Information-theoretic generalisation bounds and applications to learning linear threshold functions

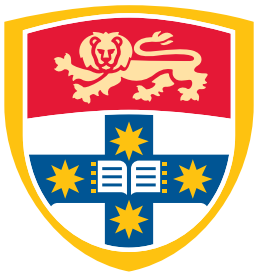
ALEXANDER TAN
SID: 490379282

Supervisor: Dr. Clément Canonne

This thesis is submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Advanced Computing (Computer Science) (Honours),
Bachelor of Science (Mathematics)

School of Computer Science
The University of Sydney
Australia

9 November 2023



THE UNIVERSITY OF
SYDNEY

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Alexander Tan

Signature: Alexander Tan **Date**: November 9, 2023

Abstract

A fundamental question in statistical learning theory is in determining whether a machine learning algorithm produces a model that generalises to the underlying distribution, as opposed to one that overfits to chance patterns that occur in the training data. We review some classical results in the literature that address this question including the Vapnik-Chervonenkis dimension and compression schemes, and summarise a recent line of work that addresses this question from an information-theoretic point of view. We apply this information-theoretic perspective in analysing a simple algorithm that learns linear threshold functions $f : \{-1,1\}^n \to \{-1,1\}$ by estimating their degree 0 and degree 1 Fourier coefficients from samples uniformly distributed over $\{-1,1\}^n$ and labelled by a linear threshold function f. In particular we show that this algorithm achieves an expected generalisation error of $O\left(\sqrt{\frac{n \log m}{m}}\right)$. We then show that a similar analysis can be applied to the algorithm of Linial et al. (1993) and that their algorithm has an expected generalisation bound of $O\left(\sqrt{\frac{|\mathcal{F}| \log m}{m}}\right)$ where $\mathcal{F}$ are subsets for which the Fourier weights of f are ε concentrated on. Finally, we consider a more general setting of learning linear threshold functions of the form $f : \mathbb{R}^n \to \{-1,1\}$ from samples drawn from the n-dimensional Gaussian density and labelled according to f. We again consider a simple algorithm that learns f by estimating its degree 0 and degree 1 Hermite coefficients, and show from the perspective of information theory that this algorithm has expected generalisation error $O\left(\sqrt{\frac{n}{m}}\right)$. We also show how to reconstruct f to within $O(\varepsilon)$ accuracy, given Hermite estimates that are within L_2 norm ε of the true Hermite coefficients.

Acknowledgements

First and foremost, I'd like to thank my supervisor Dr. Clément Canonne for guiding me along my honours journey, and more importantly, for being a super awesome and friendly person in general! I've learnt a lot over this past year, thanks to your guidance and advice. Also, I hope your pot plants grow well.

I'd also like to thank Associate Professor Joseph Lizier for your great course CSYS5030 on information theory and complex systems. It was very fascinating to see how information theory could be used in a totally different perspective to solve interesting problems. Thanks also for the interesting conversations we've had.

More generally I'd like to thank all my lecturers, tutors, friends, peers, and everyone else who have helped shape my experiences at university these past five years. It was not without its challenges — experiencing a pandemic in the middle was not part of the plan — but I'm proud that I've made it to the end in spite of it.

Finally, I'd like to thank Lillie, Oden, Marco, George, Ryan, James, and Ezra on the Discord server for helping me out on my thesis, whether you may have realised it or not. I'd also like to thank all my other friends and family for supporting me through this journey.

CONTENTS

List of Figures

CHAPTER 1

Introduction

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and statistical models that are able to learn and make predictions from data without explicitly being programmed to do so. In recent years, there has been impressive practical success in machine learning, particularly with certain deep learning models. However, how do we know if the training process outputs a model that is reflective of the underlying population that the training data was drawn from, as opposed to overfitting the chance patterns that may arise in the data? This is a fundamental question in the field of statistical machine learning.

A vast array of methods have been proposed in the literature to address this question. Most notably, the theory of Vapnik and Chervonenkis (1971) shows that if the output model is not "too complex" as formalised by the Vapnik-Chervonenkis (VC) dimension, then it cannot overfit too much in the sense that if a model perfectly fits the training data (i.e., it has zero empirical risk), then that model also has close to zero population risk, meaning that it will perform well even on data it has not seen before. A much more recent line of work (Xu and Raginsky, 2017; Bu et al., 2020; Steinke and Zakynthinou, 2020) has used the framework of information theory to bound the *expected generalisation error*, which is the expected difference between the empirical risk and population risk. In this thesis we study some applications of these information-theoretic results for learning linear threshold functions a.k.a. halfspaces, a fundamental family of classifiers.

1.1 Main results

We consider a simple algorithm that learns linear threshold functions (LTFs) $f : \{-1,1\}^n \to \{-1,1\}$ over the uniform Boolean hypercube $\mathcal{U}(\{-1,1\}^n)$ by estimating the degree 0 and degree 1 Fourier coefficients and using a result by O'Donnell and Servedio (2008) to reconstruct the LTF. In particular,

we show that this algorithm achieves an expected generalisation error of $O\left(\sqrt{\frac{n \log m}{m}}\right)$ where m is the number of samples (Theorem 14).

We show that a similar analysis can be applied to the algorithm of Linial et al. (1993) which leanrs Boolean functions whose Fourier spectrum is "concentrated" on relatively few coefficients. In particular, we show that their algorithm has an expected generalisation error of $O\left(\sqrt{\frac{|\mathcal{F}| \log m}{m}}\right)$ where $\mathcal{F}$ are the subsets for which the Fourier weights of f are ε concentrated on (Theorem 22).

Finally, we consider a more general setting of learning LTFs of the form $f : \mathbb{R}^n \to \{-1, 1\}$ over the n-dimensional Gaussian density. We again consider a simple algorithm that learns f by estimating its degree 0 and degree 1 Hermite coefficients, the Gaussian analogue of Fourier coefficients, and show that this algorithm has expected generalisation error $O\left(\sqrt{\frac{n}{m}}\right)$ (Theorem 25). We also show how to reconstruct an LTF to within $O(\varepsilon)$ accuracy, given Hermite estimates that are within L_2 norm ε of the true Hermite coefficients (Theorem 29).

1.2 Thesis outline

In Chapter 2, we formalise the setting of statistical learning theory and rigorously define what we mean by a good learning algorithm via the idea of "probably approximately correct" (PAC) learning introduced by Valiant (1984). We then review some classical results that can be used to determine the sample complexity required for PAC learning a hypothesis class, including VC dimension (Vapnik and Chervonenkis, 1971) and compression schemes (Littlestone and Warmuth, 1986).

In Chapter 3, we give a brief self-contained introduction to information theory, and summarise some recent work that uses information-theoretic analyses to bound the expected generalisation error of a learning algorithm (Russo and Zou, 2016; Xu and Raginsky, 2017; Bu et al., 2020; Steinke and Zakynthinou, 2020; Esposito et al., 2020a,b).

In Chapter 4, we review some properties of Boolean functions and their Fourier analysis, before applying the information-theoretic tools in the previous chapter to the algorithm described in Section 1.1 that learns linear threshold functions over $\{-1, 1\}^n$ by estimating the degree 0 and degree 1 Fourier coefficients. We show this algorithm achieves expected generalisation error $O\left(\sqrt{\frac{n \log m}{m}}\right)$.

In Chapter 5, we show how our analysis can also be applied to an algorithm by Linial et al. (1993). We derive a novel result that their algorithm, when learning a Boolean function f, has expected generalisation error $O\left(\sqrt{\frac{|\mathcal{F}| \log m}{m}}\right)$. We review some known applications of the algorithm by Linial et al. (1993) and discuss the expected generalisation error implied by our analysis in those applications.

In Chapter 6, we generalise the setup of learning LTFs to the continuous setting. In particular, the aim is to learn LTFs over $\mathbb{R}^n$ from samples drawn from the n-dimensional standard Gaussian density. We consider the algorithm that learns LTFs by estimating their degree 0 and degree 1 *Hermite* coefficients, and show that this learning algorithm has expected generalisation error $O\left(\frac{n}{m}\right)$. We also show how to reconstruct an LTF to within $O(\varepsilon)$ accuracy, given Hermite estimates that are within L_2 norm ε of the true Hermite coefficients.

We conclude our thesis in Chapter 7 and provide directions for future work.

1.3 Notation

We outline the notation we will be using in this thesis.

- We write $[n] := \{1, 2, \cdots, n\}$ and $\mathbb{N} := \{0, 1, 2, \cdots\}$.
- Random variables are typically denoted by capital letters, such as X, Y, Z, and realisations of random variables are typically denote in lowercase such as x, y, z.
- We write $X \sim p$ if the random variable X is distributed according to probability mass function or density p.
- We write "i.i.d." as shorthand for "independently and identically distributed". This is typically used for statements like: "let $X_1, \cdots, X_n \sim p$ be i.i.d." which means that the variables are mutually independent and are each distributed identically according to probability mass function or density p.
- We write $\mathop{\mathbf{E}}_{X \sim p}[X]$ for the expectation of the random variable X, where X is distributed according to probability mass function (or density) p. Where it is clear from context, we write $\mathop{\mathbf{E}}_{X}[X]$ or even $\mathbf{E}[X]$.
- Similarly, we write $\mathop{\mathbf{P}}_{X \sim p}[A]$ for the probability of event A occuring when X is distributed according to p, which we often write as $\mathop{\mathbf{P}}_{X}[A]$ or even $\mathbf{P}[A]$.
- The indicator random variable over the set A is written $\mathbb{1}\{A\}$.

- We use $\otimes$ to denote the product of two distributions. For example if $Z \sim p_X \otimes p_Y$ then $Z = (X, Y)$ is a random vector where $X \sim p_X$ independently of $Y \sim p_Y$. We overload this notation slightly so we can write something like $Z \sim p^{\otimes n}$ to mean that Z is a random vector of dimension n where each component is i.i.d. according to p.
- We write $\mathcal{N}(0, 1)$ for a standard normal random variable and more generally $\mathcal{N}(\mu, \sigma^2)$ for a normal random variable with mean μ and variance σ^2. We define $\varphi(x) := \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$ for the density of a $\mathcal{N}(0, 1)$ random variable, and $\Phi(x) := \int_{-\infty}^{x} \varphi(t)\, dt$ for its cumulative distribution function. The n-dimensional case is written $\varphi_n(x) := \frac{1}{\sqrt{2\pi^n}} \exp\left(-\frac{1}{2}x^T x\right)$.
- We write $\mathcal{U}[A]$ for the uniform distribution over the set A. For example $\mathcal{U}([-1, 1])$ is the continuous uniform distribution over the interval $[-1, 1]$ and $\mathcal{U}\left(\{-1, 1\}^n\right)$ is the discrete uniform distribution over the Boolean hypercube $\{-1, 1\}^n$.

Assume all sets are measurable.

Assume all functions are measurable with respect to counting measure or Lebesgue measure in the discrete or continuous cases respectively.

We will use the following abbreviations in this thesis.

- i.i.d. — independently and identically distributed
- p.m.f. — probability mass function
- p.d.f. — probability density function
- c.d.f. — cumulative distribution function
- PAC — probably approximately correct (see Chapter 2)
- ERM — empirical risk minimisation (see Chapter 2)
- VC dimension/theory — Vapnik Chervonenkis dimension/theory (see Chapter 2)
- DNF — disjunctive normal form (see Chapter 2)
- CNF — conjunctive normal form (see Chapter 2)
- SVM — support vector machine (see Chapter 2)
- KL divergence — Kullback-Leibler divergence (see Chapter 3)
- MI — mutual information (see Chapter 3)
- CMI — conditional mutual information (see Chapter 3)
- LTF — linear threshold function (see Chapter 4)
- PTF — polynomial threshold function (see Chapter 4)

- LMN algorithm — the algorithm by Linial, Mansour, and Nisan (1993) (see Chapter 5)

CHAPTER 2

Classical results in learning theory

2.1 Preliminaries

In the standard setting of statistical learning theory, we are given a **training set** consisting of m i.i.d. samples

$$Z = \left(Z^{(1)}, \cdots, Z^{(m)}\right) \tag{2.1}$$

with each $Z^{(i)}$ drawn i.i.d. from a fixed distribution $\mathcal{D}$ and of the form $Z^{(i)} := \left(X^{(i)}, Y^{(i)}\right)$. The $X^{(i)}$ are known as the **features** and range over the **feature space** $\mathcal{X}$. For each feature $X^{(i)}$ we are given an associated **label** $Y^{(i)}$ which range over the **label space** $\mathcal{Y}$. Let $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$.

In the simplest scenario, the relationship between the features $X^{(i)}$ and labels $Y^{(i)}$ is given by a deterministic but unknown function $c : \mathcal{X} \to \mathcal{Y}$ called a **concept**, so that $Y^{(i)} = c(X^{(i)})$ for all $i \in \{1, \cdots, m\}$. The aim is to learn what the function c is using the training set Z. The catch is that by attempting to determine what c is, you implicitly need to determine how it behaves on features that are *not contained in the training set*! This is of course in general impossible without making some assumptions on c, and so the assumption we will make is that c comes from a fixed predetermined class of functions $\mathcal{C}$ called the **concept class**.

A **learning algorithm** is a (possibly randomized) algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ that takes as input a training set Z, and outputs a guess $h : \mathcal{X} \to \mathcal{Y}$ of the target concept, which we call a **hypothesis**. Note that the set of all hypothesis concepts $\mathcal{H}$ that the algorithm may output, which we call the **hypothesis class**, may not necessarily coincide with the set of concepts $\mathcal{C}$ which the true concept resides in.

As an example, suppose our concept class $\mathcal{C}$ consists of threshold functions $I_\theta : [-1, 1] \to \{0, 1\}$ of the form $I_\theta : x \mapsto \mathbb{1}\{x \geq \theta\}$ for $\theta \in [-1, 1]$. Here the feature space is $\mathcal{X} = [-1, 1]$ and the label space is binary $\mathcal{Y} = \{0, 1\}$. Further suppose that the feature space distribution is uniform $\mathcal{D}_\mathcal{X} = \mathcal{U}([-1, 1])$. Then the task is: given samples $(X^{(1)}, Y^{(1)}), \cdots, (X^{(m)}, Y^{(m)})$ where $X^{(i)} \sim \mathcal{U}([-1, 1])$ i.i.d. and

$Y^{(i)} = I_{\theta^*}(X^{(i)})$ for an unknown $I_{\theta^*} \in \mathcal{C}$, try to determine the concept I_{θ^*}. We can't hope to predict the exact value of θ^* as it is an element of the reals, but we may hope to at least get close. An intuitive learning algorithm might be one that finds the smallest $X^{(i)}$ with a label of $Y^{(i)} = 1$ and output that $X^{(i)}$. This is an example of a learning algorithm where the hypothesis space $\mathcal{H}$ *does* agree with the concept space $\mathcal{C}$.

To measure how good the hypothesis output by a particular learning algorithm is, we introduce a **loss function** $\ell : \mathcal{H} \times \mathcal{Z} \to [0, \infty)$, which takes as input a learnt hypothesis $h \in \mathcal{H}$, as well as a particular feature vector and label pair $(x, y) \in \mathcal{Z}$, and outputs a value capturing how close the hypothesis' predicted label $h(x)$ is to the true target label y, with lower values being better. The loss is typically zero if the hypothesis correctly predicts the label, i.e,., if $h(x) = y$ then $\ell(h, (x, y)) = 0$.

The **population risk** of a hypothesis $h \in \mathcal{H}$ compared to the true concept $c \in \mathcal{C}$, with respect to a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to [0, \infty)$ and distribution $\mathcal{D}$ over $\mathcal{Z}$, is defined as

$$R_{c,\mathcal{D}}(h) := \mathop{\mathbf{E}}_{(X,Y)\sim\mathcal{D}} \left[\ell(h, (X, Y))\right], \tag{2.2}$$

namely the expected loss when (X, Y) is drawn according to $\mathcal{D}$. This definition formalises what it means to have a "good" hypothesis, in that it should have $R_{c,\mathcal{D}}(h) \approx 0$. When it is clear from context we will drop the subscript and write $R(h)$ to mean $R_{c,\mathcal{D}}(h)$.

Note that this definition captures the idea that in order for a hypothesis to do well in terms of having low population risk, it must do more than simply perform well on the given training data — it must minimize the expected loss over *all* of $\mathcal{D}$ even on features it has not encountered in the training set!

Hence a tentative definition for a good learning algorithm $\mathcal{A}$ is one that consistently outputs a hypothesis with close to zero population risk, i.e.,

$$R(\mathcal{A}(Z)) \approx 0 \tag{2.3}$$

for all possible Z.

Alas, this may be too much to hope for, because the samples Z are generated probabilistically and so there is a chance that it may be wholly unrepresentative of its underlying distribution $\mathcal{D}$.

Continuing with our threshold functions example, the concept could be $\theta^* = 0$, but if the samples are such that $X^{(i)} \geq \frac{1}{2}$ for all $i \in [m]$ then the samples do not provide much useful information other than that $\theta^* \leq \frac{1}{2}$ and so no reasonable algorithm could come close to learning a hypothesis with close to

zero population loss. However, this sample, although possible, is unlikely to occur. Most samples would have a roughly even number of positive $X^{(i)}$ as well as negative $X^{(i)}$.

This motivates us to define a good learning algorithm $\mathcal{A}$ as one that, given Z, outputs a hypothesis with close to zero population loss *with high probability over* Z. This is the idea behind the definition of the "probably approximately correct" (PAC) learning framework of Valiant (1984).

2.2 PAC learning

To introduce PAC learning, we will assume for the remainder of this chapter that the **label space is binary**, i.e., $\mathcal{Y} = \{0, 1\}$. Although this is quite a restrictive assumption, we will see that the theory for this special case is already very rich. In this case, the only reasonable loss function to use is the zero-one loss,

$$\ell_{0/1}(h, (x, y)) := \mathbb{1}\{h(x) \neq y\}, \tag{2.4}$$

so that

$$R(h) = \mathop{\mathbf{E}}_{(X,Y)\sim\mathcal{D}} \left[\ell_{0/1}(h, (X, Y))\right] = \mathop{\mathbf{P}}_{(X,Y)\sim\mathcal{D}} \left[h(X) \neq Y\right]. \tag{2.5}$$

We will also assume that $\mathcal{H}$ always contains a hypothesis with zero population risk. This is called the **realisability assumption**.

DEFINITION 1 (Realisability assumption). *We say that the realisability assumption holds for a hypothesis class $\mathcal{H}$ with respect to concept class $\mathcal{C}$ if for any concept $c \in \mathcal{C}$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, there always exists $h \in \mathcal{H}$ with $R_{c,\mathcal{D}}(h) = 0$.*

Finally, we also assume that the learning algorithm outputs $h \in \mathcal{C}$, or in other words, $\mathcal{H} = \mathcal{C}$. This is called **proper learning**. We will discuss the improper case, where the learning algorithm is allowed to output $h \notin \mathcal{C}$ in Section 2.2.1.

We are now ready to introduce the definition of PAC learning.

DEFINITION 2 (Proper realisable PAC learning (Valiant, 1984)). *Assume that the realisability assumption (Definition 1) holds, that $\mathcal{H} = \mathcal{C}$, and that $\mathcal{Y} = \{0, 1\}$. Then $\mathcal{H}$ is said to be PAC learnable (in the proper realisable setting) if there exists a learning algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ and a polynomial function poly$(\cdot, \cdot)$ such that for any $c \in \mathcal{C}$, any distribution $\mathcal{D}$ over $\mathcal{Z}$, any $\varepsilon > 0$ and $\delta > 0$,*

$$\mathop{\mathbf{P}}_{Z\sim\mathcal{D}^{\otimes m}} \left[R(\mathcal{A}(Z)) \leq \varepsilon\right] \geq 1 - \delta \tag{2.6}$$

given $m \geq poly(1/\varepsilon, 1/\delta)$ samples. We say $\mathcal{H}$ is efficiently *PAC learnable if the running time of $\mathcal{A}$ is $poly(1/\varepsilon, 1/\delta)$ and we call $\mathcal{A}$ an* efficient *PAC learner for $\mathcal{H}$.*

In other words, given enough (but only polynomially many) samples m, the learning algorithm will, with arbitrarily high probability $1 - \delta$, output a hypothesis with arbitrary low population risk $R(\mathcal{A}(Z)) \leq \varepsilon$. Note that ε plays the role of guaranteeing that the hypothesis h is "approximately correct" in that it agrees with c when evaluated on most inputs, whilst δ ensures this will "probably" happen when learning from the sample Z.

Different learning algorithms $\mathcal{A}$ will have different functions $\text{poly}(\cdot, \cdot)$. The **sample complexity** of $\mathcal{H}$ is then defined as the smallest possible function.

2.2.1 Proper and improper learners

In the definition of PAC learning we defined above, we required that the learning algorithm must output a hypothesis $h \in \mathcal{C}$, i.e., we have $\mathcal{H} = \mathcal{C}$. Any learning algorithm satisfying this behaviour is called a **proper** learner. However, we may sometimes wish to consider a learning algorithm that outputs a hypothesis $h \notin \mathcal{C}$, i.e., we have $\mathcal{H} \supsetneq \mathcal{C}$. This can often make the learning problem much easier and is called **improper** learning.

For example, consider the concept class $\mathcal{C}$ consisting of 3-term disjunctive normal form (DNF) formulae. This is the set of all disjunctions $T_1 \vee T_2 \vee T_3$ where each T_i is a conjunction of literals over Boolean variables $x_1, \cdots, x_n$. It can be shown that $\mathcal{C}$ is not *efficiently* PAC learnable under standard complexity assumptions (NP $\neq$ RP)[1] using any *proper* learner, i.e., if we require the learning algorithm to output a 3-term DNF (Pitt and Valiant, 1988; Kearns and Vazirani, 1994). However, note that by distributivity, we can rewrite any 3-term DNF formula as

$$T_1 \vee T_2 \vee T_3 = \bigwedge_{u \in T_1, v \in T_2, w \in T_3} (u \vee v \vee w). \tag{2.7}$$

Hence 3-term DNFs are a subset of the class of conjunctive normal form (CNF) formulae where each clause is a disjunction of at most 3 literals, as in the right hand side of Equation 2.7 above — we will call this form 3-CNF. It turns out that if the learning algorithm for 3-term DNFs is allowed to be *improper*, then 3-term DNFs *can be learned in polynomial time* by outputting them in the form of a 3-CNF (Pitt and Valiant, 1988; Kearns and Vazirani, 1994).

[1] `https://complexityzoo.net/Complexity_Zoo`

2.2.2 Agnostic learning

Another direction to extend our original definition of PAC learning is to relax the realisability assumption (Definition 1). We no longer assume that the samples are perfectly labelled according to some concept $c \in C$, or that a target concept exists at all. Instead, we just assume that we are getting labelled data from some process, which may even introduce noise into the labels. This more closely resembles real world data, which is typically not perfectly labelled. The goal now is to find the model that best explains the data from a predetermined hypothesis class such as linear threshold functions, decision trees, neural networks, etc.

In this case the population risk $R(h)$ of any hypothesis cannot be made arbitrarily small, but is instead lower bounded by $\inf_{h \in \mathcal{H}} R(h)$. We define a good learning algorithm as one that outputs a hypothesis whose population risk can become arbitrarily close to $\inf_{h \in \mathcal{H}} R(h)$.

DEFINITION 3 (Proper agnostic PAC learning (Haussler, 1992)). *Assume $\mathcal{H} = \mathcal{C}$ and $\mathcal{Y} = \{0, 1\}$. Then $\mathcal{H}$ is PAC learnable (in the proper agnostic setting) if there exists a learning algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ and a polynomial function poly$(\cdot, \cdot)$ such that for any distribution $\mathcal{D}$ over $\mathcal{Z}$, any $\varepsilon > 0$ and $\delta > 0$,*

$$\mathop{\mathbf{P}}_{Z \sim \mathcal{D}^{\otimes m}} \left[R(A(Z)) < \inf_{h \in \mathcal{H}} R(h) + \varepsilon \right] \geq 1 - \delta \tag{2.8}$$

given $m \geq$ poly$(1/\varepsilon, 1/\delta)$ samples. We say $\mathcal{H}$ is efficiently *agnostically PAC learnable if the running time of $\mathcal{A}$ is poly$(1/\varepsilon, 1/\delta)$ and we call $\mathcal{A}$ an* efficient *agnostic PAC learner for $\mathcal{H}$.*

Learning in the agnostic setting is typically much harder. For example, Kanade (2017) show that linear threshold functions are not efficiently agnostically learnable under standard complexity assumptions (NP $\neq$ RP). On the other hand, LTFs can be learnt in polynomial time in the realizable setting by linear programming or support vector machines.

We can also make a distinction between proper and improper learners in the agnostic setting. For improper learners, we have $\mathcal{H} \supsetneq \mathcal{C}$ and we only require that the algorithm outputs a hypothesis whose population risk can become arbitrarily close to $\inf_{c \in C} R(c)$. In other words, we modify Equation 2.8 to

$$\mathop{\mathbf{P}}_{Z \sim \mathcal{D}^{\otimes m}} \left[R(A(Z)) < \inf_{c \in \mathcal{C}} R(c) + \varepsilon \right] \geq 1 - \delta. \tag{2.9}$$

2.3 Empirical risk minimization

We have now formally defined what it means to have a "good" learning algorithm via the definition of PAC learning. But how does one find such an algorithm? Even in the realizable setting, the target concept c and the underlying distribution $\mathcal{D}$ is unknown, so we cannot minimize the population risk directly. However a simple idea would be to find a hypothesis h that instead minimizes the empirical risk

$$\widehat{R}_Z(h) := \frac{1}{m}\sum_{i=1}^{m} \ell(h, (X_i, Y_i)), \tag{2.10}$$

since the expected value of the empirical risk is equal to the population risk, and the empirical risk is easy to compute. This is the empirical risk minimisation (ERM) algorithm.

