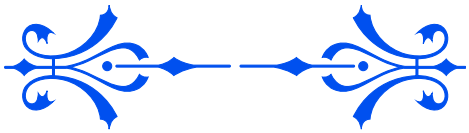
PH.D. THESIS

DESIGNING QUANTUM LEARNING ALGORITHMS FOR CLASSICAL OBJECTS

SAGNIK CHATTERJEE

2025

Designing Quantum Learning Algorithms for Classical Objects



THESIS

submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Sagnik Chatterjee

PHD19007

Indraprastha Institute of Technology, Delhi

New Delhi- 110020

Designing Quantum Learning Algorithms for Classical Objects

Candidate: Sagnik Chatterjee

Advisor: Prof. Debajyoti Bera (IIIT-Delhi)

Thesis Committee:

Prof. Debajyoti Bera **(Chair)**
(IIIT-Delhi)

Prof. Ambuj Tewari
(University of Michigan)

Prof. Piyush Srivastava
(TIFR-Mumbai)

Dr. Min Hsiu-Hsieh
(Hon Hai Quantum Computing Research Center)

Ph.D. in Computer Science and Engineering

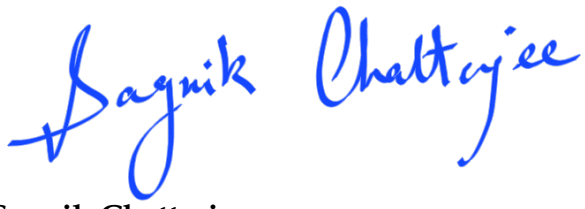
Indraprastha Institute of Information Technology, Delhi

To Maa and Baba,

and to their unconditional love and support that made this journey possible.

Declaration

I declare that I am the author of the thesis titled **Designing Quantum Learning Algorithms for Classical Objects**. This dissertation contains the research I have conducted under the guidance of Prof. Debajyoti Bera (CSE, IIIT Delhi). To the best of my knowledge, it is an original work, and has not been submitted elsewhere, in parts or in full, for a degree.



Sagnik Chatterjee
Doctoral Candidate,
Dept. of Computer Science and Engineering,
IIIT Delhi, 110020.

Place: 11.06.25

Date: New Delhi

Thesis Certificate

This is to certify that the thesis titled **Designing Quantum Learning Algorithms for Classical Objects**, submitted by **Sagnik Chatterjee**, to the Indraprastha Institute of Technology, Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Debajyoti Bera
Thesis Supervisor,
Associate Professor,
Dept. of Computer Science and Engineering,
IIIT Delhi, 110020.

Place: New Delhi

Date: 4th June, 2025

Acknowledgements

At the outset, I want to thank my advisor, Prof. Debajyoti Bera[1], for being the best mentor any budding researcher could ask for. Without his trust and unconditional support throughout my PhD, I would never have made it this far into my research journey. His door was (and still is, to this day) always open for my incessant questions, something that I used to take for granted but have come to appreciate more as I have matured in academia. I am truly grateful that at the beginning of my PhD, his patience was unending whenever my focus faltered (a recurring occurrence). I am also grateful that once I had become comfortable in my skin, he was always there to reign me in when I was plunging all my digits in as many different pies[2] as possible (another recurring occurrence).

The next person I want to acknowledge is Prof. Manuj Mukherjee. When talking about him, two things immediately come to mind - math and coffee. Even though we crossed paths quite late in my PhD, his influence on my PhD is second only to my advisor. He has been instrumental to my growth as a researcher, and the second part of my thesis was only possible due to a fruitful collaboration with him. Our long talks/heated debates over (and also centered around) freshly brewed coffee and other essential topics ranging from citation styles to artistic outlooks, etc.[3] are some of my fondest memories during my PhD. Last but not least, I want to thank him for rekindling the love of pure math in me, something I thought had died a long time ago.

Prof. Andris Ambainis drastically changed the trajectory of my PhD journey by inviting me to Riga for a talk and extending an offer to host me for a postdoctoral position after graduation. He has been a constant presence during the final year of my PhD and an endless source of encouragement for my often outlandish research directions. Our research meetings *always* leave me with a sense of awe and a yearning desire to learn more. Thinking aloud in his presence is always a joy, especially when

[1] a.k.a. Dbera, as he is more fondly known throughout the student body.
[2] Read as “starting a multitude of research projects".
[3] I am choosing to exclude some of the more esoteric topics from this text.

he effortlessly molds and refines your argument into the best possible form.

I am grateful to the members of my Ph.D. committee Prof. Ambuj Tewari, Prof Piyush Srivastava, and Dr. Min-Hsiu Hsieh for agreeing to pe part of the committee and providing helpful and encouraging comments on my thesis. I would also like to thank Prof. Bapi Chatterjee, Prof. Saket Anand, and Prof. Vinayak Abrol for agreeing to serve on my comprehensive and yearly review committees and providing me with feedback and encouragement throughout my PhD journey.

A big part of my PhD journey was the transition from a systems engineering background to theoretical computer science, which would have been impossible without some of the best educators in the field, with whom I have had the fortune of interacting. I would like to thank Prof. Syamantak Das, Prof. Rohit Vaish, and Prof. Subhabrata Samajder especially for being *excellent* teachers and widening my horizons in different areas of theoretical computer science. I learned almost all I know about theoretical computer science in their classrooms or in front of their office whiteboards. I am also grateful to Prof. Syamantak Das (once again[4]), Prof. Nikhil Gupta, and Prof. Ravi Anand for always having their doors open for a quick chat on any topic - ranging from rants over visa procedures, and research trends, to extremely detailed and helpful career advice.

Prof. Abhranil Chatterjee (Abhranil*da*) first introduced me to the A.C.M.U. unit at the Indian Statistical Unit when he invited me for a talk at ISI. Following this, Prof. Sourav Chakraborty graciously agreed to host me at ISI in the fall of 2024 when I was writing my thesis. I often reflect fondly on some of our discussions regarding the “big open questions" early in the morning over a freshly brewed cup of coffee. One of the most interesting things at ISI was the constant stream of talks by visiting researchers and the accompanying follow-up discussions. The highlight of my ISI stint was the mid-morning tea sessions and lunches at *Bapida's canteen* with Abhranil*da*, Santanu, Debarshi, Uddalok, and many others. I am also grateful to Dr. Vyacheslav Kungurtsev (Slava) and Prof. Jakub Marecek for hosting me at CTU in Prague during the summer of 2023. I also want to express my gratitude towards Mrs. Priti Patwal - the manager of CSE 5th floor, for making my PhD as frictionless with respect to administrative issues.

My friends, peers, and fellow PhD students at the theory lab in IIIT Delhi deserve my unreserved thanks for constantly being there for me at various points in the journey - Rahul*da*, Tharrma, Atish, Yash, Karamjeet, Mohit, Suryendu, Ritesh, Nakul, Jaya, Sanchita, Sudatta, Mahesh, Arpit, Deepika, Purbasha, Debashish, Saurav Awasthi, and Saurav Das. I would like to thank Alhad for being a wonderful collaborator and an

[4] I finally thank Prof. Syamantak Das for the third time for taking me out for coffee, lunches, and dinners *many many times*, and introducing me to Kashmiri cuisine in particular.

excellent mentee. Despite him being a *mere* undergrad (/s), I learned more from him than I ever taught him - in fact, his brilliance made up for any faulty advice I ever gave him. The latter part of the sentiment also extends to Mudit, Farhan, and Devansh, another set of budding theory superstars, with all of whom I had the fortune of interacting with in their formative years. Ballari, words alone cannot express how much your support means to me. The journey alone would not have been possible without you beside me, let alone the outcome.

Finally, my parents. I would never even have thought about doing a PhD if it was not for their constant encouragement. They sacrificed too much for my sake at every turn and never doubted that I could accomplish whatever I had set out to do, even at times when I had completely given up. They are always there for me with unbridled acceptance, love, and support, making even the toughest challenges seem surmountable. Thank you for everything and more.

Abstract

In the first part of this dissertation, we focus on identifying the advantage afforded by quantum algorithms in the field of learning theory, which began as a mathematical framework in the mid-20th century to understand how algorithms can generalize from data [VC71; VC74; Vap82; Val84]. We specifically focus on the supervised setting under *unbounded* label-noise.

Classically, efficient algorithms for various learning problems in this particular setting are often allowed to query Membership Query (MQ) oracles, which have historically been criticised for being too strong, both in a theoretical and practical sense. A long-standing open question in learning theory originating with the seminal work of Ehrenfeucht and Haussler [EH89] is as follows: Do there exist efficient learning algorithms for Boolean functions that have polynomial-sized decision tree representations under *unbounded label noise* and *without Membership query access*, assuming uniform marginal distribution over the instances? In the first part of our thesis, we answer the above question in the affirmative, by showing that there exists a quantum learning algorithm for the same.

In the second part of our thesis, we introduce techniques to analyze the generalization behavior of statistical learning algorithms (both quantum *and* classical) trained on non-i.i.d. data under bounded and unbounded label noise. Our proof techniques involve generalizing a specific variant of the well known Online-to-Batch conversion paradigm [LN23] to the setting where the underlying data is drawn from β and ϕ mixing stochastic processes.

Contents

Prologue **1**

1 Introduction **3**
1.1 Feynman to Grover, and beyond! . . . 3
1.2 A Brief Overview of Learning Theory . . . 4
1.3 Our Contributions . . . 5
1.4 Bibliographic Note . . . 13

2 Preliminaries **15**
2.1 Basic Notation . . . 16
2.2 Useful Mathematical Results . . . 17
2.3 Boolean functions . . . 19
2.4 Basics of Computational Learning Theory . . . 22
2.5 Basics of Quantum Computing . . . 30

I Quantum Weak Learners **41**

3 Quantum and Classical Oracle Models **43**
3.1 Introduction . . . 44
3.2 Example Query Oracles . . . 44
3.3 Membership Query Oracles . . . 47
3.4 Exact and Approximate Simulatibility of Membership Query Oracles . . . 50
3.5 Conclusion and Future Work . . . 53

4 Quantum Weak Learners **55**
4.1 Introduction . . . 56
4.2 Weak and Strong Learning . . . 56
4.3 Designing Weak Learners . . . 57
4.4 Decision Trees . . . 61
4.5 Weak Learning Decision Trees w/o Membership Query (MQ) . . . 62
4.6 Discussion and Conclusion . . . 69

II Quantum Ensemble Learning 71

5 Realizable Quantum Boosting for Domain Partitioning Weak Learners 73
5.1 Introduction . . . 74
5.2 The ADABOOST algorithm and its variants . . . 77
5.3 The QREALBOOST algorithm . . . 82
5.4 Discussion . . . 96

6 Quantum Agnostic Boosting 99
6.1 Introduction . . . 100
6.2 The Quantum Agnostic Boosting Algorithm . . . 100
6.3 Discussion . . . 112

III Generalization Error Bounds 115

7 Generalization Error and Empirical Behaviour of QREALBOOST 117
7.1 Generalization Bounds on QREALBOOST . . . 118
7.2 Empirical evaluations of QREALBOOST . . . 120
7.3 Discussion and Future Work . . . 128

8 Generalization Error Bounds for Dependent Data 129
8.1 Introduction . . . 130
8.2 Generalization Error Bounds . . . 142
8.3 Discussion and Future Work . . . 148

IV Epilogue 151

9 The Quantum Learning Zoo 153
9.1 Introduction . . . 154
9.2 Sample Complexity separations . . . 154
9.3 Time Complexity separations . . . 158
9.4 Discussion and Conclusion . . . 170

Appendix 171

A Omitted Definitions 172
A.1 Group Theory . . . 172

B Omitted Algorithms 173
B.1 A General Framework for Realizable Boosting . . . 174
B.2 The ADABOOST algorithm . . . 176
B.3 The REALBOOST algorithm . . . 177
B.4 The SMOOTHBOOST algorithm . . . 178
B.5 The POTENTIALBOOST algorithm . . . 180

C Omitted Proofs **181**
C.1 Proofs from Chapter 5 181
C.2 Proofs from Chapter 7 184

D List of Figures **187**

E List of Tables **188**

F List of definitions **189**

Bibliography **191**

Prologue

The secret of getting ahead is getting started.

MARK TWAIN

Chapter 1

Introduction

§ 1.1 Feynman to Grover, and beyond!

In the early 1980s, researchers conceived the idea of computing devices that could leverage quantum mechanical principles [Fey82; Fey85; Deu85]. Their vision came to fruition as several groundbreaking quantum algorithms were proposed that solved *classical* computational tasks either provably faster than classical computers [Gro96] or performed tasks (*still*) not known to be solvable by classical computers [Sho94]. Such advances opened the proverbial floodgates for attacks on various areas of traditional computer science, previously thought to have reached their computational barriers.

One such affected area in theoretical computer science was learning theory, a mathematical framework introduced in the 1970s [VC71; VC74; Vap82; Val84]; for understanding how algorithms can generalize from data. Learning theory is closely connected to fields like cryptography, coding theory, and circuit complexity, making it an extremely interesting area of research. Towards the close of the twentieth century, researchers started encountering theoretical barriers to *efficient learning*[1], even for extremely simple classes of Boolean functions.

Soon after the breakthrough results of Shor and Grover, a promising new avenue of research in learning theory was discovered when Bshouty and Jackson [BJ95] circumvented certain classical barriers for learning DNF formulas by using *quantum query algorithms*. This result effectively established the new field of quantum learning theory, and since then, quantum learning theory has grown in tandem with the advances in quantum computing and learning theory.

[1] There are various notions of learning and efficient learning, many of which will be formally introduced (starting from Chapter 2) and extensively discussed throughout this thesis.

In this dissertation, we concern ourselves with quantum *advantage* in the context of *classical* supervised learning tasks (e.g., learning Boolean functions) under various extensions to the traditional PAC learning setup, which include (but are not limited to) the following scenarios:

- efficient learning under bounded and unbounded label noise,
- efficient learning with access to weaker query models,
- generalization behavior of learners from a margin optimization viewpoint, and
- generalization behavior of learners trained on extremely dependent data.

§ 1.1.1 Structure and Organization

In the following sections, we detail the contributions of our thesis, along with short overviews of the proof techniques involved.

1.2 A Brief Overview of Learning Theory 4
1.3 Our Contributions 5
1.3.1 Sampling Queries are weaker than Active Queries 5
1.3.1.1 Summary of our results 6
1.3.2 Learning Fourier-Sparse Boolean functions 7
1.3.2.1 Summary of our results 9
1.3.3 Generalization Error Bounds of Statistical Learners 9
1.3.3.1 Summary of our results 12
1.4 Bibliographic Note 13

§ 1.2 A Brief Overview of Learning Theory

The overarching goal of computational learning theory is to efficiently construct a model (known as a **learning algorithm**, or simply, a **learner**) given a set of possibly noisy data drawn from some fixed yet unknown distribution, to make accurate predictions on unseen data drawn from the same underlying distribution.

In this dissertation, we are concerned with supervised learning (only under label noise) setups for discriminative learning tasks, which may be formulated as follows[2]. Suppose $c : \mathcal{X} \mapsto \mathcal{Y}$ is an unknown labeling function (or concept) belonging to a class of labeling functions $\mathcal{C}$, that labels the instance space $\mathcal{X}$ with elements from $\mathcal{Y}$. The goal of learning is to *efficiently* construct a (randomized or quantum) algorithm A, s.t., for

[2] See Section 2.4.1 for a formal introduction to computational learning theory.

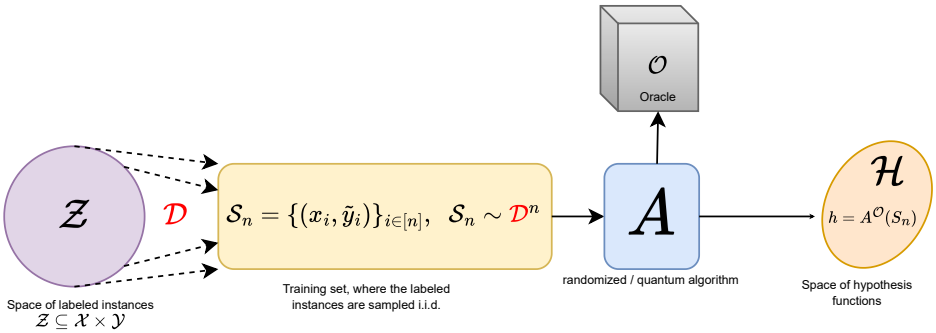


Figure 1.1: Learning with queries.

$\varepsilon \in (0, 1/2)$, $\delta \in (0, 1]$,

$$\Pr\left[\Pr_{x\sim\mathfrak{D}}\left[c(x) \neq h_{A,S_n,\mathcal{O}}(x)\right] \leqslant \varepsilon\right] \geqslant 1-\delta \tag{1.1}$$

where $h_{A,S_n,\mathcal{O}} = A^{\mathcal{O}}(S_n)$ is the hypothesis produced when the learner A is provided access to a labeled set [3] of instances $S = \{(x_i, \tilde{y}_i)\}_{i\in[n]} \sim \mathfrak{D}^n$, and can make black-box queries to an oracle $\mathcal{O}$ to obtain labels of instances. The outer probability in Eq. (1.1) is over the choice of the training set S_n and any internal randomization of A[4].

Remark 1.2.1. The notions of *efficiency*, *accuracy*, and *noise* are subject to the choice of the underlying computational and/or learning model. We shall explore and formalize these models, along with the above concepts, in more detail later in Chapter 2.

§ 1.3 Our Contributions

We now give a high-level outline of our contributions as presented in this dissertation.

§ 1.3.1 Sampling Queries are weaker than Active Queries

There are primarily two types of oracle queries a learner can make - **active queries** and **sampling queries**. Every sampling query by the learner results in (possibly an ensemble of) labeled instances $(x, y) \sim \mathfrak{D}$. Here the learner has no control over *what* samples gets queried. On the contrary, with active queries, the learner queries the oracle with an instance and obtains its label. Hence, in the active query setup, unlike the sampling query setup, the learner has more fine-grained control over the labeled instances it has access to. We enumerate two key sampling oracle models now.

[3] The tilde notation denotes that the labels are possibly noisy.

[4] This setup extends the original PAC learning setup [Val84]. We also note that in Eq. (1.1), the set of labeled instances is drawn i.i.d. from some distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$. This assumption will be relaxed later.

1. The Example Query (EX) oracle. Querying the classical random example oracle $\mathrm{EX}(\mathfrak{D}, c)$ generates a single labeled example $(x, c(x))$ where $x \sim \mathfrak{D}$.
2. The Quantum Example Query (QEX) oracle.

$$\mathrm{QEX}(\mathfrak{D}, c) : |0, 0\rangle \mapsto \sum_{x \in \mathbb{F}_2^n} \sqrt{\mathfrak{D}(x)}\, |x, c(x)\rangle .$$

It is straightforward to observe that the QEX oracle is a quantum generalization of the EX oracle. We now turn our attention to a specific active oracle model. In the context of Boolean function learning over the Boolean hypercube, i.e., for $\mathcal{X} = \{0,1\}^n, \mathcal{Y} = \{\pm 1\}$,[5] the Membership Query (MQ) oracle is a central construct and the key ingredient behind many efficient learning algorithms. Let c be the unknown concept that we are trying to learn. Querying the MQ oracle MEM_c with an instance $x \in \mathbb{F}_2^n$ returns the instance's label $c(x)$ directly. As we can see, the MQ oracle allows us to make very powerful *out-of-distribution* queries. In fact, Bshouty [Bsh95] proved that any n-bit Boolean function is *efficiently* learnable with MQ access in polynomial time in n and its Disjunctive Normal Form (DNF) size.

§ 1.3.1.1 Summary of our results

In Chapter 3, we start by providing a proof for the following folklore result:

> **Theorem (Informal).** A PAC learner A for $c \in \mathcal{C}$ with access to MEM_c has polynomially smaller query complexity compared to A with access to an $\mathrm{EX}(\mathfrak{D}, c)$ oracle.

We then show that MQ queries are stronger than quantum sampling queries w.r.t. the uniform distribution. Suppose we are given access to a (quantum) sampling oracle $\mathcal{O}(\mathfrak{D}, h)$ with the promise that $h = f$ or $h = g$, where $f, g \in \mathcal{F}$ belong to the same Boolean function family, s.t. $\|f - g\|_1 \leqslant \varepsilon$ for some $\varepsilon > 0$. The sampling oracle $\mathcal{O}$ can be used to **approximately simulate** MQ for $\mathcal{F}$ w.r.t. some distribution $\mathfrak{D}$ if there exists an efficient (quantum) algorithm A that can figure out if $h = f$ or $h = g$ with query access to $\mathcal{O}(\mathfrak{D}, h)$.

