto the set Ω.The last equality is because for any vector a, we have

$$\begin{aligned}\operatorname*{argmin}_{w\in\Omega} \langle a, w\rangle + \frac{1}{\eta}\|w\|_2^2 &= \operatorname*{argmin}_{w\in\Omega} \frac{1}{2\eta}\|w + \eta a\|_2^2 - \eta\|a\|_2^2 &(11.80)\\ &= \operatorname*{argmin}_{w\in\Omega} \|w + \eta a\|_2^2 &(11.81)\\ &= \operatorname*{argmin}_{w\in\Omega} \|w - (-\eta a)\|_2^2 &(11.82)\\ &= \Pi_\Omega(-\eta a). &(11.83)\end{aligned}$$

[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:

$$u_t = u_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{11.84}$$

$$w_t = \Pi_\Omega(u_t). \tag{11.85}$$

Similarly, we can define gradient descent with “eager” projection (which can get similar regret bounds):

$$u_t = w_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{11.86}$$

$$w_t = \Pi_\Omega(u_t). \tag{11.87}$$

This concludes our discussion of online learning in this course.

Bibliography

Animashree Anandkumar, Rong Ge, and Majid Janzamin. Learning overcomplete latent variable models through tensor methods. In *Proceedings of the Conference on Learning Theory (COLT), Paris, France*, 2015.

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.

Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arpit17a.html`.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90014-2. URL `http://dx.doi.org/10.1016/0893-6080(89)90014-2`.

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. *arXiv preprint arXiv:1904.09080*, 2019.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Kohd, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020.

Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 8, 2018.

Fan Chung. Four proofs for the cheeger inequality and graph partition algorithms. In *Proceedings of ICCM*, volume 2, page 378. Citeseer, 2007.

Alex Damian, Tengyu Ma, and Jason Lee. Label noise sgd provably prefers flat global minimizers, 2021.

Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of EM suffice for mixtures of two gaussians. *arXiv preprint arXiv:1609.00368*, 2016.

Simon Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. In *International Conference on Machine Learning*, pages 1655–1664. PMLR, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 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, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, June 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.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018.

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 and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16067.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. *arXiv preprint arXiv:1712.09203*, pages 2–47, 2017.

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.

Shiyu Liang, Ruoyu Sun, Jason D Lee, and R Srikant. Adding one neuron can eliminate all bad local minima. *Neural Information Processing Systems (NIPS)*, 2018.

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.

Anand Louis and Konstantin Makarychev. Approximation algorithm for sparsest k-partitioning. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1244–1255. SIAM, 2014.

Haipeng Luo. Introduction to online learning, 2017. URL https://haipeng-luo.net/courses/CSCI699/.

Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2016. URL http://arxiv.org/abs/1610.01980.

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.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14:849–856, 2001.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

John A. Rice. *Mathematical Statistics and Data Analysis.* Belmont, CA: Duxbury Press., third edition, 2006.

Tselil Schramm and David Steurer. Fast and robust tensor decomposition with applications to dictionary learning. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1760–1793. PMLR, 07–10 Jul 2017. URL `https://proceedings.mlr.press/v65/schramm17a.html`.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

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.

Matus Telgarsky. Deep learning theory lecture notes. `https://mjt.cs.illinois.edu/dlt/`, 2021. Version: 2021-10-27 v0.0-e7150f2d (alpha).

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. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. In *Advances in Neural Information Processing Systems*, pages 9722–9733, 2019a.

Colin Wei and Tengyu Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*, 2019b.

Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel, 2020.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. *arXiv preprint arXiv:2002.09277*, 2020.

Ji Xu, Daniel J Hsu, and Arian Maleki. Global analysis of expectation maximization for mixtures of two gaussians. In *Advances in Neural Information Processing Systems*, pages 2676–2684, 2016.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.