For example, under the realizable assumption, the minimum empirical risk is zero. If we are able to find such a h, will that same h also minimize the population risk? Unfortunately, this is not true in general. If the hypothesis class $\mathcal{H}$ is sufficiently complex, it could be the case that h has simply "memorized" the sample Z. As an extreme example, the learnt hypothesis could simply output 1 when its input exactly matches one of the training data samples that had a label of 1. This would clearly have zero empirical risk, but would generalise extremely poorly to unseen data, and is an example of **overfitting**.

This example shows that a hypothesis class that contains such functions is clearly too complex to prevent overfitting. Hence a natural question is to ask when exactly a particular hypothesis class is simple enough so that the learnt hypothesis will generalise well.

2.4 Finite hypothesis classes

A very simple sufficient condition to prevent overfitting is for the hypothesis class $\mathcal{H}$ to be finite. Intuitively, this is because there are only finitely many hypotheses that are consistent with any given sample S. As the sample size increases, the number of such hypotheses will shrink, so given enough samples it is unlikely that a hypothesis that is perfectly consistent with the data would exist, unless that hypothesis is the target concept.

This is stated more formally in the realizable setting by the following result.

THEOREM 1 (Mohri et al. (2018)). *Suppose the realisability assumption holds, $\mathcal{Y} = \{0, 1\}$, the loss is $\ell_{0/1}$, and $|\mathcal{H}| < \infty$. Let $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ be an ERM, i.e., $\widehat{R}_Z(\mathcal{A}(Z)) = 0$. Then*

$$\mathop{\mathbb{P}}_{Z \sim \mathcal{D}^{\otimes m}}[R(\mathcal{A}(Z)) \leq \varepsilon] \geq 1 - \delta$$

for

$$m \geq \frac{1}{\varepsilon}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right).$$

In other words, $\mathcal{H}$ is PAC learnable.

The above bound, however, becomes vacuous when the hypothesis space $\mathcal{H}$ is infinite, which is typically the case for most machine learning problems. For example any problem that requires learning a real number falls into this category. Nevertheless, there are many examples of learning algorithms that generalise well even when the hypothesis class is infinite. Hence we need a more nuanced way to measure complexity.

2.5 VC dimension

A most elegant result in learning theory states that a certain combinatorial quantity of $\mathcal{H}$ called the *VC dimension* (see Definition 4 below) exactly captures the nuances about the PAC learnability of $\mathcal{H}$. Moreover, this quantity even determines the sample complexity.

THEOREM 2 (Fundamental theorem of statistical learning (Blumer et al., 1989; Shalev-Shwartz and Ben-David, 2014)). *Suppose that the realisability assumption holds, $\mathcal{Y} = \{0, 1\}$, and the loss is $\ell_{0/1}$. Let d be the VC dimension of hypothesis class $\mathcal{H}$. Then the following are equivalent.*

(1) *Any proper ERM algorithm is a proper PAC learner for $\mathcal{H}$.*
(2) *$\mathcal{H}$ is properly PAC learnable.*
(3) *$d < \infty$.*

Moreover the proper sample complexity $m(\varepsilon, \delta)$ of $\mathcal{H}$ satisfies

$$\Omega\left(\frac{d + \log(1/\delta)}{\varepsilon}\right) \leq m(\varepsilon, \delta) \leq O\left(\frac{d\log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right), \tag{2.11}$$

with the upper bound being attained by any proper ERM algorithm.

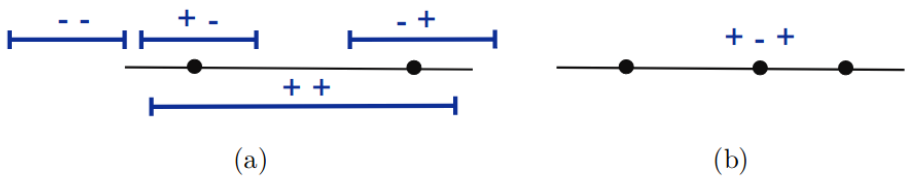


FIGURE 2.1. Illustration showing that the VC dimension of real intervals is 2 (figure due to Mohri et al. (2018)). (a) For any two points, all four classifications are attainable using intervals. (b) No sample of three points can have the classification of $(+, -, +)$ using intervals.

We now formally define the VC dimension of a hypothesis class, which is named after the two authors Vapnik and Chervonenkis (1971).

DEFINITION 4 (VC dimension (Vapnik and Chervonenkis, 1971)). *The VC dimension of a hypothesis class $\mathcal{H}$ is the size of the largest set of features $X \subseteq \mathcal{X}$ for which all $2^{|X|}$ possible labellings of X are attainable using classifiers in $\mathcal{H}$.*

For example, let $\mathcal{H}$ consist of all intervals over the real line. Then for any two distinct points, all four possible classifications, i.e., $(+,+), (+,-), (-,+), (-,-)$ are attainable using classifiers from $\mathcal{H}$. However, no sample of three distinct points can have the classification of $(+,-,+)$. This is illustrated in Figure 2.1. Hence the VC dimension of intervals over the real line is 2.

2.5.1 Sample complexity of improper and proper learning

There is a logarithmic term of $\log(1/\varepsilon)$ between the lower and upper bounds of the sample complexity bounds in Equation 2.11. It turns out these bounds are tight for *proper learners in general* in that for certain hypothesis classes the extra logarithmic term is necessary. Under some general conditions on $\mathcal{H}$, Bousquet et al. (2020) showed that the $\log(1/\varepsilon)$ term can be removed for proper learners if and only if a combinatorial parameter of $\mathcal{H}$ known as the *dual Helly number* is finite.

On the other hand, a landmark result by Hanneke (2016) showed that this logarithmic term can be removed altogether for *improper learners*, namely that the *improper* sample complexity is always

$$m(\varepsilon, \delta) = \Theta\left(\frac{d + \log 1/\delta}{\varepsilon}\right). \tag{2.12}$$

This again shows that improper learning is often much easier than proper learning.

2.5.2 Agnostic learning

Similar ideas continue to hold when we consider learning in the agnostic setting instead. In fact the fundamental theorem of statistical learning continues to hold except that the sample complexities have a quadratic dependence on $1/\varepsilon$, which agrees with intuition that agnostic learning is much harder than the realizable setting.

THEOREM 3 (Fundamental theorem of statistical version, agnostic setting (Blumer et al., 1989; Shalev-Shwartz and Ben-David, 2014)). *Suppose that $\mathcal{Y} = \{0, 1\}$, and the loss is $\ell_{0/1}$. Let d be the VC dimension of hypothesis class $\mathcal{H}$. Then the following are equivalent.*

(1) *Any proper ERM algorithm is a proper agnostic PAC learner for $\mathcal{H}$.*
(2) *$\mathcal{H}$ is properly agnostically PAC learnable.*
(3) $d < \infty$.

Moreover the proper agnostic sample complexity $m(\varepsilon, \delta)$ of $\mathcal{H}$ satisfies

$$m = \Theta\left(\frac{d + \log 1/\delta}{\varepsilon^2}\right) \tag{2.13}$$

which is attained by any proper agnostic ERM algorithm.

Note that unlike the realisable setting, there is no $\log 1/\varepsilon$ gap in the sample complexity for the agnostic setting.

2.6 Compression schemes

VC theory is very elegant in that it tells us exactly when a hypothesis class is PAC learnable, what its optimal sample complexity is, and how well any ERM algorithm will perform. However, it can often be difficult to compute the VC dimension of a particular hypothesis class.

The theory of *compression schemes* provides an alternative way to determine how well an ERM algorithm performs that is much simpler to compute and is dependent on the learning algorithm $\mathcal{A}$ instead of the hypothesis class $\mathcal{H}$.

DEFINITION 5 (Compression scheme (Littlestone and Warmuth, 1986)). *A learning algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ is said to have a* compression scheme *of size k for $k < m$ if $\mathcal{A}$ can be decomposed into a "compression algorithm" $\kappa : \mathcal{Z}^m \to \mathcal{Z}^k$ and an "encoding algorithm" $\rho : \mathcal{Z}^k \to \mathcal{H}$ where:*

(1) *The compression algorithm κ takes as input $Z \in \mathcal{Z}^m$ and returns a subsequence of Z of length k.*
(2) *The encoding algorithm ρ takes as input the compressed subsequence $\kappa(Z)$ and reconstructs the original classifier trained on the full sample Z, that is, $\rho(\kappa(Z)) = \mathcal{A}(Z)$.*

One such example is the support vector machine (SVM) algorithm. Recall that the SVM algorithm finds the largest margin hyperplane, that is, the hyperplane for which the distance to the closest positive and negative labels in the sample are maximized (see Figure 2.2).

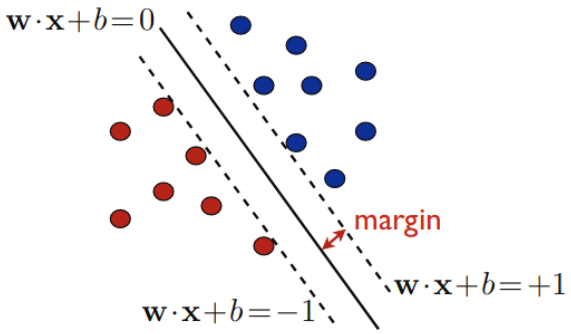


FIGURE 2.2. Illustration of the SVM algorithm, which maximises the margin (distance to the positive and negative samples) of the separating hyperplane (figure due to Mohri et al. (2018)).

The SVM algorithm admits a compression scheme because the separating hyperplane is dependent only on the *support vectors*, i.e., the points lying on the margin.

The following result shows that any ERM that admits a compression scheme is a PAC learner.

THEOREM 4 (Littlestone and Warmuth (1986)). *Suppose that the realisability assumption holds, $\mathcal{Y} = \{0, 1\}$, and the loss is $\ell_{0/1}$. Further suppose that $\mathcal{A} : Z^m \to \mathcal{H}$ is an ERM with a compression scheme of size k. Then,*

$$\mathop{\mathbb{P}}_{Z\sim\mathcal{D}^{\otimes m}}[R(\mathcal{A}(Z)) \leq \varepsilon] \geq 1 - \delta \tag{2.14}$$

for

$$m \geq O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta} + \frac{k}{\varepsilon}\log\frac{k}{\varepsilon} + k\right). \tag{2.15}$$

CHAPTER 3

Information-theoretic generalisation bounds

We have seen that the PAC learnability of a hypothesis class is characterised by its VC dimension and that we can get generalisation bounds for any ERM algorithm. However, VC theory does not tell us how well any *general* algorithm will perform outside the class of ERMs. This is to be expected since VC dimension is a measure of the hypothesis class $\mathcal{H}$ only, and is independent of any learning algorithm for $\mathcal{H}$. Moreover, although any ERM algorithm is sample-optimal, solving the ERM problem can be NP hard for certain hypothesis classes (Feldman et al., 2012). Hence we would like tools to analyse learning algorithms *in general*.

Recently, information-theoretic tools to study generalisation have gathered attention (Russo and Zou, 2016; Xu and Raginsky, 2017; Steinke and Zakynthinou, 2020; Bu et al., 2020). These methods produce *algorithm-dependent* generalisation bounds, meaning they can be used to tell us how well *any* particular algorithm will generalise. The high level idea is that the *mutual information* between the input $Z \in \mathcal{Z}^m$ of the learning algorithm and its output hypothesis $\mathcal{A}(Z) \in \mathcal{H}$ is informative of the algorithm's generalisation ability. To make these ideas rigorous, we will need to take a rather long detour into information theory to formalise what we mean by "information".

3.1 Shannon information theory

In this section we introduce the area of information theory. In particular we will introduce Shannon entropy, Kullback-Leibler divergence and mutual information for discrete random variables, and some of their important properties. We will then extend these ideas to the continuous case and discuss the similarities and differences between the discrete and continuous variants. Proofs of some results will be given only where we think the proofs are relevant and informative. For a more comprehensive treatment of the topic, the reader is invited to consult a reference on information theory such as Cover and Thomas

(2006). On the other hand, the reader who is already informed about this topic, for example after having read Chapter 2 of Cover and Thomas (2006), is invited to skip to Section 3.2.

Information theory has its roots in the landmark paper by Shannon (1948), who was motivated by communication theory. One of his motivating questions was determining the ultimate data compression rate in communication. The answer turns out to be the entropy H, a fundamental quantity which we will discuss shortly, and which is connected with mutual information. Since Shannon's work, information theory has made fundamental contributions to physics, computer science, statistics, and much more (Cover and Thomas, 2006).

3.1.1 Entropy

To motivate the definition of entropy, suppose that you are given a random integer between 1 and 16, with each choice being equally likely, i.e., having probability $p = 1/16$. How many yes/no questions are required to determine what integer you are given? The answer is $\log_2(1/p) = 4$ by effectively performing a binary search using the questions. More generally, for an outcome x that occurs with probability $p(x)$, it takes approximately $\log_2(1/p(x))$ yes/no questions, which can be thought of as "bits" of information, to distinguish it from other outcomes having the same probability. Note $\log_2(1/p(x))$ is only "approximately" correct because of the cases when $1/p(x)$ is not an exact power of 2. The quantity $\log_2(1/p(x))$ can be alternatively interpreted as a measure of surprise or uncertainty in seeing the outcome x — outcomes that have a lower probability $p(x)$ of occurring have a higher $\log_2(1/p(x))$.

Now consider a discrete random variable X taking values $x \in \mathcal{X}$, each with probability $p(x) := \mathbf{P}[X = x]$. The *entropy* of X is defined as the *average* value of $\log_2(1/p(x))$ — it is the average surprise or bits of information to distinguish the realisations of X.

DEFINITION 6 (Entropy). *Let X be a discrete random variable/vector. Then the (Shannon) entropy of X is*

$$H(X) := \mathop{\mathbf{E}}_{X\sim p}\left[\log_2 \frac{1}{p(X)}\right] = \sum_x p(x) \log_2 \frac{1}{p(x)} \tag{3.1}$$

in bits, where we define $0 \log \frac{1}{0} := 0$ as $\lim_{p\to 0} p \log_2 \frac{1}{p} = 0$.

For example consider a Bernoulli random variable X that takes value 1 with probability p and 0 with probability $1 - p$. Then its entropy is

$$H(X) = -p \log_2 p - (1 - p) \log_2(1 - p) \tag{3.2}$$

bits.

It can be easily checked that the above expression is maximised at $p = 1/2$ in which case the entropy is exactly one bit. This makes intuitive sense as the outcome of a coin toss is most uncertain or surprising when the coin is unbiased. On the other hand, we would not be very uncertain or surprised in seeing the outcome of the toss of a coin that is heavily biased towards heads or tails. Correspondingly, as $p \to 0$ or $p \to 1$, the random variable becomes deterministic and Equation 3.2 approaches zero. See Figure 3.1 below for a diagram.

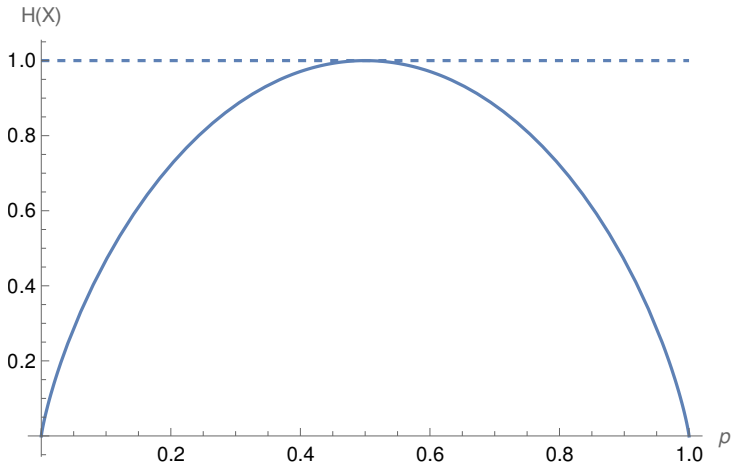


FIGURE 3.1. Graph of $H(X)$ where $X \sim \text{Bernoulli}(p)$ as a function of p. Entropy is maximised when $p = 1/2$.

More generally, entropy is always non-negative, and is equal to zero exactly when the random variable is deterministic (a constant).

LEMMA 1 (Entropy is non-negative). *$H(X) \geq 0$ with equality if and only if X is deterministic.*

PROOF. For $p(x) \in (0, 1]$ we have $\log_2 \frac{1}{p(x)} > 0$. When $p(x) = 0$, by definition $p(x) \log_2 \frac{1}{p(x)} = 0$. Hence $\sum_x p(x) \log_2 \frac{1}{p(x)} \geq 0$. For equality to occur, we must have either $p(x) = 0$ or $p(x) = 1$ for all x since these are the only two cases for which $p(x) \log_2 \frac{1}{p(x)} = 0$. □

Also, observe that the definition of entropy is dependent only on the probability mass function $p(x)$, and not on what values the random variable attains. Suppose we apply a function f to a random variable X. Then any realisation $x \in \mathcal{X}$ of X is mapped to $f(x)$. Moreover, if the function f is injective, then the value $f(x)$ could only have come from the original realisation x, which occurs with probability $p(x)$. So

after applying f, the resulting random variable attains value $f(x)$ with probability $p(x)$ for each $x \in \mathcal{X}$. Hence the entropy, which is dependent on only the probabilities $p(x)$ remains unchanged.

LEMMA 2. *Let f be an injective function and X a random variable/vector. Then*

$$H(f(X)) = H(X). \tag{3.3}$$

For example adding any constant to a random variable or multiplying a random variable by any non-zero constant does not change its entropy.

Before introducing our next definition, let us make a quick notational remark. When it is clear from context, we will use notation like $p(x \mid y)$ to denote conditional probability mass functions and $p(x)$ to denote the marginal probability mass functions, where it is clear from context what random variable(s) the probabilities are being taken with respect to. To be clear, in the two examples, $p(x \mid y)$ is short for $\mathbf{P}[X = x \mid Y = y]$ and $p(x)$ is short for $\mathbf{P}[X = x]$. With this in mind, we now define conditional entropy, which describes the average uncertainty or surprise in a random variable X, *conditioned* on another random variable Y.

DEFINITION 7 (Conditional entropy). *Let X and Y be two jointly distributed discrete random variables. The conditional (Shannon) entropy of X given Y is*

$$H(X \mid Y) := \mathop{\mathbf{E}}_{y \sim p_Y} [H(X \mid Y = y)] \tag{3.4}$$

$$= \sum_y p(y) \sum_x p(x \mid y) \log \frac{1}{p(x \mid y)} \tag{3.5}$$

in bits.

In other words, conditional entropy is the entropy of the random variable $X \mid Y = y$, averaged over the values of y.

An important fact is that entropy can never increase after conditioning on another random variable.

LEMMA 3 (Conditioning cannot increase entropy). *$H(X \mid Y) \leq H(Y)$ with equality if and only if X and Y are independent.*

Intuitively, this is saying that knowing some extra information Y can never increase uncertainty about X on average.

Finally, we introduce a *chain rule* for entropy that lets us break up the entropy of a random *vector* $H(X, Y)$ as the sum of the entropy of a single variable $H(X)$ plus the conditional entropy of the second given the first $H(Y \mid X)$.

LEMMA 4.

$$H(X, Y) = H(X) + H(Y \mid X). \tag{3.6}$$

Applying Lemma 3 to this result gives the following.

LEMMA 5.

$$H(X, Y) \leq H(X) + H(Y) \tag{3.7}$$

with equality if and only if X and Y are independent.

3.1.2 KL divergence

Next, we introduce a measure of the distance between two probability distributions.

DEFINITION 8 (Kullback-Leibler divergence). *Let $p(x)$ and $q(x)$ be two probability mass functions. Then the Kullback-Leibler (KL) divergence between p and q is*

$$D(p \parallel q) := \sum_x p(x) \log_2 \frac{p(x)}{q(x)} \tag{3.8}$$

in bits.

It can be shown that $D(p \parallel q)$ is a measure of the "inefficiency" of assuming the distribution is q when the true distribution is p.

Note that KL divergence lacks some properties we would expect of a distance. In particular, it is not symmetric and does not satisfy the triangle inequality. Hence it is more appropriate to call it a "divergence" than a true "distance". Nonetheless, KL divergence does have the important property that it is always non-negative, and is equal to zero if and only if the two probability distributions are equal.

THEOREM 5 (Gibbs' inequality).

$$D(p \parallel q) \geq 0 \tag{3.9}$$

with equality if and only if $p = q$.

PROOF. Let $P := \{x \in \mathcal{X} : p(x) > 0\}$ be the support of p. Then

$$D(p \parallel q) = \sum_{x \in P} p(x) \log_2 \frac{p(x)}{q(x)} \tag{3.10}$$

$$= \mathop{\mathbf{E}}_{X \sim p} \left[-\log_2 \frac{q(X)}{p(X} \right] \tag{3.11}$$

$$\geq -\log_2 \mathop{\mathbf{E}}_{X \sim p} \left[\frac{q(X)}{p(X)} \right] \tag{3.12}$$

$$= -\log_2 \left(\sum_{x \in P} p(x) \frac{q(x)}{p(x)} \right) \tag{3.13}$$

$$= -\log_2 \left(\sum_{x} q(x) \right) \tag{3.14}$$

$$\geq -\log_2 1 \tag{3.15}$$

$$= 0, \tag{3.16}$$

where Equation 3.12 is by Jensen's inequality (Theorem 32) since $x \mapsto -\log_2(x)$ is convex. □

3.1.3 Mutual information

We are now finally ready to introduce mutual information.

DEFINITION 9 (Mutual information). *Let X and Y be two jointly distributed discrete random variables. Then the mutual information between X and Y is*

$$I(X;Y) := D(p_{XY} \parallel p_X \otimes p_Y) \tag{3.17}$$

$$= \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \tag{3.18}$$

Intuitively, this is a measure of similarity between the joint distribution p_{XY} and the product of the marginals $p_X \otimes p_Y$, i.e., the distribution *if X and Y were independent*. In other words, it is a measure of how "close" X and Y are to being independent. Indeed, as a direct consequence of the non-negativity of KL divergence (Theorem 5), we have the following.

THEOREM 6. *For any two random variables X, Y,*

$$I(X;Y) \geq 0 \tag{3.19}$$

with equality if and only if $p_{XY} = p_X \otimes p_Y$, i.e., X and Y are independent.

Intuitively speaking, random variables that are close to being independent do not carry much information about each other, and indeed the mutual information is close to zero; on the other hand, random variables that are far from being independent must be tightly coupled with each other, and the mutual information between them is high.

The next lemma states that mutual information is symmetric, hence the name *mutual* information, and provides an alternate interpretation of mutual information in terms of *entropy*.

LEMMA 6. *For any two random variables X, Y,*

$$I(X;Y) = I(Y;X) \tag{3.20}$$

$$= H(Y) - H(Y \mid X) \tag{3.21}$$

$$= H(X) - H(X \mid Y) \tag{3.22}$$

Combining the non-negativity of KL divergence (Theorem 5) and the above result shows that $H(X \mid Y) \leq H(X)$ with equality if and only if X and Y are independent, which is Lemma 3.

Equation 3.21 and Equation 3.22 show that mutual information may alternatively be thought of as how much the entropy of one variable, say X, goes down *when conditioned on a second variable* Y. Intuitively, after seeing Y, we have gained $H(X) - H(X \mid Y) = I(X;Y)$ bits of information about X since our uncertainty about X has been reduced by that amount.

Next, we show that scaling a variable does not change the mutual information.

LEMMA 7.

$$I(aX;Y) = I(X;Y) \tag{3.23}$$

for any $a \neq 0$.

PROOF.

$$I(aX;Y) = H(aX) - H(aX \mid Y) \tag{3.24}$$

$$= H(aX) - \mathop{\mathbf{E}}_{y\sim p_Y}[H(aX \mid Y = y)] \tag{3.25}$$

$$= H(X) - \mathop{\mathbf{E}}_{y\sim p_Y}[H(X \mid Y = y)] \tag{3.26}$$

$$= H(X) - H(X \mid Y) \tag{3.27}$$

$$= I(X;Y) \tag{3.28}$$

where Equation 3.24 is due to Lemma 6, Equation 3.25 by definition of conditional entropy, Equation 3.26 by Lemma 2 since $x \mapsto ax$ is injective for $a \neq 0$, Equation 3.27 by definition of conditional entropy, and Equation 3.28 by Lemma 6. □

Note that symmetry of mutual information also implies $I(X;aY) = I(X;Y)$, or more generally that $I(aX;bY) = I(X;Y)$ for $a, b \neq 0$.

Our last property of mutual information states that mutual information between two variables cannot increase as a result of processing any of the two variables. To state this result, let us first make a definition.

DEFINITION 10. *Random variables X, Y, Z are said to form a Markov chain, denoted $X \to Y \to Z$ if the conditional distribution of Z depends only on Y and is conditionally independent of X, i.e.,*

$$p(z \mid y, x) = p(z \mid y) \tag{3.29}$$

for all x, y, z.

Intuitively, this is saying that all the "information" from X and Y that can be used to determine Z can be found in Y alone, where the term "information" is being used in a loose sense. Any "information" from X that can be used to determine Z always passes through Y first, and consequently, having conditioned on Y, there is no "information" between X and Z, i.e., X and Z are conditionally independent given Y. In fact, it can be shown that $X \to Y \to Z$ if and only if X and Z are conditionally independent given Y.