> **Theorem (Informal).** QEX cannot approximately simulate MQ w.r.t. the uniform distribution over the Boolean hypercube.

This extends an earlier result by [BJ95], who showed a separation for exact simulatibility, i.e., when f and g differ on exactly one instance.

[5] We will be using the sets $\{0,1\}$ and $\mathbb{Z}_2$, and the sets $\{\pm 1\}$ and $\mathbb{F}_2$ interchangeably.

§ 1.3.2 Learning Fourier-Sparse Boolean functions

Sometimes, it is notoriously hard to construct efficient learning algorithms, even for extremely simple classes of Boolean functions. For example, a major open question in learning theory revolves around whether decision tree representations of Boolean functions can be learned efficiently. Constructing decision tree representations for functions is an important algorithmic problem dating back to the work of Hyafil and Rivest [HR76], who showed that this problem is NP-complete.

Despite this discouraging start, Ehrenfeucht and Haussler [EH89] managed to show that given any n-bit Boolean function f which can be represented by a decision tree with at most t leaves, there exists an algorithm to construct a quasi-polynomial sized decision tree representation that *approximates* f to high accuracy in quasi-polynomial time. This sparked a long line of research centered around efforts to shave off the super-constant exponent. However, recently Koch et al. [KST23], showed superpolynomial lower bounds for proper decision tree learning, indicating that the result by [EH89] was nearly-optimal.

Even though learning decision tree representations of boolean functions efficiently is impossible (as discussed above), nothing in the preceding discussion precludes us from learning the function f itself efficiently. Alternatively, we can pose the following research question.

> **Research Question 1.** Consider the class $\mathcal{T}$ of decision trees on n-bit instances that have at most poly (n) leaves. Can we *efficiently learn* any unknown $T \in \mathcal{T}$?

Note that in Research Question 1, we only care about approximating the unknown decision tree T by constructing any suitable model/learning algorithm A **instead of** *constructing another decision tree* $T' \in \mathcal{T}$ that approximates T.

Remark 1.3.1. For reasons detailed in Chapters 2 and 4, the class of Boolean functions captured by $\mathcal{T}$ are known as Fourier-sparse Boolean functions.

This viewpoint on decision tree learning turns out to be much more profitable for research on efficient learning algorithms, as was demonstrated early on by Kushilevitz and Mansour [KM91], who showed that the class of n-bit decision trees can be learned with respect to the uniform distribution and with access to Membership Query (MQ) oracles. We see that **two additional restrictions** were required for efficient learnability in the result of Kushilevitz and Mansour [KM91] :

1. The (noiseless) data is assumed to be sampled from the uniform distribution;
2. The learning algorithm needs access to active oracle queries.

In Section 3.3, we shall explore why access to membership queries is undesirable simultaneously from a theoretical and a practical viewpoint. For the time being, we limit our discussions to center around the fact that we either want to eliminate the MQ assumption or, at the very least, meaningfully weaken the assumption. This brings us to the next major research problem.

> **Research Question 2.** Consider the class $\mathcal{T}$ of decision trees on n-bit instances that have at most poly (n) leaves. Can we efficiently learn any unknown $T \in \mathcal{T}$ *without query access to MQ oracles*?

We recall our earlier discussion on sampling queries, which are provably weaker than active queries. In this vein, Bshouty and Jackson [BJ95] answered Research Question 2 affirmatively by constructing a (quantum) learning algorithm for DNF learning that only makes queries to QEX oracles. This immediately implies that n-bit decision trees can *also* be quantumly learned with respect to uniform distribution and access to QEX queries.

▷ Decision Tree learning under label noise.

Unfortunately, the above results on decision tree learning do not always hold when the training data is subject to *label noise*, which we introduce now and formally define later in Section 2.4.3.2. Given a set $S = \{(x_i, y_i = f(x_i))\}_{i\in[n]}$ of n training examples, where the instances are labeled by some unknown binary labeling function f, we want to construct an algorithm A that *learns* f. Suppose there is an adversary who flips the labels $y_i \mapsto \tilde{y}_i$ with some probability $p_i > 0, \forall i \in [n]$, i.e., the learning algorithm A now has access to a noisy training set $S = \{(x_i, \tilde{y}_i)\}_{i\in[n]}$ s.t.

$$\tilde{y}_i = \begin{cases} \overline{y_i} = \overline{f(x_i)}, & \text{w.p. } p_i, \\ y_i = f(x_i), & \text{w.p. } 1 - p_i. \end{cases}$$

When $0 < p_i \leqslant p < 1/2$, it is said to be the bounded noise model. When $0 < p_i \leqslant 1/2 - \varepsilon_i$ for any $\varepsilon_i > 0$, $i \in [n]$; i.e., each label can be flipped w.p. (dependent on the instance) arbitrarily close to $1/2$, we are in an unbounded noise model. The unbounded noise model is also known as the **agnostic setting**, which in some sense is the hardest label noise setting. We now pose an updated version of Research Question 2.

> **Research Question 3.** Consider the class $\mathcal{T}$ of decision trees on n-bit instances with at most poly (n) leaves. Can we efficiently learn any unknown $T \in \mathcal{T}$ *without query access to MQ oracles* w.r.t. (possibly unbounded) label noise?

In the first part of this dissertation, we answer Research Question 3 in the affirmative by giving a quantum learning algorithm for the same. We note here that prior to the results outlined in this thesis, Research Question 3 has been an open question for almost four decades.

§ 1.3.2.1 Summary of our results

Our main contributions in this part are as follows:

1. In Chapter 4, we show how to efficiently construct *agnostic* quantum weak learners for learning Boolean functions with polynomial-sized decision tree representations under unbounded label noise. Our weak learners only make queries to *noisy* quantum oracles, which are provably weaker than Membership query oracles.
2. In Chapter 6, we give the first efficient quantum agnostic boosting algorithm, which answers an open question posed by [IW23].

Combining the above three results gives us a solution to Research Question 3. We also show the following results.

3. In Chapter 4, we also show how to obtain similar results in the noiseless and bounded noise settings by combining the Fourier Sampling algorithm [BV97] with noiseless quantum boosting algorithms.
4. On this note, in Chapter 5, we also show how to design efficient quantum boosting algorithms (in the noiseless setting) that can boost weak learners that output domain-partitioning hypotheses. This answers an open question posed by [AM20; IW23].

§ 1.3.3 Generalization Error Bounds of Statistical Learners

In the second part of this dissertation, we turn our attention to the generalization error behavior of statistical learning algorithms (both quantum and classical). In the earlier setup, the learner A simply mapped the training set S_n to a particular hypothesis $h \in \mathcal{H}$, with the aim that h approximates some target labeling function f. A statistical learning algorithm A, on the other hand, takes as input a set S_n of (possibly noisy) instances and *induces a distribution* $P_{A(S_n)}$ on the hypothesis set $\mathcal{H}$. The goal of the learner is to ensure that $h \sim P_{A(S_n)}$ approximates the unknown labeling function f, in expectation. Note that similar to the earlier definition of learning, both quantum and randomized learners can be modeled as statistical learning algorithms. See Fig. 1.2 for a visual representation of statistical learners.

Traditionally, a lot of emphasis has been placed on the **empirical risk minimization**

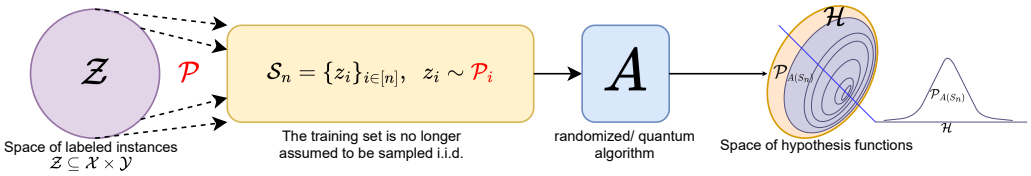


Figure 1.2: A visual representation of a statistical learning algorithm.

framework under which the main design emphasis of statistical learning algorithms is on minimizing some appropriately chosen loss function over their training set. If the training set is "large enough" w.r.t. to the size of the instance space, one can show that the learning algorithm *generalizes well* on instances not seen before.

However, the above framework comes with certain downsides. Sometimes, in order to satisfy empirical risk minimization, statistical learners end up *fitting* to their training data too well. While this is usually a desirable property, sometimes this leads to overfitting issues - where the error incurred by the learner on unseen data increases as the error on the training set decreases. While normally overfitting is a concern, in Chapter 7, we prove that our noiseless quantum-boosting algorithm **induces large margins** over the training set - a property allowing *test error to reduce even after training error has converged to zero*.

> **Theorem (Informal).** QREALBOOST induces large margins on its training set.

This indicates that contrary to traditional theories on overfitting in machine learning algorithms, our quantum boosting algorithm can continue to generalize well after the training error has converged to zero. We provide empirical evidence of the above hypothesis with experiments on the `Breast Cancer Wisconsin` and `MNIST` datasets, where we contrast the empirical behavior of QREALBOOST against various existing classical and quantum boosting algorithms.

The issues with traditional theories on generalization behavior in machine learning models run much deeper. In the modern regime of overparameterized models that have billions of tunable parameters, generalization error bounds based on complexity measures of the hypothesis space are usually vacuous. Another issue in generalization error bounds is the implicit i.i.d. assumption, which rarely holds in a realistic setting, especially when it comes to distributed learning setups such as federated learning. To this end, we consider the following research question:

Research Question 4. Can we obtain generalization error bounds for statistical learning algorithms (both quantum and classical) that are trained on *non-i.i.d. data without relying on combinatorial measures* of the learner's hypothesis space, under bounded and unbounded label noise?

Classically, the generalization error of an offline learner A has been characterized in terms of the VC-dimension or the Rademacher complexity of $\mathcal{H}_A$ [Yu94; Mei00]. However, recently, with the advent of the over-parameterized regime, it has been noted that the traditional models of generalization error are often vacuous w.r.t. the generalization behavior of massively large neural networks which have billions of tunable parameters [Zha+17]. In an effort to explain this discrepancy, instead of focusing solely on complexity measures of the learner's hypothesis space, researchers have proposed algorithm-dependent generalization error bounds, such as bounds due to stability [BE02], information-theoretic properties [RZ16; XR17], or PAC-Bayesian bounds [Hel+24; Alq24]. Most of these bounds, however, assume that the training and test samples are drawn i.i.d. from the same (unknown) underlying distribution.

▷ Generalization Bounds for Batch Learners in the Non-i.i.d. setting.

In many real-world applications, such as in learning from time-series dependent data such as stock prediction tasks, the i.i.d. assumption does not hold [Vid13]. A more "modern" example of a learning setting in which the i.i.d. assumption becomes untenable is federated learning, where the presence of non-i.i.d. data has been shown to be a crucial barrier in real-world deployment of architectures [Ami+22; Xio+22; Iye24; Li+24].

In learning with non-i.i.d. data, we usually assume that the data is drawn from a mixing random process. Initially, techniques based on uniform convergence over complexity measures of the hypothesis space [Yu94; Mei00; MR08] were used to bound the generalization error of learning algorithms. Later, techniques based on algorithmic stability applied to different notions of mixing processes [MR10; Zha+19; Fu+23] were used to study the generalization performance of learning algorithms. These works usually enforce strong stationarity conditions on the mixing process and/or require stability conditions directly on the batch learner itself, unlike our technique.

Concurrent to our work, Abeles et al. [ACN24] independently addressed the same problem. Both works derive generalization bounds of the same order, i.e., $\frac{\text{regret}}{n} + O(\frac{1}{\sqrt{n}})$, but involve different assumptions, and hence vastly different techniques, and also differ in the formulation of the Online-to-Batch framework. While the two generalization bounds are extremely similar, there are subtle differences when considering the PAC-Bayesian instantiations of the bounds. In our bounds, we incur an additive

term arising due to the specific definition of our mixing process, while in [ACN24], they incur a multiplicative factor of $\log n$, which arises due to their use of a delay in the OtB technique. This can potentially lead to worse PAC-Bayesian instantiations compared to our bounds.

§ 1.3.3.1 Summary of our results

In Chapter 8, we extend the Online-to-Batch (**OtB**) conversion framework of [LN23] to bound the generalization error of statistical learning algorithms (both quantum and classical) that are trained on data sampled from weak mixing stochastic processes. We assume that the statistical learners are trained on data sampled from a stochastic process that is "mixing" to a stationary distribution. Mixing is a fairly natural assumption in which the dependence between data samples weakens over time and has been the setting of choice for analyzing the performance of optimization and learning algorithms trained on non-i.i.d. data (see for example [Yu94; Mei00; LKS05; MR07; MR10; AD11; Duc+12; KM17; Zha+19; Fu+23]). Since our bounds are PAC-Bayesian in nature and do not depend on combinatorial measures of the learner's hypothesis space, such as VC-dimension, the following theorem answers Research Question 4.

> **Theorem (Informal)** (Generalization Bounds for Statistical Learning Algorithms)**.** Consider any statistical learning algorithm A trained on samples drawn from a suitably mixing random process. Then, for any arbitrary Wasserstein-stable online learner $\mathcal{L}$ obtained from a corresponding n-round OTB game played w.r.t. A, the expected generalization error of A is upper bounded by $\frac{1}{n}\mathbb{E}\left[\text{regret}_{\mathcal{L}}\right] + O\left(\frac{1}{n}\right)$. Furthermore, the generalization error of the learner A is upper bounded by $\frac{1}{n}\text{regret}_{\mathcal{L}} + O\left(\sqrt{\frac{1}{n}\cdot\log\left(1/\delta\right)}\right)$ with probability at least $1-\delta$.

Remark 1.3.2. Our bounds are non-vacuous and can be instantiated via the well-known class of EWA learners, which leads to PAC-Bayesian bounds for generalization error that are almost optimal w.r.t. the i.i.d. setting upto *logarithmic factors*. We refer the reader to Chatterjee et al. [CMS25] for a detailed discussion on this topic.

▷ Connection to results in Sections 1.3.1 and 1.3.2.

At this point, we recall that statistical learners induce a distribution over the hypothesis space, and we make no assumptions regarding how such a distribution is induced; therefore, the statistical learners can themselves be randomized or quantum algorithms (and are even allowed to make queries). Furthermore, the generalization error of statistical learners is evaluated with respect to positive real-valued loss functions. This generalizes the $0/1$ loss function used to obtain error bounds for the quantum

learners (both realizable and agnostic) in Section 1.3.2. Finally, the generalization error bounds in Chapter 8 hold for any real-valued loss functions and hypothesis classes that satisfy Assumptions 8.1 to 8.3, that are straightforwardly satisfied by the hypothesis classes of different representations of Boolean functions, and the 0/1 loss metrics used for obtaining quantum weak and strong learners described in Section 1.3.2.

§ 1.4 Bibliographic Note

The majority of the results in this thesis are part of the following papers.

[Cha+23] Sagnik Chatterjee, Rohan Bhatia, Parmeet Singh Chani, and Debajyoti Bera. "Quantum boosting using domain-partitioning hypotheses". In: *Quantum Machine Intelligence* 5.2 (July 2023), p. 33. ISSN: 2524-4914. DOI: 10.1007/s42484-023-00122-3 (cit. on p. 13).

[CMS25] Sagnik Chatterjee, Manuj Mukherjee, and Alhad Sethi. "Generalization Bounds for Dependent Data using Online-to-Batch Conversion". In: *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*. accepted. 2025 (cit. on p. 13).

[CSB24] Sagnik Chatterjee, Tharrmashastha SAPV, and Debajyoti Bera. "Efficient Quantum Agnostic Improper Learning of Decision Trees". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li. Vol. 238. Proceedings of Machine Learning Research. PMLR, May 2024, pp. 514–522. URL: https://proceedings.mlr.press/v238/chatterjee24a.html (cit. on p. 13).

The author of this dissertation is the main contributor of [Cha+23; CSB24]. The co-authorship for [CMS25] is equally shared.

Let's start at the very beginning,
a very good place to start.

THE SOUND OF MUSIC

Chapter 2

Background and Preliminaries

Abstract

This chapter presents a preliminary introduction to the concepts that we require throughout the thesis. We start by introducing basic mathematical notation for sets, vector spaces, random variables, and distributions. We then introduce a few important concepts from group theory and some common concentration inequalities for both independent and dependent random variables. Subsequently, we provide a brief overview of Boolean functions, computational learning theory, Fourier analytic techniques for learning theory, and quantum computing. The overview presented in this chapter aims to be comprehensive with respect to the contents of this thesis. For a more detailed exposition of these topics, we refer the reader to

1. [KV94b; MRT12; SSBD14] for computational learning theory,
2. [SF12; FF12] for an overview of ensemble learning,
3. [NC10; Wol23] for quantum computation, and
4. [Man94; O'D14] for Fourier analytic techniques in learning theory.

Contents

2.1 Basic Notation 16
2.2 Useful Mathematical Results 17
2.3 Boolean functions 19
2.3.1 Fourier Analysis of Boolean functions 21
2.4 Basics of Computational Learning Theory 22
2.4.1 PAC-learning 26
2.4.2 Learning with queries 28
2.4.3 Taxonomy of PAC learning 28
2.4.3.1 Proper and Improper PAC learning 28
2.4.3.2 PAC learning under various Label Noise Models 29
2.4.3.3 Weak and Strong Learning 30
2.5 Basics of Quantum Computing 30
2.5.1 The Quantum RAM model 36
2.5.2 Important Quantum Subroutines 37

§ 2.1 Basic Notation

▷ Sets, Vector Spaces, and Strings.

We denote the set of Natural numbers, Real numbers, Complex numbers, and Integers by the symbols $\mathbb{N}$, $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{Z}$, respectively. Sets are denoted by calligraphic fonts such as $\mathcal{A}, \mathcal{S}, \mathcal{X}$. For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. The finite field of order 2 is denoted by $\mathbb{F}_2 = \{-1, +1\}$ or $\mathbb{Z}_2 = \{0, 1\}$, and $\mathbb{F}_2^n$ (resp. $\mathbb{Z}_2^n$) denotes an n-dimensional vector space over $\mathbb{F}_2$ (resp. $\mathbb{Z}_2$).

A string $x \in \mathbb{F}_2^n$ is an n-dimensional vector in $\mathbb{F}_2^n$, where x_i denotes the i-th bit of x. A Boolean literal is a bit x_i or its negation $\overline{x}_i$. For a string $x \in \mathbb{F}_2^n$, the string $\overline{x} \in \mathbb{F}_2^n$ is the string given by $\overline{x}_i = 1 \oplus x_i$, where $\oplus$ is the bit-wise sum operation. For $a, b \in \mathbb{F}_2^n$ (or $\mathbb{Z}_2^n$), any bit-wise binary operation $\odot$ results in a string $(a \odot b) \in \mathbb{F}_2^n$ (or $\mathbb{Z}_2^n$ respectively). The bit-wise AND, OR, and XOR operations are denoted as $a \wedge b$, $a \vee b$, and $a \oplus b$, respectively. The Hamming weight of $x \in \mathbb{F}_2^n$ is denoted by $|x|$.

▷ Random Variables and Distributions.

We denote by $\mathop{\mathbb{E}}_{\mathfrak{D}}[\mathbf{X}]$ the expectation of $\mathbf{X}$ over distribution $\mathfrak{D}$, and by $\mathbb{E}[\mathbf{X}]$ the expectation of $\mathbf{X}$ over the uniform distribution. We will sometimes interchangeably denote the expectation of a random variable $\mathbf{X}$ w.r.t. a distribution $\mathfrak{D}$, i.e., $\mathop{\mathbb{E}}_{\mathfrak{D}}[\mathbb{X}]$ as $\langle \mathfrak{D}, \mathbb{X} \rangle$.

The uniform distribution over a set $\mathcal{S}$ is denoted by $\mathcal{U}(\mathcal{S})$. The support of a distribution $\mathfrak{D}$ over the Boolean hypercube $\mathbb{F}_2^n$ is denoted as $\text{supp}(\mathfrak{D}) = \{z \in \mathbb{F}_2^n \mid \mathfrak{D}(z) \neq 0\}$. $x \sim \mathfrak{D}$ denotes a string sampled from a probability distribution $\mathfrak{D}$, i.e. $\Pr(\mathbf{X} = x) = \mathfrak{D}(x)$. Similarly, $x \sim \mathcal{S} \iff x \sim \mathcal{U}(\mathcal{S})$, where $\mathcal{S} \subseteq [n]$. We use $\mathcal{G} \sim \mathfrak{D}^k$ to denote that $\mathcal{G}$ is a set of k elements that are sampled i.i.d. from $\mathfrak{D}$. Unless explicitly stated otherwise (see Chapter 8), we assume that sampling is i.i.d. We denote by $\Delta_{\mathcal{H}}$ the set of all distributions over any arbitrary set $\mathcal{H}$.

▷ Order Notation.

Let $c > 0$ be a constant, and n_0 be an integer. For $f, g : \mathbb{N} \mapsto \mathbb{R}$, $f(n) = O(g(n))$ denotes that there exists some $c, n_0 > 0$, s.t. $f(n) \leqslant c \cdot g(n)$ for all $n > n_0$. $f(n) = \Omega(g(n))$ denotes that there exists some $c, n_0 > 0$, s.t. $f(n) \geqslant c \cdot g(n)$ for all $n > n_0$. Finally, we use $f(n) = \Theta(g(n))$ to denote that $f(n) = O(g(n))$ and $g(n) = O(f(n))$. For any $c > 0$, $f(n) = o(g(n)) \implies f(n) \leqslant c \cdot g(n)$ and $f(n) = \omega(g(n)) \implies f(n) \geqslant c \cdot g(n)$, for $n > n_0$. We use the notation $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$, $\tilde{\Theta}(\cdot)$ to denote that poly-logarithmic terms are hidden in the order notation. For example, $f(n) = \tilde{O}(g(n)) \implies \exists c > 0$, s.t., $f(n) \leqslant c \cdot g(n) \operatorname{polylog}(g(n))$.

▷ Miscellaneous.

We represent the logarithm to base 2 by log and logarithm to the base e by ln. The indicator random variable for an event E is denoted by $\mathbb{1}_{[E]}$. For $n \in \mathbb{N}$, any positive polynomial function in n is denoted by $\text{poly}(n)$. A multivariate polynomial over n-variables is denoted as $\text{poly}(x_1, x_2, \ldots, x_n)$.

§ 2.2 Useful Mathematical Results

▷ Concentration Inequalities.

We use various forms of concentration inequalities throughout this thesis to bound the deviations of functions of random variables from their expectation. In the case of i.i.d. random variables, we use Chernoff-Hoeffding bounds as stated below.

Lemma 2.1 (Chernoff-Hoeffding bounds)**.** Let $\gamma \in (0, \frac{1}{2})$ and $\mathbf{X}_1, \ldots, \mathbf{X}_m$ be independent r.v.'s with mean $\mathbb{E}[\mathbf{X}_i] = \mu_i$, s.t., $\forall i \in [m]$, $a_i \leqslant X_i \leqslant b_i$. Then,

$$\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \frac{1}{m}\sum_{i=1}^{m}\mu_i\right| > \gamma\right) \leqslant 2\exp\left\{\frac{-2\gamma^2}{\sum_{i=1}^{m}(a_i - b_i)^2}\right\}.$$

We can also obtain multiplicative bounds. For $\gamma \in (0, 1)$, i.i.d. r.v.'s $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$,

with $\mathbf{X} = \sum_i \mathbf{X}_i$ and $\mu = \mathbb{E}[\mathbf{X}]$, we have $\Pr(|\mathbf{X} - \mu| \geqslant \gamma\mu) \leqslant 2 \cdot \exp\{-\mu\gamma^2/3\}$.

In Chapter 8, we also have to deal with concentration inequalities on functions of *dependent* random variables using the Azuma-Hoeffding inequality defined below. Consider any probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and let $(\mathcal{F}_t)_{t\in\mathbb{N}}$ be a filtration. Then, a random process $(M_t)_{t\in\mathbb{N}}$ adapted to the filtration $\mathcal{F}_t$ is said to be a *martingle difference sequence* if $\mathbb{E}[|M_t|] < \infty$ and $\mathbb{E}[M_t \mid \mathcal{F}_{t-1}] = 0$ almost surely. We state below the Azuma-Hoeffding inequality , which bounds the probability of the sum of the first T terms of a martingale difference sequence exceeding some constant.

Lemma 2.2 (Azuma-Hoeffding inequality)**.** Let $(M_t)_{t\in\mathbb{N}}$ be a martingale difference sequence with respect to the filtration $(\mathcal{F}_t)_{t\in\mathbb{N}}$. Let there be constants $c_t \in (0, \infty)$ such that $\forall t \geqslant 1$, almost surely $|M_t| \leqslant c_t$. Then, $\forall \gamma > 0$,

$$\Pr\left(\sum_{t=1}^{T} M_t \geqslant \gamma\right) \leqslant \exp\left(-\frac{\gamma^2}{2\sum_{t=1}^{T} c_t^2}\right).$$

▷ Information-theoretic Inequalities.

Let P and Q be two distributions on the same probability space $(\Omega, \mathcal{F})$, having densities p and q, respectively, with respect to an underlying measure μ. Then the Total Variation distance $d_{TV}(\cdot, \cdot)$ is defined w.r.t. $\mathfrak{P}$ and $\mathfrak{Q}$ as

$$d_{TV}(P, Q) := \sup_{A\in\mathcal{F}} |P(A) - Q(A)| = \frac{1}{2}\int_{\omega\in\Omega} |p(\omega) - q(\omega)|\, d\mu(\omega).$$

The Kullback–Leibler (KL) divergence between P and Q, denoted by $D(P\|Q)$ is defined as

$$D(P\|Q) = \mathop{\mathbb{E}}_{X\sim P}\left[\ln\left(\frac{dP}{dQ}(\mathbf{X})\right)\right].$$

The relation between total variation distance and KL divergence is captured by Pinsker's inequality.

Lemma 2.3 (Pinsker's Inequality)**.** Every P, Q on $(\Omega, \mathcal{F})$ satisfies

$$d_{TV}(P, Q) \leqslant \sqrt{\frac{1}{2} D(P\|Q)}.$$

The KL divergence can also be expressed as a variational form.

Lemma 2.4 (Donsker-Varadhan variational form)**.** Let X be a real-valued integrable random variable. Then for every $\lambda \in \mathbb{R}$,

$$\log \mathop{\mathbb{E}}_{P}\left[e^{\lambda(X-\mathop{\mathbb{E}}_{P}[X])}\right] = \sup_{Q \ll P}\left[\lambda \langle Q - P, X\rangle - D(Q\|P)\right].$$

See Theorem 4.19 and Corollary 4.14 of [BLM13] for a proof of Lemma 2.3 and Lemma 2.4, respectively.

§ 2.3 Boolean functions

A function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is known as a Boolean function. At this point, we recall that for the purposes of this thesis, the finite field of order 2 is denoted by $\mathbb{F}_2 = \{-1, +1\}$ or $\mathbb{Z}_2 = \{0, 1\}$, and $\mathbb{F}_2^n$ (resp. $\mathbb{Z}_2^n$) denotes an n-dimensional vector space over $\mathbb{F}_2$ (resp. $\mathbb{Z}_2$).

For the purposes of this thesis, we are concerned with decision problems, i.e., binary-valued functions, and therefore mostly consider Boolean functions of the form $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$, unless explicitly stated otherwise.

Example 2.1. A simple example of the type of Boolean functions we are concerned with is a binary-valued function on an n-bit binary-valued string, where the convention is to denote the output 0 as false and the output 1 as true.

▷ Representation of Boolean functions.

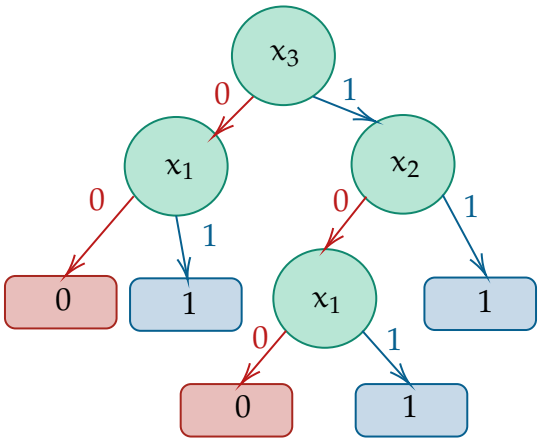
There are many semantically equivalent representations of Boolean functions. In this thesis, we are concerned with three main types of representations of Boolean functions, as listed below. We refer the reader to Fig. 2.1 for a quick example of different representations of the same Boolean function.

- **Truth Table**: In this representation, we explicitly list the value of the Boolean function for all of its 2^n possible inputs.
- **Boolean Circuits**: A Boolean circuit is a finite Directed Acyclic Graph (DAG) with clearly designated input nodes, internal nodes, and an output node. For the purposes of this thesis, every node has a fan-in of at most 2. The input nodes are the bits of the input string, and have fan-in 0; the internal nodes consist of rudimentary Boolean functions belonging to some basis set of Boolean functions (eg. ANDs, ORs, and NOTs), and the output node (with fan-out 0) stores a binary value corresponding to the output of the evaluation of the input bits.

- **Boolean Formulas / Propositional Formulas**: We can express any arbitrary Boolean function f as an algebraic formula consisting of rudimentary Boolean functions belonging to some basis set of Boolean functions, applied to its input bits. Formally, any Boolean circuit where every node has fan-out at most 1 is a Boolean formula. Some well-known canonical representations of Boolean functions are the DNF and Conjunctive Normal Form (CNF) representations, defined w.r.t. ORs and ANDs over the input literals. Unlike Truth tables or Circuits, Boolean formulas correspond to a covering of $f^{-1}(0)$ or $f^{-1}(1)$.

x_3	x_2	x_1	$f(x)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(a) Truth Table of f.



Decision Tree Representation of f (size 5 and depth 3).

$$\overline{(\overline{x_3} \wedge \overline{x_1})} \vee (\overline{x_3} \wedge x_1) \vee \overline{(x_3 \wedge \overline{x_2} \wedge \overline{x_1})} \vee (x_3 \wedge \overline{x_2} \wedge x_1) \vee (x_1 \wedge x_2)$$

DNF Representation of f (size 5 and width 3).

(b) Decision tree and DNF representations of f.

Figure 2.1: Different representations of the same Boolean function f.

Remark 2.3.1. Given a universal basis set $\mathcal{B}$ for Boolean functions, we can define Boolean circuits such that the internal nodes only contain gates from $\mathcal{B}$. For example, consider the basis sets $\mathcal{B}_1 = \{\text{NAND}\}$ or $\mathcal{B}_1 = \{\text{NOR}\}$. Similar to Boolean circuits, we can also write a formula in terms of different universal basis sets over Boolean functions. However, for the purposes of this thesis, the canonical basis set for classical Boolean circuits and formulas is $\{\text{AND}, \text{OR}, \text{NOT}\}$.

▷ Properties of Boolean functions.

A Boolean function is said to be **constant** if its output is a fixed value and does not depend on its inputs. A Boolean function is **monotone** if changing any input bit from 0 to 1 can only force the output to change from 0 to 1 and not from 1 to 0. Formally, for an n-bit monotone Boolean function, if $a_i \leqslant b_i$, for $a_i, b_i \in \mathbb{F}_2, i \in [n]$, then $f(a) \leqslant f(b)$.

A Boolean function is **symmetric** if its output does not depend on the order of its input bits. A Boolean function is **linear** or **affine** if $f(x) = y_0 \oplus (x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \oplus \ldots \oplus (x_n \wedge y_n)$, $x \in \mathbb{F}_2^n, y \in \mathbb{F}^{n+1}$.

Example 2.2 (Some Boolean functions)**.**

- The truth table with all 0's or all 1's is a constant function.
- $f(x) = \wedge_{i\in[n]} x_i$ or $f(x) = \vee_{i\in[n]} x_i$ is a monotonic Boolean function.
- The **Parity function** $f(x) = x_1 \oplus x_2 \oplus \ldots x_n$ is both linear and symmetric.

Our primary object of study in this thesis are decision trees, which are formally defined in Section 4.4. See Fig. 2.1 for an intuitive understanding of decision trees.

§ 2.3.1 Fourier Analysis of Boolean functions

Fourier-analytic techniques constitute a very powerful family of tools for designing learning algorithms in general and particularly for learning over the Boolean hypercube [LMN93]. In this section, we will limit our discussions to Fourier analysis for ± 1-valued or $\{0,1\}$-valued functions, but most results will also apply to real-valued functions over the Boolean hypercube. Every Boolean function $f(x)$, can be *uniquely* represented as a multilinear polynomial known as the **Fourier representation** of f, as follows.

$$f(x) = \sum_{S\subseteq[n]} \hat{f}(S)\,\chi_S(x), \quad f : \mathbb{F}_2^n \mapsto \mathbb{F}_2. \tag{2.1}$$

In Eq. (2.1), $\chi_S(x) = \prod_{i\in S} x_i$ (we define explicitly $\chi_S(\varnothing) = -1$) is known as a **parity monomial**, and $\hat{f}(S) = \mathbb{E}\,[f \cdot \chi_S]$ is known as the **Fourier coefficient** of f w.r.t. to the subset of literals S.

Remark 2.3.2. A Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$ can be transformed into an equivalent function $\tilde{f} : \{\pm 1\}^n \mapsto \{\pm 1\}$ using the following conversion:

$$\tilde{f}(x_1, x_2, \ldots, x_n) = 1 - 2 \cdot f\left(\frac{1-x_1}{2}, \frac{1-x_2}{2}, \ldots, \frac{1-x_n}{2}\right), \quad x_i \in \{\pm 1\}\ \forall i \in [n].$$

Remark 2.3.3. If $f : \{0,1\}^n \mapsto \mathbb{F}_2$, then $\chi_S(x) = (-1)^{\sum_{i\in S} x_i}$. Sometimes, it's useful to consider this particular semantic, for example, while defining the quantity $\chi_\varnothing = -1$.

The set of all parity monomials $\{\chi_S \mid S \subseteq [n]\}$ forms a basis for all real-valued functions on the Boolean hypercube $\mathbb{F}_2^n$. The vector space spanned by $\mathbb{P}$ forms an inner product space as follows.

Definition 2.1 (Inner Product of Boolean functions)**.** For any $f, g : \mathbb{F}_2^n \mapsto \mathbb{F}_2$,

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x) \cdot g(x) = \mathop{\mathbb{E}}_{\mathcal{U}(\mathbb{F}_2^n)} [\, f(x) \cdot g(x) \,] = \mathrm{corr}_{\mathcal{U}}(f, g) = 1 - 2 \mathop{\Pr}_{\mathcal{U}(\mathbb{F}_2^n)} [f \neq g]\,. \tag{2.2}$$

The inner product of two Boolean functions captures the notion of distance/ **correlation** between them.[1] The above notion of inner product also shows us that the set of all parity monomials forms an orthonormal basis set, since $\langle \chi_S, \chi_S \rangle = 1$ for all $S \subseteq [n]$, and $\langle \chi_S, \chi_T \rangle = 0$ if $S \neq T$. This allows us to state the following two important lemmas.

Lemma 2.5 (Plancherel's Theorem)**.** For any $f, g : \mathbb{F}_2^n \mapsto \mathbb{F}_2$, $\langle f, g \rangle = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \hat{g}(S)$.

Lemma 2.6 (Parseval's Theorem)**.** For any $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$, $\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1$.

Observation 2.3.1 (Fourier spectrum of a Boolean function)**.** For any Boolean function f, we see that the set of squared Fourier coefficients $\left\{ \hat{f}(S)^2 \right\}_{S \subseteq [n]}$ forms a probability distribution (see Lemma 2.6). This is known as the **Fourier spectrum** or **Fourier distribution** of f, and denoted by $\mathrm{FS}(f)$.

§ 2.4 Basics of Computational Learning Theory

▷ Concept Class, Hypothesis Class, and Representation Size.

Let $\mathcal{X}_n \subseteq \mathbb{F}_2^n$ be a set of n-bit instances (or strings), and $\mathcal{Y}$ be the set of labels. For $n \in \mathbb{N}$, we denote by $\mathcal{C} = \cup_{n>0} \mathcal{C}_n$, where $\mathcal{C}_n = \{c_n \mid c_n : \mathcal{X}_n \mapsto \mathcal{Y}\}$, a class of functions that maps instances to labels, known as the **concept class**. Similarly, a **hypothesis class** $\mathcal{H} = \cup_{n>0} \mathcal{H}_n$, where $\mathcal{H}_n = \{h_n \mid h_n : \mathcal{X}_n \mapsto \mathcal{Y}\}$, is also a class of functions that maps instances to labels. When it is obvious from the context, we drop the subscript n from the definitions. For most of this thesis, we assume that the label set is binary. Therefore, the concepts and the hypotheses discussed above are Boolean functions.

Since multiple classes of Boolean circuits can compute exactly the same Boolean function (as we have discussed earlier), the choice of representation of the concept or hypothesis becomes important. We assume that the representation of any concept class or hypothesis class is *fixed*. The representation size of a Boolean function can be

[1] See Definition 2.4 for a semantic understanding of correlation.

defined depending on the underlying choice of representation. Some standard representation choices include the depth of the minimal Boolean circuit representation or the minimal decision tree representation depth. For the purposes of this thesis, the choice of the underlying representation is the concept class $\mathcal{C}$ / hypothesis class $\mathcal{H}$ under consideration, and its corresponding size denoted by $\text{size}(\mathcal{C})$ and $\text{size}(\mathcal{H})$ respectively, is the maximal cost of the representation of $\mathcal{C}$ and $\mathcal{H}$ respectively.

▷ Learning Algorithm.

A learning algorithm (or learner) A is a randomized map from a set of instances[2] (sampled according to an unknown distribution $\mathfrak{D}$ and labeled by an unknown *target* concept $c \in \mathcal{C}$) to a hypothesis function $h \in \mathcal{H}$.[3] Since the label set is binary, most of the algorithms discussed in this thesis perform *binary classification*. We note here that the learner A can be either a classical randomized algorithm or a quantum algorithm.[4] Below, we motivate the above setup with an example.

Example 2.3. Suppose we wish to impress a person $\mathbb{X}$ by taking them out for pizza. However, given the variety of pizza joints in the city catering to different niches, we would like to be sure that our choice of pizzeria serves the kind of pizza $\mathbb{X}$ enjoys. Hence, we would like to "learn" the types of pizza $\mathbb{X}$ finds tasty. Let us assume that all pizzas have three main attributes - **topping**, **size**, and **crust** thickness. After observing the food habits of $\mathbb{X}$ for a few days, suppose we (or a learner A) formulate the following hypothesis h : **topping** $\times$ **size** $\times$ **crust** $\mapsto$ {Tasty, Not-tasty}: $\mathbb{X}$ only prefers **non-veg** pizzas $>$ 16 **cm**, or **veg** pizzas $>$ 16 **cm** with **thin crusts**. Since our formulated hypothesis h is a Boolean function, we can have different equivalent representations, such as DNF or CNF forms, as shown below:

$$h^{(\text{DNF})} := (\underline{\textbf{Non-Veg}} \wedge \underline{> \textbf{16}}) \vee (\underline{\textbf{Veg}} \wedge \underline{\textbf{Thin}} \wedge \underline{> \textbf{16}}).$$

$$h^{(\text{CNF})} := (\underline{> \textbf{16}}) \wedge (\underline{\textbf{Non-Veg}} \vee \underline{\textbf{Thin}}).$$

Computational learning theory tries to answer the following three questions:

Question 2.1. How **accurate** is our hypothesis h?

For example, can we accurately predict if person $\mathbb{X}$ finds a new instance $x \in$ **topping** $\times$ **size** $\times$ **crust** tasty or not in Example 2.3?

[2] Such a set S is referred to as the training set for A.
[3] See Chapter 8 for a more general definition of statistical learning algorithms.
[4] We will formally define the notion of quantum algorithms in Section 2.5.

Question 2.2. Which representation of h is more **useful** in practice?

Note that in Example 2.3, $h^{(CNF)}$ is computationally easier to evaluate than $h^{(DNF)}$. We do not know if this is necessarily true when considering a hypothesis for another person $\mathcal{Y}$, who might present us with a different set of rules.