With this definition in mind, we can now state the data processing inequality.

THEOREM 7 (Data processing inequality). *Suppose $X \to Y \to Z$ form a Markov chain. Then*

$$I(X;Y) \geq I(X;Z). \tag{3.30}$$

For example, if $Z = f(Y)$ is a deterministic function of Y, then $X \to Y \to Z = f(X)$ form a Markov chain and so the data processing inequality implies

$$I(X;Y) \geq I(X;f(Y)). \tag{3.31}$$

Finally, we define a conditional variant of mutual information, following the same ideas as for the definition of conditional entropy.

DEFINITION 11 (Conditional mutual information). *Let X, Y, Z be three jointly distributed discrete random variables. The conditional mutual information between X and Y given Z is*

$$I(X;Y \mid Z) := \mathop{\mathbf{E}}_{z \sim p_Z} [I(X \mid Z = z; Y \mid Z = z)] \tag{3.32}$$

$$= \sum_z p(z) \sum_x \sum_y p(x,y \mid z) \log \frac{p(x,y \mid z)}{p(x \mid z)p(y \mid z)} \tag{3.33}$$

In other words, conditional mutual information is the mutual information between the random variables $X \mid Z = z$ and $Y \mid Z = z$, averaged over all possible values of z.

3.1.4 Differential entropy

All the above quantities can be defined for continuous random variables by replacing sums with integrals, and probability mass functions with probability density functions. In the continuous case, it is most natural to take logarithms base e instead of base 2 as we were doing previously, which leads to entropy and related quantities to be given in *nats* instead of bits. The continuous analogue for Shannon entropy is *differential entropy*.

DEFINITION 12 (Differential entropy). *Let X be a **continuous** random variable/vector with probability **density** function $f(x)$ and support $\mathcal{X} := \{x : f(x) > 0\}$. Then the differential entropy of X is*

$$h(X) := \int_{\mathcal{X}} f(x) \ln \frac{1}{f(x)} \, dx \tag{3.34}$$

in nats.

Note that we use lower case h for differential entropy, to distinguish it from Shannon entropy H.

Great care needs to be taken when manipulating differential entropy as some properties that hold for Shannon entropy do not necessarily carry over to differential entropy. As an example, let us compute the differential entropy of a random variable X taking values uniformly over the real interval $[0, a]$ for $a > 0$. Then X has p.d.f. $f(x) = 1/a$ over $[0, a]$ and so its differential entropy is

$$h(X) = \int_0^a \frac{1}{a} \ln a \, dx = \ln a. \tag{3.35}$$

Observe that taking $a = 1$ tells us that that the differential entropy of a uniform $[0, 1]$ random variable is zero. Hence, unlike Shannon entropy, a differential entropy of zero does **not** correspond to a deterministic random variable. Determinism occurs in the limit as $a \to 0$ in which case the differential entropy $\ln a$ limits to $-\infty$. This also shows that *differential entropy can be negative*, unlike Shannon entropy. More generally, taking any $0 < a < 1$ will result in $H(X) = \ln a < 0$. Compare these observations with Lemma 1. The reason that Lemma 1 no longer holds is because the proof relied on the probability mass function satisfying $p(x) \leq 1$ which subsequently implies that $\log_2 \frac{1}{p(x)} > 0$. However, for a density f, it is not necessarily true that $f(x) \leq 1$, and so if $f(x) > 1$ it is possible for differential entropy to become negative.

Furthermore observe that scaling X by any positive $a \neq 1$ **changes** its differential entropy. Generally, we have the following result.

LEMMA 8. *Let X be a continuous random variable/vector and $a \neq 0$. Then*

$$h(aX) = h(X) + \ln a. \tag{3.36}$$

PROOF. First assume $a > 0$. The c.d.f. of the random variable aX is

$$F_{aX}(x) := \mathbf{P}\left[aX \leq x\right] \tag{3.37}$$

$$= \mathbf{P}\left[X \leq \frac{x}{a}\right] \tag{3.38}$$

$$= F_X\left(\frac{x}{a}\right) \tag{3.39}$$

where F_X is the c.d.f. of X. Differentiating both sides with respect to x, we obtain that the density of aX is

$$f_{aX}(x) = \frac{1}{a} f_X\left(\frac{x}{a}\right) \tag{3.40}$$

where f_X is the density of X. Hence,

$$h(aX) = -\int \frac{1}{a} f_X\left(\frac{x}{a}\right) \ln\left(\frac{1}{a} f_X\left(\frac{x}{a}\right)\right) \, dx \tag{3.41}$$

$$= \ln a \int \frac{1}{a} f_X\left(\frac{x}{a}\right) \, dx - \int \frac{1}{a} f_X\left(\frac{x}{a}\right) \ln f_X\left(\frac{x}{a}\right) \, dx \tag{3.42}$$

$$= \ln a \int f_X\left(\frac{x}{a}\right) \, d\left(\frac{x}{a}\right) - \int f_X\left(\frac{x}{a}\right) \ln f_X\left(\frac{x}{a}\right) \, d\left(\frac{x}{a}\right) \tag{3.43}$$

$$= \ln a + h(X). \tag{3.44}$$

For the case $a < 0$, the c.d.f. of aX is given by

$$F_{aX}(x) := \mathbf{P}\left[aX \leq x\right] \tag{3.45}$$

$$= \mathbf{P}\left[X \geq \frac{x}{a}\right] \tag{3.46}$$

$$= 1 - F_X\left(\frac{x}{a}\right) \tag{3.47}$$

and so the density is

$$f_{aX}(x) = -\frac{1}{a} f_X\left(\frac{x}{a}\right). \tag{3.48}$$

Proceeding with similar steps to the case $a > 0$ gives the result. □

This is markedly different behaviour to Shannon entropy as the Shannon entropy of a random variable remains unchanged after applying any injective function to it (see Lemma 2). The one saving grace of differential entropy is that it is invariant to translations. This can be easily seen by using a change of variables in the definition of differential entropy.

LEMMA 9. *Let X be a continuous random variable/vector and c a constant. Then*

$$h(X + c) = h(X). \tag{3.49}$$

Finally, let us conclude our discussion on differential entropy with the important example of a normal random variable. In light of the above result, we will look at normal random variables with mean zero only.

THEOREM 8. *Let $X \sim \mathcal{N}(0, \sigma^2)$. Then*

$$h(X) = \frac{1}{2} \ln\left(2\pi e \sigma^2\right). \tag{3.50}$$

PROOF. For the case $\sigma = 1$ we have

$$h(X) = -\int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \ln\left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)\right] dx \tag{3.51}$$

$$= \frac{1}{2} \ln 2\pi \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) dx \tag{3.52}$$

$$+ \frac{1}{2} \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} x^2 \exp\left(-\frac{1}{2}x^2\right) dx$$

$$= \frac{1}{2} \ln 2\pi + \frac{1}{2} \tag{3.53}$$

$$= \frac{1}{2} \ln 2\pi e, \tag{3.54}$$

where Equation 3.53 follows because densities integrate to one, and $\mathbf{E}[X^2] = \sigma^2 = 1$. For general σ, the result follows by Lemma 8. □

3.1.5 Differential KL divergence

Because of the many undesirable properties of differential entropy, it is sometimes useful to study KL divergence instead. Just like differential entropy, this is defined by taking the definition in the discrete case and replacing probability mass functions with densities, sums with integrals, and, for convenience, base 2 logarithms with natural logarithms.

DEFINITION 13 ((Differential) Kullback-Leibler divergence). *Let $f(x)$ and $g(x)$ be two probability* ***density*** *functions. Then the KL divergence between f and g is*

$$D(f \parallel g) := \int_{\mathcal{X}} f(x) \ln \frac{f(x)}{g(x)} \, dx \tag{3.55}$$

where $\mathcal{X} = \{x : f(x) > 0\}$ is the support of f.

Unlike differential entropy however, KL divergence is always non-negative even in the continuous case, due to Jensen's inequality continuing to hold in the continuous case. In other words, Gibbs' inequality (Theorem 5) continues to hold.

For this reason, instead of studying the entropy of a density f, it can be useful to instead study the KL divergence between f and a fixed reference density such as the normal density, i.e., the quantity

$$D\left(f(x) \,\middle\|\, \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}x^2\right)\right) \geq 0. \tag{3.56}$$

We will see in Section 6.4.2 that the normal density **maximises** differential entropy over all continuous random variables with a given variance. Intuitively, having Equation 3.56 close to zero implies that f is close to the normal density, and so we might expect f has close to maximum differential entropy; conversely having large values of Equation 3.56 implies that f is far from being normal, and so we might expect that its differential entropy is much lower compared to that of a normal. In fact, as we will see in Section 6.4.2, if σ^2 is chosen to be the variance of (the random variable associated with) f, then

$$D\left(f(x) \,\middle\|\, \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}x^2\right)\right) = \frac{1}{2}\ln\left(2\pi e\sigma^2\right) - h(f). \tag{3.57}$$

In other words Equation 3.56 reduces to the difference between the differential entropy of f to that of the corresponding normal with the same variance. For this reason, KL divergence is often called *relative entropy*; it measures the entropy of f, *relative* to another reference distribution.

Note that in the discrete case when the support is finite, one has an analogous statement when taking the KL divergence with respect to the uniform distribution over the support.

3.1.6 Differential mutual information

Finally, we can extend mutual information to the continuous case in the obvious way.

DEFINITION 14. *The mutual information between two continuous random variables/vectors X and Y is*

$$I(X;Y) = D(f_{XY} \parallel f_X \otimes f_Y) \tag{3.58}$$

$$= \iint f_{XY}(x,y) \ln \frac{f_{XY}(x,y)}{f_X(x) f_Y(y)}\, dx\, dy. \tag{3.59}$$

Because KL divergence is non-negative in the continuous case, so too is mutual information, with mutual information being zero if and only if the two variables are independent, i.e., Theorem 6 continues to hold for the continuous case.

Additionally, all the properties that that mutual information satisfied in the discrete case continue to carry over to the continuous case. Namely, (differential) mutual information is symmetric, $I(X;Y) = I(Y;X)$. It is the reduction in entropy after conditioning on the second variable, $I(X;Y) = H(X) - H(X \mid Y)$. It is invariant to scaling in either variable, $I(aX;bY) = I(X;Y)$ for $a, b \neq 0$. Finally, it continues to satisfy the data processing inequality.

The upshot of all this is that KL divergence, and in particular mutual information, shares the same properties in both the discrete and continuous cases, and so for the most part we will not make too great a distinction between the two cases. However, the same is categorically not true for the continuous analogue of entropy, and so great care must be taken when working with this quantity. This is why we slightly overload notation and write $D(f \parallel g)$ and $I(X;Y)$ for both the discrete and continuous cases, but we make a notational distinction between differential entropy h and Shannon entropy H.

3.2 Mutual information generalisation bounds

Having rigorously introduced some foundational concepts in information theory, we are finally ready to see how this ties in with deriving generalisation bounds for a learning algorithm. As discussed at the very beginning of this chapter, the high level idea is to look at

$$I(Z;\mathcal{A}(Z)), \tag{3.60}$$

the mutual information between the training samples $Z \in \mathcal{Z}^m$ given to the learning algorithm $\mathcal{A}$, and the hypothesis that the algorithm outputs $\mathcal{A}(Z) \in \mathcal{H}$. Note that Z is a random vector consisting of m i.i.d. samples from distribution $\mathcal{D}$, and so $A(Z)$ is also a random variable/vector.

If the mutual information between the two quantities is very low, then we know that they are close to being independent (see Theorem 6). Intuitively, this means the learnt hypothesis $\mathcal{A}(Z)$ is not very dependent on the training data Z, so modifying Z will not change $\mathcal{A}(Z)$ too much, and hence the learnt hypothesis is unlikely to overfit to its training data. On the other hand, if the mutual information is very high, this suggests the learnt hypothesis is highly sensitive to the input samples, and so it may be overfitting. These ideas were made formal in the work of Russo and Zou (2016) and Xu and Raginsky (2017) who showed that $I(Z;\mathcal{A}(Z))$ can be used to bound the *expected generalisation error*, that is, the expected difference between the empirical risk and population risk.

THEOREM 9 (Russo and Zou (2016); Xu and Raginsky (2017)). *Suppose $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ is a (possibly randomised) learning algorithm, and that the loss function has bounded range, i.e., is of the form $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$. Then*

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \Big[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \Big] \right| \leq \sqrt{\frac{1}{2m} I(Z;\mathcal{A}(Z))}. \tag{3.61}$$

The mutual information quantity above is assumed to be given in nats, i.e., all logarithms are taken in base e, *even in the discrete case* where we have previously defined all information-theoretic quantities using base 2. Since $\log_2 x = \frac{\ln x}{\ln 2}$, if we have computed $I(Z; \mathcal{A}(Z))$ in bits, we can convert it to nats by multiplying by $\ln 2$.

Let us make a few remarks about Theorem 9. Firstly, note that there is no requirement for $\mathcal{Y} = \{0, 1\}$, as was the case for VC theory and compression schemes. Consequently, there is no restriction on what loss function we use, only that it needs to output values in the range $[0, 1]$.

Secondly, we allow the learning algorithm $\mathcal{A}$ to be randomised. In fact, if the feature vectors are continuous (e.g., $\mathcal{X} = \mathbb{R}^n$), then the learning algorithm *must* be randomised in order to get non-vacuous bounds in Theorem 9. This is because in the continuous case,

$$I(Z; \mathcal{A}(Z)) = h(\mathcal{A}(Z)) - h(\mathcal{A}(Z) \mid Z) \tag{3.62}$$

$$= h(\mathcal{A}(Z)) - \mathop{\mathbf{E}}_{z \sim p_Z} \left[h(\mathcal{A}(Z) \mid Z = z) \right]. \tag{3.63}$$

However, if $\mathcal{A}$ is deterministic, the random variable/vector $\mathcal{A}(Z) \mid Z = z$ is deterministic for each fixed value z and so its differential entropy $h(\mathcal{A}(Z) \mid Z = z)$ is $-\infty$, hence the mutual information $I(Z; \mathcal{A}(Z))$ is infinite. In the discrete case, this is not a problem because the *Shannon* entropy of a deterministic *discrete* random variable is zero and so

$$I(Z; \mathcal{A}(Z)) = H(\mathcal{A}(Z)). \tag{3.64}$$

Indeed we will see these observations play out in Chapter 6.

On the flip side, note that it is possible for $I(Z; \mathcal{A}(Z)) = 0$. As we have seen before, mutual information is always non-negative, and is zero if and only if the two arguments, here Z and $\mathcal{A}(Z)$, are independent (Theorem 6). For this to be the case, the learning algorithm must ignore the inputs Z that it is given. For example, a "learning" algorithm that always outputs the same hypothesis h, regardless of its inputs Z, would have $I(Z; \mathcal{A}(Z)) = I(Z; h) = 0$. Theorem 9 then implies the expected generalisation error is zero, but clearly this is a terrible learning algorithm, if it can be called a learning algorithm at all. There is no contradiction here, however. Although the empirical and population risks are the same (in expectation), the population risk $R(h)$ is in general terrible, and $R(h)$ is what we ultimately want to minimize. This example shows that *having low (expected) generalisation error is not sufficient to guarantee a good learning algorithm*; we also require one of either $R(\mathcal{A}(Z))$ or $\widehat{R}_Z(\mathcal{A}(Z))$ to be low as well.

Finally, if $\mathcal{A}$ is an ERM and the realisability assumption holds, then $\widehat{R}_Z(\mathcal{A}(Z)) = 0$ and so Theorem 9 reduces to

$$\mathop{\mathbf{E}}_{Z}[R(\mathcal{A}(Z))] \leq \sqrt{\frac{1}{2m} I(Z;\mathcal{A}(Z))}. \tag{3.65}$$

By Markov's inequality (Theorem 33), this then implies

$$\mathbf{P}[R(\mathcal{A}(Z)) \geq \varepsilon] \leq \sqrt{\frac{1}{2m\varepsilon^2} I(Z;\mathcal{A}(Z))}. \tag{3.66}$$

Letting $\delta := \sqrt{\frac{1}{2m\varepsilon^2} I(Z;\mathcal{A}(Z))}$ and solving for m implies that

$$\mathbf{P}[R(\mathcal{A}(Z)) \geq \varepsilon] \leq \delta, \tag{3.67}$$

or equivalently that

$$\mathbf{P}[R(\mathcal{A}(Z)) < \varepsilon] \geq 1 - \delta, \tag{3.68}$$

for

$$m \geq \frac{I(Z;\mathcal{A}(Z))}{2\delta^2\varepsilon^2}. \tag{3.69}$$

Note that Equation 3.68 is in the form of the definition for PAC learning (see Equation 2.6) and so Equation 3.69 gives us sample complexity bounds for the learning algorithm $\mathcal{A}$. Unfortunately, the dependence on δ and ε is suboptimal, since VC theory already tells us that the sample complexity of any ERM $\mathcal{A}$ is no more than $O\left(\frac{d\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\right)$ (see Equation 2.11). This is because Theorem 9 only provides bounds *in expectation*, and so applying Markov's inequality produces quite weak bounds. In this sense, the expected generalisation error bounds are qualitatively weaker compared to the PAC learning sample complexity bounds attained by VC theory. On the other hand, this is the price to pay for generality — the framework here allows for arbitrary $\mathcal{Y}$, more general loss functions ℓ, and works for *any* algorithm $\mathcal{A}$, even randomised ones.

Some recent work has showed that if we a *different* measure of information such as Sibson's α-mutual information $I_\alpha(Z;\mathcal{A}(Z))$, a generalisation of mutual information (Sibson, 1969; Verdú, 2015), or maximal leakage (Issa et al., 2016), then this *can* be used to obtain *high probability bounds*, that is, sample complexity bounds that are polylogarithmic in $1/\delta$ instead of polynomial in $1/\delta$ (Esposito et al., 2020a,b). For example, Esposito et al. (2020b) show that Sibson's α mutual information can be used to derive a *high probability* bound that results in sample complexity

$$m \geq \frac{I_\alpha(Z;\mathcal{A}(Z)) + \ln 2 + \frac{\alpha}{\alpha-1}\log\frac{1}{\delta}}{2\varepsilon^2}. \tag{3.70}$$

Unfortunately, computing α mutual information is much more difficult as many of the natural properties that mutual information satisfies do not carry over. In particular, α mutual information is not symmetric and does not satisfy a chain rule (Steinke, 2023).

3.3 Individual sample mutual information generalisation bounds

We have seen that $I(Z;\mathcal{A}(Z))$ is infinite when Z is continuous and $\mathcal{A}$ deterministic. This is highly undesirable, as many learning algorithms fall into this category despite having good generalisation guarantees. For example, the SVM algorithm with $\mathcal{X} = \mathbb{R}^n$ falls into this category. This is despite the fact that, in the realisable case, SVM is an ERM and contains a compression scheme, so we can bound its sample complexity using either the results of VC theory or compression schemes.

There are a few ways to address this issue. An idea explored in Bu et al. (2020) is to look at the mutual information between *one* sample $Z^{(i)}$ and the output hypothesis $\mathcal{A}(Z)$, i.e., the quantity

$$I(Z^{(i)};\mathcal{A}(Z)), \tag{3.71}$$

instead of the mutual information between the *entire* sample and the output hypothesis $I(Z;\mathcal{A}(Z))$.

The advantage of this approach is that even if Z is continuous and $\mathcal{A}$ is a deterministic function of Z, it is often the case that $\mathcal{A}$ is *not* a deterministic function of a single sample $Z^{(i)}$ alone. This is because, as long as the algorithm is making use of the other $Z^{(j)}$ for $j \neq i$, those $Z^{(j)}$ induce randomness in the algorithm $\mathcal{A}$, when viewed as taking $Z^{(i)}$ alone as input. In this case, $I(Z^{(i)};\mathcal{A}(Z))$ is not necessarily infinite anymore. We will see an example of this distinction playing out in Chapter 6.

The following result shows that, what is essentially the mean of $I(\mathcal{A}(Z);Z^{(i)})$ over all samples i, can be used to bound the expected generalisation error.

THEOREM 10 (Bu et al. (2020)). *Suppose $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ is a (possibly randomised) learning algorithm, and that the loss function satisfies $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$. Then*

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \Big[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \Big] \right| \leq \frac{1}{m} \sum_{i=1}^{m} \sqrt{\frac{1}{2} I(Z^{(i)};\mathcal{A}(Z))}. \tag{3.72}$$

3.4 Conditional mutual information generalisation bounds

Steinke and Zakynthinou (2020) explore an alternative way to address the problem of $I(Z; \mathcal{A}(Z))$ being infinite. The idea is to look at a *conditional* mutual information quantity, where the conditioning is done in a clever way so that the conditional mutual information is always finite.

More specifically, we are given $2m$ i.i.d. samples $\widetilde{Z} \in \mathcal{Z}^{2\times m}$ instead of the usual m. It will be helpful to think of this as m *pairs* of samples. From each of the these pairs, one of the two data points are picked uniformly at random. Let $S \in \{0,1\}^m$ be a random variable denoting the m indices of which data point is being picked, and let $\widetilde{Z}_S \in \mathcal{Z}^m$ denote the selected m points. See Figure 3.2 for an illustration.

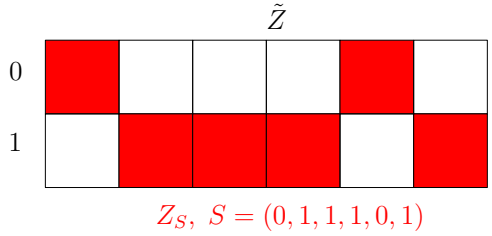


FIGURE 3.2. Illustration of the conditional mutual information framework. The two rows of squares represents $\widetilde{Z}$ with $m = 6$. The red squares represent $\widetilde{Z}_S$ for a particular choice of S.

We look at the mutual information between $\mathcal{A}(\widetilde{Z}_S)$, the output hypothesis of the learning algorithm when given the selected m points, and S, the indices used to select the points, conditioned on the $2m$ samples $\widetilde{Z}$, i.e., the quantity

$$I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right). \tag{3.73}$$

Intuitively, the above quantity measures how well we can *distinguish* the m data points the learning algorithm was trained on from the m "spurious" data points which were not used in the algorithm, by looking at the output hypothesis of the learning algorithm. In contrast, the (unconditional) mutual information $I(\mathcal{A}(Z); Z)$ measures how well we can *reconstruct* the input to the learning algorithm, by looking at the output hypothesis. Importantly, the conditional variant is always bounded in that $I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right) \leq m \ln 2$ nats.

This quantity can be used to bound the expected generalisation error, in a similar way to Theorem 9.

THEOREM 11 (Steinke and Zakynthinou (2020)). *Suppose $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ is a (possibly randomized) learning algorithm, and that the loss function satisfies $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$. Then*

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \left[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \right] \right| \leq \sqrt{\frac{2}{m} I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right)} \tag{3.74}$$

Note that of course the generalisation bound is still vacuous when $I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right)$ is close to its maximum value $m \ln 2$. This is unavoidable because some learning algorithms indeed overfit and so the generalisation error necessarily must be vacuous in that case. However, Steinke and Zakynthinou (2020) illustrate with the example of learning threshold functions over $\mathbb{R}$ that $I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right)$ can be $\Theta(1)$ even when $I(Z; \mathcal{A}(Z))$ is infinite.

3.5 Learning algorithms with low information

We have seen that a variety of information-theoretic quantities can be used to bound the expected generalisation error of a learning algorithm. However, are there simple ways to compute these information-theoretic quantities? For certain classes of learning algorithms, bounds on these quantities are known.

3.5.1 Differentially private algorithms

One such class of learning algorithms are those that are *privacy-preserving* in the sense that we define below.

DEFINITION 15 (Differential privacy (Dwork et al., 2006)). *A randomized learning algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ is* (ε, δ)-differentially private *if for any two training sets $Z, Z' \in \mathcal{Z}^m$ that differ in a single element (i.e. $Z^{(i)} \neq Z'^{(i)}$ for some i, and $Z^{(j)} = Z'^{(j)}$ for all $j \neq i$) and for any set of hypotheses $H \subseteq \mathcal{H}$,*

$$\mathbf{P}\left[\mathcal{A}(Z) \in H\right] \leq \exp(\varepsilon) \mathbf{P}\left[\mathcal{A}(Z') \in \mathcal{H}\right] + \delta. \tag{3.75}$$

Intuitively, differential privacy is saying that if one of the training data points is changed, the distribution in the hypotheses that the learning algorithm outputs will not be changed by too much. It is natural to expect that such learning algorithms also have low mutual information because if changing the inputs to $\mathcal{A}$ does not significantly affect its output, then the input does not provide much information in determining the output either.

Indeed, it has been shown (McGregor et al., 2010; Bun and Steinke, 2016) that if $\mathcal{A}$ is a randomised differentially private learning algorithms with $\delta = 0$, then

$$I(Z; \mathcal{A}(Z)) \leq \frac{1}{2}\varepsilon^2 m. \tag{3.76}$$

Furthermore, Steinke and Zakynthinou (2020) establish the analogous result for the conditional variant, i.e.,

$$I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right) \leq \frac{1}{2}\varepsilon^2 m. \tag{3.77}$$

3.5.2 Compression schemes

Recall from Section 2.6 that a learning algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{H}$ has a compression scheme of size k if only k of its m inputs are used in determining the output hypothesis. This is another situation where we might expect to get low information since not all the information in the inputs are used by the algorithm. Indeed, Steinke and Zakynthinou (2020) show that for such algorithms,

$$I\left(S; \mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right) \leq k \ln(2m). \tag{3.78}$$

These two examples also show that the conditional mutual information approach to generalisation bounds encompass existing techniques. Dwork et al. (2015) show directly (i.e., without using the information theoretic framework described here) that differentially private algorithms generalise well in the context of adaptive data analysis, and of course we saw in Section 2.6 that algorithms which admit a compression scheme also generalise well.