Question 2.3. How *efficiently* can we generalize the learning algorithm?

How **fast** can we come up with some hypothesis h' for a different distribution $\mathfrak{D}$ and concept c from the same concept class $\mathcal{C}$? In terms of Example 2.3, can we formulate a similarly accurate hypothesis for another person $\mathcal{Y}$?

Question 2.1 is the most straightforward question to answer out of the above three. Throughout most of this work, we focus on the following metrics to quantify the performance of the output hypothesis- error and correlation. There are two notions of error, and both capture the number of mistakes made by the hypothesis h w.r.t. the concept c in different forms.

Definition 2.2 (Training Error)**.** The training error of a hypothesis $h \in \mathcal{H} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ w.r.t. an unknown concept $c \in \mathcal{C} : \mathbb{F}_2^n \mapsto \{-1, +1\}$ that labels a training set $\mathcal{S}$ with m labeled instances is defined as:

$$\widehat{\text{err}}_{\mathcal{S}}(h) := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[(h(x_i) \neq c(x_i))]}. \tag{2.3}$$

Definition 2.3 (Test Error)**.** The error (or test error) of a hypothesis $h \in \mathcal{H} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ w.r.t. an unknown concept $c \in \mathcal{C} : \mathbb{F}_2^n \mapsto \{-1, +1\}$ and an unknown underlying distribution $\mathfrak{D}$ is defined as:

$$\text{err}_{\mathfrak{D},c}(h) := \mathop{\mathbb{E}}_{x \sim \mathfrak{D}} \left[\mathbb{1}_{[(h(x) \neq c(x))]} \right] = \Pr_{x \sim \mathfrak{D}} [h(x) \neq c(x)]. \tag{2.4}$$

The optimal error of a hypothesis class $\mathcal{H}$ w.r.t. an unknown concept $c \in \mathcal{C}$ w.r.t. to an underlying distribution $\mathfrak{D}$ is defined as $\text{opterr}_{\mathfrak{D},c}(\mathcal{H}) := \min_{h \in \mathcal{H}} \text{err}_{\mathfrak{D},c}(h)$.

Observation 2.4.1. From Definitions 2.2 and 2.3, we have $\mathop{\mathbb{E}}_{\mathcal{S} \sim \mathfrak{D}^m} [\widehat{\text{err}}_{\mathcal{S}}(h)] = \text{err}_{\mathfrak{D},c}(h)$.

The third metric to quantify the performance of a hypothesis is correlation. Informally, the correlation between a hypothesis h and a concept c measures how accurate h is with respect to c.

Definition 2.4 (Correlation)**.** The correlation of a hypothesis $h \in \mathcal{H} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ w.r.t. an unknown concept $c \in \mathcal{C} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ and an unknown underlying distribution $\mathfrak{D}$ is defined as:

$$\operatorname{corr}_{\mathfrak{D},c}(h) := \mathop{\mathbb{E}}_{\mathfrak{D}}[h(x) \cdot c(x)] = 1 - 2\operatorname{err}_{\mathfrak{D},c}(h). \tag{2.5}$$

The optimal correlation of a hypothesis class $\mathcal{H}$ w.r.t. an unknown concept $c \in \mathcal{C}$ and distribution $\mathfrak{D}$ is defined as $\operatorname{optcorr}_{\mathfrak{D},c}(\mathcal{H}) := \max_{h \in \mathcal{H}} \operatorname{corr}_{\mathfrak{D},c}(h)$. Sometimes, one or more of the subscripts are dropped from the notation of $\operatorname{err}(h)$, $\operatorname{opterr}(h)$, $\operatorname{corr}(h)$, and $\operatorname{optcorr}(h)$ when the underlying distribution and concept are clear from the context.

Remark 2.4.1. Sometimes a hypothesis with $\operatorname{err}(h) = 0$ is called the **ideal hypothesis**.

▷ The Hypothesis Class associated with a Learner.

An implicit point central to the study of computational aspects of learning theory is that every learning algorithm A has its own associated hypothesis class $\mathcal{H}_A$. The learner A can only output a hypothesis from $\mathcal{H}_A$.[5] We also assume here that all hypothesis classes $\mathcal{H}_A$ are **polynomially evaluatable** by A, as defined below.

Definition 2.5 (Polynomial Evaluatibility)**.** Given any $x \in \mathcal{X}$ and $h \in \mathcal{H}_A$, if A can compute $h(x)$ in time $\operatorname{poly}(n, \operatorname{size}(\mathcal{H}_A))$, then $\mathcal{H}_A$ is said to be polynomially evaluatable by A.

We note that certain learning algorithms are *more expressive* than others, courtesy of their respective associated hypothesis classes (recall the notion of different representations introduced in Example 2.3). However, a learning algorithm with the most expressive hypothesis class may not always be the best choice since learners with larger hypothesis classes would need more computational resources to find the optimal hypothesis in its associated hypothesis class. This is the crux of Question 2.2, and is discussed throughout this thesis, starting from Section 2.4.1.

Similarly, we consider Question 2.3. Given an arbitrary choice of distribution $\mathfrak{D}$ over the data and some unknown labeling concept $c \in \mathcal{C}$, assuming that a learning algorithm A can output an ideal hypothesis is not always realistic. For example, consider scenarios when $\mathcal{H}_A$ is extremely large or the concept c does not belong to $\mathcal{H}_A$. Later, we discuss settings where obtaining an ideal hypothesis is both computationally and mathematically impossible. In such cases, the main objective of learning theory is to *efficiently* output a hypothesis h such that $h \approx c$ w.r.t. some accuracy metric such as error or correlation. Such a notion is introduced in the next section.

[5] We drop the subscript from the hypothesis class when the choice of the learner is clear.

§ 2.4.1 PAC-learning

In this section, we attempt to formalize our answers to Questions 2.2 and 2.3 by considering various learning models. We start by defining the Probably Approximately Correct (PAC) model of statistical learning, which was introduced by Valiant [Val84].

Definition 2.6 ((ε, δ)-PAC learnability)**.** A learning algorithm A (ε, δ)-PAC learns an unknown concept class $\mathcal{C}$ over n-bit instances using hypothesis class $\mathcal{H}_A$, if for every distribution $\mathfrak{D}$ over $\mathcal{X}$, and every target concept $c \in \mathcal{C}$, there exists a function $m_0 : (0,1)^2 \mapsto \mathbb{N}$, such that for every $\varepsilon, \delta \in [0,1]$, A takes as input a training set $\mathcal{S}$ consisting of $m > m_0(\varepsilon, \delta)$ sampled i.i.d. according to $\mathfrak{D}$ and labeled by c and outputs $h \in \mathcal{H}_A$ s.t.

$$\Pr\left[\mathrm{err}_{\mathfrak{D},c}(h) \leqslant \varepsilon\right] \geqslant 1 - \delta. \tag{2.6}$$

Equivalently,

$$\Pr\left[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant 1 - 2\varepsilon\right] \geqslant 1 - \delta. \tag{2.7}$$

The probability in Eqs. (2.6) and (2.7) is over the choice of the training set $\mathcal{S}$ and any internal randomization of the learner A. We also note that in Definition 2.6, there is an implicit *assumption of realizability*, i.e., with probability 1 over $\mathcal{S}$, there exists $h_{\mathrm{opt}} \in \mathcal{H}$ s.t. $\mathrm{err}_{\mathfrak{D},c}(h_{\mathrm{opt}}) = 0$.

▷ Sample Complexity and Learnability.

In Definition 2.6, m is the **sample complexity** of the learner A for learning $\mathcal{C}$. If $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$, then the concert class $\mathcal{C}$ is said to be PAC-learnable by A using the hypothesis class $\mathcal{H}_A$. A concept class $\mathcal{C}$ is *learnable* or PAC-learnable using $\mathcal{H}_A$ if there exists some ε, δ, m_0 for which Eqs. (2.6) and (2.7) hold.

▷ Time Complexity and Efficient Learnability.

If A (ε, δ)-PAC learns $\mathcal{C}$ in polynomial time, i.e., in time $\mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$, then we say that the concept class $\mathcal{C}$ is **efficiently PAC learnable** (or simply, **efficiently learnable**) by A using the hypothesis class $\mathcal{H}_A$. For efficient learners, $\mathrm{size}(\mathcal{H}_A) = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{C}))$. We note here that not all *learnable* concept classes w.r.t. a learner A are *efficiently learnable* w.r.t. A.

Remark 2.4.2. We are following the definition of [SSBD14], where we are decoupling the notion of sample complexity from the definition of PAC learning, in order to then drive home the distinction between learnability and efficient learnability. In other standard textbooks and sources, it is common to find $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ directly in Definition 2.6 itself.

Remark 2.4.3. All Boolean functions are efficiently PAC learnable if the representation scheme of the underlying concept class is extremely expressive/ verbose - for example, the concept class is described using truth tables. The interesting case is when the underlying concept is represented using Boolean functions which are believed to have *limited expressivity* such as Decision trees, since extremely simple Boolean circuits can compute them.

▷ VC dimension and Bounds on Test Error.

The notion of learnability of concept classes $\mathcal{C}$ becomes non-trivial when considering learning algorithms whose associated hypothesis spaces are infinite, since it becomes hard to argue about polynomial evaluability for infinite hypothesis spaces. We now introduce and define a combinatorial notion - the VC dimension [VC71] of a hypothesis space $\mathcal{H}$, that can be used to obtain test error guarantees for learning algorithms even when $\mathcal{H}$ is infinite. We first define the notions of *growth functions* and *shattering*, which we use to define a VC dimension formally.

The **growth function** $\Pi_{\mathcal{H}} : \mathbb{N} \mapsto \mathbb{N}$ of a hypothesis space $\mathcal{H}$ is defined as the maximum number of distinct ways to classify m instances using hypotheses in $\mathcal{H}$.

$$\Pi_{\mathcal{H}}(m) := \max_{\{x_1,\ldots,x_m\}\subseteq\mathcal{X}} |\{h(x_1),\ldots,h(x_m) \mid h \in \mathcal{H}\}|, \quad \forall m \in \mathbb{N}.$$

The total number of ways we can classify a set $\mathcal{S}$ using a label set $\mathcal{Y}$ is $|\mathcal{Y}|^{|\mathcal{S}|}$. For binary classification, it is simply $2^{|\mathcal{S}|}$. A set of m points is said to be **shattered** by a hypothesis set $\mathcal{H}$ if $\Pi_{\mathcal{H}}(m) = 2^m$, i.e., all possible labelings of the set are realizable by $\mathcal{H}$.

Definition 2.7 (VC dimension of a Hypothesis Space [VC71])**.** The VC dimension of a hypothesis space $\mathcal{H}$, denoted by $d_{\mathcal{H}}$, is the largest set of points that can be shattered by $\mathcal{H}$, i.e. $d_{\mathcal{H}} := \max\{m \mid \Pi_{\mathcal{H}}(m) = 2^m\}$.

If a hypothesis space $\mathcal{H}$ has VC dimension $d_{\mathcal{H}}$, there exists a set of size $d_{\mathcal{H}}$ that can be shattered by $\mathcal{H}$. We now state an upper bound on the error of a hypothesis in terms of the VC dimension of the corresponding hypothesis space.

Lemma 2.7 (Error Bounds in terms of VC dimension)**.** Let $\delta > 0$ be a constant, and $\mathcal{H}$ be a hypothesis class with VC dimension $d_{\mathcal{H}}$. Then for any distribution $\mathfrak{D}$ over the instances, any concept $c \in \mathcal{C}$, and any $h \in \mathcal{H}$, we have

$$\Pr\left[\mathrm{err}_{\mathfrak{D},c}(h) \leqslant \widehat{\mathrm{err}}_{\mathcal{S}\sim\mathfrak{D}^m}(h) + O\left(\sqrt{\frac{\log(m/d_{\mathcal{H}})}{m/d_{\mathcal{H}}}}\right)\right] \geqslant 1-\delta.$$

There are other versions of Lemma 2.7, based on different ways of characterizing the complexity of the Hypothesis space. We will provide a novel way of characterizing the upper bounds on the error of a hypothesis later in Chapters 7 and 8.

Fact 2.1. $d_{\mathcal{H}}$ is always upper-bounded by size $(\mathcal{H})$.

§ 2.4.2 Learning with queries

The learning algorithm A in Definition 2.6 (ε, δ)-PAC learns $\mathcal{C}$ with respect to random examples (provided in the form of the training set S). Alternatively, we can also assume that the learner A can make black-box queries, i.e., the learner has access to an oracle $\mathcal{O}$. In this generalized model, we redefine the notion of PAC learning as follows:

> **Definition 2.8** (PAC learning with Oracle access)**.** A learner A (ε, δ)-PAC learns concept class $\mathcal{C}$ over n-bit instances, with query access to oracle $\mathcal{O}$, using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0,1)$, for every distribution $\mathfrak{D}$ over $\mathcal{X}$, and every target concept $c \in \mathcal{C}$, A takes as input a training set S consisting of $m = \text{poly}(1/\varepsilon, 1/\delta, n, \text{size}(\mathcal{H}_A))$ sampled i.i.d. according to $\mathfrak{D}$ and labeled by c, and makes at most m queries to $\mathcal{O}$ and outputs h s.t. the test error of h is at most ε with probability at least $1 - \delta$.

Throughout this thesis, we shall be exploring the setting of learning with queries in detail. In Chapter 3, we shall define different oracles[6] and instantiate Definition 2.8 for each of them. Later in Chapters 4 and 9, we shall discuss why exploring learning under different oracles is an important topic in statistical learning theory.

§ 2.4.3 Taxonomy of PAC learning

There are many possible ways to classify PAC learning algorithms. In this thesis, we consider three broad classification umbrellas - the representation of the output hypothesis, the label noise, and the accuracy of the output hypothesis. See Fig. 2.2 for a brief overview of these categories.

§ 2.4.3.1 Proper and Improper PAC learning

Recall Remark 2.4.3 and the preceding discussions while defining the notions of learnability and efficient learnability. In Definition 2.6, the choice of representation of the output hypothesis $h \in \mathcal{H}_A$ produced by a learning algorithm A need not necessarily be the same as the unknown target concept class $\mathcal{C}$ it is trying to learn. If the choice of representation of $\mathcal{H}_A$ is the same as the choice of representation as $\mathcal{C}$, i.e., $h \in \mathcal{C}$, then

[6] Both classical and quantum oracles will be explored.

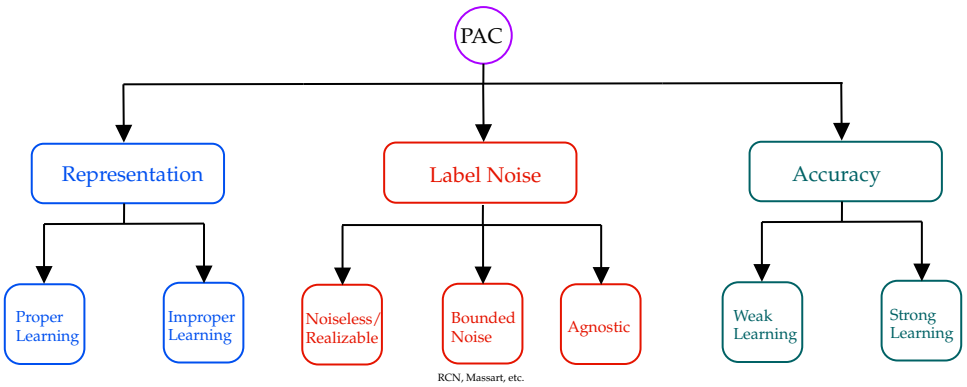


Figure 2.2: Broad classifications of PAC learning algorithms.

the learner A is a **proper PAC learner** for $\mathcal{C}$. Otherwise, when $h \notin \mathcal{C}$, A is considered an **improper PAC learner** for $\mathcal{C}$.[7]

> **Example 2.4.** Suppose the true underlying concept in Example 2.3 is represented using a DNF formula. If A outputs $h^{(\mathrm{DNF})}$, A properly learns the pizza labeling concept for person $\mathbb{X}$. Otherwise, A is an improper learner.

§ 2.4.3.2 PAC learning under various Label Noise Models

▷ The Realizable / Noiseless setting.

The learning setup introduced in Section 2.4 is called the **realizable setting** since we assume that there always exists a target concept $c \in \mathcal{C}$ that **realizes** the labeling of the instances. We also denote this setting as the **noiseless setting** since the labels in the training set are available to the learner without change.

▷ Bounded Noise setting.

We can generalize the realizable setting to account for more realistic learning scenarios as follows. Let $S = \{(x_1, c(x_1)), \ldots, (x_m, c(x_m))\}$ be the training set used to train a learning algorithm A. Suppose an adversary *corrupts* the labels of the training set S before it is obtained by the learner A and used to formulate the hypothesis. It is straightforward to see that the sample complexity and time complexity guarantees of (ε, δ)-PAC learnability would have to depend on the noise rate.

In the Random Classification Noise (RCN) setting introduced by [AL88], we assume that the labels of the training set are flipped with probability $0 \leqslant \eta < 1/2$, i.e., given training set $S = (x_i, y_i)_{i \in [m]}$, set each $y_i = \overline{y_i}$ independently with probability η. In this setting, the sample and time complexity of a PAC learner depends on

[7] (Im)Proper learning is also known as representation-(in)dependent learning.

poly $(1/\varepsilon, 1/\delta, n, \text{size}(c), 1/1-2\eta)$. In Chapter 3, we will redefine the RCN setting in terms of noisy oracles. An example of a more challenging class of bounded noise is **Massart noise**, where each label y_i is flipped with probability $\eta_i \leqslant \eta < 1/2$.

▷ Unbounded Noise setting.

A far more challenging noise model is the agnostic PAC setting [Hau92; KSS92]. In this noise model, each label y_i is corrupted with probability $\eta_i < 1/2$. Here, we note that there is no bound η s.t. $\eta_i < \eta, \forall i \in [|S|]$. If the rate of corruption of labels is unbounded, it is unclear whether a learning algorithm A, which was an efficient learner for $\mathcal{C}$, even remains a PAC learner for $\mathcal{C}$ after the noise has been added to the training set. We shall defer our discussions on PAC learning in the agnostic setting to Chapters 3, 4 and 6. In the unbounded noise model, we no longer make the assumption that there is a fixed underlying concept $c \in \mathcal{C}$ that labels the instances $x \in \mathcal{X}$, since the training set itself can be inconsistent.[8][9] Rather, we assume that the labeled instances (x, y) are jointly sampled from a distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$. Therefore, the learning algorithm is *agnostic* to the choice of the target concept.

§ 2.4.3.3 Weak and Strong Learning

Directly designing a PAC learner for an unknown target concept w.r.t. to an unknown distribution is often challenging. In practice, it is easier to design a *weak* PAC learner - something that succeeds with ~ 51% accuracy, and then amplify the success probability of such learners to be arbitrarily close to 100%. In other words: Given an ensemble of learning algorithms that are PAC learners in a *weak* sense, we can combine the above ensemble of *weak* learners using **ensemble learning algorithms** to obtain *strong* PAC learners[10]. In Chapter 4, we formalize the notions of *weak* and *strong* PAC learners under the various noise models explored earlier. Later in Chapters 5 and 6, we show how to design ensemble learners in the realizable and agnostic settings, respectively.

§ 2.5 Basics of Quantum Computing

▷ Qubits and Superpositions.

The quantum bit or *qubit* is the fundamental building of quantum computation, analogous to its classical counterpart - the bit. Formally, a qubit is a vector in a complex

[8] If, for some $i \in [|S|]$, the noise rate η_i is very close to $1/2$, the training set can contain samples $\{x_i, +1\}$ and $\{x_i, -1\}$. Hence, the labeling of instances is no longer considered a function.

[9] The same inconsistency can arise in the bounded noise model, but can be resolved easily by majority vote.

[10] In the realizable setting, strong PAC learners satisfy (ε, δ)-PAC learnability.

Hilbert space of dimension 2, represented as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Unlike classical bits, however, qubits are allowed to be in **superposition**, i.e., linear combinations of $|0\rangle$ and $|1\rangle$ are valid quantum states:

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha_0 |0\rangle + \alpha_1 |1\rangle \ , \ |\alpha_0|^2 + |\alpha_1|^2 = 1 \, , \ \alpha_0, \alpha_1 \in \mathbb{C}.$$

The vectors $|0\rangle$ and $|1\rangle$ form an orthonormal basis for all 1-qubit quantum states known as the **computational basis** or **standard basis**. The vectors

$$|+\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, |-\rangle = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

are another set of basis vectors known as the **Hadamard basis**. Note that we can combine different Hilbert spaces using the tensor product to obtain quantum states corresponding to multiple qubits: if $\mathbb{H}_M$ is a Hilbert space of dimension M and $\mathbb{H}_N$ is a Hilbert space of dimension N, then $\mathbb{H} = \mathbb{H}_M \otimes \mathbb{H}_N$ is a Hilbert space of dimension $M \times N$. Therefore, an n-qubit quantum state is a vector in a complex-valued Hilbert space of dimension 2^n, $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$. An n-qubit **pure state** can be expressed as

$$|\psi\rangle = \sum_{x \in \mathbb{Z}_2^n} \alpha_x |x\rangle \, , \text{ s.t., } \sum_{x \in \mathbb{Z}_2^n} |\alpha_x|^2 = 1.$$

The complex numbers α_x are known as the **amplitudes** of the basis vectors $|x\rangle$. The computational basis states for a n-qubit system are of the form $|x\rangle = |x_1 x_2 \ldots x_n\rangle$, $x \in \mathbb{Z}_2^n$, where 2^n amplitude vectors uniquely specify each basis state. The **complex conjugate** of a quantum state $|\psi\rangle$ is denoted by the row vector

$$\langle\psi| = \sum_{x \in \mathbb{Z}_2^n} \alpha_x^* \langle x| \text{ s.t. } \sum_{x \in \mathbb{Z}_2^n} |\alpha_x|^2 = 1.$$

The **inner product** of two n-qubit quantum states $|\psi_i\rangle$ and $|\psi_j\rangle$ is denoted by $\langle\psi_i|\psi_j\rangle$ which gives a complex number and the outer product of two n-qubit quantum states $|\psi_i\rangle$ and $|\psi_j\rangle$ is denoted by $|\psi_i\rangle\langle\psi_j|$, which forms a Hermitian matrix $M \in \mathbb{C}^{2^n \times 2^n}$.

▷ Mixed states.

The notion of pure states can be generalized by considering classical probability en-

sembles of pure states, i.e., probability distributions over pure states. These resulting quantum states are known as **mixed states** and are represented by **density matrices**. A density matrix ρ corresponding to a n-qubit quantum state is a $n \times n$ positive semi-definite (PSD) matrix with trace 1: $\rho = \sum_{i\in[n]} p_i |\psi_i\rangle \langle\psi_i|$, where $\{|\psi_i\rangle\}_{i\in[n]}$ are pure states and $\{p_i\}_{i\in[n]}$ form a probability distribution. Density matrices corresponding to pure states have rank 1, i.e., $\rho = |\psi\rangle \langle\psi|$.

▷ Similarity between Quantum states.

The **norm** of a quantum state $|\psi\rangle$ is denoted by $\||\psi\rangle\|^2 = \langle\psi|\psi\rangle$. The **Schatten k-norm** of a density matrix ρ is defined as $\|\rho\|_k = \left(\sum_i |p_i|^k\right)^{1/k}$, where $\{p_i\}_i$ are eigenvalues of ρ. The Schatten 1-norm of a general PSD ρ with eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant 0$, (also known as the **trace norm** of ρ) is denoted by $\|\rho\|_1 = \mathrm{Tr}(|\rho|) = \sum_i |p_i|$. For a density matrix $\|\rho\|_1 = \sum_i |\lambda_i| = 1$ (as stated earlier).

Definition 2.9 (Trace Distance)**.** The trace distance between two quantum states ρ and σ is denoted by $D(\rho, \sigma) = \frac{1}{2}\|\rho - \sigma\|_1$. When ρ and σ correspond to pure states $|\phi\rangle$ and $|\psi\rangle$ the trace distance is expressed as $D(|\phi\rangle, |\psi\rangle) = \sqrt{1 - |\langle\phi|\psi\rangle|^2}$.

Fact 2.2 (Distinguishing quantum states)**.** Suppose an algorithm is tasked with distinguishing between between two mixed states ρ_0 and ρ_1. If the algorithm is given ρ_b, where b is picked uniformly at random from $\{0, 1\}$, then the best success probability of distinguishing between ρ_0 and ρ_1 is $\frac{1}{2} + \frac{1}{2}D(\rho_0, \rho_1) = \frac{1}{2} + \frac{1}{2}\left(\frac{1}{2}\|\rho_0 - \rho_1\|_1\right)$.

▷ Unitary Evolution, Quantum Gates.

One way of manipulating vectors is through linear transformations. In quantum mechanics, for a linear transformation to be a valid quantum operation, it must ensure that it preserves the 2-norm of the vector. i.e.,

$$U|\psi\rangle = U\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\alpha'\\ \beta'\end{pmatrix}, \quad |\alpha'|^2 + |\beta'|^2 = |\alpha|^2 + |\beta|^2 = 1.$$

Hence, all valid quantum operations are unitary transformations U, $UU^\dagger = I$. We also refer to unitary operations as **quantum gates**. See Fig. 2.3 for some important one qubit gates. The C-U or the controlled-U gate takes as input a two qubit state $|\psi_1\rangle |\psi_2\rangle$, and applies U^{ψ_1} to the second qubit. Note that U is only applied to $|\psi_2\rangle$ if $|\psi_1\rangle = |1\rangle$. Examples of the controlled-U gate include the C-NOT gate and the controlled-rotation gate. Similarly, we can also construct multi-controlled unitary gates such as

$$\text{C-C-C-NOT} : |a, b, c, d\rangle \mapsto |a, b, c, d \oplus (a \cdot b \cdot c)\rangle \ \ \forall a, b, c, d \in \mathbb{Z}_2^n.$$

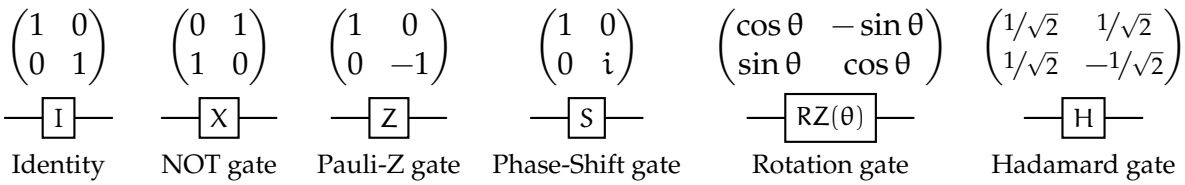


Figure 2.3: Important 1-qubit gates

Some interesting two-qubit gates are given in Fig. 2.4. The unitary evolution of mixed

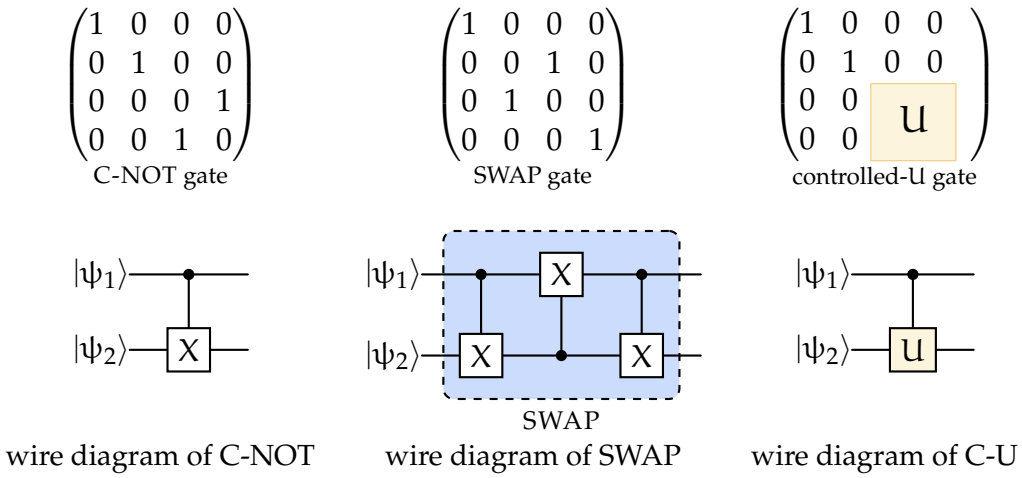


Figure 2.4: Important 2-qubit unitaries

states is as follows: $U : \rho \mapsto U\rho U^\dagger = \sum_{i\in[n]} p_i\, U\,|\psi_i\rangle\langle\psi_i|\,U^\dagger$.

▷ Measurement in Standard Basis.

Performing a measurement on a quantum system extracts classical information about its quantum state.

A quantum state $|\psi\rangle$ can be expressed as a superposition of an arbitrary set of orthogonal basis vectors $\mathcal{B}$ that span the corresponding Hilbert space $\mathbb{H}_\mathcal{B}$, s.t., measuring a quantum state in the basis $\mathcal{B}$ only yields a classical state corresponding to one of the possible vectors in $\mathcal{B}$.[11]

Fact 2.3 (Born's rule)**.** Measuring an arbitrary n-qubit quantum state $|\psi\rangle$ in any orthonormal basis $\mathcal{B} = \{|\mathcal{B}_0\rangle, \ldots, |\mathcal{B}_{2^n-1}\rangle\}$ yields the outcome $|\mathcal{B}_i\rangle$ with probability $|\langle\mathcal{B}_i|\psi\rangle|^2$.

To measure in any arbitrary orthonormal basis $\mathcal{B} = \{|\mathcal{B}_0\rangle, \ldots, |\mathcal{B}_{2^n-1}\rangle\}$ instead of the computational basis, we first apply a unitary transformation $U_\mathcal{B}$ s.t. $U_\mathcal{B}\,|\mathcal{B}_x\rangle = |x\rangle$ for all $x \in \mathbb{Z}_2^n$, and then measure in the standard basis. See Fig. 2.5 for a circuit that measures the first qubit in the standard basis and the second qubit in the basis $\mathcal{B}$.

[11] Hence, measuring $|\psi\rangle$ *collapses* the quantum state to a classical state and the *quantum information* encoded in the amplitudes is gone.

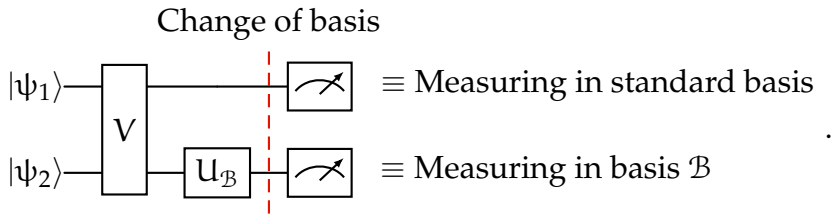


Figure 2.5: Change of Basis in Measurement

▷ Projective Measurement.

A more general notion of measurement compared to measurement in the standard basis is **projective measurement**. Given a collection of k projectors (i.e., Hermitian and positive semidefinite matrices with $0-1$ eigenvalues) $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$ s.t. $\sum_{i\in[k]} \mathcal{B}_i = I$, we can measure any arbitrary pure quantum state $|\psi\rangle$ in this projector basis to obtain a k-outcome quantum measurement, instead of the 2^n-outcome measurement in the standard basis. Given, a quantum state ρ, we obtain the outcome i w.p. $p_i = \mathrm{Tr}(\mathcal{B}_i\rho)$.

▷ POVMs.

POVMs (positive operator valued measurements) are a generalization of projective measurements. Here, we consider a collection of k PSD matrices $\Pi = \{\Pi_1, \ldots, \Pi_k\}$ s.t. $\sum_{i\in[k]} \Pi_i = I$. When the POVM Π is applied to a quantum state $|\psi\rangle$ it collapses to the state $\frac{|\psi_j\rangle}{\||\psi_j\rangle\|}$ with probability $\langle\psi|\Pi_j|\psi\rangle$, where $j \in [k]$. When measuring a mixed state ρ, we obtain the state $\frac{\Pi_i\rho\Pi_i}{p_i}$ w.p. $p_i = \mathrm{Tr}(\Pi_i\rho)$. Given a set of POVMs Π, the POVM leading to the largest difference in measurement outcomes between two quantum states is the trace distance between the two states.

$$D(\rho,\sigma) = \frac{1}{2}\|\rho-\sigma\|_1 = \frac{1}{2}\max_{\Pi_i\in\Pi}\sum_i |\mathrm{Tr}\{\Pi_i\rho\} - \mathrm{Tr}\{\Pi_i\sigma\}|. \tag{2.8}$$

In other words, the trace distance between two quantum states is the largest difference in probability that the two states can give to the same measurement outcome.

▷ Quantum Circuits.

Classical Boolean circuits are finite directed acyclic graphs with AND, OR, and NOT gates which take n-input bits and outputs m-output bits to compute some function $f \in \mathbb{Z}_2^n \mapsto \mathbb{Z}_2^m$. Classical circuits take n-bits as input and output m-bits. A quantum circuit is a generalization of a classical circuit where the input is an n-qubit state, the output is an m-qubit state, and the AND, OR, and NOT gates are replaced by *quantum gates*. In this thesis, we consider the following augmented (non-minimal) universal gate set as the canonical quantum basis gates: $\{X, Y, Z, H, \mathrm{CNOT}, S, T\}$.

▷ Quantum Query Model.

Sometimes, in quantum circuits, we assume that we are given black-box access to a unitary operator (also known as an **oracle**), which encodes some information only accessible by queries. See Fig. 2.6 for examples of oracles.

Phase Oracles. $O_x^{\pm} : |i, a\rangle \mapsto (-1)^{a.x_i} |i, a\rangle$

Function Oracles. $O_f : |x, a\rangle \mapsto |x, a \oplus f(x)\rangle$

Probability Oracles. $O_p : |x, 0, 0\rangle \mapsto \sqrt{p_x} |x, 1, \psi_x^1\rangle + \sqrt{1-p_x} |x, 0, \psi_x^0\rangle$

Figure 2.6: Some common oracles used in quantum algorithms

Example 2.5 (String Oracle / Address Oracle)**.** A secret string $s = s_1 s_2 \dots s_n$ is encoded into some oracle O_s, and information about the bits of s can be extracted through querying O_s as: $O_s : |i, a\rangle \mapsto |i\rangle |a \oplus s_i\rangle$. The "string" oracle in this example is a type of function oracle as given in Fig. 2.6.

In the standard quantum query model formally introduced by Beals et al. [Bea+01], a quantum query algorithm has 3 sets of registers - ancillas $\mathcal{A}$, workspace $\mathcal{W}$, and query registers $\mathcal{Q}$ all initialized to 0. A k-query quantum query algorithm proceeds by making k rounds of interleaved unitary operations and calls to some quantum query oracle. The algorithm ends by measuring the ancilla register. See Fig. 2.7 for a schematic of the quantum query model.

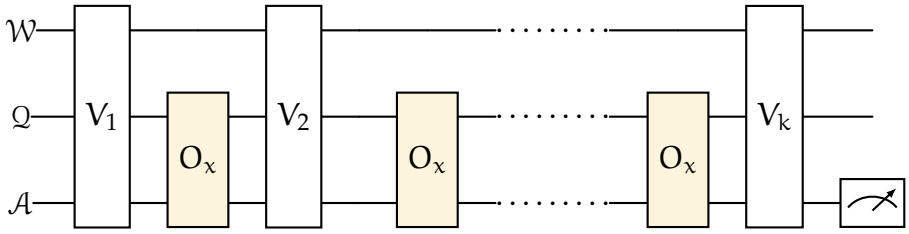


Figure 2.7: The Quantum Query Model

▷ Quantum Algorithms and BQP.

Given a Boolean function f, a quantum circuit computes f with error $\varepsilon > 0$, if $\forall x \in \mathbb{F}_2^n$, the quantum circuit Q can compute $f(x)$ with probability $1 - \varepsilon$.[12] The bounded error query complexity of computing f is the smallest number of calls to any oracle O_x s.t. Q computes f w.p. $\geqslant 1 - \varepsilon$. The bounded error time complexity also takes into

[12] By compute, we mean that the outcome of the measurement in the ancilla is $f(x)$.

account the time taken to implement the unitaries $V_1, \ldots, V_k$. The complexity class BQP is the class of decision problems efficiently solvable by a quantum computer and is defined as the class of languages $L \subseteq \{0,1\}^*$ that are decidable, with a bounded error probability of at most $1/3$ by a poly-time uniform family of quantum circuits $\{Q_n\}_n$.

Remark 2.5.1. Throughout this thesis, when we say that a quantum algorithm exists that performs a certain task, we mean that there is a uniform family of quantum query circuits that performs said task.

§ 2.5.1 The Quantum RAM model

Quantum advantage in the fields of algorithms, optimization, machine learning, or even cryptography[13] is crucially dependent on the ability of quantum algorithms to manipulate data in superposition states. This requires access to a specialized memory, which may not always be a foregone conclusion, especially in the era of Noisy Intermediate-Scale Quantum (NISQ) devices. In this section, we discuss an often overlooked yet critical assumption on memory access made by quantum algorithms. First we start with a brief overview of classical memory access models, which we generalize later to the quantum setting.

▷ Classical Memory Access.

The Random Access Memory (RAM) is a specialized memory for storing information (as bits) that can be accessed with low latency and constant overhead regardless of the physical location of information. Suppose we had memory access to M bits of memory classically. This memory could be accessed either by sequential memory access models such as Turing machines or gate-based models such as circuits. The worst-case time complexity (or gate-complexity) for sequential models (circuit models, respectively) to access any arbitrary memory bit would be $\Omega(M)$. However, in the (classical) **RAM model**, we assume that the worst-case cost of accessing a memory bit is $O(\log M)$. A classical RAM model can determined by the following $(m + \log m + 1)$ tuple:

$$\Big(\underbrace{y_1, y_2, \ldots, y_{\log m}}_{\text{ADDRESS BITS}}, \underbrace{b}_{\text{I/O BIT}}, \underbrace{z_1, z_2, \ldots, z_m}_{\text{MEMORY BITS}} \Big).$$

We now define a reversible operation $\mathcal{O}_{r/w}$ that can be used to read or write from the classical RAM model:

$$\mathcal{O}_{r/w} : \big(y_1, \ldots, y_{\log m}, b, z_1, \ldots, z_y, \ldots, z_m \big) \longmapsto \big(y_1, \ldots, y_{\log m}, z_y, z_1, \ldots, b, \ldots, z_m \big)$$

[13] See section III of the survey by Jaques and Rattew [JR23] for a detailed list of references.

▷ Quantum Memory Access.

We can generalize the above definition of a classical RAM to a Quantum Random Access Memory (QRAM) in primarily two ways listed below. We refer the reader to the surveys [JR23; PCG23] for specific implementations of QRAMs.

1. Quantum Random Access for Quantum Memory (QQRAM): The memory stores information as qubits. In the QQRAM model, the memory can become entangled with the quantum circuit accessing the memory.

$$U_{r/w} : \left|y_1, \ldots, y_{\log m}, b, z_1, \ldots, z_y, \ldots, z_m\right\rangle \longmapsto \left|y_1, \ldots, y_{\log m}, z_y, z_1, \ldots, b, \ldots, z_m\right\rangle$$

2. Quantum Random Access for Classical Memory (CQRAM): The memory stores information as classical bits. An example of a CQRAM would be the address oracle discussed in Example 2.5. We can perform read operations from the memory bits in superposition, but we can only perform classical writes. The unitary corresponding to the read operation is

$$U_{\text{read}} : \left|y_1, \ldots, y_{\log m}, b, \ldots, z_y, \ldots, z_m\right\rangle \longmapsto \left|y_1, \ldots, y_{\log m}, b \oplus z_y, \ldots, z_y, \ldots, z_m\right\rangle$$

All algorithms in this thesis are assumed to be in the CQRAM model. Henceforth, we shall mean CQRAM when we refer to the QRAM model unless explicitly stated otherwise. Access to the CQRAM by itself is not too strong of an assumption if we assume that the CQRAM is empty and that the input must be loaded into the CQRAM before any quantum circuit can access it. This setting guarantees, at most, a polynomial speedup over settings in which there is no access to a CQRAM.

§ 2.5.2 Important Quantum Subroutines

In this section, we list some important quantum subroutines that have historically provided most of the speedups in quantum learning theory and also serve as core building blocks of the algorithms outlined in Chapters 4 to 6 and 9.