CHAPTER 4

Applications to learning linear threshold functions

We have seen that various information-theoretic quantities can be used to derive expected generalisation error bounds. For certain classes of learning algorithms, such as those that admit compression schemes, or are differentially private, analytic upper bounds on the information-theoretic quantities exist, as discussed in Section 3.5.

In this chapter, we will derive novel bounds on the discussed information-theoretic quantities for an algorithm that learns linear threshold functions (LTFs) over the Boolean hypercube. The main idea is that LTFs are uniquely characterized by $n + 1$ parameters known as the *Chow parameters*, and we can easily estimate these parameters by computing a series of sample means. Our main result in this chapter is Theorem 14 which bounds the expected generalisation error of this algorithm by $O\left(\sqrt{\frac{n \log m}{m}}\right)$.

4.1 Boolean functions and their Fourier expansion

We now formalise the above discussions by introducing Boolean functions and some of their analysis. Specifically, we introduce the *Fourier expansion*, an alternative way to represent Boolean functions. The material in this section is taken from Chapter 1 of O'Donnell (2021) and summarised here for completeness. The reader already familiar with the text may wish to skip to Section 4.2.

A Boolean function is a function $f : \{-1, 1\}^n \to \{-1, 1\}$. As an example, a Boolean function for $n = 2$ could be the function defined by

$$f(+1, +1) = +1,$$
$$f(-1, +1) = +1,$$
$$f(+1, -1) = +1,$$
$$f(-1, -1) = -1,$$

which simply takes the maximum of its two inputs. Interestingly, this function f has another representation:

$$f(x_1, x_2) = \frac{1}{2} + \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_1x_2. \tag{4.1}$$

It is easy to check that, when evaluated on inputs $(x_1, x_2) \in \{-1, 1\}^2$, the two representations are equivalent. Moreover, it can be shown that the above representation is unique.

The form in Equation 4.1 is known as a *multilinear polynomial* — when viewed as a function of any *single* variable x_i alone and treating all other variables as constants, the expression becomes linear. Perhaps surprisingly, *any* Boolean function $f : \{-1, 1\}^n \to \{-1, 1\}$ can be uniquely expressed as a multilinear polynomial over n variables.

To see why, we need to consider a more general class of functions, the *real-valued Boolean functions* $f : \{-1, 1\}^n \to \mathbb{R}$. The set of all such functions forms a vector space V over $\mathbb{R}$ with dimension $\dim V = 2^n$. Informally, this is because we may treat each function f as a 2^n-dimensional "vector", where each component of the vector corresponds to what f evaluates to for a particular input; adding two vectors together corresponds to adding the two corresponding functions together, and multiplying a vector by a scalar corresponds to multiplying the corresponding function by a scalar.

Notice that for any fixed $a = (a_1, \cdots, a_n) \in \{-1, 1\}^n$ the polynomial

$$\mathbb{1}_a(x) := \prod_{i=1}^{n} \frac{1}{2}(1 + a_i x_i) \tag{4.2}$$

is an indicator function that outputs 1 when $x = a$ and 0 otherwise. Hence f can be written as

$$\begin{aligned}
f(x) &= \sum_{a \in \{-1,1\}^n} f(a)\mathbb{1}_a(x) && (4.3)\\
&= \sum_{a \in \{-1,1\}^n} \frac{1}{2^n} f(a) \prod_{i=1}^{n} (1 + a_i x_i) && (4.4)\\
&= \sum_{a \in \{-1,1\}^n} \frac{1}{2^n} f(a) \sum_{S \subseteq [n]} \prod_{i \in S} a_i x_i && (4.5)\\
&= \sum_{S \subseteq [n]} \sum_{a \in \{-1,1\}^n} \frac{1}{2^n} f(a) \prod_{i \in S} a_i \prod_{i \in S} x_i && (4.6)\\
&= \sum_{S \subseteq [n]} \left(\sum_{a \in \{-1,1\}^n} \frac{1}{2^n} f(a)\chi_S(a) \right) \chi_S(x) && (4.7)
\end{aligned}$$

where

$$\chi_S(x) := \prod_{i \in S} x_i \tag{4.8}$$

is the *parity function* of the bits $(x_i)_{i \in S}$ and a multilinear polynomial. Equation 4.7 is a linear combination of multilinear polynomials which is itself a multilinear polynomial, hence showing that any $f : \{-1,1\}^n \to \mathbb{R}$ can be expressed as a multilinear polynomial. Moreover, $(\chi_S)_{S \subseteq [n]}$ must form a basis for V because Equation 4.7 shows that every f can be expressed as a linear combination of χ_S of which there are 2^n many, and $\dim V = 2^n$. Consequently this shows that the coefficient in front of χ_S is unique, and hence the multilinear polynomial representation of f is unique.

This unique multilinear polynomial representation of f has a name: the "Fourier expansion" of f, and the coefficients in front of χ_S are the "Fourier coefficients" of f, denoted $\widehat{f}(S)$. Hence we have proved the following.

THEOREM 12. *Every function $f : \{-1,1\}^n \to \mathbb{R}$ has unique Fourier expansion*

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S(x) \tag{4.9}$$

where the Fourier coefficients are

$$\widehat{f}(S) = \sum_{x \in \{-1,1\}^n} \frac{1}{2^n} f(x) \chi_S(x) = \mathop{\mathbf{E}}_{X \sim \{-1,1\}^n} [f(X) \chi_S(X)] . \tag{4.10}$$

The notation $X \sim \{-1,1\}^n$ means that X is distributed uniformly over the set $\{-1,1\}^n$. Equivalently, X is a vector of n independent Rademacher ± 1 random variables.

In particular, this shows that *Boolean* functions $f : \{-1,1\}^n \to \{-1,1\}$, a special case of real-valued Boolean functions, have unique Fourier expansion. Crucially, this means we can characterise them using their Fourier coefficients $\widehat{f}(S)$, instead of the more traditional characterisation using their truth tables, i.e., a list of the values $f(x)$ for all possible $x \in \{-1,1\}^n$.

At first glance, this seems a bit pointless, as there are 2^n many Fourier coefficients $\widehat{f}(S)$, one for each subset $S \subseteq [n]$, which is just as many values as we would need to list out in the truth table representation of f. However, for certain classes of Boolean functions, namely linear threshold functions and more generally polynomial threshold functions, such functions are entirely determined by only a very small number of Fourier coefficients; hence to learn such functions, it suffices to only learn those particular Fourier coefficients. We will explore this idea in the current chapter.

For other classes of Boolean functions, it can also be shown that most of the large Fourier coefficients are concentrated on a small number of subsets, and learning only those Fourier coefficients allows us to learn the function to sufficiently high accuracy. We will explore this idea in Chapter 5.

4.2 Linear threshold functions and the Chow parameters

A linear threshold function is any Boolean function which can be represented as a linear function $x \mapsto w_0 + w^T x$, then taking the sign of the output.

DEFINITION 16. *A linear threshold function (LTF) is a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ such that*

$$f(x) = sgn(w_0 + w_1 x_1 + \cdots + w_n x_n), \tag{4.11}$$

for $w_0, \cdots, w_n \in \mathbb{R}$.

As alluded to above, LTFs are special in that they are entirely determined by only a small number of Fourier coefficients, namely those $\widehat{f}(S)$ such that $|S| \leq 1$. This result is known as Chow's theorem, which was proved independently by Chow (1961) and Tannenbaum (1961).

THEOREM 13 (Chow's theorem (Chow, 1961; Tannenbaum, 1961; O'Donnell, 2021)). *Suppose $f, g : \{-1,1\}^n \to \{-1,1\}$ are two LTFs. If $\widehat{f}(S) = \widehat{g}(S)$ for all $|S| \leq 1$ then $f = g$.*

We will prove this theorem in Section 4.5. In light of this result, we can learn LTFs by only learning $\widehat{f}(\emptyset), \widehat{f}(\{1\}), \cdots, \widehat{f}(\{n\})$, which we will abuse notation slightly and write as $\widehat{f}(0), \widehat{f}(1), \cdots, \widehat{f}(n)$ respectively. These are often referred to as the "Chow parameters" of f. How can the Chow parameters be learnt?

Suppose we have m i.i.d. samples $X^{(1)}, \cdots, X^{(m)}$, each drawn uniformly from $\{-1,1\}^n$, and their associated labels $Y^{(1)}, \cdots, Y^{(m)}$. Since $\widehat{f}(S) = \mathbf{E}[f(X)\chi_S(X)]$, we in particular have $\widehat{f}(\emptyset) = \mathbf{E}[f(X)]$ and $\widehat{f}(j) = \mathbf{E}[f(X)X_j]$ for $j \in \{1, \cdots, n\}$. We can approximate these quantities via the sample mean

$$\widehat{\mu}_0 := \frac{1}{m}\sum_{i=1}^{m} Y^{(i)}, \tag{4.12}$$

$$\widehat{\mu}_j := \frac{1}{m}\sum_{i=1}^{m} Y^{(i)} X_j^{(i)}, \quad j \in \{1, \cdots, n\}. \tag{4.13}$$

By the weak law of large numbers (Theorem 30), these estimates converge in probability to their expected value as $m \to \infty$ which are precisely the Chow parameters. In other words we have

$$\widehat{\mu}_j \xrightarrow{P} \widehat{f}(j), \quad j \in \{0, \cdots, n\}. \tag{4.14}$$

Of course, for any finite m, the estimates are not in general exactly equal to the Chow parameters, so Chow's theorem does not apply to the estimated Chow parameters and the corresponding function with those Fourier coefficients is not necessarily a linear threshold function. Furthermore, because Chow's theorem does not apply, there is no guarantee that the function with such Fourier coefficients is unique, as only $n+1$ out of the 2^n Fourier coefficients are specified. What is more, this function may not even output values in $\{-1, 1\}$, but instead output values in $\mathbb{R}$.

Fortunately, we will see in Section 4.6 that there is an algorithm that, given approximate Chow parameters, can construct an LTF that is close to the true LTF, provided the Chow parameters can be estimated to sufficiently high accuracy as will be the case for large enough m.

4.3 Mutual information bound

We will now turn to our information-theoretic framework to derive generalisation bounds for our algorithm that learns an LTF through approximating the Chow parameters. Formally, our algorithm takes as input $Z = ((X^{(1)}, Y^{(1)}), \cdots, (X^{(m)}, Y^{(m)}))$ as described above and computes $\widehat{\mu} := (\widehat{\mu}_0, \cdots, \widehat{\mu}_n)$ as in Equation 4.12 and Equation 4.13. This is then fed into a "black box" $\mathcal{A}(\widehat{\mu})$, which we will discuss in Section 4.6, that outputs an LTF corresponding to the approximate Chow parameters $\widehat{\mu}$.

Our main result of this chapter, stated below, is a bound on the expected generalisation error of this algorithm.

THEOREM 14 (**Our result**). *The Chow parameter LTF learner $\mathcal{A}$ described above has expected generalisation error*

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \Big[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \Big] \right| \leq O\left(\sqrt{\frac{n \log m}{m}} \right). \tag{4.15}$$

This result is derived by using the mutual information framework described in Section 3.2 and bounding the quantity $I(Z; \mathcal{A}(\widehat{\mu}(Z)))$. We will now go through the details of proving this statement.

First, note that by the data processing inequality (Theorem 7), we have

$$I(Z; \mathcal{A}(\widehat{\mu}(Z))) \leq I(Z; \widehat{\mu}(Z)). \tag{4.16}$$

But since the feature space $\mathcal{X} = \{-1, 1\}^n$ is discrete and the learning algorithm $\widehat{\mu}$ deterministic, we have

$$I(Z; \widehat{\mu}(Z)) = H(\widehat{\mu}), \tag{4.17}$$

by the arguments made in Equation 3.64.

This can be further simplified since

$$H(\widehat{\mu}) \leq \sum_{j=0}^{n} H(\widehat{\mu}_j) \tag{4.18}$$

$$= \sum_{j=0}^{n} H(m\widehat{\mu}_j) \tag{4.19}$$

by Lemma 5 and Lemma 2 respectively. Note that $m\widehat{\mu}_j$ are binomial random variables with m trials and success probability equal to $\underset{X \sim \{-1,1\}^n}{\mathbf{P}}[f(X) = 1]$ for $j = 0$ and $\underset{X \sim \{-1,1\}^n}{\mathbf{P}}[f(X)X_j = 1]$ for $j \in \{1, \cdots, n\}$.

Unfortunately, there is no exact form for the entropy of a binomial random variable. One way to get around this issue is to settle for approximations or asymptotic analyses of the entropy instead. A very natural idea then is to note that by the central limit theorem (Theorem 31), binomial random variables (appropriately scaled) converge in distribution to a normal distribution as $m \to \infty$. This leads us to suspect that there might be some relationship between the differential entropy of a normal random variable and the Shannon entropy of a binomial random variable, at least asymptotically.

In the *continuous* case, we will show later on (see Theorem 27) that the *differential* entropy of any continuous random variable is upper bounded by that of a normal having the same variance. We will now show that an analogous result is also true in the discrete case.

The following is an argument by Massey (1988). Let X be a discrete random variable taking integer values (e.g. a binomial random variable) and let $p(k) := \mathbf{P}[X = k]$. Define the *continuous* random variable $\widetilde{X}$ as the random variable having density $f_{\widetilde{X}}(x) = p(k)$ whenever $x \in (k - \frac{1}{2}, k + \frac{1}{2}]$ for every $k \in \mathbb{Z}$. Clearly $h(\widetilde{X}) = H(X)$ by definitions of differential and Shannon entropy. However, note that since $\widetilde{X}$ is now continuous, we can bound $h(\widetilde{X})$ using Theorem 27, which then gives us a bound on

$H(X)$ as well. To do so, we need to compute the variance of $\widetilde{X}$, which we will do by computing its first and second moments. The first moment is

$$\begin{aligned}
\mathbf{E}[\widetilde{X}] &= \int_{\mathbb{R}} x f_{X'}(x)\, dx && (4.20)\\
&= \sum_{k\in\mathbb{Z}} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} xp(k)dx && (4.21)\\
&= \sum_{k\in\mathbb{Z}} p(k)\frac{1}{2}\left[\left(k+\frac{1}{2}\right)^2 - \left(k-\frac{1}{2}\right)^2\right] && (4.22)\\
&= \sum_{k\in\mathbb{Z}} p(k)\frac{1}{2}[2k] && (4.23)\\
&= \mathbf{E}[X]. && (4.24)
\end{aligned}$$

Similarly, the second moment is

$$\begin{aligned}
\mathbf{E}[\widetilde{X}^2] &= \sum_{k\in\mathbb{Z}} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} x^2 p(k)dx && (4.25)\\
&= \sum_{k\in\mathbb{Z}} p(k)\frac{1}{3}\left[\left(k+\frac{1}{2}\right)^3 - \left(k-\frac{1}{2}\right)^3\right] && (4.26)\\
&= \sum_{k\in\mathbb{Z}} p(k)\frac{1}{3}\left[3k^2+\frac{1}{4}\right] && (4.27)\\
&= \mathbf{E}[X^2] + \frac{1}{12}. && (4.28)
\end{aligned}$$

Hence,

$$\mathrm{Var}[\widetilde{X}] = \mathbf{E}[X^2] + \frac{1}{12} - \mathbf{E}[X]^2 = \mathrm{Var}[X] + \frac{1}{12}. \tag{4.29}$$

The significance of the $\frac{1}{12}$ term is that it is the variance of a uniform distribution over an interval of size 1. In fact, we can derive the above result in a different way that more clearly demonstrates this idea.

Notice that the distribution of $\widetilde{X}$ can be written as $\mathcal{U}((X-\frac{1}{2}, X+\frac{1}{2}])$. In particular, this implies

$$\mathbf{E}[\widetilde{X} \mid X] = X. \tag{4.30}$$

Then by the law of total expectation,

$$\mathbf{E}[\widetilde{X}] = \mathbf{E}[\mathbf{E}[\widetilde{X} \mid X]] = \mathbf{E}[X], \tag{4.31}$$

but also, by the law of total variance,

$$\operatorname{Var}[\widetilde{X}] = \mathbf{E}[\operatorname{Var}[\widetilde{X} \mid X]] + \operatorname{Var}[\mathbf{E}[\widetilde{X} \mid X]] \tag{4.32}$$

$$= \mathbf{E}\left[\frac{1}{12}\right] + \operatorname{Var}[X] \tag{4.33}$$

where the first term comes from the fact that $\widetilde{X} \mid X$ is uniform on an interval of length 1.

Finally, we can apply Theorem 27 to bound the differential entropy of $\widetilde{X}$, giving us the following result.

THEOREM 15. *Let X be a integer-valued discrete random variable with variance σ^2. Then*

$$H(X) < \frac{1}{2}\ln\left(2\pi e\left(\sigma^2 + \frac{1}{12}\right)\right). \tag{4.34}$$

Moreover, if $X \sim \text{Binomial}(m, p)$ then $\operatorname{Var}[X] = mp(1-p) \leq m/4$ and so

$$H(X) < \frac{1}{2}\ln\left(2\pi e\left(\frac{m}{4} + \frac{1}{12}\right)\right). \tag{4.35}$$

Applying this to our original problem of the Chow estimates, we obtain

$$I(Z; \widehat{\mu}(Z)) = \sum_{j=0}^{n} H(m\widehat{\mu}_j) < \frac{n+1}{2}\ln\left(2\pi e\left(\frac{m}{4} + \frac{1}{12}\right)\right) \tag{4.36}$$

$$= O\left(n \log m\right). \tag{4.37}$$

Combining this with Equation 4.16 and Theorem 9, gives us the desired result of Theorem 14.

4.4 Conditional mutual information bound

In this section we explore if we can get a better bound by using the conditional mutual information framework (Steinke and Zakynthinou, 2020) discussed in Section 3.4.

We have,

$$I\left(S;\widehat{\mu}(\widetilde{Z}_S)\mid\widetilde{Z}\right) := \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[I(S|\widetilde{Z}=\widetilde{z};\widehat{\mu}(\widetilde{Z}_S)|\widetilde{Z}=\widetilde{z})\right] \tag{4.38}$$

$$= \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[I(S;\widehat{\mu}(\widetilde{z}_S))\right] \tag{4.39}$$

$$= \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[H(\widehat{\mu}(\widetilde{z}_S)) - H(\widehat{\mu}(\widetilde{z}_S)|S)\right] \tag{4.40}$$

$$= \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[H(\widehat{\mu}(\widetilde{z}_S)) - \mathop{\mathbf{E}}_{s\sim\{-1,1\}^n}\left[H(\widehat{\mu}(\widetilde{z}_S)|S=s)\right]\right] \tag{4.41}$$

$$= \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[H(\widehat{\mu}(\widetilde{z}_S)) - 0\right] \tag{4.42}$$

$$\leq \sum_{j=0}^{n}\mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[H(\widehat{\mu}_j(\widetilde{z}_S))\right] \tag{4.43}$$

$$= \sum_{j=0}^{n}\mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[H(m\widehat{\mu}_j(\widetilde{z}_S))\right]. \tag{4.44}$$

where Equation 4.40 is by Lemma 6, Equation 4.42 is because $\widehat{\mu}$ is deterministic, Equation 4.43 is by Lemma 5, and Equation 4.44 is by Lemma 2.

Let $\widetilde{z}\in\mathcal{Z}^{2\times m}$ be fixed and write

$$\widetilde{z} = \begin{pmatrix}(\widetilde{x}^{(0,1)},\widetilde{y}^{(0,1)}) & \cdots & (\widetilde{x}^{(0,m)},\widetilde{y}^{(0,m)})\\ (\widetilde{x}^{(1,1)},\widetilde{y}^{(1,1)}) & \cdots & (\widetilde{x}^{(1,m)},\widetilde{y}^{(1,m)})\end{pmatrix} \tag{4.45}$$

so that $m\widehat{\mu}_j(\widetilde{z}_S)$ can be expressed as

$$m\widehat{\mu}_j(\widetilde{z}_S) = \sum_{i=1}^{n}\widetilde{y}^{(S_i,i)}\widetilde{x}_j^{(S_i,i)} \tag{4.46}$$

for $j\in\{1,\cdots,m\}$ and similarly for the case $j=0$. Now, for those i such that $\widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} = \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)}$, the random value of S_i does not affect the corresponding summand. The randomness in S only affects i for which $\widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} \neq \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)}$ and so

$$\begin{aligned} m\widehat{\mu}_j(\widetilde{z}_S) \sim{}& \left|\left\{i\in[m] : \widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} = \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)} = 1\right\}\right| \\ &- \left|\left\{i\in[m] : \widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} = \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)} = -1\right\}\right| \\ &+ \text{Binomial}\left(\left|\left\{i\in[m] : \widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} \neq \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)}\right\}\right|, \frac{1}{2}\right). \end{aligned} \tag{4.47}$$

The first two lines in the right hand side are constants, which does not affect the entropy of $m\widehat{\mu}_j(\widetilde{z}_S)$. By our work from the previous section (in particular Equation 4.35), we get

$$H\left(m\widehat{\mu}_j(\widetilde{z}_S)\right) < \frac{1}{2}\ln\left(2\pi e\left(\frac{c_j(\widetilde{z})}{4}+\frac{1}{12}\right)\right) \tag{4.48}$$

where we define

$$c_j(\widetilde{z}) := \left|\left\{i \in [m] : \widetilde{y}^{(0,i)}\widetilde{x}_j^{(0,i)} \neq \widetilde{y}^{(1,i)}\widetilde{x}_j^{(1,i)}\right\}\right|. \tag{4.49}$$

Hence, from Equation 4.44,

$$I\left(S;\widehat{\mu}(\widetilde{Z}_S) \mid \widetilde{Z}\right) < \frac{n+1}{2}\ln(2\pi e) + \frac{1}{2}\sum_{j=0}^{n} \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[\ln\left(\frac{c_j(\widetilde{z})}{4}+\frac{1}{12}\right)\right]. \tag{4.50}$$

Then by Jensen's inequality,

$$I\left(S;\widehat{\mu}(\widetilde{Z}_S) \mid \widetilde{Z}\right) < \frac{n+1}{2}\ln(2\pi e) + \frac{1}{2}\sum_{j=0}^{n} \ln\left(\mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[\frac{c_j(\widetilde{z})}{4}+\frac{1}{12}\right]\right). \tag{4.51}$$

But since the samples in $\widetilde{z}$ are i.i.d., we have

$$\mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}}\left[c_j(\widetilde{z})\right] = m \mathop{\mathbf{P}}_{x,x'\sim\{-1,1\}^n}\left[f(x)x_j \neq f(x')x'_j\right] =: mp_j^f \tag{4.52}$$

where the probability is taken over two i.i.d. samples x and x'. Hence we get

$$I\left(S;\widehat{\mu}(\widetilde{Z}_S) \mid \widetilde{Z}\right) < \frac{n+1}{2}\ln(2\pi e) + \frac{1}{2}\sum_{j=0}^{n} \ln\left(\frac{p_j^f}{4}m+\frac{1}{12}\right). \tag{4.53}$$

Using the very loose bound $p_j^f \leq 1$ we get

$$I\left(S;\widehat{\mu}(\widetilde{Z}_S) \mid \widetilde{Z}\right) < \frac{n+1}{2}\ln(2\pi e) + \frac{1}{2}\sum_{j=0}^{n} \ln\left(\frac{m}{4}+\frac{1}{12}\right) = O(n\log m) \tag{4.54}$$

which recovers the same (asymptotic) bound as we got using the mutual information based analysis in the previous section, and hence resulting in the same (asymptotic) expected generalisation error bound of $O\left(\sqrt{\frac{n\log m}{m}}\right)$ by Theorem 11.

However, Equation 4.53 allows us to perform a more fine-grained analysis that is based on the behaviour of the particular function f. An interesting direction for further work could be to identify classes of LTFs for which this analysis results in tighter bounds.

4.5 Proof of Chow's theorem

In this section, we will prove Chow's theorem (Theorem 13), which is the key result in motivating this line of work. Again, the material in this section is taken from O'Donnell (2021) and is summarised here for completeness.

First, we introduce an inner product between pairs of functions.

DEFINITION 17. *We define an inner product $\langle \cdot, \cdot \rangle$ on pairs of functions $f, g : \{-1,1\}^n \to \mathbb{R}$ by*

$$\langle f, g \rangle := \sum_{x \in \{-1,1\}^n} \frac{1}{2^n} f(x)g(x) = \mathop{\mathbf{E}}_{X \sim \{-1,1\}^n} [f(X)g(X)] . \tag{4.55}$$

The inner product is defined in such a way because this results in the bases χ_S being *orthonormal*.