▷ Amplitude Amplification.

Suppose we have access to a unitary U that prepares a quantum state

$$|\psi\rangle = \sqrt{\alpha_{\text{good}}}\left|\psi_{\text{good}}\right\rangle + \sqrt{1-\alpha_{\text{good}}}\,|\alpha_{\text{bad}}\rangle\,,$$

and we wanted to obtain the "good" state $\left|\psi_{\text{good}}\right\rangle$ with high probability. Simply measuring $|\psi\rangle$ would only give us $\left|\psi_{\text{good}}\right\rangle$ with probability $\left|\alpha_{\text{good}}\right|^2$. Hence, on expecta-

tion it would take us $1/|\alpha_{good}|^2$ independent measurements to obtain $\left|\psi_{good}\right\rangle$ with high probability. However, Brassard et al. [Bra+02] showed that this can be improved using Amplitude Amplification as stated below.

Lemma 2.8 (Amplitude Amplification)**.** Let $p > 0$ be a constant, and U be a unitary operator s.t. $U|0\rangle = \sqrt{p_0}|\phi_0\rangle|0\rangle + \sqrt{1-p_0}|\phi_1\rangle|1\rangle$ for an unknown $p_0 \geqslant p > 0$. There exists a quantum algorithm that makes $\Theta(\sqrt{p'/p})$ expected number of calls to U and U^{-1} and outputs the state $|\phi_0\rangle$ with a probability $p' > 0$.

Let $|\phi_0\rangle = \left|\psi_{good}\right\rangle$, and $p_0 = \left|\alpha_{good}\right|^2$ in Lemma 2.8. Then, setting $p' = 1$ in Lemma 2.8 gives us $\left|\psi_{good}\right\rangle$ by making $O\left(1/|\alpha_{good}|\right)$ queries to U and U^{-1} in expectation. This gives us a quadratic speedup over the classical case.

▷ Amplitude estimation and Mean estimation.

One issue with Amplitude Amplification is that we may not know p_0 and hence, how many times we need to invoke U and U^{-1} in order to obtain $\left|\psi_{good}\right\rangle$. We use the following result by Ambainis [Amb08] to obtain an estimate of p_0 with relative error.

Lemma 2.9 (Relative Amplitude Estimation)**.** Given a constant $p > 0$, an error parameter $\varepsilon > 0$, a constant $k \geqslant 1$, and a unitary U such that $U|0\rangle = \sqrt{p_0}|\phi_0\rangle|0\rangle + \sqrt{1-p_0}|\phi_1\rangle|1\rangle$ where either $p_0 \geqslant p$ or $p_0 = 0$. Then there exists a quantum algorithm, that produces an estimate $\tilde{p}_0$ of the success probability p_0 with probability at least $1 - 1/2^k$ such that $|p_0 - \tilde{p}_0| \leqslant \varepsilon \cdot p_0$ when $p_0 \geqslant p$, and makes $O\left(\frac{k}{\varepsilon\sqrt{p}}\left(1 + \log\log\frac{1}{p}\right)\right)$ calls to U and U^{-1} in expectation.

Classically, we can use Lemma 2.1 to perform mean estimation with relative error by sampling $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ independent copies of $U|0\rangle$ and setting $p = O(1/m)$, where $|\phi_0\rangle$ is a superposition over m basis states. Hence, using Relative Amplitude Estimation, we can perform mean estimation with relative error ε with a quadratic speedup over classical mean estimation techniques in terms of dependence on ε and m.

▷ The Quantum Fourier Transform, and Fourier Sampling.

The Quantum Fourier Transform (QFT) is a unitary that performs the Discrete Fourier transform on an N-dimensional Hilbert space as follows

$$\mathrm{QFT}_N|j\rangle = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\exp\left\{\frac{2\pi\iota\langle j,k\rangle}{N}\right\}|k\rangle$$

If $N = 2^n$, we can approximate QFT very closely using $O(n\log n)$ gates. Classically performing the Discrete Fourier transform takes $\Theta(n2^n)$ operations using the Fast

Fourier transform algorithm. Quantum algorithms tend to use QFT in the following algorithmic framework.

- Start with the uniform superposition state $|+^n\rangle$ over all inputs.
- Apply some (black-box or classically efficient) function U_f to $|+^n\rangle$.
- Apply QFT_{2^n} to $U_f |+^n\rangle$ and measure.

Keeping the above framework in mind, we state the Fourier Sampling lemma by Bernstein and Vazirani [BV97].

Lemma 2.10 (Fourier Sampling)**.** Given access to a function oracle O_f, there exists a quantum algorithm (see Algorithm 2.1) that produces the state $\sum_{S\in[n]} \hat{f}(S)\,|S\rangle$, using only 1 query to O_f and $O(n)$ gates.

Algorithm 2.1: The Fourier Sampling Algorithm

Input: Oracle $O_f : |x\rangle \mapsto f(x)\,|x\rangle$, providing black-box access to f.

1 Start with $|0^n\rangle$ and apply $H^{\otimes n}$ to obtain $|+^{\otimes n}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{F}_2^n} |x\rangle$.

2 Query O_f to obtain the state $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{F}_2^n} f(x)\,|x\rangle$.

3 Apply $H^{\otimes n}$ to $|\psi\rangle$.

Output: The state $\sum_{S\in[n]} \hat{f}(S)\,|S\rangle$.

Remark 2.5.2. Unlike in the quantum case, explicitly constructing the distribution $\mathrm{FS}(f)$ is harder classically, since any $\hat{f}(S) = \mathbb{E}\,[f \cdot \chi_S]$ depends on all 2^n values of f.

We note that Line 3 in Algorithm 2.1 performs the following operation:

$$H^{\otimes n}|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{F}_2^n} f(x) \left(\frac{1}{\sqrt{2^n}} \sum_{S\in\mathbb{Z}_2^n} (-1)^{x\cdot S} |S\rangle \right) \tag{2.9}$$

$$= \frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} \sum_{S\in\mathbb{F}_2^n} f(x)\cdot(-1)^{x\cdot S} |S\rangle \tag{2.10}$$

$$= \sum_{S\in\mathbb{F}_2^n} \frac{1}{2^n} \sum_{x\in\mathbb{F}_2^n} f(x)\cdot(-1)^{\sum_{i\in S} x_i} |S\rangle = \sum_{S\in[n]} \hat{f}(S)\,|S\rangle\,. \tag{2.11}$$

The state produced in Fourier Sampling allows us to sample from $\mathrm{FS}(f)$, i.e., sample S with probability $\hat{f}(S)^2$. We can use Fourier Sampling to learn linear functions f using one query to O_f [BV97], whereas, classically, this takes $O(n)$ queries.

Part I

Quantum Weak Learners

Every great man is an idol, an oracle of inquiry.
Don't aspire to know the former, but aspire to know the deity in his soul.

MICHAEL BASSEY JOHNSON

Chapter 3

Quantum and Classical Oracle Models

Abstract

In this chapter, we show that PAC learning under various label noise settings can be redefined as PAC learning with access to corresponding query oracles (as in Definition 2.8) in both the classical and quantum settings. After introducing various classical and quantum example oracles, we discuss the concept of **simulatability** introduced by Bshouty and Jackson [BJ95], which helps us characterize the relative strengths of various classical and quantum oracle-access learning algorithms and helps us form a hierarchy of oracles. We end the chapter by generalizing a result by Bshouty and Jackson [BJ95] to show that classical membership query oracles are not even *approximately* simulatable by quantum example oracles.

Contents

3.1 Introduction . . . 44
3.2 Example Query Oracles . . . 44
3.2.1 Classical Example Query Oracles . . . 44
3.2.2 Quantum Example Query Oracles . . . 45
3.3 Membership Query Oracles . . . 47
3.3.1 Criticisms of MQ oracles and other variants . . . 48
3.3.2 Biased Quantum Membership Query Oracles . . . 49
3.4 Exact and Approximate Simulatibility of Membership Query Oracles . . . 50
3.5 Conclusion and Future Work . . . 53

§ 3.1 Introduction

There are many natural concept classes that are not known to be efficiently PAC learnable, sometimes even in the noiseless setting (see [AL88; Bsh95; Val84]). However, if we have the ability to query certain types of function oracles, many such concept classes can be PAC learned efficiently. This poses the following question:

> **Question 3.1.** Given a concept class $\mathcal{C}$ that is not known to be PAC learnable, what is the weakest possible oracle a learning algorithm can query in order to satisfy efficient (ε, δ)-PAC learnability for $\mathcal{C}$?

In this chapter, we introduce the various oracle models that are of interest throughout this thesis and show how to form a partial hierarchy among them.

▷ Structure and Organisation.

We introduce example query oracles in both the classical (see Section 3.2.1) and quantum (see Section 3.2.2) settings that capture the notion of PAC learning under various noise models through the lens of Definition 2.8. We then introduce a particularly powerful class of oracles in Section 3.3 - the MQ oracle and show that MQ oracles are not *approximately simulatable* by quantum example query oracles even in the noiseless setting (see Section 3.4).

§ 3.2 Example Query Oracles

§ 3.2.1 Classical Example Query Oracles

▷ The EX Oracle.

Querying the classical random **example oracle** $\mathrm{EX}(\mathfrak{D}, c)$ generates labeled examples of the form $(x, c(x))$ where $x \sim \mathfrak{D}$. Using the EX oracle, we can redefine the sample complexity and time complexity of PAC learning concept $c \in \mathcal{C}$ using A in terms of the number of queries A makes to $\mathrm{EX}(\mathfrak{D}, c)$ as follows.

> **Definition 3.1** ((ε, δ)-PAC learning with access to EX oracle)**.** A learner A (ε, δ)-PAC learns concept class $\mathcal{C}$ over n-bit instances, with query access to $\mathrm{EX}(\mathfrak{D}, c)$, using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0, 1)$, for every distribution $\mathfrak{D}$ over $\mathcal{X}$, and every target concept $c \in \mathcal{C}$, A makes at most $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ queries to $\mathrm{EX}(\mathfrak{D}, c)$ and outputs h s.t. $\Pr[\mathrm{err}_{\mathfrak{D},c}(h) \leqslant \varepsilon] \geqslant 1 - \delta$. Equivalently, in terms of correlation, $\Pr[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant 1 - 2\varepsilon] \geqslant 1 - \delta$.

▷ The Random Noise Example Query (REX) Oracle.

Querying an $\mathrm{REX}^{\eta}(\mathfrak{D}, c)$ oracle generates labeled examples of the form $\{(x_i, y_i')\}_{i \in [m]}$,

where $y'_i = \overline{y_i}$ w.p. η and $y'_i = y_i$ w.p. $1-\eta$. Hence, PAC learning in the RCN setting can be thought of as the learner A making queries to an REX oracle.

▷ The Agnostic Example Query (AEX) Oracle.

Similar to the RCN setting, PAC learning in the agnostic setting can thought of as the learner A making queries to an AEX oracle, where invoking an $\mathrm{AEX}(\mathfrak{D})$ oracle generates samples $(x, y) \sim \mathfrak{D}$ according to some unknown joint distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$. The notion of learnability in the agnostic case was defined as follows [KMV08].

Definition 3.2 (β-optimal (ε, δ)-agnostic PAC learning)**.** A learner A β-optimally (ε, δ)-agnostic PAC learns a benchmark concept class $\mathcal{C}$ over n-bit instances, with query access to $\mathrm{AEX}(\mathfrak{D})$, using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0,1)$, for some $\beta \in [0, 1/2)$, for every distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$, A makes at most $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ queries to $\mathrm{AEX}(\mathfrak{D})$ and outputs h s.t. $\Pr\left[\mathrm{err}_{\mathfrak{D}}(h) \leqslant \mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) + \beta + \varepsilon\right] \geqslant 1 - \delta$ for some $\beta \in [0, 1/2)$. Equivalently, in terms of correlation, $\Pr\left[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant \mathrm{optcorr}_{\mathfrak{D}}(\mathcal{C}) - 2\beta - 2\varepsilon\right] \geqslant 1 - \delta$..

Learnability (and efficient learnability) in Definition 3.2 implies that the number of queries made by A to $\mathrm{AEX}(\mathfrak{D})$ (time taken by A to output h, respectively) is polynomially bounded, similar to Definition 3.1. In Definition 3.2, if there exists a concept $c \in \mathcal{C}$ s.t. $\mathrm{err}_{\mathfrak{D}}(c) := \Pr_{(x,y)\sim\mathfrak{D}}[c(x) \neq y] = 0$, then $\mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) = \beta$, then we are almost back in the noiseless PAC learning setting.

The concept class $\mathcal{C}$ serves as a *benchmark* for the learner A in the agnostic setting. If we choose a very bad benchmark concept class, i.e., $\mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) > 1/2$, then the best agnostic learner will produce a hypothesis with accuracy less than $1/2$. Similar to the PAC setting, if A outputs a hypothesis from the benchmark class $h \in \mathcal{C}$, then A is a proper agnostic learner for $\mathcal{C}$. Otherwise, A is an improper agnostic learner for $\mathcal{C}$.

§ 3.2.2 Quantum Example Query Oracles

▷ The QEX Oracle.

Bshouty and Jackson [BJ95] generalized the notion of (ε, δ)-PAC learning with access to EX oracle to the quantum setting by introducing the QEX oracle (parameterized as $\mathrm{QEX}(\mathfrak{D}, c)$), and defining quantum PAC learning with access to a QEX oracle as follows.

$$\mathrm{QEX}(\mathfrak{D}, c) : |0,0\rangle \mapsto \sum_{x \in \mathbb{F}_2^n} \sqrt{\mathfrak{D}(x)}\,|x, c(x)\rangle\,. \tag{3.1}$$

▷ The Quantum RCN Example Query (QREX) Oracle.

We can define a quantum example oracle that captures the RCN setting as follows.

$$\mathrm{QREX}^{p}(\mathfrak{D}, c) : |0,0\rangle \mapsto \sum_{x \in \mathbb{F}_2^n} \sqrt{(1-p)\cdot \mathfrak{D}(x)}\,|x, f(x)\rangle + \sum_{x \in \mathbb{F}_2^n} \sqrt{p \cdot \mathfrak{D}(x)}\,\left|x, \overline{f(x)}\right\rangle . \quad (3.2)$$

Definition 3.3 (Quantum PAC learning with QEX (resp. QREX) oracle)**.** A quantum algorithm A (ε, δ)-quantum PAC learns concept class $\mathcal{C}$ over n-bit instances, with query access to $\mathrm{QEX}(\mathfrak{D}, c)$ (resp. $\mathrm{QREX}^{p}(\mathfrak{D}, c)$), using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0,1)$, for every distribution $\mathfrak{D}$ over $\mathcal{X}$, and every target concept $c \in \mathcal{C}$, A makes at most $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ queries to $\mathrm{QEX}(\mathfrak{D}, c)$ (resp. $\mathrm{QREX}^{p}(\mathfrak{D}, c)$) and outputs h s.t. $\Pr[\mathrm{err}_{\mathfrak{D},c}(h) \leqslant \varepsilon] \geqslant 1-\delta$. Equivalently, in terms of correlation, $\Pr[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant 1-2\varepsilon] \geqslant 1-\delta$.

▷ The Quantum Agnostic Example Query (QAEX) Oracle.

Arunachalam and Wolf [AW17a] generalized the Agnostic Example Query (AEX) oracle and the notion of β-optimal (ε, δ)-agnostic PAC learning to the quantum setting as follows:

$$\mathrm{QAEX}(\mathfrak{D}) : |0,0\rangle \mapsto \sum_{(x,y) \in \mathbb{F}_2^{n+1}} \sqrt{\mathfrak{D}(x,y)}\,|x,y\rangle . \quad (3.3)$$

Definition 3.4 (Quantum agnostic PAC learning with QAEX oracle)**.** A quantum learner A β-optimally (ε, δ)-PAC learns a benchmark concept class $\mathcal{C}$ over n-bit instances, with query access to $\mathrm{QAEX}(\mathfrak{D})$, using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0,1)$, for some $\beta \in [0, 1/2)$, for every distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$, A makes at most $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ queries to $\mathrm{QAEX}(\mathfrak{D})$ and outputs h s.t. $\Pr\left[\mathrm{err}_{\mathfrak{D}}(h) \leqslant \mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) + \beta + \varepsilon\right] \geqslant 1-\delta$ for some $\beta \in [0, 1/2)$. Equivalently, in terms of correlation, $\Pr\left[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant \mathrm{optcorr}_{\mathfrak{D}}(\mathcal{C}) - 2\beta - 2\varepsilon\right] \geqslant 1-\delta$..

Observation 3.2.1. If the learner is classical but the query oracles are quantum, the only recourse left to the learner is to measure the state in Eqs. (3.1) to (3.3) to obtain one labeled sample. Therefore, the $\mathrm{EX}(\mathfrak{D}, c)$ and $\mathrm{QEX}(\mathfrak{D}, c)$ oracles (and their noisy versions, respectively) are equivalent to a classical learning algorithm.

Remark 3.2.1. In Definitions 3.3 and 3.4, when we say that the quantum learner outputs a hypothesis, we mean that the quantum learner performs a POVM measurement such that each outcome of the measurement is associated with a hypothesis.

Remark 3.2.2. The concepts of learnability and efficient learnability in the quantum PAC setting are similar to those in the classical setting. The notions of complexity that

we care about in the quantum PAC setting are, once again, the sample (and/or query) complexity and the time complexity of the learner.

Remark 3.2.3. We use $A^{\mathcal{O}}$ to denote that learner A makes queries to oracle $\mathcal{O}$.

§ 3.3 Membership Query Oracles

One of the most important oracles for PAC learning is the Membership Query oracle MEM_c. Querying the MEM_c oracle with an instance $x \in \mathbb{F}_2^n$ returns the instance's label $c(x)$ directly. As we can see, the MQ oracle allows us to make very powerful *out-of-distribution* queries. The quantum MQ oracle $\mathrm{QMEM}(f) : |x\rangle \mapsto |x\rangle |f(x)\rangle$ is simply a function oracle (see Fig. 2.6), and ubiquitous in designing quantum algorithms such as Algorithm 2.1. We shall explore how to use classical and quantum MQ oracles to design weak learners in Chapter 4.

Learning with the membership queries is an *active* model of learning, where the learner can choose to query information about any instance-label pair it wishes, compared to the *passive* learning models in which the learner can only obtain instance-label pairs that are sampled according to an unknown distribution. We state two results below that give some intuition into the relative strength between the two query models of learning. Bshouty [Bsh95] proved the following result.

> **Lemma 3.1** (Learnability of Boolean functions)**.** Any n-bit Boolean function is PAC learnable with Membership queries in polynomial time in n and its DNF size or its decision tree size.

Recall that (ε, δ)-PAC learning with access to EX oracle is nothing but vanilla PAC learning, reformulated using query access to EX oracle. Hence, Lemma 3.1 immediately implies a possible learning separation between PAC learning with EX queries and MQ queries for DNF, which is not known to be PAC learnable with EX queries. The following proposition, which follows from similar arguments by [AW17b; AG98], gives an unconditional (albeit polynomial) separation between MQ oracles and EX oracles.

> **Proposition 3.1** (MQ vs EX oracles)**.** Suppose a (publicly known) set of instances $\mathcal{S}$ is the largest set shattered by a binary concept class $\mathcal{C}$, and let $\mathfrak{D} : \mathcal{S} \mapsto [0, 1]$ be a distribution that puts most of its weight on a particular instance as follows:
>
> $$\mathfrak{D}(x) = \begin{cases} 1 - 3\varepsilon & , x = x', \varepsilon = 1/\mathrm{poly}(|\mathcal{S}|) \\ \frac{3\varepsilon}{|\mathcal{S}|}, & , x \in \mathcal{S} \setminus \{x'\}. \end{cases}$$

> Then, any (ε, δ)PAC learner for c w.r.t. $\mathfrak{D}$ making queries to a MEM_c oracle has polynomially smaller query complexity than a learner making queries to an $\mathrm{EX}(\mathfrak{D}, c)$ oracle.