THEOREM 16. *The 2^n parity functions $\chi_S : \{-1,1\}^n \to \{-1,1\}$ form an orthonormal basis for V, i.e.,*

$$\langle \chi_S, \chi_T \rangle = \mathbb{1}\{S = T\} \tag{4.56}$$

PROOF. Suppose $S \neq T$. Letting $S \triangle T := (S \setminus T) \cup (T \setminus S) \neq \emptyset$ denote the symmetric difference between two sets, we have

$$\begin{aligned} \chi_S(X)\chi_T(X) &= \prod_{i \in S} X_i \prod_{i \in T} X_i && (4.57)\\ &= \prod_{i \in S \cap T} X_i^2 \prod_{i \in S \triangle T} X_i && (4.58)\\ &= \prod_{i \in S \triangle T} X_i. && (4.59) \end{aligned}$$

Taking expectations we obtain

$$\begin{aligned} \mathbf{E}[\chi_S(x)\chi_T(X)] &= \mathbf{E}\left[\prod_{i \in S \triangle T} X_i\right] && (4.60)\\ &= \prod_{i \in S \triangle T} \mathbf{E}[X_i], && (4.61) \end{aligned}$$

where Equation 4.61 follows by independence of each X_i. Now, X_i are Rademacher ± 1 random variables so $\mathbf{E}[X_i] = 0$. Hence $\mathbf{E}[\chi_S(X)\chi_S(T)] = 0$.

On the other hand, if $S = T$, then $\chi_S(X)\chi_T(X) = 1$ and so $\mathbf{E}[\chi_S(X)\chi_T(X)] = 1$. □

Having defined an inner product between two functions, we can define the *norm* of a function.

DEFINITION 18. *The (L_2) norm of a function $f : \{-1,1\}^n \to \mathbb{R}$ is*

$$\|f\|_2 := \sqrt{\langle f, f\rangle} = \sqrt{\mathop{\mathbf{E}}_{X\sim\{-1,1\}^n}[f(X)^2]}. \tag{4.62}$$

Note that for *Boolean-valued* functions $f : \{-1,1\}^n \to \{-1,1\}$ we always have $\|f\|_2 = 1$ since $f(X)^2$ is identically 1.

We can now prove Plancherel's theorem, which expresses the inner product between two functions as the sum of the product of the Fourier coefficients between the two functions.

THEOREM 17 (Plancherel's theorem). *For any $f, g : \{-1,1\}^n \to \mathbb{R}$,*

$$\langle f, g\rangle = \sum_{S\subseteq[n]} \widehat{f}(S)\widehat{g}(S). \tag{4.63}$$

PROOF.

$$\begin{aligned}
\langle f, g\rangle &= \mathbf{E}[f(X)g(X)] && (4.64)\\
&= \mathbf{E}\left[\left(\sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(X)\right)\left(\sum_{T\subseteq[n]} \widehat{g}(T)\chi_T(X)\right)\right] && (4.65)\\
&= \sum_{S\subseteq[n]}\sum_{T\subseteq[n]} \widehat{f}(S)\widehat{g}(T)\mathbf{E}[\chi_S(X)\chi_T(X)] && (4.66)\\
&= \sum_{S\subseteq[n]}\sum_{T\subseteq[n]} \widehat{f}(S)\widehat{g}(T)\mathbb{1}\{S = T\} && (4.67)\\
&= \sum_{S\subseteq[n]} \widehat{f}(S)\widehat{g}(S) && (4.68)
\end{aligned}$$

where Equation 4.65 is due to the Fourier expansion of f and g (Theorem 12), Equation 4.66 is due to linearity of expectation, and Equation 4.67 is due to the orthonormality of χ_S (Theorem 16). □

By taking $g = f$ in Plancherel's theorem gives an expression for the norm of f. This result is called Parseval's theorem.

THEOREM 18 (Parseval's theorem). *For any $f : \{-1,1\}^n \to \mathbb{R}$,*

$$\|f\|_2^2 = \mathop{\mathbf{E}}_{X\sim\{-1,1\}^n}\left[f(X)^2\right] = \sum_{S\subseteq[n]} \widehat{f}(S)^2. \tag{4.69}$$

Note that for *Boolean-valued* functions $f : \{-1,1\}^n \to \{-1,1\}$ we have $\|f\|_2 = 1$ and hence

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = 1. \tag{4.70}$$

We are now ready to prove Chow's theorem (Theorem 13). Recall the statement says that if $f, g : \{-1,1\}^n \to \{-1,1\}$ are two LTFs with $\widehat{f}(S) = \widehat{g}(S)$ for all $|S| \leq 1$ then $f = g$.

PROOF OF CHOW'S THEOREM. (O'Donnell, 2021) Since f is an LTF, then by definition we can write $f(x) = \text{sgn}(\ell(x))$ where $\ell : \{-1,1\}^n \to \mathbb{R}$ is given by $\ell(x) = w_0 + w_1 x_1 + \cdots + w_n x_n$. Without loss of generality, we may assume that ℓ is never 0 on $\{-1,1\}^n$ because if it is, we can perturb it slightly without changing the behaviour of f. For any $x \in \{-1,1\}^n$ we have

$$f(x)\ell(x) = \text{sgn}(\ell(x))\ell(x) \tag{4.71}$$

$$= |\ell(x)| \tag{4.72}$$

$$\geq g(x)\ell(x) \tag{4.73}$$

where the last inequality follows because $g(x) \in \{-1,1\}$. Hence it follows that

$$\mathbf{E}[f(X)\ell(X)] \geq \mathbf{E}[g(X)\ell(X)]. \tag{4.74}$$

Applying Parseval's theorem (Theorem 18) to both sides of the inequality implies that

$$\sum_{S \subseteq [n]} \widehat{f}(S)\widehat{\ell}(S) = \mathbf{E}[f(X)\ell(X)] \geq \mathbf{E}[g(X)\ell(X)] = \sum_{S \subseteq [n]} \widehat{g}(S)\widehat{\ell}(S). \tag{4.75}$$

But by assumption, $\widehat{f}(S) = \widehat{g}(S)$ for $|S| \leq 1$. On the other hand, because $\ell(x) = w_0 + w_1 x_1 + \cdots + w_n x_n$, we have $\widehat{\ell}(S) = 0$ for $|S| > 1$. Hence we must have equality in Equation 4.75, i.e.,

$$\mathbf{E}[f(X)\ell(X)] = \mathbf{E}[g(X)\ell(X)]. \tag{4.76}$$

At the same time, we also know that $f(x)\ell(x) \geq g(x)\ell(x)$ from Equation 4.73, but because the expectations are equal, we must have

$$f(x)\ell(x) = g(x)\ell(x) \tag{4.77}$$

for all $x \in \{-1,1\}^n$. This then implies $f(x) = g(x)$ for all $x \in \{-1,1\}^n$ because we assumed $\ell(x)$ is never zero on $\{-1,1\}^n$. □

4.6 From Chow estimates to an LTF

In Section 4.3 and Section 4.4, we successfully derived expected generalisation bounds for learning LTFs via the Chow parameters. Our analysis was focused on the estimates of the Chow parameters from the given samples, but we skimmed over the details of how to transform our Chow estimates into an LTF. In this section, we go over these details.

Recall that the learnt Chow parameters $\widehat{\mu}$ are only approximations of the true Chow parameters $\widehat{f}$, and so Chow's theorem does not apply to $\widehat{\mu}$. Moreover even if the theorem were to apply, the proof is nonconstructive in the sense that it does not tell us how to construct the LTF from its Chow parameters, which is what we would ultimately like to do.

Fortunately, the following result by O'Donnell and Servedio (2008) shows that given sufficiently accurate estimates of the Chow parameters, one can indeed construct an LTF that is very close to the true LTF, though this process is highly nontrivial.

THEOREM 19 (O'Donnell and Servedio (2008)). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be an LTF and $\widehat{f}$ its Chow parameters. There is a randomized algorithm $\mathcal{A}$ such that, when given approximate Chow parameters $\widehat{\mu}$ satisfying*

$$\|\widehat{\mu} - \widehat{f}\|_2 \leq 2^{-\widetilde{O}(1/\varepsilon^2)}, \tag{4.78}$$

outputs the weights-based representation of a LTF h that with probability at least $1-\delta'$ over $\mathcal{A}$ satisfies

$$\mathop{\mathbf{P}}_{X\sim\{-1,1\}^n}[f(X) \neq h(X)] \leq \varepsilon. \tag{4.79}$$

and has time complexity

$$2^{poly\left(2^{\widetilde{O}(1/\varepsilon^2)}\right)} n^2 \log n \log \frac{n}{\delta'}. \tag{4.80}$$

To use this result, let us use enough samples for $\widehat{\mu}$ so that Equation 4.78 is satisfied with failure probability no more than $\delta/2$ and let us run $\mathcal{A}$ until its failure probability is no more than $\delta/2$ as well (i.e. $\delta' = \delta/2$). By the union bound, this results in an algorithm that, with failure probability no more than δ, outputs an LTF h with

$$\mathop{\mathbf{P}}_{X\sim\{-1,1\}^n}[f(X) \neq h(X)] \leq \varepsilon. \tag{4.81}$$

In other words, the process of computing the estimates $\widehat{\mu}$ and applying Theorem 19 is a proper PAC learner for LTFs in the realisable setting *under the uniform Boolean hypercube*, i.e., $\mathcal{D}_{\mathcal{X}} = \mathcal{U}(\{-1,1\}^n)$,

though of course it is not an *efficient* one because the dependence on $1/\varepsilon$ is doubly exponential. However, we emphasise that for a fixed ε, the time complexity dependence on n is quadratic.

How many samples do we need to satisfy Equation 4.78 with failure probability no more than $\frac{\delta}{2}$? For each $j \in \{0, 1, \cdots, n\}$, let us ensure $\widehat{\mu}_j$ is within an additive $(n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)}$ to $\widehat{f}(j)$ with failure probability no greater than $\frac{\delta}{2(n+1)}$. In other words,

$$\mathbf{P}\left[\left|\widehat{\mu}_j - \widehat{f}(j)\right| \leq (n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)}\right] \geq 1 - \frac{\delta}{2(n+1)}. \tag{4.82}$$

By Theorem 35 this can be done with no more than

$$m = O\left(n \cdot 2^{\widetilde{O}(1/\varepsilon^2)} \cdot \log \frac{n}{\delta}\right) \tag{4.83}$$

samples.

The probability that *all* $n+1$ estimates are within the prescribed additive range is then

$$\mathbf{P}\left[\bigcap_{j=0}^{n}\left\{\left|\widehat{\mu}_j - \widehat{f}(j)\right| \leq (n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)}\right\}\right] \tag{4.84}$$

$$= 1 - \mathbf{P}\left[\bigcup_{j=0}^{n}\left\{\left|\widehat{\mu}_j - \widehat{f}(j)\right| > (n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)}\right\}\right] \tag{4.85}$$

$$\geq 1 - \sum_{j=0}^{n}\mathbf{P}\left[\left|\widehat{\mu}_j - \widehat{f}(j)\right| > (n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)}\right] \tag{4.86}$$

$$\geq 1 - (n+1)\cdot\frac{\delta}{2(n+1)} \tag{4.87}$$

$$= 1 - \frac{\delta}{2}. \tag{4.88}$$

But if

$$\left|\widehat{\mu}_j - \widehat{f}(j)\right| \leq (n+1)^{-1/2}2^{-\widetilde{O}(1/\varepsilon^2)} \tag{4.89}$$

for all $j \in \{0, \cdots, n\}$ then this implies

$$\sum_{j=0}^{n}\left(\widehat{\mu}_j - \widehat{f}(j)\right)^2 \leq 2^{-\widetilde{O}(1/\varepsilon^2)} \tag{4.90}$$

and so

$$\|\widehat{\mu} - f\|_2 \leq 2^{-\widetilde{O}(1/\varepsilon^2)} \tag{4.91}$$

as required.

From Equation 4.83, we see the sample complexity dependence is exponential in $1/\varepsilon$, but we emphasize that the dependence on n is near-linear.

4.7 Extension to polynomial threshold functions

The ideas in this chapter can be trivially generalised to *polynomial threshold functions* (PTFs). A Boolean-valued function f is a PTF of degree at most k if it is expressible as

$$f(x) = \text{sgn}(p(x)) \tag{4.92}$$

for some real polynomial $p(x)$ of degree at most k.

Chow's theorem can be generalised to PTFs, with the main ideas in the proof staying the same.

THEOREM 20 (O'Donnell (2021)). *Let $f, g : \{-1,1\}^n \to \{-1,1\}$ be PTFs of degree at most k. Suppose that $\widehat{f}(S) = \widehat{g}(S)$ for all $|S| \leq k$. Then $f = g$.*

Just like we did for the case of LTFs, we can estimate the Fourier coefficients $\widehat{\mu}(S)$ for $|S| \leq k$. A result by Diakonikolas and Kane (2019) generalises Theorem 19 to the case of PTFs. Specifically, it states there is an algorithm that can approximately reconstruct f from $\widehat{\mu}(S)$ provided the Fourier estimates $\widehat{\mu}(S)$ are close enough to $\widehat{f}(S)$.

Already for the case of LTFs we saw that the distance requirement $\|\widehat{\mu} - \widehat{f}\|_2$ is inversely exponential in $1/\varepsilon$. It should be no surprise then that for the general case of PTFs, this distance requirement becomes staggeringly small and turns out to be inversely proportional to the Ackermann function (Diakonikolas and Kane, 2019).

However, from our information-theoretic framework, the analysis remains the same as before except we estimate $O(n^k)$ Fourier coefficients instead of $n+1$. Thus the algorithm that estimates the Fourier coefficients and reconstructs the PTF has expected generalisation error

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \Big[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \Big] \right| \leq O\left(\sqrt{\frac{n^k \log m}{m}} \right). \tag{4.93}$$

CHAPTER 5

Applications to the LMN algorithm

In the previous chapter we saw that we could learn LTFs over the Boolean hypercube via estimating the Chow parameters. Using the information theory toolkit developed in Chapter 3, we showed that the expected generalisation error $\left|\mathop{\mathbf{E}}_{Z,\mathcal{A}}\left[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z))\right]\right|$ is bounded by $O\left(\sqrt{\frac{n \log m}{m}}\right)$, however learning to population risk $R(\mathcal{A}(Z)) < \varepsilon$ had sample complexity m exponential in $1/\varepsilon$ and time complexity doubly exponential in $1/\varepsilon$.

In this short chapter, we will show that our analysis can be applied to a well known learning algorithm by Linial et al. (1993). The high level idea of their algorithm is to estimate the Fourier coefficients of functions for which the large coefficients are provably located on a small number of subsets. Our main result in this chapter is Theorem 22.

5.1 LMN algorithm

To study this idea in more detail, recall that by Parseval's theorem (Theorem 18),

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = 1 \tag{5.1}$$

for any Boolean-valued $f : \{-1, 1\}^n \to \{-1, 1\}$.

The squared Fourier coefficients $\widehat{f}(S)^2$ are called the *Fourier weights*. Suppose that the Fourier weights are concentrated on a small number of subsets $\mathcal{F}$, as formalized by the following definition.

DEFINITION 19. *Let $\mathcal{F}$ be a collection of subsets of $[n]$. We say the Fourier weights of $f : \{-1, 1\}^n \to \{-1, 1\}$ are ε-concentrated on $\mathcal{F}$ if*

$$\sum_{S \subseteq [n], S \notin \mathcal{F}} \widehat{f}(S)^2 \leq \varepsilon. \tag{5.2}$$

We know that f has Fourier expansion

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x). \tag{5.3}$$

A simple idea by Linial, Mansour, and Nisan (1993) to learn f is the following. We can estimate the Fourier coefficients $\widehat{f}(S)$ for $S \in \mathcal{F}$ like we did for estimating the Chow parameters. Let $\widehat{\mu}(S)$ denote these estimates. Since the Fourier weight of f is ε concentrated on $\mathcal{F}$, we know that $\widehat{f}(S)$ for $S \notin \mathcal{F}$ is close to zero, so let us use the approximation $\widehat{\mu}(S) := 0$ for $S \notin \mathcal{F}$. We can then construct an approximate Fourier expansion of f given by

$$g(x) := \sum_{S \in \mathcal{F}} \widehat{\mu}(S)\chi_S(x) + \sum_{S \notin \mathcal{F}} 0 \cdot \chi_S(x). \tag{5.4}$$

The function g does not necessarily output values in $\{-1, 1\}$ so let us take the *sign* of g instead. These steps are summarized in Algorithm 1.

Algorithm 1 LMN Algorithm (Linial, Mansour, and Nisan, 1993)

1: Estimate the Fourier coefficients for each $S \in \mathcal{F}$, i.e., compute the sample means

$$\widehat{\mu}(S) := \frac{1}{m}\sum_{i=1}^{m} f(X^{(i)})\chi_S(X^{(i)}) = \frac{1}{m}\sum_{i=1}^{m} Y^{(i)} \prod_{j \in S} X_j^{(i)}. \tag{5.5}$$

2: Form the function $g : \{-1, 1\}^n \to \mathbb{R}$ given by

$$g(x) := \sum_{S \in \mathcal{F}} \widehat{\mu}(S)\chi_S(x). \tag{5.6}$$

3: Output

$$h(x) := \text{sgn}(g(x)). \tag{5.7}$$

It turns out that this algorithm is a PAC learner for f, provided we can find such a set $\mathcal{F}$.

THEOREM 21 (Linial et al. (1993); O'Donnell (2021)). *Suppose $f : \{-1, 1\}^n \to \{-1, 1\}$ is $\varepsilon/2$ concentrated on $\mathcal{F}$. Then, Algorithm 1, when given*

$$m \geq O\left(\frac{|\mathcal{F}|}{\varepsilon} \log \frac{|\mathcal{F}|}{\delta}\right) \tag{5.8}$$

i.i.d. samples over $\{-1, 1\}^n$ labelled by f, outputs a function h that with probability at least $1 - \delta$ satisfies

$$\mathbf{P}[f(X) \neq h(X)] \leq \varepsilon. \tag{5.9}$$

In other words, the LMN algorithm is a proper PAC learner in the realisable setting *under the uniform Boolean hypercube*, i.e., $\mathcal{D}_{\mathcal{X}} = \mathcal{U}(\{-1,1\}^n)$.

We will prove this result in Section 5.3.

For now, notice that the Fourier coefficients are approximated in the exact same way as we did for the Chow coefficients, except that we do this more generally for sets in $\mathcal{F}$ rather than only for those sets S with $|S| \leq 1$. However, the key difference to the LTF case is that although the degree 0 and degree 1 Fourier coefficients uniquely determine an LTF, those sets are **not** an ε-concentration for the Fourier weights. Hence computing $g(x)$ and $h(x)$ as do we here would not be a good approximation for an LTF, and indeed recovering the LTF from the Chow parameters is a lot more difficult as we saw in Section 4.6.

From the perspective of our information-theoretic framework however, the analysis of the LMN algorithm is almost identical to Section 4.3 except that we have $|\mathcal{F}|$ estimates instead of $n+1$ and hence

$$I(Z;\widehat{\mu}(Z)) \leq \sum_{S\in\mathcal{F}} H(m\widehat{\mu}(S)) \tag{5.10}$$

$$< \frac{|\mathcal{F}|}{2}\ln\left(2\pi e\left(\frac{m}{4}+\frac{1}{12}\right)\right) \tag{5.11}$$

$$= O(|\mathcal{F}|\log m). \tag{5.12}$$

Let $\mathcal{A}(\widehat{\mu}(Z))$ denote the entire LMN algorithm. Again by the data processing inequality

$$I(Z;\mathcal{A}(\widehat{\mu}(Z))) \leq I(Z;\widehat{\mu}(Z)) \tag{5.13}$$

and so by Theorem 9 we arrive at the following result.

THEOREM 22 (**Our result**). *The LMN algorithm $\mathcal{A}$ has expected generalisation error*

$$\left|\mathop{\mathbf{E}}_{Z,\mathcal{A}}\left[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z))\right]\right| = O\left(\sqrt{\frac{|\mathcal{F}|\log m}{m}}\right) \tag{5.14}$$

where $\mathcal{F}$ are the subsets for which the Fourier weights are ε-concentrated on.

5.2 Functions with concentrated Fourier weights

In the previous section we saw that if we could identify sets $\mathcal{F}$ for which the Fourier weights of f are $\varepsilon/2$ concentrated in, then this could be used in a learning algorithm for f that had sample complexity

dependence $O\left(\frac{|\mathcal{F}|}{\varepsilon}\log\frac{|\mathcal{F}|}{\delta}\right)$ and expected generalisation error $O\left(\sqrt{\frac{|\mathcal{F}|\log m}{m}}\right)$. This is great news if we could somehow find such a set $\mathcal{F}$. In this section, we summarize some material from O'Donnell (2021) stating that for various classes of functions, $\mathcal{F}$ consists of all sets with low cardinality, i.e., $\mathcal{F} = \{S \subseteq [n] : |S| \leq k\}$ for some k, and hence $|\mathcal{F}| = O(n^k)$. These results then allow us to very easily bound the expected generalisation error of the LMN algorithm for these classes of functions via Equation 5.14.

5.2.1 Functions with low influence

The influence of coordinate i on a Boolean-valued function $f : \{-1,1\}^n \to \{-1,1\}$ is the probability that changing the ith bit changes the output of the function,

$$\text{Inf}_i[f] := \mathop{\mathbf{P}}_{X\sim\{-1,1\}^n}\left[f(X) \neq f(X^{\oplus i})\right] \tag{5.15}$$

where $X^{\oplus i}$ means X but with the ith bit flipped.

The **total influence** of $f : \{-1,1\}^n \to \{-1,1\}$ is the sum of all its influences

$$I[f] := \sum_{i=1}^{n} \text{Inf}_i[f]. \tag{5.16}$$

The following lemma relates the influence of a function f to the size of the sets that f is ε concentrated on.

LEMMA 10. *For any $f : \{-1,1\}^n \to \{-1,1\}$ and any $\varepsilon > 0$, the Fourier weights of f are ε concentrated on sets with cardinality up to $I[f]/\varepsilon$.*

Consider the concept class $\mathcal{C}$ of Boolean functions with $I[f] \leq t$. By the above result, every function in $\mathcal{C}$ has its Fourier weights $\varepsilon/2$ concentrated on sets with cardinality up to $k = 2t/\varepsilon$. Hence the LMN algorithm can be used to learn $\mathcal{C}$ by setting $\mathcal{F} := \{S \subseteq [n] : |S| \leq \frac{2t}{\varepsilon}\}$. This achieves sample complexity

$$m = \frac{n^{O(t/\varepsilon)}}{\varepsilon}\log\frac{n^{O(t/\varepsilon)}}{\delta}, \tag{5.17}$$

and expected generalisation error

$$\sqrt{\frac{n^{O(t/\varepsilon)}\log m}{m}}. \tag{5.18}$$

Notice that the expected generalisation error here is dependent on ε. This is because, in order to learn to smaller ε, we need to set $\mathcal{F}$ to contain sets with larger cardinality — in particular the maximum cardinality of sets in $\mathcal{F}$ scales with $1/\varepsilon$. Hence $|\mathcal{F}|$ is a function of ε and so the expected generalisation error is also a function of ε.

5.2.2 Monotone functions

An important class of functions with low influence are the class of monotone functions. Such functions are quite common and natural, and in particular encompass most "reasonable" voting rules, i.e., where switching votes from one candidate to the other cannot cause the second to lose the election.

DEFINITION 20. *A function $f : \{-1,1\}^n \to \{-1,1\}$ is monotone if $f(x) \leq f(y)$ whenever $x \leq y$ coordinate-wise.*

LEMMA 11. *For monotone f,*

$$I[f] \leq \sqrt{\frac{2}{\pi}} n^{1/2} + O(n^{-1/2}) \tag{5.19}$$

with equality if f is the majority function over n elements.

Consequently, by Lemma 10, monotone functions have Fourier weights $\varepsilon/2$ concentrated on sets up to degree

$$\frac{2}{\varepsilon}\left(\sqrt{\frac{2}{\pi}} n^{1/2} + O(n^{-1/2})\right) = O\left(\frac{\sqrt{n}}{\varepsilon}\right). \tag{5.20}$$

Hence the LMN algorithm, when learning monotone functions, has sample complexity

$$\frac{n^{O(\sqrt{n}/\varepsilon)}}{\varepsilon} \log \frac{n^{O(\sqrt{n}/\varepsilon)}}{\delta} \tag{5.21}$$

and expected generalisation error no more than

$$\sqrt{\frac{n^{O(\sqrt{n}/\varepsilon)} \log m}{m}}. \tag{5.22}$$

5.2.3 Functions with low noise sensitivity

Another technique to show Fourier weight concentration is to look at the *sensitivity* of a function to noise in its input.

DEFINITION 21. *For $f : \{-1,1\}^n \to \{-1,1\}$, the noise sensitivity of f at γ, denoted $NS_\gamma[f]$ is the probability that $f(X) \neq f(Y)$ when $X \sim \{-1,1\}^n$ and Y is formed by flipping each bit in X independently with probability γ.*

The following lemma relates a function's noise sensitivity to the size of the subsets that its Fourier weights are concentrated on.

LEMMA 12. *The Fourier weights of $f : \{-1,1\}^n \to \{-1,1\}$ are $3NS_\gamma[f]$ concentrated on sets with cardinality at most $1/\gamma$.*

Using this result, we can learn functions with low noise sensitivity. Consider the concept class $\mathcal{C}$ of Boolean functions f having $\mathrm{NS}_\gamma[f] \leq \varepsilon/6$. Then f is $3\mathrm{NS}_\gamma[f] \leq \varepsilon/2$ concentrated on sets with cardinality at most $1/\gamma$ and so the LMN algorithm when used to learn $\mathcal{C}$ has sample complexity

$$O\left(\frac{n^{1/\gamma}}{\varepsilon}\log\frac{n^{1/\gamma}}{\delta}\right), \tag{5.23}$$

and by our result, has expected generalisation error no more than

$$O\left(\sqrt{\frac{n^{1/\gamma}\log m}{m}}\right). \tag{5.24}$$

5.2.4 Peres' theorem and LTFs revisited

A result by Peres (2021) bounds the noise stability of LTFs and hence by the previous result we can learn LTFs *via the LMN algorithm*.