Proof. **Learning w.r.t. MQ oracle:** Since $\mathcal{C}$ has VC-dim $|\mathcal{S}|$, there exists a concept $c \in \mathcal{C}$, s.t. $c(x') = 0$, and assigns some labeling in $2^{|\mathcal{S}|-1}$ to other elements of the instance space. Let A be a PAC learner for c w.r.t. $\mathfrak{D}$ that makes at most Q_{MQ} queries to an MEM_c oracle. A requires $Q = O\left(\frac{2}{3}|\mathcal{S}|\right)$ queries to PAC learn c w.r.t. $\mathfrak{D}$, since we need to learn the labels of at most $\frac{2}{3}|\mathcal{S}|$ instances to ε-approximate c.
Learning w.r.t. EX oracle: From Lemma 12 of Arunachalam and Wolf [AW17b], we obtain that $Q_{EX} = \mathrm{poly}(|\mathcal{S}|)$ queries, straightforwardly. This is polynomially larger than the MQ case. □

Below, we give a brief proof sketch of the main idea of the query complexity of PAC learning with EX oracle query access. *Proof Sketch.* [Learning w.r.t. EX oracle] Let A be a (ε, δ)-PAC learner for c w.r.t. $\mathfrak{D}$ that makes at most Q_{EX} queries to a $\mathrm{EX}(\mathfrak{D}, c)$ oracle. To ε-approximate the concept c, A would need to learn the labeling of at least $\frac{2}{3}|\mathcal{S}|$ number of instances. To learn all possible $2^{|\mathcal{S}|}$ labelings, we need Q_{EX} queries to yield $\Omega(|\mathcal{S}|)$ bits of information about c. However, each query only yields $O(\varepsilon)$ bits of information about c, since every instance (apart from x') is sampled only with probability $\frac{3\varepsilon}{|\mathcal{S}|}$. Since Q_{EX} queries would yield at most $O(Q_{EX}\varepsilon)$ bits of information about c, we need $Q_{EX} \cdot \varepsilon \geqslant |\mathcal{S}|$. Therefore, A needs to make $Q_{EX} = \Omega\left(\frac{|\mathcal{S}|}{\varepsilon}\right) = \mathrm{poly}(|\mathcal{S}|)$ queries. □

Both Lemma 3.1 and Proposition 3.1 show why active learning with MQ oracles is more powerful than relying on sampling queries for learning. Despite its advantages, there have been criticisms of MQ oracles, which we discuss in Section 3.3.1.

§ 3.3.1 Criticisms of MQ oracles and other variants

Even though a significant portion of the learning theory literature has been dedicated to constructing PAC learning algorithms with access to MQ oracles for various Boolean concept classes, it has not been the subject of much practical work. This is possibly due to the paradoxical nature of MQ oracles themselves: If it was possible to construct MQ oracles for some target concept class $\mathcal{C}$ in practice, there would be no need to construct PAC learners for $\mathcal{C}$. On the other hand, sampling query oracles such as EX, REX, AEX (and their quantum counterparts) simply extend the usual notion of supervised learning by reframing the definitions in terms of oracle queries.
Remark 3.3.1. An infamous experiment was conducted by Baum and Lang [BL92]

where they tried to learn a linear classifier for handwritten characters and digits using human annotators as MQ oracles. It was seen that in addition to making out-of-distribution queries, sometimes the learner tended to make out-of-domain queries as well. This may be noted as anecdotal evidence in support of the hypothesis that MQ oracles may not be useful in practice.

In order to address concerns levied against the *unrestricted* nature of queries the learner can make to an MQ oracle, Awasthi et al. [AFK13] introduced the local MQ oracle model, where the learner can only query the MQ oracle on some neighborhood[1] of a training set S provided to the learner. Unfortunately, it was shown by Bary-Weisberg et al. [BWDSS20] that even though learning with q-local MQ, for any constant $q > 0$ is stronger than vanilla PAC learning, it is not sufficient to learn relatively simple classes of Boolean functions such as juntas or Decision Trees in a distribution-free setting. A similar oracle model is the Random Walk Query (RWQ) oracle model introduced by Bshouty et al. [Bsh+03], where the first point given to the learner is uniformly sampled from $\mathbb{F}_2^n$, while every succeeding point is obtained by a uniform random walk over $\mathbb{F}_2^n$.

Remark 3.3.2. We note here that local MQ and RWQ oracles can be used to obtain efficient PAC learners for DNF w.r.t. the uniform distribution [AFK13; Bsh+03].

§ 3.3.2 Biased Quantum Membership Query Oracles

Before moving on to the next section, we introduce a nomenclature for quantum membership query oracles that will be useful for us in Chapter 4.

Definition 3.5 (ε-Biased oracles)**.** A quantum example query oracle $\mathcal{O}_f^\varepsilon$ corresponding to a Boolean function f is called an ε-biased for some $0 < \varepsilon < \frac{1}{2}$, when

$$\mathcal{O}_f^\varepsilon : |0\rangle^{\otimes n} |0\rangle \mapsto \sum_{x \in \mathbb{F}_2^n} \alpha |x\rangle |f(x)\rangle + \sqrt{1-\alpha^2} |x\rangle \left|\overline{f(x)}\right\rangle, \quad \alpha > \frac{1}{2} + \varepsilon.$$

QREX oracles is an ε-biased oracle. An unbiased oracle has $\alpha = 1$. Iwama et al. [IRY05] proved the following lemma:

Lemma 3.2 (Simulatability of Unbiased Oracles by ε-Biased Oracles)**.** For any $O(T)$ query quantum algorithm that solves a problem with high probability with query access to an unbiased oracle, there exists an $O\left(\frac{T}{\varepsilon}\right)$ query quantum algo-

[1] Fix some metric over the instance space $\mathcal{X}$, e.g., consider Hamming distance d when instances are sampled from the Boolean hypercube. Now, we are only allowed to query samples that are within an open ball of radius d around any instance in the training set.

rithm that solves the same problem with high probability with query access to an ϵ-biased oracle.

We now define an oracle that is noisier than ε-Biased oracles.

Definition 3.6 (Strongly Biased oracles)**.** A quantum example query oracle $\mathcal{O}_f^{\varepsilon_x}$ corresponding to a Boolean function f is called a strongly biased for some $0 < \varepsilon_x < \frac{1}{2}, \forall x \in \mathbb{F}_2^n$, when

$$\mathcal{O}_f^{\varepsilon_x} : |0\rangle^{\otimes n} |0\rangle \mapsto \sum_{x \in \mathbb{F}_2^n} \alpha_x |x\rangle |f(x)\rangle + \sqrt{1-\alpha_x^2}\, |x\rangle \left|\overline{f(x)}\right\rangle, \quad \alpha_x > \frac{1}{2} + \varepsilon_x, \forall x \in \mathbb{F}_2^n.$$

Note that learning under the sampling oracle QAEX is a stronger model of learning than learning under a strongly biased oracle. In Chapter 4, we shall see how to convert a QAEX oracle to a strongly biased oracle.

§ 3.4 Exact and Approximate Simulatibility of Membership Query Oracles

We first reprove a result of Bshouty and Jackson [BJ95] showing that QEX oracles cannot exactly *simulate* MQ oracles w.r.t. the uniform distribution. First, we start by defining the notion of exact simulatability and then extending this to the notion of approximate simulatability. Then, we generalize the result of Bshouty and Jackson [BJ95] to show that QEX oracles cannot even approximately *simulate* MQ oracles w.r.t. the uniform distribution.

Definition 3.7 (Exact Simulatibility of MQ)**.** A class of quantum (resp. classical) example oracles $\mathcal{O}$ can exactly simulate MQ for a Boolean concept class $\mathcal{F}$ w.r.t. a distribution $\mathcal{D}$, if there exists a BQP (resp. BPP) algorithm A s.t. for all $f \in \mathcal{F}$ and all $x \in \mathbb{F}_2^n$, running A with access to $\mathcal{O}(\mathfrak{D}, f)$ on x can produce $f(x)$.

Theorem 3.1 ([BJ95])**.** QEX cannot exactly simulate MQ w.r.t. $\mathcal{U}\left(\mathbb{F}_2^n\right)$.

Proof. Consider two Boolean functions f and g that differ only in a single string $x' \in \mathbb{F}_2^n$, e.g., let $f(x) = 0$ for all $x \in \mathbb{F}_2^n$, and

$$g(x) = \begin{cases} 1 & , x = x', \\ 0 & , \text{otherwise.} \end{cases}$$

Consider a Boolean function h with the promise that it is either f or g. Suppose a quantum learner A makes Q queries to a QEX oracle $\mathrm{QEX}(\mathcal{U}, h)$ and wants to figure out which oracle it is querying. This task is the same as trying to discriminate between the following pair of states:

$$|\psi_f\rangle = \left(\sum_{x\in\mathbb{F}_2^n} \sqrt{\mathcal{U}(x)}\,|x, f(x)\rangle\right)^{\otimes Q} \quad , \quad |\psi_g\rangle = \left(\sum_{x\in\mathbb{F}_2^n} \sqrt{\mathcal{U}(x)}\,|x, g(x)\rangle\right)^{\otimes Q} . \tag{3.4}$$

From Fact 2.2, we know that the best success probability of distinguishing between $|\psi_f\rangle$ and $|\psi_g\rangle$ is $\frac{1}{2} + \frac{1}{2}\mathrm{D}\left(|\psi_f\rangle, |\psi_g\rangle\right)$. If this probability is greater than $1-\delta$ for any $\delta \in (0, 1/2)$, then $\langle\psi_f|\psi_g\rangle \leqslant 2\sqrt{\delta\cdot(1-\delta)}$. Therefore,

$$\langle\psi_f|\psi_g\rangle \leqslant 2\sqrt{\delta\cdot(1-\delta)} \tag{3.5}$$

$$\implies \left(\sum_{x\neq x'} \sqrt{\mathcal{U}(x)}\right)^{Q} \leqslant 2\sqrt{\delta\cdot(1-\delta)} \tag{3.6}$$

$$\implies \left(1-\mathcal{U}(x')\right)^{Q/2} \leqslant 2\sqrt{\delta\cdot(1-\delta)}. \tag{3.7}$$

Let $\varepsilon = \mathcal{U}(x') = \frac{1}{2^n}$. Then, we have,

$$(1-\varepsilon)^{Q/2} \leqslant 2\sqrt{\delta\cdot(1-\delta)} \tag{3.8}$$

$$\implies (1-\varepsilon)^{Q} \leqslant 4\delta \tag{3.9}$$

$$\implies Q\log(1-\varepsilon) \leqslant \log(4\delta) \tag{3.10}$$

$$\implies \frac{-Q\varepsilon}{1-\varepsilon} \leqslant \log(4\delta)\,. \tag{3.11}$$

The inequality in Eq. (3.11) follows from applying the inequality $\log(1+x) \geqslant x/1+x$, $\forall x > -1$ to the LHS of Eq. (3.10). Hence, we have that $Q = \Omega\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$, i.e., the number of queries is superpolynomial. Therefore, there exists no query-efficient learner A^{QEX} that can distinguish between $|\psi_f\rangle$ and $|\psi_g\rangle$ with prob $\geqslant 1-\delta$. □

Remark 3.4.1 (Unconditional separation between MQ and QEX w.r.t. uniform)**.** Even though MQ cannot be exactly simulated by QEX w.r.t. uniform, we can simulate the QEX oracle by 1 query to the MQ oracle as follows:

$$\mathrm{MEM}_f\, H^{\otimes n}|0\rangle^{\otimes n}|0\rangle = \sum_{x\in\mathbb{F}_2^n} \frac{1}{2^{n/2}}|x, f(x)\rangle \tag{3.12}$$

This shows us that at least w.r.t. the uniform distribution, MQ queries are unequivo-

cally stronger than QEX queries, and hence QREX or QAEX queries.

The proof of Theorem 3.1 shows us that if there exist two functions that differ on exactly one coordinate, it is not possible for any polynomial query quantum algorithm to distinguish the corresponding quantum states. We now formalize this notion to define the concept of approximate simulatability.

Definition 3.8 (Approximate Simulatability)**.** Let $f, g \in \mathcal{F}$ be two Boolean functions s.t. f and g only disagree on an ε-fraction of the domain, for some $\varepsilon > 0$. Let h be a Boolean function with the promise that h is either f or g. Then a class of quantum (resp. classical) example oracles $\mathcal{O}$ can (ε, δ)-simulate (or approximately simulate) MQ for a Boolean concept class $\mathcal{F}$ w.r.t. a distribution $\mathcal{D}$, if there exists a BQP (resp. BPP) algorithm A with query access to any $\mathcal{O}(\mathcal{D}, h)$, that can figure out if $h = f$ or if $h = g$, with probability at least $1 - \delta$, for all $f, g \in \mathcal{F}$ which are ε-far.

We see that for any Boolean function f, exact simulatibility (as in Definition 3.7) means being able to $(1/2^n, \delta)$-simulate f. We now prove the following theorem:

Theorem 3.2. QEX cannot approximately simulate MQ w.r.t. $\mathcal{U}\left(\mathbb{F}_2^n\right)$.

Proof. Suppose f and g are Boolean functions that differ on a polynomial sized subset $S \subset \mathbb{F}_2^n$, i.e., $|S| = \text{poly}(n)$. Therefore, $\|f - g\|_1 = \frac{1}{2^n}\sum_x |f(x) - g(x)| = \frac{|S|}{2^n} = \text{poly}(n)/2^n = \varepsilon$. Suppose a quantum learner A makes Q queries to a QEX oracle $\text{QEX}(\mathcal{U}, h)$ and wants to discriminate between the states $|\psi_f\rangle$ and $|\psi_g\rangle$ as described in Eq. (3.4). From Fact 2.2, we know that the best success probability of distinguishing between $|\psi_f\rangle$ and $|\psi_g\rangle$ is $\frac{1}{2} + \frac{1}{2}D\left(|\psi_f\rangle, |\psi_g\rangle\right)$. If this probability is greater than $1 - \delta$ for any $\delta \in (0, 1/2)$, then $\langle\psi_f|\psi_g\rangle \leqslant 2\sqrt{\delta \cdot (1-\delta)}$. Therefore,

$$\langle\psi_f|\psi_g\rangle \leqslant 2\sqrt{\delta \cdot (1-\delta)} \tag{3.13}$$

$$\implies \left(\sum_{x \notin S} \sqrt{\mathcal{U}(x)}\right)^Q \leqslant 2\sqrt{\delta \cdot (1-\delta)} \tag{3.14}$$

$$\implies \left(1 - \sum_{x \in S} \sqrt{\mathcal{U}(x)}\right)^Q \leqslant 2\sqrt{\delta \cdot (1-\delta)} \tag{3.15}$$

$$\implies (1-\varepsilon)^Q \leqslant 2\sqrt{\delta \cdot (1-\delta)} \tag{3.16}$$

$$\implies (1-\varepsilon)^{2Q} \leqslant 4\delta \tag{3.17}$$

$$\implies 2Q \log(1-\varepsilon) \leqslant \log(4\delta) \tag{3.18}$$

$$\implies \frac{-2Q\varepsilon}{1-\varepsilon} \leqslant \log(4\delta)\,. \tag{3.19}$$

The inequality in Eq. (3.19) follows from applying the inequality $\log(1+x) \geqslant x/1+x$, $\forall x > -1$ to the LHS of Eq. (3.18). Plugging in the value of ε, we see that $Q = \Omega\left(\frac{2^n}{\text{poly}(n)} \log\left(\frac{1}{\delta}\right)\right)$. Hence, there does not exist any query efficient quantum learner A^{QEX} that can distinguish between $|\psi_f\rangle$ and $|\psi_g\rangle$ with probability $\geqslant 1-\delta$. □

§ 3.5 Conclusion and Future Work

Suppose we generalize the RWQ oracle model [Bsh+03] to a Continuous-time Quantum Walk Query (CQW) oracle model. Childs et al. [Chi+03] showed that there is an exponential separation between RWQ and CQWQ when performing graph traversal over certain classes of finite graphs. It is known, however, that such speedups do not exist over $\mathbb{F}_2^n$. In this light, we ask the following open question.

Open Problem 1. Can we find a natural learning task where there is an exponential separation between PAC learning with RWQ (or local MQ) queries and PAC learning with CQW queries?

In any moment of decision, the best thing you can do is the right thing,
the next best thing is the wrong thing,
and the worst thing you can do is nothing.

THEODORE ROOSEVELT

Chapter 4

Quantum Weak Learners for Decision Trees

Abstract

The agnostic setting is the hardest generalization of the PAC model since it is akin to learning with adversarial noise. In this chapter, we design the first quantum weak learning algorithm for polynomial-sized decision trees in the agnostic setting w.r.t. uniform marginals over the instance space. Unlike existing weak learners for polynomial-sized decision trees, our weak learners only make queries to quantum example oracles, which are weaker than Membership Query (MQ) oracles w.r.t. uniform distribution, as we proved earlier in Chapter 3. We also show how to design weak quantum learners for poly-sized decision trees in the bounded-noise and noiseless settings without access to MQ oracles. A significant part of this chapter is based on the following publication:

- Sagnik Chatterjee et al. "Efficient Quantum Agnostic Improper Learning of Decision Trees". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta et al. Vol. 238. Proceedings of Machine Learning Research. PMLR, May 2024, pp. 514–522. URL: https://proceedings.mlr.press/v238/chatterjee24a.html

Contents

4.1 Introduction 56
4.2 Weak and Strong Learning 56
4.2.1 Weak and Strong Learning in the Realizable setting 56
4.2.2 Weak and Strong Learning in the Agnostic setting 57
4.3 Designing Weak Learners 57
4.3.1 Bridging Fourier-Analysis and Learning 57
4.3.2 Weak Learners via Spectral Concentration 58

4.4 Decision Trees 61
4.5 Weak Learning Decision Trees w/o MQ 62
4.5.1 Realizable Setting 63
4.5.2 The RCN setting 64
4.5.3 The Agnostic setting with Uniform Marginals 64
4.6 Discussion and Conclusion 69

§ 4.1 Introduction

Even though very strong learners are ultimately desired, it might not always be prudent to use highly accurate learners for many reasons - longer running times, overfitting, and lack of model interpretability/explainability.[1] On the other hand, many well-known off-the-shelf learning algorithms such as decision stumps, naïve Bayes over a single variable, and clustering algorithms with a fixed number of clusters, are easy to create and use, and are extremely interpretable, but with a loss in accuracy. The latter type of learners are commonly referred to as *weak learners*. In this chapter, we focus on constructing weak learners for Boolean function classes with respect to the Benchmark class of polynomial-sized decision trees.

▷ Structure and Organization.

In Section 4.2, we define the notions of *weak* and *strong* PAC learners. Following this, we introduce Fourier-analytic tools for constructing *weak* PAC learners in Section 4.3. We then formally introduce the concept class of Decision trees and discuss a few important properties in Section 4.4. Finally, in Section 4.5, we show how to combine all the concepts introduced thus far to construct weak quantum learners for decision trees in various noise models w.r.t. uniform distributions without invoking MQ oracles.

§ 4.2 Weak and Strong Learning

§ 4.2.1 Weak and Strong Learning in the Realizable setting

Before moving on to the definitions of weak and strong learning, we introduce the notion of the concept of **bias** in PAC learners. The **bias** of an (ε, δ)-PAC learner A is denoted by $\gamma = 1/2 - \varepsilon$ for some $\varepsilon \in (0, 1/2)$; i.e., the advantage of learner A vs. a learning algorithm randomly guessing the labels. We now define the notions of weak and strong learnability.

[1] We refer the reader to [Mol20] for a concrete discussion on model interpretability / explainability.

Definition 4.1 (γ-Weak Learner)**.** For any $k > 0$ and $\gamma = 1/\text{poly}(n)$, a learner A is a γ-weak learner for concept class $\mathcal{C}$ if it is a $\left(\frac{1}{2} - \gamma, \delta\right)$ PAC learner for concept class $\mathcal{C}$. Equivalently, a γ-weak learner for $\mathcal{C}$ has $\text{corr}_{\mathfrak{D},c}(h) = O(\gamma) = 1/\text{poly}(n)$.

Definition 4.2 (Strong Learner)**.** A learner A is a strong learner for concept class $\mathcal{C}$ if it is a $\left(\leqslant \frac{1}{3}, \delta\right)$ PAC learner for concept class $\mathcal{C}$. Equivalently, a strong learner A outputs h s.t. $\text{corr}_{\mathfrak{D},c}(h) \geqslant 2/3$, with high probability.

In Chapter 5, we shall see how to convert a γ-Weak Learner into a Strong Learner.