THEOREM 23 (Peres (2021)). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be an LTF. Then*

$$NS_\gamma[f] \leq O(\sqrt{\gamma}). \tag{5.25}$$

By Lemma 12, this implies that LTFs have their Fourier weights $\varepsilon/2$ concentrated on sets up to degree $O(1/\varepsilon^2)$ and so learning LTFs via this approach yields a sample complexity of

$$m = \frac{n^{O(1/\varepsilon^2)}}{\varepsilon}\log\frac{n^{O(1/\varepsilon^2)}}{\delta} \tag{5.26}$$

and an expected generalisation error of

$$\sqrt{\frac{n^{O(1/\varepsilon^2)} \log m}{m}}. \tag{5.27}$$

Unfortunately, this is not any better than the approach discussed in Chapter 4 where we learnt LTFs via the Chow parameters.

However, the noise sensitivity approach is much more flexible as it allows us to handle *compositions* of elementary functions such as LTFs. For example, consider the concept class $\mathcal{C}$ consisting of functions of the form $h = g(f_1, \cdots, f_s)$ where $f_1, \cdots, f_s : \{-1,1\}^n \to \{-1,1\}$ are LTFs. Peres' theorem can be used to show that $\mathrm{NS}_\gamma[h] \leq O(s\sqrt{\gamma})$. Hence $\mathcal{C}$ can be learnt via the LMN algorithm using

$$m = \frac{n^{O(s^2/\varepsilon^2)}}{\varepsilon} \log \frac{n^{O(s^2/\varepsilon^2)}}{\delta} \tag{5.28}$$

samples. This is the only known way of showing that a conjunction of two LTFs is learnable to constant error ε in time poly(n) (O'Donnell, 2021).

Our information-theoretic analysis complements this result by showing that this algorithm has expected generalisation error no more than

$$\sqrt{\frac{n^{O(s^2/\varepsilon^2)} \log m}{m}}. \tag{5.29}$$

5.2.5 Functions with constant Fourier degree

As our last example, we study functions $f(x)$ whose Fourier expansion $\sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x)$ has (polynomial) degree at most k, which is equivalent to requiring that $\widehat{f}(S) = 0$ for $|S| > k$. For example, it can be shown that decision trees of depth at most k satisfy this condition.

Such functions satisfy the property that each $\widehat{f}(S)$ is an integer multiple of 2^{1-k}. Consequently, if for each S with $|S| \leq k$, we learn $\widehat{f}(S)$ to within 2^{-k} then we can round our estimate to the nearest multiple of 2^{1-k} and return a hypothesis that has *zero* error. By Hoeffding's inequality (Theorem 35), this can be done with failure probability no more than δ using

$$m = O\left(2^{2k} \log \frac{2k}{\delta}\right) \tag{5.30}$$

samples. The expected generalisation error is then no more than

$$O\left(\sqrt{\frac{2^k \log m}{m}}\right). \tag{5.31}$$

To the best of our knowledge, this bound does not appear in the literature, and the ease with which we were able to obtain it demonstrates the versatility of our result.

5.3 Sample complexity of the LMN algorithm

We end this chapter by proving that the sample complexity of the LMN algorithm is $O\left(\frac{|\mathcal{F}|}{\varepsilon}\log\frac{|\mathcal{F}|}{\delta}\right)$ as stated in Theorem 21. The material in this section is taken from O'Donnell (2021) and we summarize it here for completeness.

PROOF OF THEOREM 21. We start the proof similarly to our analysis in Section 4.6, except that we want $\widehat{\mu}_S$ to be within an additive $\frac{1}{2}\sqrt{\frac{\varepsilon}{|\mathcal{F}|}}$ of $\widehat{f}_S$ with failure probability no greater than $\frac{\delta}{|\mathcal{F}|}$ for each $S \in \mathcal{F}$. Again, by Theorem 35, this can be achieved using the number of samples given in Equation 5.8.

Using the same union bound argument as in Section 4.6, this implies that

$$\mathbf{P}\left[\bigcap_{S\in\mathcal{F}}\left\{\left|\widehat{\mu}_S - \widehat{f}_S\right| \leq \frac{1}{2}\sqrt{\frac{\varepsilon}{|\mathcal{F}|}}\right\}\right] \geq 1-\delta. \tag{5.32}$$

Recall that the LMN algorithm computes $g(x) = \sum_{S\in\mathcal{F}}\widehat{\mu}(S)\chi_S(x)$. By Parseval's theorem (Theorem 18),

$$\|f-g\|_2^2 = \sum_{S\subseteq[n]}\widehat{f-g}(S)^2 \tag{5.33}$$

$$= \sum_{S\in\mathcal{F}}\left(\widehat{f}(S)-\widehat{\mu}(S)\right)^2 + \sum_{S\notin\mathcal{F}}\left(\widehat{f}(S)-0\right)^2. \tag{5.34}$$

By assumption, the Fourier weights of f are $\varepsilon/2$ concentrated on $\mathcal{F}$, so by definition $\sum_{S\notin\mathcal{F}}\widehat{f}(S)^2 \leq \varepsilon/2$. Furthermore, by Equation 5.32, with probability at least $1-\delta$ we have $\left(\widehat{f}(S)-\widehat{\mu}(S)\right)^2 \leq \frac{\varepsilon}{4|\mathcal{F}|}$ for all $S\in\mathcal{F}$ hence $\sum_{S\in\mathcal{F}}\left(\widehat{f}(S)-\widehat{\mu}(S)\right)^2 \leq \frac{\varepsilon}{4}$. Thus,

$$\|f-g\|_2^2 < \frac{\varepsilon}{4}+\frac{\varepsilon}{2} < \varepsilon \tag{5.35}$$

with probability at least $1-\delta$.

Finally, recall that the LMN algoritihm outputs $h(x) := \text{sgn}(g(x))$. Now,

$$\mathbf{P}[f(X) \neq h(X)] = \mathbf{E}\left[\mathbb{1}\left\{f(X) \neq \text{sgn}(g(X))\right\}\right]. \tag{5.36}$$

Note that if $f(X) \neq \text{sgn}(g(X))$ then either $f(X) = 1$ and $g(X) < 0$, or $f(X) = -1$ and $g(X) \geq 0$. In either case, $|f(X) - g(X)| \geq 1$ which implies

$$(f(X) - g(X))^2 \geq 1 = \mathbb{1}\left\{f(X) \neq \text{sgn}(g(X))\right\}. \tag{5.37}$$

Otherwise if $f(X) = \text{sgn}(g(X))$ then clearly

$$(f(X) - g(X))^2 \geq 0 = \mathbb{1}\left\{f(X) \neq \text{sgn}(g(X))\right\}. \tag{5.38}$$

Hence for all X, we have the inequality

$$\mathbb{1}\left\{f(X) \neq \text{sgn}(g(X))\right\} \leq (f(X) - g(X))^2 \tag{5.39}$$

and so,

$$\mathbf{E}\left[\mathbb{1}\left\{f(X) \neq \text{sgn}(g(X))\right\}\right] \leq \mathbf{E}\left[(f(X) - g(X))^2\right] \tag{5.40}$$

$$= \|f - g\|_2^2. \tag{5.41}$$

But we know from Equation 5.35 that $\|f - g\|_2^2 < \varepsilon$ with probability at least $1 - \delta$, completing the proof. □

CHAPTER 6

Extension to learning LTFs over $\mathbb{R}^n$

In Chapter 4 we have seen that LTFs $f : \{-1,1\}^n \to \{-1,1\}$ are characterised by their degree 0 and 1 Fourier coefficients which we can easily learn to get an approximation of f. In this chapter, we generalise this idea to *real-valued* LTFs instead, i.e., functions of the form

$$\begin{aligned} f : \mathbb{R}^n &\to \{-1,1\} \\ x &\mapsto \operatorname{sgn}(w_0 + w_1 x_1 + \cdots + w_n x_n). \end{aligned} \tag{6.1}$$

It turns out that many of the properties from the discrete case have analogues in this setting, if we assume that the distribution over $\mathcal{X}$ is the n-dimensional standard normal density instead of the uniform distribution over $\{-1,1\}^n$.

In particular, LTFs in this setting are again characterised by their degree 0 and degree 1 Hermite coefficients, the Gaussian analogue of Fourier coefficients. We again analyse a learning algorithm that learns LTFs by approximating their Hermite coefficients then reconstructing the LTF. Our main result in this chapter is Theorem 25, which states that this algorithm has an expected generalisation error of $O\left(\sqrt{\frac{n}{m}}\right)$. We also prove a secondary result, Theorem 29, which shows that reconstructing the LTF from Hermite estimates is much easier in this setting.

6.1 L^2 integrable functions and Hermite analysis

Instead of analysing functions of the form $f : \{-1,1\}^n \to \mathbb{R}$ as we did in Chapter 4, we will now analyse functions of the form $f : \mathbb{R}^n \to \mathbb{R}$ that are L^2 integrable under the n-dimensional standard Gaussian density φ_n, i.e., the set of functions in the function space

$$L^2(\mathbb{R}^n, \varphi_n) := \left\{ f : \mathbb{R}^n \to \mathbb{R} \mid \int_{\mathbb{R}^n} [f(x)]^2 \, \varphi_n(x) dx < \infty \right\}. \tag{6.2}$$

Many of the properties in the discrete uniform case continue to hold in this setting, which we summarise in this section. The results in this section are taken from O'Donnell (2021).

We define an analogous inner product between two functions in this space,

$$\langle f, g \rangle := \mathop{\mathbf{E}}_{X \sim \mathcal{N}(0,1)^{\otimes n}} [f(X)g(X)] = \int_{\mathbb{R}^n} f(x)g(x)\varphi_n(x)dx. \tag{6.3}$$

In the discrete case, we saw that the parity functions $\chi_S(x) := \prod_{i \in S} x_i$, which are just products of linear polynomials x_i for $i \in [n]$, formed an orthonormal basis over the set of functions $f : \{-1, 1\}^n \to \mathbb{R}$. In the Gaussian setting, it turns out that products of a certain class of polynomials known as the *Hermite polynomials* form an orthonormal basis for $L^2(\mathbb{R}^n, \varphi_n)$.

DEFINITION 22 (Hermite polynomials). *The Hermite polynomials are the univariate polynomials defined as*

$$h_n(x) := (-1)^n \exp\left(\frac{1}{2}x^2\right) \frac{d^n}{dx^n} \left[\exp\left(-\frac{1}{2}x^2\right)\right] \tag{6.4}$$

for $n \in \mathbb{N}_{\geq 0}$.

Some low order Hermite polynomials are

$$h_0(x) = 1, \tag{6.5}$$

$$h_1(x) = x, \tag{6.6}$$

$$h_2(x) = x^2 - 1, \tag{6.7}$$

$$h_3(x) = x^3 - 3x. \tag{6.8}$$

It can be shown that products of these polynomials form an orthonormal basis for $L^2(\mathbb{R}^n, \varphi_n)$.

THEOREM 24. *The family of multivariate polynomials*

$$H_S(x) := \prod_{i=1}^{n} h_{S_i}(x_i) \tag{6.9}$$

for $S \in \mathbb{N}^n_{\geq 0}$ *forms an orthonormal basis for* $L^2(\mathbb{R}^n, \varphi_n)$ *under the inner product defined in Equation 6.3.*

Moreover, as in the Boolean-valued case, any function $f \in L^2(\mathbb{R}^n, \varphi_n)$ can be written uniquely as

$$f(x) = \sum_{S \in \mathbb{N}^n} \widehat{f}(S) H_S(x) \tag{6.10}$$

where

$$\widehat{f}(S) := \langle f, H_S \rangle = \int_{\mathbb{R}^n} f(x) H_S(x) \varphi_n(x) dx = \mathop{\mathbf{E}}_{X \sim \mathcal{N}(0,1)^{\otimes n}} [f(X) H_S(X)] \tag{6.11}$$

is the *Hermite coefficient* of f relative to S.

6.2 Hermite analysis of LTFs

We continue to analyse LTFs in this setting, but for simplicity we restrict ourselves to only *origin-centred* LTFs, i.e., LTFs of the form

$$f(x) = \text{sgn}(w^T x) = \text{sgn}(w_1 x_1 + \cdots + w_n x_n), \tag{6.12}$$

with constant coefficient w_0 equal to zero. Without loss of generality, we assume $\|w\|_2 = 1$. Our ideas will also work for the non origin-centred case, but the analysis becomes a bit more involved.

The Hermite coefficient relative to the zero vector is, by definition,

$$H_0 = \mathbf{E}[f(X)] \tag{6.13}$$

$$= \mathbf{E}[\text{sgn}(w^T X)]. \tag{6.14}$$

Because X is n-dimensional standard normal, each component X_i is standard normal and so $w^T X \sim \mathcal{N}(0,1)$ because $\|w\|_2 = 1$. Hence $\text{sgn}(w^T X)$ is 1 with probability $1/2$ and -1 with probability $1/2$ and so $H_0 = 0$.

The Hermite coefficient relative to the standard unit vectors e_i (the vector having 1 in component i and 0s elsewhere) are

$$H_{e_i} = \mathbf{E}[f(X) X_i] \tag{6.15}$$

$$= \mathbf{E}[\text{sgn}(w^T X) X_i]. \tag{6.16}$$

Now $w^T X$ and X_i are jointly bivariate normal, and have covariance

$$\text{Cov}[w^T X, X_i] = \mathbf{E}[(w^T X) X_i] - \mathbf{E}[w^T X]\mathbf{E}[X_i] \tag{6.17}$$

$$= w_i \mathbf{E}[X_i^2] + \sum_{j \neq i} w_j \mathbf{E}[X_j X_i] - 0 \tag{6.18}$$

$$= w_i \tag{6.19}$$

where the last inequality is because X_i^2 is chi squared with one degree of freedom, which has mean one, and because X_i and X_j are independent which implies $\mathbf{E}[X_i X_j] = \mathbf{E}[X_i]\mathbf{E}[X_j] = 0$.

Consequently, $(w^T X, X_i)$ has the same distribution as $\left(U, w_i U + \sqrt{1-w_i^2}V\right)$ where U, V are *independent* standard normal. Hence,

$$\mathbf{E}[\text{sgn}(w^T X)X_i] = \mathbf{E}\left[\text{sgn}(U)\left(w_i U + \sqrt{1-w_i^2}V\right)\right] \tag{6.20}$$

$$= w_i \mathbf{E}[|U|] + \sqrt{1-w_i^2}\mathbf{E}[\text{sgn}(U)]\mathbf{E}[V] \tag{6.21}$$

$$= \sqrt{\frac{2}{\pi}} w_i + 0. \tag{6.22}$$

In other words the Hermite coefficients H_{e_i} are just a rescaling of the weights w_i by $\sqrt{\frac{2}{\pi}}$. Hence, just like Chow's theorem in the discrete case, here the Hermite coefficients H_{e_i} also uniquely determine the LTF. Moreover it is trivial to reconstruct the LTF given H_{e_i}, which is very much different to the discrete case, and in this sense the Gaussian setting is much easier to deal with compared to the discrete setting.

Given m samples $(X^{(1)}, Y^{(1)}), \cdots, (X^{(m)}, Y^{(m)})$ where the X_i are i.i.d. n-dimensional standard Gaussian and $Y_i = f(X^{(i)}) = \text{sgn}(w^T X^{(i)})$, we can estimate the Hermite coefficients H_{e_i} in the exact same way that we estimated the Chow coefficients, namely by computing

$$\widehat{\mu}_j := \sum_{i=1}^{m} Y^{(i)} X_j^{(i)} \tag{6.23}$$

for $j \in [n]$. We can estimate the weights by rescaling $\widehat{\mu}$ by $\sqrt{\frac{\pi}{2}}$ and outputting the hypothesis

$$h(x) = \text{sgn}\left(\sqrt{\frac{\pi}{2}}\widehat{\mu}^T x\right) \tag{6.24}$$

which is equivalent to just outputting

$$h(x) = \text{sgn}(\widehat{\mu}^T x). \tag{6.25}$$

In Section 6.5, we will show that if $\widehat{\mu}$ is estimated to within L_2 norm ε of the vector of Hermite coefficients, then h defined above is $O(\varepsilon)$ close to the true LTF f in the sense that

$$\mathbf{P}[f(X) \neq h(X)] < O(\varepsilon). \tag{6.26}$$

6.3 MI and CMI bounds

For the time-being however, we aim to use our information-theoretic toolkit to analyse the expected generalisation error of this learning algorithm. Like in the discrete case, the algorithm is comprised of two parts. First estimating the Hermite coefficients, i.e., computing $\widehat{\mu}(Z)$, then constructing the approximate LTF h from the Hermite estimates, which we will denote by $\mathcal{A}(\widehat{\mu}(Z))$. As before, our information-theoretic analysis will be focused on $\widehat{\mu}(Z)$; we will then apply the data processing inequality to get a bound on the entire algorithm $\mathcal{A}(\widehat{\mu}(Z))$.

A first idea is to use the mutual information framework in Section 3.2. However, as we discussed there, the quantity $I(Z;\widehat{\mu}(Z))$ is infinite in this case because the feature space $\mathcal{Z} = \mathbb{R}^n$ is continuous and the algorithm $\widehat{\mu}$ deterministic.

So a natural second idea is to compute the conditional mutual information quantity $I\left(S;\mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right)$ instead as we know this is always upper bounded by $m \ln 2$. We can follow the exact same steps as in the discrete case in Section 4.4, getting that

$$I\left(S;\mathcal{A}(\widetilde{Z}_S) \mid \widetilde{Z}\right) \leq \sum_{j=0}^{n} \mathop{\mathbf{E}}_{\widetilde{z}\sim\mathcal{D}^{\otimes 2m}} \left[H(m\widehat{\mu}_j(\widetilde{z}_S))\right]. \tag{6.27}$$

As before, we fix $\widetilde{z} \in \mathcal{Z}^{2\times m}$ and write

$$\widetilde{z} = \begin{pmatrix} (\widetilde{x}^{(0,1)}, \widetilde{y}^{(0,1)}) & \dots & (\widetilde{x}^{(0,m)}, \widetilde{y}^{(0,m)}) \\ (\widetilde{x}^{(1,1)}, \widetilde{y}^{(1,1)}) & \dots & (\widetilde{x}^{(1,m)}, \widetilde{y}^{(1,m)}) \end{pmatrix} \tag{6.28}$$

so that $\widehat{\mu}_j(\widetilde{z}_S)$ can be expressed as

$$m\widehat{\mu}_j(\widetilde{z}_S) = \sum_{i=1}^{n} \widetilde{y}^{(S_i,i)} \widetilde{x}_j^{(S_i,i)} \tag{6.29}$$

for $j \in [m]$. In the discrete case $\widetilde{x}_j^{(S_i,i)} \in \{-1,1\}$ and so $\widetilde{y}^{(S_i,i)}\widetilde{x}_j^{(S_i,i)} \in \{-1,1\}$, hence $m\widehat{\mu}_j(\widetilde{z}_S)$ was a binomial (plus a constant). However, in the continuous case, $\widetilde{x}_j^{(S_i,i)} \in \{-1,1\} \in \mathbb{R}$ and so $\widetilde{y}^{(S_i,i)}\widetilde{x}_j^{(S_i,i)} \in \mathbb{R}$. With probability one over S, each $m\widehat{\mu}_j(\widetilde{z}_S)$ will have a different real value for each S, and so $m\widehat{\mu}_j(\widetilde{z}_S)$ attains the maximum entropy of $m \ln 2$, making the conditional mutual information generalisation bound vacuous as well.

6.4 Individual sample mutual information bound

Thankfully, this is a situation where the individual sample mutual information discussed in Section 3.3 is useful. Although $\widehat{\mu}$ is a deterministic function of Z, it is not a deterministic function of any one $Z^{(i)}$, since the inclusion of the other $Z^{(j)}$ for $j \neq i$ introduce randomness. Our main result for this chapter is in showing that this approach allows us to prove that the Hermite-based LTF learning algorithm achieves an expected generalisation bound of $O\left(\sqrt{\frac{n}{m}}\right)$.

THEOREM 25 (**Our result**). *The Hermite parameter LTF learner $\mathcal{A}$ described in the previous section has expected generalisation error*

$$\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \left[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \right] \right| \leq O\left(\sqrt{\frac{n}{m}}\right). \tag{6.30}$$

We now prove this result in the remainder of this section.

Our end goal is to bound the quantity $I(X^{(i)}, Y^{(i)}; \widehat{\mu})$. Note that since $\widehat{\mu}_j = \frac{1}{m}\sum_{i=1}^{m} \text{sgn}(w^T X^{(i)}) X_j^{(i)}$, in vector form we have $\widehat{\mu} = \frac{1}{m}\sum_{i=1}^{m} \text{sgn}(w^T X^{(i)}) X^{(i)}$. Note also that the individual sample mutual information $I(X^{(i)}, Y^{(i)}; \widehat{\mu})$ is invariant to the choice of i, hence without loss of generality we will take $i = 1$. Then,

$$\begin{aligned}
I\left(X^{(i)}, Y^{(i)}; \widehat{\mu}\right) &= I\left(X^{(1)}, Y^{(1)}; \widehat{\mu}\right) && (6.31)\\
&= I(X^{(1)}; \widehat{\mu}) && (6.32)\\
&= I(X^{(1)}; m\widehat{\mu}) && (6.33)\\
&= I\left(X^{(1)}; \sum_{k=1}^{m} \text{sgn}(w^T X^{(k)}) X^{(k)}\right) && (6.34)\\
&= I\left(X^{(1)}; \text{sgn}(w^T X^{(1)}) X^{(1)} + \sum_{k=2}^{m} \text{sgn}(w^T X^{(k)}) X^{(k)}\right). && (6.35)
\end{aligned}$$

where Equation 6.33 is due to mutual information being invariant to scaling (Lemma 7).

Notice that in Equation 6.35 we want to compute the mutual information between a random variable $X^{(1)}$ and a deterministic function of that random variable $\text{sgn}(w^T X^{(1)}) X^{(1)}$ plus independent "noise" $\sum_{k=2}^{m} \text{sgn}(w^T X^{(k)}) X^{(k)}$. Notice that as m increases, the magnitude of the noise increases, and so we would expect the mutual information to be decreasing in m. This is good news, as this implies the expected generalisation error bound we get is also decreasing in m (see Theorem 10). The exact rate

of decrease, however, is not *a priori* clear, and needs to be workoed out, which we do over the next subsections.

To progress further, we first simplify Equation 6.35 using the following lemma.

LEMMA 13. *Let X, Y be continuous independent random variables and f a deterministic function. Then*

$$I(X; f(X) + Y) = h(f(X) + Y) - h(Y). \tag{6.36}$$

PROOF.

$$\begin{aligned} I(X; f(X) + Y) &= h(f(X) + Y) - h(f(X) + Y \mid X) && (6.37)\\ &= h(f(X) + Y) - \mathop{\mathbf{E}}_{x \sim p_X} [h(f(X) + Y \mid X = x)] && (6.38)\\ &= h(f(X) + Y) - \mathop{\mathbf{E}}_{x \sim p_X} [h(f(x) + Y \mid X = x)] && (6.39)\\ &= h(f(X) + Y) - \mathop{\mathbf{E}}_{x \sim p_X} [h(Y \mid X = x)] && (6.40)\\ &= H(f(X) + Y) - H(Y) && (6.41) \end{aligned}$$

where Equation 6.37 is due to Lemma 6, and Equation 6.40 due to differential entropy being invariant to translations (Lemma 9). □

Note the exact same argument also works for Shannon entropy in the case of discrete random variables, but we will not use this result here.

Applying the above lemma to Equation 6.35 results in the simplified expression

$$\begin{aligned} & I\left(X^{(1)}; \operatorname{sgn}(w^T X^{(1)}) X^{(1)} + \sum_{k=2}^{m} \operatorname{sgn}(w^T X^{(k)}) X^{(k)}\right) \\ &= h\left(\sum_{k=1}^{m} \operatorname{sgn}(w^T X^{(k)}) X^{(k)}\right) - h\left(\sum_{k=2}^{m} \operatorname{sgn}(w^T X^{(k)}) X^{(k)}\right) && (6.42)\\ &= h\left(\sum_{k=1}^{m} \operatorname{sgn}(w^T X^{(k)}) X^{(k)}\right) - h\left(\sum_{k=1}^{m-1} \operatorname{sgn}(w^T X^{(k)}) X^{(k)}\right). && (6.43) \end{aligned}$$

6.4.1 Differential entropy of the sample Hermite means

Let us now analyse the distribution of $\sum_{k=1}^{m} \text{sgn}(w^T X^{(k)}) X^{(k)}$. For the simple case of $m = 1$, this reduces to $\text{sgn}(w^T X^{(1)}) X^{(1)}$. Consider an arbitrary point $x \in \mathbb{R}^n$ such that $w^T x \geq 0$, and let $g(x) := \text{sgn}(w^T x)x$. Notice that

$$g(-x) = \text{sgn}(w^T(-x))(-x) \tag{6.44}$$

$$= -\text{sgn}(w^T x)(-x) \tag{6.45}$$

$$= \text{sgn}(w^T x)x \tag{6.46}$$

$$= g(x). \tag{6.47}$$

In other words any $-x$ with $w^T(-x) \leq 0$, is mapped to the same point that x, which has $w^T x \geq 0$, is mapped to. Moreover, by rotational symmetry of the n-dimensional standard Gaussian density, $-x$ has the same Gaussian density as x. Hence $\text{sgn}(w^T X^{(1)}) X^{(1)}$ has support on the set $\left\{x : w^T x \geq 0\right\}$ and the density there is twice the normal density, i.e.,

$$2\varphi_n(x) \mathbb{1}\left\{w^T x \geq 0\right\}. \tag{6.48}$$

Note that the differential entropy of the above p.d.f. is invariant to w by the rotational symmetry of the Gaussian density. Hence, for the purposes of computing differential entropy we may take $w = e_1$, i.e.,

$$h(\text{sgn}(w^T X^{(1)}) X^{(1)}) = h(\text{sgn}(X_1^{(1)}) X^{(1)}). \tag{6.49}$$

The same argument shows that for the case of general m,

$$h\left(\sum_{k=1}^{m} \text{sgn}(w^T X^{(k)}) X^{(k)}\right) = h\left(\sum_{k=1}^{m} \text{sgn}(e_1^T X^{(k)}) X^{(k)}\right) \tag{6.50}$$

$$= h\left(\sum_{k=1}^{m} \text{sgn}(X_1^{(k)}) X^{(k)}\right) \tag{6.51}$$

$$= h\left(\sum_{k=1}^{m} \left|X_1^{(k)}\right|, \sum_{k=1}^{m} \text{sgn}(X_1^{(k)}) X_2^{(k)}, \ldots, \sum_{k=1}^{m} \text{sgn}(X_1^{(k)}) X_n^{(k)}\right). \tag{6.52}$$

We now show that the n random variables inside the differential entropy are mutually independent. We prove this only for the case $m = 1$, because the independence of the $X^{(k)}$ over k imply the case of general m.

Recall that by definition, n random variables are independent if their joint c.d.f. factors into the product of the marginal c.d.f.s. Let $\mathcal{E} := \left\{\left|X_1^{(1)}\right| \leq x_1, \text{sgn}(X_1^{(1)})X_2^{(1)} \leq x_2, \ldots, \text{sgn}(X_1^{(1)})X_n^{(1)} \leq x_n\right\}$. For $x_1 \geq 0$ and $x_2, \ldots, x_n \in \mathbb{R}$, the joint c.d.f. is

$$\begin{aligned}
\mathbf{P}[\mathcal{E}] &= \mathbf{P}[\mathcal{E}, \text{sgn}(X_1^{(1)}) = 1] + \mathbf{P}[\mathcal{E}, \text{sgn}(X_1^{(1)}) = -1] && (6.53)\\
&= \mathbf{P}\left[0 \leq X_1^{(1)} \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n\right] && (6.54)\\
&+ \mathbf{P}\left[0 \geq X_1 \geq -x_1, X_2 \geq -x_2, \ldots, X_n \geq -x_n\right] \\
&= 2\mathbf{P}\left[0 \leq X_1^{(1)} \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n\right] && (6.55)\\
&= 2\left(\Phi(x_1) - \frac{1}{2}\right)\prod_{k=2}^{n}\Phi(x_k). && (6.56)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbf{P}[|X_1| \leq x_1] &= \mathbf{P}[-x_1 \leq X_1 \leq x_1] && (6.57)\\
&= \Phi(x_1) - \Phi(-x_1) && (6.58)\\
&= 2\Phi(x_1) - 1, && (6.59)
\end{aligned}$$

and for $j > 1$,

$$\begin{aligned}
\mathbf{P}[\text{sgn}(X_1)X_k \leq x_k] &= \mathbf{P}[\text{sgn}(X_1)X_k \leq x_k, X_1 \geq 0] && (6.60)\\
&+ \mathbf{P}[\text{sgn}(X_1)X_k \leq x_k, X_1 < 0] \\
&= \mathbf{P}[X_k \leq x_k, X_1 \geq 0] && (6.61)\\
&+ \mathbf{P}[X_k \geq -x_k, X_1 < 0] \\
&= \frac{1}{2}\Phi(x_k) + \frac{1}{2}\Phi(x_k) && (6.62)\\
&= \Phi(x_k). && (6.63)
\end{aligned}$$

Hence the random variables $|X_1^{(1)}|, \operatorname{sgn}(X_1^{(1)})X_2^{(1)}, \ldots, \operatorname{sgn}(X_1^{(1)})X_n^{(1)}$ are mutually independent, and thus so too are $\sum_{k=1}^m \left|X_1^{(k)}\right|, \sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_2^{(k)}, \ldots, \sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_n^{(k)}$. Consequently their differential entropy is

$$\begin{aligned} &h\left(\sum_{k=1}^m \left|X_1^{(k)}\right|, \sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_2^{(k)}, \ldots, \sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_n^{(k)}\right) \\ &= h\left(\sum_{k=1}^m |X_1^{(k)}|\right) + \sum_{j=2}^n h\left(\sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_j^{(k)}\right). \end{aligned} \tag{6.64}$$

Notice that $\operatorname{sgn}(X_1^{(k)})X_j^{(k)}$ can be understood as independently and uniformly flipping the sign of $X_j^{(k)}$; clearly this does not change the distribution. Hence $\sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_j^{(k)} \sim \mathcal{N}(0, m)$ and so

$$\sum_{j=2}^n h\left(\sum_{k=1}^m \operatorname{sgn}(X_1^{(k)})X_j^{(k)}\right) = (n-1) \cdot \frac{1}{2}\log(2\pi e m). \tag{6.65}$$

Tying things together, this shows that

$$h\left(\sum_{k=1}^m \operatorname{sgn}(w^T X^{(k)})X^{(k)}\right) = h\left(\sum_{k=1}^m |X_1^{(k)}|\right) + \frac{n-1}{2}\log(2\pi e m). \tag{6.66}$$

Hence,

$$\begin{aligned} &h\left(\sum_{k=1}^m \operatorname{sgn}(w^T X^{(k)})X^{(k)}\right) - h\left(\sum_{k=1}^{m-1} \operatorname{sgn}(w^T X^{(k)})X^{(k)}\right) \\ &= h\left(\sum_{i=1}^m |X_1^{(i)}|\right) - h\left(\sum_{i=1}^{m-1} |X_1^{(i)}|\right) + \frac{n-1}{2}\log\left(1 + \frac{1}{m-1}\right) \end{aligned} \tag{6.67}$$

6.4.2 Differential entropy of the sum of half-normals

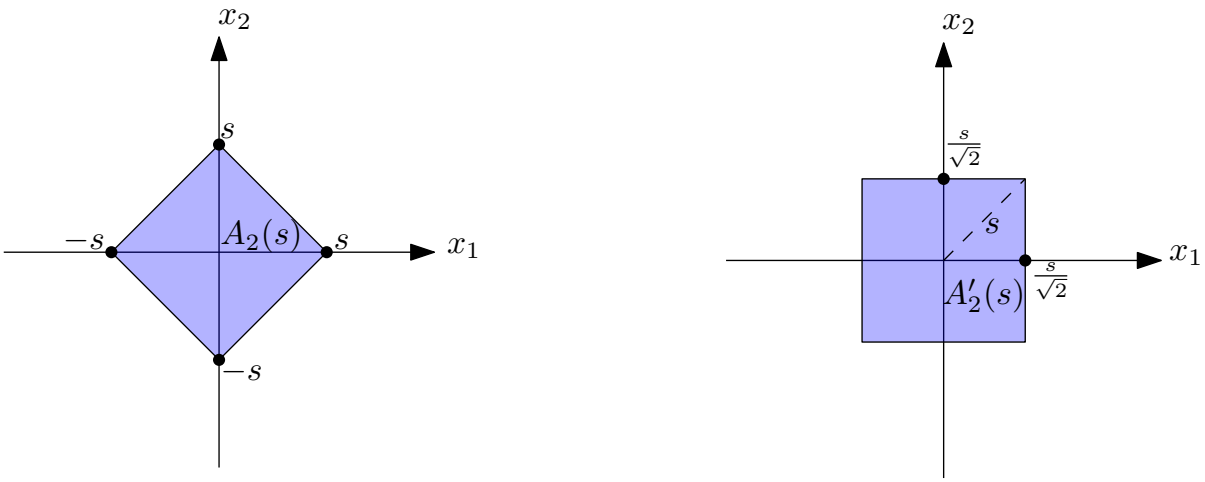
To continue, we would like to write down an expression for $h\left(\sum_{k=1}^m |X_1^{(k)}|\right)$. For notational simplicity in this section, we will rewrite this as $h\left(\sum_{k=1}^m |X_k|\right)$ where each X_k is understood to be i.i.d. $\mathcal{N}(0,1)$. As a first attempt, let us try compute the p.d.f. of $S_m := \sum_{k=1}^m |X_k|$ analytically.

The case $m = 2$ has been analysed by Mark (2013). The c.d.f. of $S_2 := |X_1| + |X_2|$ is

$$F_{S_2}(s) = \mathbf{P}[|X_1| + |X_2| \leq s] \tag{6.68}$$

$$= \mathbf{P}[(X_1, X_2) \in A_2(s)] \tag{6.69}$$

where $A_2(s) := \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| + |x_2| \leq s\}$. The situation is illustrated diagrammatically in to Figure 6.1a. Note that $A_2(s)$ is a rotated square with a half-width of $\frac{s}{\sqrt{2}}$. Because $X^{(1)}, X^{(2)}$



(A) $A_2(s) := \{(x_1, x_2) \in \mathbb{R}^2 : |x_1| + |x_2| \leq s\}$.

(B) $A_2'(s) = \left[-\frac{s}{\sqrt{2}}, \frac{s}{\sqrt{2}}\right]^2$.

FIGURE 6.1. Illustration of the region $A_2(s)$ and its rotation $A_2'(s)$.

are independent standard Gaussian, their joint density has circular symmetry, and so we can rotate the square $A_2(s)$ so that it is parallel to the axes. In other words,

$$\mathbf{P}\left[(X_1, X_2) \in A_2(s)\right] = \mathbf{P}\left[(X_1, X_2) \in A_2'(s)\right] \tag{6.70}$$

where $A_2'(s) := \left[-\frac{s}{\sqrt{2}}, \frac{s}{\sqrt{2}}\right]^2$ is an axis-parallel square with the same half-width. See Figure 6.1b.

By exploiting the independence of X_1 and X_2, we have

$$\begin{aligned}
\mathbf{P}\left[(X_1, X_2) \in A_2'(s)\right] &= \mathbf{P}\left[(X_1, X_2) \in \left[-\frac{s}{\sqrt{2}}, \frac{s}{\sqrt{2}}\right]^2\right] && (6.71)\\
&= \mathbf{P}\left[X_1 \in \left[-\frac{s}{\sqrt{2}}, \frac{s}{\sqrt{2}}\right]\right]^2 && (6.72)\\
&= \left[\Phi\left(\frac{s}{\sqrt{2}}\right) - \Phi\left(-\frac{s}{\sqrt{2}}\right)\right]^2 && (6.73)\\
&= \left[2\Phi\left(\frac{s}{\sqrt{2}}\right) - 1\right]^2. && (6.74)
\end{aligned}$$

Hence the density of $S_2 := |X_1| + |X_2|$ is the derivative of the above expression,

$$f_{S_2}(s) = \frac{d}{ds}\left[2\Phi\left(\frac{s}{\sqrt{2}}\right) - 1\right]^2 \tag{6.75}$$

$$= 2\sqrt{2}\left[2\Phi\left(\frac{s}{\sqrt{2}}\right) - 1\right]\varphi\left(\frac{s}{\sqrt{2}}\right). \tag{6.76}$$

Let us try to generalise this argument for $m > 2$. For $m = 3$, we have

$$F_{S_3}(s) = \mathbf{P}[(X_1, X_2, X_3) \in A_3(s)] \tag{6.77}$$

where $A_3(s) := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : |x_1| + |x_2| + |x_3| \leq s\}$. The solid $A_3(s)$ is an octahedron and visualised in Figure 6.2. Unfortunately, the same trick of rotating $A_3(s)$ does not generalise. However,

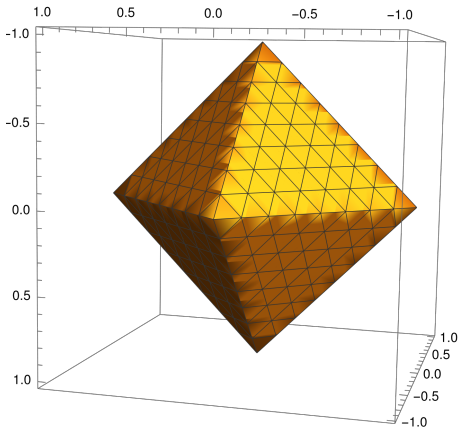


FIGURE 6.2. Illustration of the solid $A_3(s) := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : |x_1|+|x_2|+|x_3| \leq s\}$ for $s = 1$.

notice that for any horizontal "slice" of $A_3(s)$ at a fixed height $X_3 = x_3$, we can compute its differential probability as $F_{S_2}(s - |x_3|)\varphi(x_3)\,dx_3$. Then, by using the symmetry of the Gaussian density, we have

$$F_{S_3}(s) = 2\int_0^s F_{S_2}(s - x_3)\varphi(x_3)\,dx_3. \tag{6.78}$$

This argument generalises to arbitrary m in that F_{S_m} satisfies the recurrence

$$F_{S_m}(s) = 2\int_0^s F_{S_{m-1}}(s - x)\varphi(x)\,dx. \tag{6.79}$$

with initial condition $F_{S_2}(s) = \left[2\Phi\left(\frac{s}{\sqrt{2}}\right) - 1\right]^2$.

Unfortunately, it does not seem very tractable to compute the derivative of this expression for general m to get the density, and then use this to compute the differential entropy. However, at the end of the day, we would like to analyse the *asymptotic* behaviour of a learning algorithm, and so perhaps we can settle for asymptotic expressions of $h(S_m)$ instead.

Like we did in the discrete case, because we are considering the asymptotics of a sum of i.i.d. random variables, it is natural to turn to the central limit theorem (Theorem 31). It can be easily shown that a half-normal has mean $\sqrt{\frac{2}{\pi}}$ and variance $1 - \frac{2}{\pi}$. Let $Z_k := |X_k| - \sqrt{\frac{2}{\pi}}$ and consider the normalized sum

$$S_m := \frac{Z_1 + \cdots + Z_m}{\sqrt{m}} \tag{6.80}$$

which has mean 0 and variance $1 - \frac{2}{\pi}$. Then by the central limit theorem,

$$S_m \xrightarrow{d} \mathcal{N}\left(0, 1 - \frac{2}{\pi}\right) \tag{6.81}$$

hence we might expect that $h(S_m) \to h(\mathcal{N}(0,1))$. In fact, this result is indeed true, and was proven in the celebrated work of Barron (1986).

THEOREM 26 (Entropic central limit theorem (Barron, 1986)). *Let $Z_1, \cdots, Z_m$ be i.i.d. continuous random variables with mean 0 and variance σ^2, and define the normalized sum*

$$S_m := \frac{Z_1 + \cdots + Z_m}{\sqrt{m}}. \tag{6.82}$$

Then the KL divergence between S_m and $\mathcal{N}(0, \sigma^2)$ converges to zero, i.e.,

$$\lim_{m \to \infty} D(S_m \parallel \mathcal{N}(0, \sigma^2)) = 0 \tag{6.83}$$

if and only if $D(S_m \parallel \mathcal{N}(0, \sigma^2)) < \infty$ for some m.

This then implies $h(S_m) \to h(\mathcal{N}(0,\sigma^2)) = \frac{1}{2}\ln 2\pi e\sigma^2$ because, letting f_m denote the density of S_m, we have

$$D(S_m \parallel \mathcal{N}(0,\sigma^2)) = \int_{\mathbb{R}} f_m(x)\ln\left(\frac{f_m(x)}{\frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{1}{2\sigma^2}x^2)}\right)dx \tag{6.84}$$

$$= \int_{\mathbb{R}} f_m(x)\ln f_m(x)dx + \int_{\mathbb{R}} f_m(x)\ln\left(\sigma\sqrt{2\pi}\right)dx$$

$$+ \int_{\mathbb{R}} f_m(x)\frac{1}{2\sigma^2}x^2dx \tag{6.85}$$

$$= -h(f_m) + \frac{1}{2}\ln\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\sigma^2 \tag{6.86}$$

$$= -h(f_m) + \frac{1}{2}\ln\left(2\pi e\sigma^2\right) \tag{6.87}$$

$$= -h(S_m) + h(\mathcal{N}(0,\sigma^2)) \tag{6.88}$$

where Equation 6.86 follows because f_m is a density so f_m integrates to one, and because S_m has mean zero so its second moment equals its variance σ^2.

As an aside, note that the result $D(S_m \parallel \mathcal{N}(0,\sigma^2)) = -h(S_m)+h(\mathcal{N}(0,\sigma^2))$ only relied on S_m having mean zero and variance σ^2, and was independent of anything else about S_m. Hence the result is true for *any* continuous random variable with mean zero and variance σ^2. Moreover, non-negativity of KL divergence implies that

$$h(S_m) \leq h(\mathcal{N}(0,\sigma^2)) \tag{6.89}$$

with equality if and only if $S_m \sim \mathcal{N}(0,\sigma^2)$. In other words, we have shown the following result.

THEOREM 27. *The normal density uniquely maximises differential entropy over all densities with a given variance. In other words, if X is a continuous random variable with mean zero (without loss of generality) and variance σ^2, then*

$$h(X) \leq \frac{1}{2}\ln(2\pi e\sigma^2) \tag{6.90}$$

with equality if and only if $X \sim \mathcal{N}(0,\sigma^2)$.

Going back to the entropic central limit theorem, we note that the proof of the result is highly nontrivial and is in fact a stronger statement than the standard central limit theorem. The proof relies on a connection between entropy and *Fisher information* (Barron, 1986), unlike the proof of the standard central limit theorem which is based on characteristic functions.

Finally, for our original problem we have

$$h\left(\frac{|X_1|+\cdots+|X_m|}{\sqrt{m}}\right) = h\left(\frac{|X_1|+\cdots+|X_m| - m\sqrt{\frac{2}{\pi}}}{\sqrt{m}}\right) \tag{6.91}$$

since differential entropy is invariant to translations (Lemma 9). Taking limits on both sides and applying the entropic central limit theorem yields

$$\lim_{m\to\infty} h\left(\frac{|X_1|+\cdots+|X_m|}{\sqrt{m}}\right) = \frac{1}{2}\ln\left(2\pi e\left(1-\frac{2}{\pi}\right)\right). \tag{6.92}$$

By a standard result in analysis, this implies

$$\lim_{m\to\infty}\left[h\left(\frac{|X_1|+\cdots+|X_m|}{\sqrt{m}}\right) - h\left(\frac{|X_1|+\cdots+|X_{m-1}|}{\sqrt{m-1}}\right)\right] = 0. \tag{6.93}$$

We have

$$h\left(\frac{|X_1|+\cdots+|X_m|}{\sqrt{m}}\right) - h\left(\frac{|X_1|+\cdots+|X_{m-1}|}{\sqrt{m-1}}\right) \tag{6.94}$$

$$= h\left(\sum_{k=1}^{m}|X_k|\right) - \frac{1}{2}\ln m - h\left(\sum_{k=1}^{m-1}|X_k|\right) + \frac{1}{2}\ln(m-1) \tag{6.95}$$

$$= h\left(\sum_{k=1}^{m}|X_k|\right) - h\left(\sum_{k=1}^{m-1}|X_k|\right) + \frac{1}{2}\ln\left(1+\frac{1}{m-1}\right), \tag{6.96}$$

hence,

$$\lim_{m\to\infty}\left[h\left(\sum_{k=1}^{m}|X_k|\right) - h\left(\sum_{k=1}^{m-1}|X_k|\right) + \frac{1}{2}\ln\left(1+\frac{1}{m-1}\right)\right] = 0. \tag{6.97}$$

Unfortunately this does not tell us what the asymptotic behaviour of $h\left(\sum_{k=1}^{m}|X_k|\right) - h\left(\sum_{k=1}^{m-1}|X_k|\right)$ is except that it is $o(1)$. For example it does not immediately follow that $h\left(\sum_{k=1}^{m}|X_k|\right) - h\left(\sum_{k=1}^{m-1}|X_k|\right) = O\left(-\frac{1}{2}\ln\left(1+\frac{1}{m-1}\right)\right)$.

The problem is that the entropic central limit theorem does not tells us *how quickly* the convergence in entropy is, only that it does converge. Some further reading into the literature leads to a more recent result by Bobkov et al. (2013) who derive an asymptotic expansion of the quantity $D(S_m \parallel \mathcal{N}(0,\sigma^2))$. As a special case of this expansion, they derive the following result.

THEOREM 28 (Bobkov et al. (2013)). *Suppose* $\mathbf{E}Z_1^4 < \infty$. *Then*

$$D(S_m \parallel \mathcal{N}(0,\sigma^2)) = \frac{1}{12m}\left(\mathbf{E}Z_1^3\right)^2 + o\left(\frac{1}{m\log m}\right). \tag{6.98}$$

Combining this result with Equation 6.88, we get that

$$h(S_m) - h(S_{m-1}) = h(\mathcal{N}(0,\sigma^2)) - D(S_m \parallel \mathcal{N}(0,\sigma^2)) - h(\mathcal{N}(0,\sigma^2)) + D(S_{m-1} \parallel \mathcal{N}(0,\sigma^2)) \tag{6.99}$$

$$= D(S_{m-1} \parallel \mathcal{N}(0,\sigma^2)) - D(S_m \parallel \mathcal{N}(0,\sigma^2)) \tag{6.100}$$

$$= \frac{1}{12}\left(\mathbf{E}Z_1^3\right)^2 \left(\frac{1}{m-1} - \frac{1}{m}\right) + o\left(\frac{1}{m\log m}\right) \tag{6.101}$$

$$= \frac{1}{12}\left(\mathbf{E}Z_1^3\right)^2 \left(\frac{1}{m(m-1)}\right) + o\left(\frac{1}{m\log m}\right). \tag{6.102}$$

For our case where $Z_i := |X_i| - \sqrt{\frac{2}{\pi}}$ it can be shown that

$$\left(\mathbf{E}Z_1^3\right)^2 = \frac{2(\pi-4)^2}{(\pi-2)^3} \approx 0.9910. \tag{6.103}$$

By Equation 6.96,

$$h\left(\sum_{k=1}^{m}|X_k|\right) - h\left(\sum_{k=1}^{m-1}|X_k|\right) = h(S_m) - h(S_{m-1}) - \frac{1}{2}\ln\left(1+\frac{1}{m-1}\right) \tag{6.104}$$

$$= O\left(\frac{1}{m^2}\right) + o\left(\frac{1}{m\log m}\right) \tag{6.105}$$

$$= o\left(\frac{1}{m\log m}\right). \tag{6.106}$$

Finally, backtracking through all our calculations (Equation 6.67, Equation 6.43 and Equation 6.35), we can conclude that the individual sample mutual information satisfies

$$I\left(Z^{(i)};\widehat{\mu}(Z)\right) = o\left(\frac{1}{m\log m}\right) + \frac{n-1}{2}\log\left(1+\frac{1}{m-1}\right) \tag{6.107}$$

$$\leq o\left(\frac{1}{m\log m}\right) + \frac{n-1}{2}\log\left(\exp\left(\frac{1}{m-1}\right)\right) \tag{6.108}$$

$$\leq o\left(\frac{1}{m\log m}\right) + O\left(\frac{n}{m}\right) \tag{6.109}$$

$$= O\left(\frac{n}{m}\right). \tag{6.110}$$

By the data processing inequality, the individual sample mutual information for the full learning algorithm then satisfies

$$I\left(Z^{(i)};\mathcal{A}(\widehat{\mu}(Z))\right) \leq O\left(\frac{n}{m}\right), \tag{6.111}$$

and so, combining this with Theorem 10 gives the intended result of Theorem 25.

6.5 From Hermite estimates to an LTF

In Section 6.2, we stated that the Hermite estimates $\widehat{\mu}$ could easily be used to construct a good approximation of the true LTF f by simply outputting an LTF with $\widehat{\mu}$ as its weights. In this section, we formalise and prove this result.

THEOREM 29 (**Our result**). *Suppose that $\widehat{\mu} \in \mathbb{R}^n$ is such that*

$$\|\widehat{\mu} - H(e)\|_2 < \varepsilon \tag{6.112}$$

where $H(e) := (H(e_1), \cdots, H(e_n))$ are the degree 1 Hermite coefficients of an origin-centred LTF f. Then the hypothesis

$$h(x) := sgn(\widehat{\mu}^T x) \tag{6.113}$$

satisfies

$$\mathop{\mathbf{P}}_{X \sim \varphi_n}[f(X) \neq h(X)] < O(\varepsilon). \tag{6.114}$$

This result is likely known to researchers in the field (Servedio, 2023) but did not seem to be present in the literature; for the sake of copmleteness, we derive this result and provide a proof below.

PROOF. Since $H_e = \sqrt{\frac{2}{\pi}} w$, it will be helpful to define the weight estimates

$$\widehat{w} := \sqrt{\frac{\pi}{2}} \widehat{\mu}, \tag{6.115}$$

and the *normalised* weight estimates

$$\widetilde{w} := \frac{\widehat{w}}{\|\widehat{w}\|_2}. \tag{6.116}$$

Then,

$$\begin{aligned}
\mathbf{P}\left[f(X) \neq h(X)\right] &= \mathbf{P}\left[\text{sgn}(w^T X) \neq \text{sgn}(\widehat{\mu}^T X)\right] && (6.117)\\
&= \mathbf{P}\left[\text{sgn}(w^T X) \neq \text{sgn}(\widetilde{w}^T X)\right] && (6.118)\\
&= \mathbf{P}[w^T X > 0, \widetilde{w}^T X < 0] + \mathbf{P}[w^T X < 0, \widetilde{w}^T X > 0] && (6.119)\\
&= 2\mathbf{P}[w^T X > 0, \widetilde{w}^T X < 0]. && (6.120)
\end{aligned}$$

Now, because w and $\widetilde{w}$ have norm one, $w^T X$ and $\widetilde{w}^T X$ are both $\mathcal{N}(0,1)$ random variables. They are jointly distributed with covariance

$$\begin{aligned}
\text{Cov}[w^T X, \widetilde{w}^T X] &= \mathbf{E}[(w^T X)(\widetilde{w}^T X)] - \mathbf{E}[w^T X]\mathbf{E}[\widetilde{w}^T X] && (6.121)\\
&= \mathbf{E}\left[\left(\sum_{i=1}^n w_i X_i\right)\left(\sum_{i=1}^n \widetilde{w}_i X_i\right)\right] - 0 && (6.122)\\
&= \sum_{i=1}^n w_i \widetilde{w}_i \mathbf{E}[X_i^2] + \sum_{i\neq j} w_i \widetilde{w}_j \mathbf{E}[X_i X_j] && (6.123)\\
&= \sum_{i=1}^n w_i \widetilde{w_i} + 0 && (6.124)\\
&= w^T \widetilde{w}. && (6.125)
\end{aligned}$$

Hence $(w^T X, \widetilde{w}^T X)$ has the same distribution as $(Y, w^T \widetilde{w} Y + \sqrt{1-(w^T\widetilde{w})^2} Z)$ where Y and Z are *independent* $\mathcal{N}(0,1)$. Thus,

$$\begin{aligned}
2\mathbf{P}\left[w^T X > 0, \widetilde{w}^T X < 0\right] &= 2\mathbf{P}\left[Y > 0, w^T\widetilde{w} Y + \sqrt{1-(w^T\widetilde{w})^2} Z < 0\right] && (6.126)\\
&= 2\mathbf{P}[Y>0]\mathbf{P}\left[w^T\widetilde{w} Y + \sqrt{1-(w^T\widetilde{w})^2} Z < 0 \mid Y > 0\right] && (6.127)\\
&= \mathbf{P}\left[w^T\widetilde{w}|Y| + \sqrt{1-(w^T\widetilde{w^2})} Z < 0\right] && (6.128)\\
&= \mathbf{P}\left[Z < -\alpha|Y|\right] && (6.129)
\end{aligned}$$

where $\alpha := \frac{w^T\widetilde{w}}{\sqrt{1-(w^T\widetilde{w})^2}}$. This probability can be calculated by integrating the product of the marginal densities of $|Y|$ and Z over the region $A := \{(y,z) \in [0,\infty) \times \mathbb{R} \mid z < -\alpha y\}$. Refer to Figure 6.3 for a diagram of this region.

Specifically,

$$\begin{aligned}
\mathbf{P}[Z < -\alpha|Y|] &= \iint_A \frac{1}{\sqrt{2\pi}} \cdot \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y^2+z^2)\right) dy\, dz && (6.130)\\
&= \iint_A \frac{1}{\pi} \exp\left(-\frac{1}{2}(y^2+z^2)\right) dy\, dz. && (6.131)
\end{aligned}$$

To evaluate this integral, make a change of variable to polar coordinates $(y,z) = (r\cos\theta, r\sin\theta)$. Recall the Jacobian matrix of this transformation has determinant r. Hence by the multivariate change

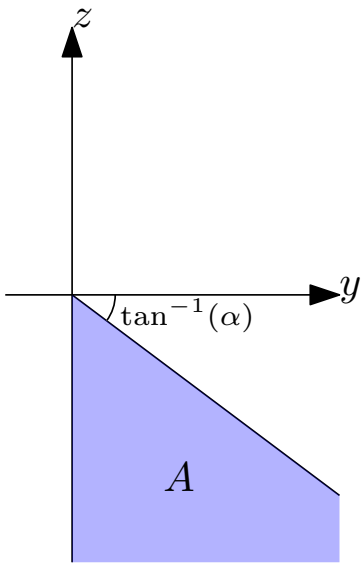


FIGURE 6.3. Illustration of the region $A := \{(y, z) \in [0, \infty) \times \mathbb{R} \mid z < -\alpha y\}$.

of variables theorem (Theorem 37),

$$\iint_A \frac{1}{\pi} \exp\left(-\frac{1}{2}(y^2 + z^2)\right) dy\, dz = \int_0^\infty \int_{-\frac{\pi}{2}}^{-\tan^{-1}(\alpha)} \frac{1}{\pi} \exp\left(-\frac{1}{2}(r^2 \cos^2\theta + r^2 \sin^2\theta)\right) r\, d\theta\, dr \tag{6.132}$$

$$= \frac{1}{\pi} \int_0^\infty \int_{-\frac{\pi}{2}}^{-\tan^{-1}(\alpha)} r \exp\left(-\frac{1}{2} r^2\right) d\theta\, dr \tag{6.133}$$

$$= \frac{1}{\pi} \int_0^\infty \left(\frac{\pi}{2} - \tan^{-1}(\alpha)\right) r \exp\left(-\frac{1}{2} r^2\right) dr \tag{6.134}$$

$$= \left(\frac{1}{2} - \frac{\tan^{-1}(\alpha)}{\pi}\right) \left[-\exp\left(-\frac{1}{2} r^2\right)\right]_{r=0}^{r=\infty} \tag{6.135}$$

$$= \frac{1}{2} - \frac{\tan^{-1}(\alpha)}{\pi} \tag{6.136}$$

which is monotonically decreasing as a function of α and approaches zero as $\alpha \to \infty$. Hence to upper bound this quantity, we seek to lower bound α which amounts to lower bounding the quantity $w^T \widetilde{w}$. To

do so, note that

$$\|w - \widetilde{w}\|_2^2 = \sum_{i=1}^{n} (w_i - \widetilde{w_i})^2 \tag{6.137}$$

$$= \sum_{i=1}^{n} w_i^2 - 2\sum_{i=1}^{n} w_i \widetilde{w}_i + \sum_{i=1}^{n} \widetilde{w}_i^2 \tag{6.138}$$

$$= \|w\|_2^2 - 2w^T \widetilde{w} + \|\widetilde{w}\|_2^2 \tag{6.139}$$

$$= 2 - 2w^T \widetilde{w}. \tag{6.140}$$

On the other hand, by the triangle inequality,

$$\|w - \widetilde{w}\|_2 \leq \|w - \widehat{w}\|_2 + \|\widehat{w} - \widetilde{w}\|_2. \tag{6.141}$$

For the second summand on the right hand side, we have

$$\|\widehat{w} - \widetilde{w}\|_2 = \|\|\widehat{w}\|_2 \cdot \widetilde{w} - \widetilde{w}\|_2 \tag{6.142}$$

$$= |\|\widehat{w}\|_2 - 1| \cdot \|\widetilde{w}\|_2 \tag{6.143}$$

$$= |\|\widehat{w}\|_2 - \|w\|_2| \tag{6.144}$$

$$\leq \|\widehat{w} - w\|_2, \tag{6.145}$$

where the second last line is because we assumed w has norm one, and the last line is due to the reverse triangle inequality. Hence,

$$\|w - \widetilde{w}\|_2 \leq 2\|\widehat{w} - w\|_2. \tag{6.146}$$

But we assumed that our Hermite estimates $\widehat{\mu}$ are within L_2 norm ε of the true Hermite coefficients. Multiplying both sides of Equation 6.112 by $\sqrt{\frac{\pi}{2}}$ gives

$$\|\widehat{w} - w\| < \sqrt{\frac{\pi}{2}}\varepsilon \tag{6.147}$$

and so

$$\|w - \widetilde{w}\|_2 < \sqrt{2\pi}\varepsilon. \tag{6.148}$$

Squaring both sides yields

$$\|w - \widetilde{w}\|_2 < 2\pi\varepsilon^2, \tag{6.149}$$

and combining this with Equation 6.140 gives

$$2 - 2w^T \widetilde{w} < 2\pi\varepsilon^2, \tag{6.150}$$

or equivalently,

$$w^T \widetilde{w} > 1 - \pi\varepsilon^2. \tag{6.151}$$

Hence we have

$$\alpha := \frac{w^T \widetilde{w}}{\sqrt{1-(w^T\widetilde{w})^2}} \tag{6.152}$$

$$> \frac{1-\pi\varepsilon^2}{\sqrt{1-(1-\pi\varepsilon^2)^2}} \tag{6.153}$$

$$= \frac{1-\pi\varepsilon^2}{\sqrt{2\pi\varepsilon^2 - \pi^2\varepsilon^4}} \tag{6.154}$$

$$=: g_1(\varepsilon). \tag{6.155}$$

We would like to determine the asymptotic behvaiour of $g_1(\varepsilon)$. We have

$$\varepsilon g_1(\varepsilon) = \frac{1-\pi\varepsilon^2}{\sqrt{2\pi - \pi^2\varepsilon^2}}. \tag{6.156}$$

If for instance, $\varepsilon \leq \frac{1}{2}$, then

$$\varepsilon g_1(\varepsilon) \geq \frac{1-\pi/4}{2\pi - \pi^2/4} \approx 0.11 \tag{6.157}$$

and so we must have

$$g_1(\varepsilon) = \Omega(1/\varepsilon). \tag{6.158}$$

We also have

$$\mathbf{P}[f(X) \neq h(X)] = \frac{1}{2} - \frac{\tan^{-1}(\alpha)}{\pi} =: g_2(\alpha). \tag{6.159}$$

To determine the asymptotic behaviour of $g_2(\alpha)$, we have, by L'Hôpital's rule (Theorem 36),

$$\lim_{\alpha\to\infty} \alpha g_2(\alpha) = \lim_{\alpha\to\infty} \frac{\frac{1}{2} - \frac{1}{\pi}\tan^{-1}(\alpha)}{\frac{1}{\alpha}} \tag{6.160}$$

$$= \lim_{\alpha\to\infty} \frac{-\frac{1}{\pi(1+\alpha^2)}}{-\frac{1}{\alpha^2}} \tag{6.161}$$

$$= \lim_{\alpha\to\infty} \frac{\alpha^2}{\pi(1+\alpha^2)} \tag{6.162}$$

$$= \frac{1}{\pi}. \tag{6.163}$$

But since $\alpha > \Omega(1/\varepsilon)$ this implies $g_2(\alpha) < O(\varepsilon)$, i.e.,

$$\mathbf{P}[f(X) \neq h(X)] < O(\varepsilon). \tag{6.164}$$

□

CHAPTER 7

Conclusion and further work

In this thesis we have looked at a variety of ideas that can be used to provably bound the generalisation error of a machine learning algorithm. In Chapter 2, we introduced the formal setting of statistical learning, and defined what it means to be a good learning algorithm — this is the definition of PAC learning, which loosely says that, given enough samples m, a good learner should be able to learn a target function f to arbitrary accuracy ε with arbitrarily high probability $1 - \delta$. We also saw how a combinatorial quantity of the hypothesis space, its VC dimension, characterises the PAC learnability of $\mathcal{H}$, and provides tight sample complexity bounds for any ERM algorithm. However, this does not give us any guarantees on any *general* learning algorithm $\mathcal{A}$.

In Chapter 3, we saw that the mutual information $I(Z; \mathcal{A}(Z))$ between the input samples Z and the output hypothesis of a learning algorithm $\mathcal{A}$, and some variations of this idea, could be used to derive bounds on the expected generalisation error of *any* learning algorithm $\mathcal{A}$, that is the quantity $\left| \mathop{\mathbf{E}}_{Z,\mathcal{A}} \left[R(\mathcal{A}(Z)) - \widehat{R}_Z(\mathcal{A}(Z)) \right] \right|$.

In Chapter 4 we applied these information-theoretic tools in analysing a relatively simple algorithm that learnt linear threshold functions over $\{-1, 1\}^n$ given samples drawn uniformly from $\{-1, 1\}^n$. We saw that LTFs were characterised by their degree 0 and degree 1 Fourier coefficients, collectively known as the Chow parameters. Based off this idea, our learning algorithm learns the Chow parameters to sufficient accuracy, then uses a result by O'Donnell and Servedio (2008) to approximately reconstruct the LTF from the estimated Chow parameters. Learning the LTF to accuracy parameter ε unfortunately required an exponential number of samples in $1/\varepsilon$ and had time complexity doubly exponential in $1/\varepsilon$ — the difficulty lies in reconstructing the LTF from the Chow estimates. Despite this, using the mutual information framework introduced in Section 3.2, we derived a novel result in Section 4.3 that this learning algorithm had expected generalisation error $O\left(\sqrt{\frac{n \log m}{m}}\right)$. Using the conditional mutual

introduced in Section 3.4, we were able to derive a slightly more fine-grained bound in Section 4.4 that depended on the particular behaviour of f.

In Chapter 5 we saw that our information-theoretic analysis could be applied to a very similar algorithm by Linial et al. (1993) and we derived a novel result that their algorithm has expected generalisation error $O\left(\sqrt{\frac{|\mathcal{F}|\log m}{m}}\right)$ where $\mathcal{F}$ is an ε concentration for the Fourier weights of f.

Finally, in Chapter 6, we generalised the setup to consider learning LTFs over $\mathbb{R}^n$, with samples now drawn from the n dimensional standard Gaussian density. We saw that many of the properties form the discrete case carried over. Importantly, we saw that the Hermite coefficients, the equivalent of the Fourier coefficients in this setting, continue to characterise LTFs, and that we can estimate the Hermite coefficients in the exact same way. Furthermore, in the continuous setting, we saw that is trivial to reconstruct an approximate LTF given approximate Hermite estimates, which is very much untrue for the discrete case. Specifically, we show that if the Hermite estimates are within L_2 norm ε of the true Hermite coefficients, then the corresponding LTF is $O(\varepsilon)$ far from the true LTF. Turning to our information-theoretic framework, we show that the mutual information and conditional mutual information framework fail to provide nontrivial generalisation bounds — this is due to having a continuous feature space $\mathbb{R}^n$ and a deterministic learning algorithm. Thankfully, using the individual sample mutual information approach described in Section 3.3, we were able to derive a novel result that the learning algorithm has an expected generalisation error of $O\left(\sqrt{\frac{n}{m}}\right)$, using a different and more involved analysis compared to the discrete case.

There are a number of directions that can be pursued further with this line of work. LTFs are one of the most fundamental building blocks in learning theory and thus often used in more advanced learning algorithms, so it would be interesting to see how our theory carries over to those cases. For example, Diakonikolas et al. (2020) study the problem of PAC learning neural networks with one hidden layer and ReLU activation under the Gaussian distribution, which basically boils down to learning multiple dependent LTFs at once. Their algorithm exploits the idea of estimating the degree 2 Hermite coefficients instead of the degree 1 coefficients which was the subject of our analysis in Chapter 6.

Another direction of work is to investigate if our analysis can be tweaked to obtain bounds on more generalised information measures that produce *high probability* generalisation guarantees such as Sibson's α mutual information $I_\alpha(Z;\mathcal{A}(Z))$ (Esposito et al., 2020a), as discussed in Section 3.2. Unfortunately,

as we discuss there, not all the properties that regular mutual information satisfies carry over, hence some new ideas are required.

A different idea is to consider if our setup can be generalised to beyond the two settings of the uniform Boolean hypercube and Gaussian density that we analysed in this thesis. Furthermore, the expected generalisation error we derived in both settings was, ignoring logarithmic factors, $\widetilde{O}\left(\sqrt{\frac{n}{m}}\right)$, however the analysis we used to attain this bound was very different between the two cases. It would be interesting to see if there is a more unified approach that lets us attain this bound.

Finally, some empirical analysis could be performed to check if our bound is tight in practice, by implementing the described learning algorithms, running them, and collecting statistics about the generalisation error empirically. This would be difficult to perform in the discrete case because the process of reconstructing the LTF from the Chow estimates described by O'Donnell and Servedio (2008) is highly nontrivial, however the continuous setting is quite straightforward to implement.

Bibliography

Andrew R. Barron. 1986. Entropy and the Central Limit Theorem. *The Annals of Probability*, 14(1):336–342.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965.

Sergey G. Bobkov, Gennadiy P. Chistyakov, and Friedrich Götze. 2013. Rate of Convergence and Edgeworth-Type Expansion in the Entropic Central Limit Theorem. *The Annals of Probability*, 41(4):2479–2512.

Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. 2020. Proper Learning, Helly Number, and an Optimal SVM Bound. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 582–609. PMLR.

Yuheng Bu, Shaofeng Zou, and Venugopal V. Veeravalli. 2020. Tightening Mutual Information-Based Bounds on Generalization Error. *IEEE Journal on Selected Areas in Information Theory*, 1(1):121–130.

Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658.

Chao-Kong Chow. 1961. On the characterization of threshold functions. In *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, pages 34–38. IEEE.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. Wiley-Interscience, Hoboken, N.J, second edition.

Ilias Diakonikolas and Daniel M. Kane. 2019. Degree-d chow parameters robustly determine degree-d PTFs (and algorithmic applications). In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 804–815. ACM, Phoenix AZ USA.

Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, and Nikos Zarifis. 2020. Algorithms and SQ Lower Bounds for PAC Learning One-Hidden-Layer ReLU Networks. In *Proceedings of Thirty Third Conference on Learning Theory*, pages 1514–1539. PMLR.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. 2015. Preserving Statistical Validity in Adaptive Data Analysis. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 117–126. ACM, Portland Oregon USA.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer.

Amedeo Roberto Esposito, Michael Gastpar, and Ibrahim Issa. 2020a. Generalization Error Bounds Via Renyi-, f-Divergences and Maximal Leakage.

Amedeo Roberto Esposito, Michael Gastpar, and Ibrahim Issa. 2020b. Robust generalization via α-Mutual information. *CoRR*, abs/2001.06399.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. 2012. Agnostic Learning of Monomials by Halfspaces Is Hard. *SIAM Journal on Computing*, 41(6):1558–1590.

Steve Hanneke. 2016. The Optimal Sample Complexity of PAC Learning. *J. Mach. Learn. Res.*, 17:38:1–38:15.

David Haussler. 1992. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150.

Ibrahim Issa, Sudeep Kamath, and Aaron B. Wagner. 2016. An operational measure of information leakage. In *2016 Annual Conference on Information Science and Systems (CISS)*, pages 234–239. IEEE.

Varun Kanade. 2017. Lecture Notes: Advanced Machine Learning, University of Oxford.

Michael J. Kearns and Umesh Vazirani. 1994. *An Introduction to Computational Learning Theory*. The MIT Press.

Nathan Linial, Yishay Mansour, and Noam Nisan. 1993. Constant depth circuits, fourier transform, and learnability. *Journal of The Acm*, 40(3):607–620.

Nick Littlestone and Manfred Warmuth. 1986. Relating data compression and learnability.

Mark. 2013. Sum of Independent Folded-Normal distributions. Mathematics Stack Exchange `https://math.stackexchange.com/questions/428781/sum-of-independent-folded-normal-distributions/428790`.

James L. Massey. 1988. On the entropy of integer-valued random variables. In *Int. Workshop on Inf. Theory*.

Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil P. Vadhan. 2010. The Limits of Two-Party Differential Privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 81–90. IEEE Computer Society.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Foundations of Machine Learning (second edition).

Ryan O'Donnell. 2021. Analysis of Boolean Functions.

Ryan O'Donnell and Rocco A. Servedio. 2008. The chow parameters problem. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 517–526. ACM, Victoria British Columbia Canada.

Yuval Peres. 2021. Noise Stability of Weighted Majority. In Maria Eulália Vares, Roberto Fernández, Luiz Renato Fontes, and Charles M. Newman, editors, *In and Out of Equilibrium 3: Celebrating*

Vladas Sidoravicius, volume 77, pages 677–682. Springer International Publishing, Cham.

Leonard Pitt and Leslie G. Valiant. 1988. Computational limitations on learning from examples. *Journal of The Acm*, 35(4):965–984.

Daniel Russo and James Zou. 2016. Controlling Bias in Adaptive Data Analysis Using Information Theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1232–1240. PMLR.

Rocco A. Servedio. 2023. Private communication.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, first edition.

Claude. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Robin Sibson. 1969. Information radius. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 14(2):149–160.

Thomas Steinke. 2023. Is there a chain rule for Sibson's mutual information? Mathematics Stack Exchange `https://math.stackexchange.com/q/4639997`.

Thomas Steinke and Lydia Zakynthinou. 2020. Reasoning About Generalization via Conditional Mutual Information. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3437–3452. PMLR.

Meyer Tannenbaum. 1961. The establishment of a unique representation for a linearly separable function. *Lockheed Missilesand Space Co., Sunnyvale, Calif., Threshold Switching Techniques Note*, 20:1–5.

Leslie. G. Valiant. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Vladimir. N. Vapnik and Alexey. Ya. Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16.2:264–280.

Sergio Verdú. 2015. α-mutual information. In *2015 Information Theory and Applications Workshop, ITA 2015, San Diego, CA, USA, February 1-6, 2015*, pages 1–6. IEEE.

Aolin Xu and Maxim Raginsky. 2017. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

APPENDIX A

Mathematical results

A.1 Probability theory

We review some basic results in probability theory.

DEFINITION 23 (Convergence in probability). *A sequence of random variables $X_1, X_2, \cdots$ converges in probability to $c \in \mathbb{R}$, written $X_n \xrightarrow{P} c$, if*

$$\lim_{n\to\infty} \mathbf{P}\left[|X_n - c| \geq \varepsilon\right] = 0. \tag{A.1}$$

DEFINITION 24 (Convergence in distribution). *A sequence of random variables $X_1, X_2, \cdots$ converges in distribution to a random variable X, written $X_n \xrightarrow{d} X$, if*

$$\lim_{n\to\infty} F_{X_n}(x) = F_X(x) \tag{A.2}$$

for all x at which $F_X(x)$ is continuous, where F_{X_n} and F_X denote the c.d.f. of X_n and X respectively.

THEOREM 30 (Weak law of large numbers). *Let $X_1, X_2, \cdots$ be i.i.d. random variables with mean $\mathbf{E}[X_1] = \mu$. Then,*

$$\frac{1}{n}\sum_{i=1}^{n} X_n \xrightarrow{P} \mu. \tag{A.3}$$

THEOREM 31 (Lindeberg-Lévy central limit theorem). *Let $X_1, X_2, \cdots$ be i.i.d. random variables with mean $\mathbf{E}[X_1] = \mu$ and variance $\mathrm{Var}[X_1] = \sigma^2 < \infty$. Then,*

$$\sqrt{n}\left(\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2). \tag{A.4}$$

THEOREM 32 (Jensen's inequality). *Let φ be a convex function, and X a random variable. Then*

$$\mathbf{E}\left[\varphi(X)\right] \geq \varphi(\mathbf{E}X). \tag{A.5}$$

THEOREM 33 (Markov's inequality). *Let $X \geq 0$ be a nonnegative random variable. Then*

$$\mathbf{P}[X \geq c] \leq \frac{\mathbf{E}[X]}{c} \tag{A.6}$$

for any $c > 0$.

THEOREM 34 (Hoeffding's inequality). *Let $X_1, \cdots, X_m$ be independent random variables such that $X_i \in [a_i, b_i]$ for $i \in [m]$, and let $S_m := \sum_{i=1}^m X_i$. Then*

$$\mathbf{P}\left(|S_m - \mathbf{E}S_m| \geq c\right) \leq 2\exp\left(-\frac{2c^2}{\sum_{i=1}^m (b_i - a_i)^2}\right).$$

Hoeffding's inequality can be used to prove the following result that states the required sample complexity required for the sample mean to be close to the true mean.

THEOREM 35. *Given m i.i.d. samples $X_1, \cdots, X_m$ of a bounded random variable taking values in $[a, b]$, the sample mean $\frac{1}{m}\sum_{i=1}^m X_i$ is within an additive $\pm\varepsilon$ of the true mean $\mathbf{E}X_1$ with probability at least $1 - \delta$ when using at least*

$$m = \frac{(b-a)^2}{2\varepsilon^2}\log\left(\frac{2}{\delta}\right) = (b-a)^2 O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right) \tag{A.7}$$

samples.

A.2 Calculus

We review some basic results in calculus.

THEOREM 36 (L'Hôpital's rule). *Suppose f, g are differentiable with $\lim_{x\to a} f(x) = \lim_{x\to a} g(x) = 0$ or $\lim_{x\to a} f(x) = \lim_{x\to a} g(x) = \infty$ for some real (possibly infinite) number a. Then,*

$$\lim_{x\to a}\frac{f(x)}{g(x)} = \lim_{x\to a}\frac{f'(x)}{g'(x)}. \tag{A.8}$$

THEOREM 37 (Multivariate change of variables theorem). *Let $\varphi : (x, y) \to (u, v)$ be a differentiable invertible map between two open subsets of $\mathbb{R}^2$. Then,*

$$\iint_A f(x,y)\,dx\,dy = \iint_{\varphi(A)} f(\varphi^{-1}(u,v))\left|\frac{d(x,y)}{d(u,v)}\right| du\,dv, \tag{A.9}$$

where,

$$\frac{d(x,y)}{d(u,v)} := \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \tag{A.10}$$

is the Jacobian matrix of φ^{-1} *and* $\left|\frac{d(x,y)}{d(u,v)}\right|$ *denotes the absolute value of the determinant of the Jacobian matrix.*