§ 4.2.2 Weak and Strong Learning in the Agnostic setting

In the agnostic setting, the optimal concept c in the benchmark class $\mathcal{C}$ can have errors worse than random guessing with respect to the true labeling of the instances, i.e., $\text{opterr}_{\mathfrak{D}}(\mathcal{C}) < 1/2$. Hence, it is easier to define the notion of weak and strong learning in the Agnostic setting using correlation w.r.t. the optimal concept in $\mathcal{C}$.

Definition 4.3 ((η, κ, δ)-Weak agnostic learner)**.** For any $\kappa = 1/\text{poly}(n)$ and $\delta > 0$, a learner A is a (η, κ, δ)-weak agnostic PAC learner for $\mathcal{C}$, if for any choice of distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$, A outputs hypothesis h s.t.

$$\Pr\left[\text{corr}_{\mathfrak{D},c}(h) \geqslant \eta \cdot \text{optcorr}_{\mathfrak{D}}(\mathcal{C}) - \kappa\right] \geqslant 1 - \delta.$$

Definition 4.3 can be interpreted as follows: A weak agnostic learner can recover some η fraction of the bias of the optimal learner. Note that since $\text{optcorr}_{\mathfrak{D}}(\mathcal{C}) := 2\gamma_{\text{opt}}$, we have, $\gamma \geqslant \eta \cdot \gamma_{\text{opt}} - 1/\text{poly}(n)$, where γ is the bias of the learner. The best any agnostic boosting algorithm can do is convert a (η, κ, δ)-Weak agnostic learner into a β-optimal (ε, δ)-agnostic PAC learning efficiently. We shall discuss how to do this in Chapter 6.

§ 4.3 Designing Weak Learners

§ 4.3.1 Bridging Fourier-Analysis and Learning

A Boolean function f is **unbiased** if $\mathop{\mathbb{E}}_{\mathfrak{D}}[f] = 0$. A Boolean function f is **unbiased** w.r.t. another Boolean function g if $\mathop{\mathbb{E}}_{\mathfrak{D}}[f \cdot g] = 0$ over the distribution $\mathfrak{D}$. As noted previously in Section 2.4.3.3 and in Definition 2.1, the notions of **bias** and **correlation** of functions are important in the context of PAC learning. We make the following observation here that connects these dots explicitly for Boolean functions.

Observation 4.3.1. If f is unbiased w.r.t. g, then f is a terrible PAC learner for g and vice-versa. If $\mathop{\mathbb{E}}_{\mathfrak{D}}[f] = \mathrm{corr}_{\mathfrak{D}}(f,1) \geqslant 1/\mathrm{poly}(n)$, then the constant parity monomial $\chi_\varnothing$ (or its negation) is a weak learner for f.

Hence, given some information about the Fourier spectrum of f, we can explicitly construct a weak learner for f. We formalize these notions by defining the concepts of Fourier weight and Spectral concentration of Boolean functions.

Definition 4.4 (Fourier Weight at degree k)**.** The **Fourier weight** of a Boolean function f at **degree** k is defined as the sum of squared Fourier coefficients of subsets of Hamming weight exactly equal to k, i.e., $W^{=k}(f) = \sum_{S\subseteq[n],|S|=k} \hat{f}(S)^2$. We can similarly define the quantities $W^{\leqslant k}(f)$ and $W^{\geqslant k}(f)$.

Definition 4.5 (Spectral concentration up to degree k)**.** A Boolean function f is ε-spectrally concentrated on a set $\mathcal{A} \subseteq 2^{[n]}$ if $\sum_{S\subseteq[n],S\notin\mathcal{A}} \hat{f}(S)^2 \leqslant \varepsilon$. Therefore, a Boolean function f is ε-spectrally concentrated on **degree** $\leqslant k$, if $W^{>k}(f) \leqslant \varepsilon$.

Linial et al. [LMN93] proved the following important result that immediately bridges Fourier analysis and Learning theory.

Lemma 4.1 (Low Degree Learning)**.** If a Boolean function f is ε-concentrated on a set $\mathcal{A}$, and we know $\mathcal{A}$ explicitly, then there exists a learning algorithm A that can (ε,δ)-PAC learn f in time poly $(|\mathcal{A}|, n, 1/\varepsilon, 1/\delta)$ with query access to a EX $(\mathcal{U}, f)$ oracle. If f is ε-concentrated up to degree k, then A can learn f in time poly $\left(n^k, 1/\varepsilon, 1/\delta\right)$ with query access to a EX $(\mathcal{U}, f)$ oracle.

Remark 4.3.1. If k is not a constant, then the learner in Lemma 4.1 does not yield an *efficient* PAC learner for f.

§ 4.3.2 Weak Learners via Spectral Concentration

One shortcoming of Lemma 4.1 is the requirement that we have explicit knowledge of the subsets of literals on which the Boolean function is spectrally concentrated. We can get around this requirement by upgrading our choice of oracle from the EX $(\mathcal{U}, f)$ oracle to the MEM_f oracle. To this end, we first present the following lemmas (due to [GL89; KM91]), where we show how to learn functions that are spectrally concentrated up to a *specified degree* (unlike Lemma 4.1 where the function is spectrally concentrated on a specific subset).

Lemma 4.2 (The Goldreich-Levin theorem)**.** Consider a $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$, s.t. $\exists v \in \mathbb{F}_2^n, \gamma, \delta > 0$ and $\Pr_{x\sim\mathcal{U}(\mathbb{F}_2^n)}[f(x) = \langle x, v\rangle] \geqslant \frac{1}{2} + \gamma$. Given access to a MEM_f oracle we can find all such v with probability at least $1-\delta$ in time poly $\left(n, \frac{1}{\gamma}, \frac{1}{\delta}\right)$.

Observe that $\langle x, v\rangle$ is the parity monomial $\chi_v(x)$. Therefore, from Definition 2.1,

$$\Pr_{x\sim\mathcal{U}(\mathbb{F}_2^n)}[f(x) = \langle x, v\rangle] \geqslant \frac{1}{2} + \gamma \iff \hat{f}(v) \geqslant \gamma. \tag{4.1}$$

The Goldreich-Levin algorithm follows from Lemma 4.2 and Eq. (4.1).

Lemma 4.3 (The Goldreich-Levin algorithm)**.** Let $\gamma, \delta > 0$, and $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$. Given access to a MEM_f oracle, there exists an algorithm that outputs a list $\mathcal{L}$ in time poly $\left(n, \frac{1}{\gamma}, \frac{1}{\delta}\right)$ s.t.

- If $\left|\hat{f}(S)\right| \geqslant \gamma$ then $S \in \mathcal{L}$ w.p. $\geqslant 1-\delta$.
- If $S \in \mathcal{L}$ then $\left|\hat{f}(S)\right| \geqslant \frac{\gamma}{2}$ w.p. $\geqslant 1-\delta$.

Observation 4.3.2. Note that $|\mathcal{L}| = O\left(\frac{1}{\gamma^2}\right)$ in Lemma 4.3.

Lemma 4.4 (The Kushilevitz-Mansour theorem)**.** Let $\varepsilon, \delta > 0$, and $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$. If f is ε-concentrated up to degree k, $\exists$ a learning algorithm A with query access to a MEM_f oracle that (ε, δ)-PAC learns f in time poly $\left(n^k, 1/\varepsilon, 1/\delta\right)$.

Proof. We know that f is ε-concentrated up to degree k. Therefore, f is ε-concentrated on a subset $\mathcal{A}$ s.t. $|\mathcal{A}| \leqslant O\left(\binom{n}{k}\right) \leqslant O\left(n^k\right)$. Fix a threshold $\tau > 0$, and use Lemma 4.3 to learn all $S \subseteq [n]$ s.t. $\left|\hat{f}(S)\right| > \tau$. Let this set be $\mathcal{L}$.

$$\sum_{S\notin\mathcal{L}} \hat{f}(S)^2 = \sum_{\substack{S\notin\mathcal{L}\\ S\notin\mathcal{A}}} \hat{f}(S)^2 + \sum_{\substack{S\notin\mathcal{L}\\ S\in\mathcal{A}}} \hat{f}(S)^2 \tag{4.2}$$

$$\implies \sum_{S\notin\mathcal{L}} \hat{f}(S)^2 \leqslant \varepsilon + |\mathcal{A}| \cdot \tau^2. \tag{4.3}$$

Setting $\tau = \sqrt{\varepsilon/|\mathcal{A}|}$, we see that f is 2ε-concentrated on the list $\mathcal{L}$ produced by Lemma 4.3. Hence, we can construct an (ε, δ)-PAC learner for f directly from $\mathcal{L}$ in time $O\left(|\mathcal{L}|, \frac{1}{\varepsilon}, \frac{1}{\delta}\right) = O\left(\frac{1}{\tau^2}, \frac{1}{\varepsilon}, \frac{1}{\delta}\right) = O\left(n^k, \frac{1}{\varepsilon}, \frac{1}{\delta}\right)$. The second inequality follows from Observation 4.3.2. □

There are two important corollaries of The Kushilevitz-Mansour theorem. The first result allows us to improve upon Lemma 4.1 by learning a function that is spectrally concentrated on a subset S, without explicit knowledge of S.

Corollary 4.1 (Learning from ε-concentrated subset of parity monomials)**.** Let $\varepsilon, \delta > 0$, and $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$. If f is ε-concentrated on a subset S, $\exists$ a learning algorithm A with query access to a MEM_f oracle that (ε, δ)-PAC learns f in time poly $(|S|, 1/\varepsilon, 1/\delta)$.

The proof follows a similar line of argument as in Lemma 4.4. The next corollary helps us define PAC learners from parity monomials that have large Fourier coefficients. Let f be a Boolean function. Then, the ℓ_1 norm of $\text{FS}(f)$ is defined as

$$\|\text{FS}(f)\|_1 = \sum_{S \subseteq [n]} \left|\hat{f}(S)\right|. \tag{4.4}$$

Corollary 4.2 (Learning from parity monomials with large Fourier coefficients)**.** Let $\varepsilon, \delta > 0$, and $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$. $\exists$ a learning algorithm A with query access to a MEM_f oracle that (ε, δ)-PAC learns f in time poly $(\|\text{FS}(f)\|_1, 1/\varepsilon, 1/\delta)$.

Proof. For any Boolean f consider the set $\mathcal{L}_1 = \left\{S \mid \left|\hat{f}(S)\right| \geqslant \frac{\varepsilon}{\|\text{FS}(f)\|_1},\ S \subseteq [n]\right\}$. We can see that f is ε-concentrated on $\mathcal{L}_1$ since

$$\sum_{S \notin \mathcal{L}_1} \hat{f}(S)^2 = \sum_{S \notin \mathcal{L}_1} \left|\hat{f}(S)\right| \cdot \left|\hat{f}(S)\right| \leqslant \sum_{S \notin \mathcal{L}_1} \left|\hat{f}(S)\right|_{\max} \cdot \left|\hat{f}(S)\right| \leqslant \sum_{S \notin \mathcal{L}_1} \frac{\varepsilon}{\|\text{FS}(f)\|_1} \cdot \left|\hat{f}(S)\right| \tag{4.5}$$

$$= \frac{\varepsilon}{\|\text{FS}(f)\|_1} \sum_{S \notin \mathcal{L}} \left|\hat{f}(S)\right| \leqslant \frac{\varepsilon}{\|\text{FS}(f)\|_1} \cdot \|\text{FS}(f)\|_1 \leqslant \varepsilon. \tag{4.6}$$

From Observation 4.3.2, we have that $|\mathcal{L}_1| = \text{poly}\left(\|\text{FS}(f)\|_1, \frac{1}{\varepsilon}\right)$. We now get the desired bound by applying Corollary 4.1. □

Observation 4.3.3. Corollary 4.2 tells us that if the ℓ_1 norm of any Boolean function is polynomially bounded, then we can construct a PAC learner for f using queries to an MQ oracle.

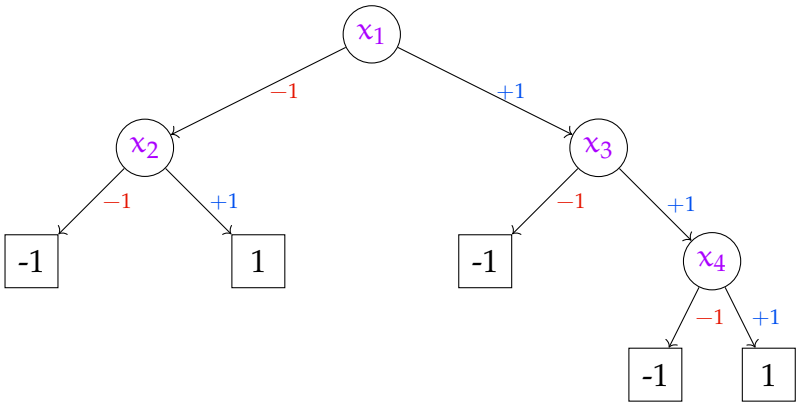
We now ask the following questions.

Question 4.1. Lemmas 4.1 and 4.4 and Corrolaries 4.1 and 4.2 give us an array of Fourier-analytic tools to construct weak learners for Boolean functions that have bounded ℓ_1 norm in the realizable setting. How can we adapt these techniques for learning Boolean functions that have bounded ℓ_1 norm under bounded and unbounded label noise?

To answer the above question, we now show how to design weak PAC learners for Boolean classes having a Fourier spectrum with bounded ℓ_1 norm using QEX oracles, ε-Biased oracles, and Strongly Biased oracles.

x_4	x_3	x_2	x_1	$f(x)$
$+1$	$+1$	$+1$	$+1$	$+1$
$+1$	$+1$	$+1$	-1	$+1$
$+1$	$+1$	-1	$+1$	$+1$
$+1$	$+1$	-1	-1	-1
$+1$	-1	$+1$	$+1$	-1
$+1$	-1	$+1$	-1	$+1$
$+1$	-1	-1	$+1$	-1
$+1$	-1	-1	-1	-1
-1	$+1$	$+1$	$+1$	-1
-1	$+1$	$+1$	-1	$+1$
-1	$+1$	-1	$+1$	-1
-1	$+1$	-1	-1	-1
-1	-1	$+1$	$+1$	-1
-1	-1	$+1$	-1	$+1$
-1	-1	-1	$+1$	-1
-1	-1	-1	-1	-1

(a) A Boolean function f



(b) Decision Tree representation of f

Figure 4.1: A decision tree computing a Boolean function on 4 bits.

§ 4.4 Decision Trees

Decision trees are an indispensable tool for the machine learning community as they are useful for both classification and regression tasks, and allow for non-linear decision boundaries and handling mixed-type data. Decision trees are also one of the flag bearers of interpretable and human-explainable machine learning and form a crucial part of ensemble models like random forests. On the other hand, decision trees play an important role in various subfields of theoretical computer science, such as query complexity, circuit complexity, analysis of Boolean functions, etc.

▷ Decision Trees on Boolean functions.

A decision tree for a Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is a binary tree such that every path from the root to any leaf corresponds to a truth table row of f. The leaf nodes are labeled by $\{0, 1\}$ or $\{\pm 1\}$, the internal nodes are labeled by the literals: variables x_i (or their negations), and every internal node has two outgoing edges labeled true and false. See Figs. 2.1 and 4.1 for examples of decision trees.

▷ Decision Trees on Relations over the Boolean Hypercube.

A decision tree is a binary tree that computes a relation $\mathcal{R} \subseteq \mathbb{F}_2^n \times \mathcal{G}$, s.t., the leaf nodes are labeled according to values in $\mathcal{G}$, while each internal node is labeled by a variable x_i (or its negation) and has two outgoing edges, labeled true and false. On input $x \in \mathbb{F}_2^n$, the tree evaluates the node x_i and follows the outgoing edge corresponding to

the truth value of x_i. The output value at the leaf $g \in \mathcal{G}$ must satisfy the relation, i.e. $(x, g) \in \mathcal{R}$. We note here that a deterministic decision tree in fact computes a function, since each input reaches exactly one leaf on the computation path of the tree. We also clarify that for the purposes of this thesis, $\mathcal{R} \subseteq \mathbb{F}_2^n \times \mathbb{F}_2$.

▷ Parity Decision Trees.

A parity decision tree is exactly like a decision tree, but each internal node is labeled by a subset $S \in [n]$ instead of a literal. The tree evaluates the node labeled by S, and follows the outgoing edge corresponding to the truth value of the parity function over the subset, i.e., it takes the edge labeled by $\oplus_{i \in S} x_i$. It can be easily shown that the parity-decision tree depth of representing any arbitrary Boolean function f is upper bounded by the decision tree depth of f.

▷ Complexity measures and Hardness of Decision trees.

The number of leaves (or the number of internal nodes) of the decision tree is known as its **size** while the longest path from the root of the decision tree to any leaf node is known as the **depth** of the decision tree.

Any arbitrary Boolean function f can be represented as a decision tree. Since the decision tree representation of a Boolean function is not unique, computationally, we would prefer to represent f in terms of the "smallest" decision tree. However, Hyafil and Rivest [HR76] showed that constructing the optimal decision tree corresponding to an arbitrary Boolean function is an NP-complete problem.

▷ Fourier Properties of Decision trees.

We end this section by stating an important result by Kushilevitz and Mansour [KM91] that gives a bound on the ℓ_1 bound of decision trees of size t. This property of decision trees becomes especially important for designing weak learners, especially in light of Corollary 4.2.

> **Lemma 4.5** (L1 bound)**.** Let $t > 0$. If a Boolean function f is computed by a decision tree with size t, then $\|\mathrm{FS}(f)\|_1 \leqslant t$. Alternatively, if a Boolean function f is computed by a decision tree with at most $O(t)$ nodes, then $\|\mathrm{FS}(f)\|_1 = O(t)$.

Note that Corollary 4.2 and Lemma 4.5 straightforwardly yield a polynomial time learning algorithm for polynomial-sized decision trees using MQ queries.

§ 4.5 Weak Learning Decision Trees w/o MQ

§ 4.5.1 Realizable Setting

We first show how to design a quantum weak learner for polynomial-sized decision trees with query access to the QEX oracle. We start by identifying the form of a realizable weak learner and then show how to extract the required weak learner from a QEX $(\mathcal{U}, f)$ oracle.

Claim 4.3. Let f be a Boolean function, s.t. $\|\mathrm{FS}(f)\|_1 \leqslant t, t > 0$. Suppose $S \subseteq [n]$ is a subset of literals s.t. $\hat{f}(S) \geqslant \frac{1}{t}$. Then, the parity monomial χ_S corresponding to S is a γ-Weak Learner for f.

Proof. Recall from Section 2.3.1 that $\mathrm{corr}_{\mathcal{U}}(f, \chi_S) = \langle f, \chi_S \rangle = \hat{f}(S) \geqslant \frac{1}{t}$. Then, χ_S is a γ-Weak Learner for f for $\gamma = \frac{1}{t}$. □

Claim 4.4. The parity monomial corresponding to the mode of the Fourier spectrum of a Boolean function f s.t $\|\mathrm{FS}(f)\|_1 \leqslant t, t > 0$, is a $\frac{1}{t}$-weak learner for f.

Proof. Let S^* be the mode of $\mathrm{FS}(f)$, i.e., $\left|\hat{f}(S^*)\right| \geqslant \left|\hat{f}(S)\right|, \forall S \subseteq [n]$. We first show that if $\|\mathrm{FS}(f)\| \leqslant t$ for $t > 0$, then $\left|\hat{f}(S^*)\right| \geqslant \frac{1}{t}$. We start by recalling Parseval's Theorem.

$$\sum_{S \subseteq [n]} \left|\hat{f}(S)\right| \cdot \left|\hat{f}(S)\right| = 1 \tag{4.7}$$

$$\implies \sum_{S \subseteq [n]} \left|\hat{f}(S)\right| \cdot \left|\hat{f}(S^*)\right| \geqslant 1 \tag{4.8}$$

$$\implies \left|\hat{f}(S^*)\right| \sum_{S \subseteq [n]} \left|\hat{f}(S)\right| \geqslant 1. \tag{4.9}$$

It follows from the final equation that $\left|\hat{f}(S^*)\right| \geqslant \frac{1}{t}$.Hence, from Claim 4.3 we have that χ_{S^*} is a $\frac{1}{t}$-weak learner for f. □

Theorem 4.5 (Realizable Quantum Weak Learner for Polynomial sized decision trees)**.** Let $t, \varepsilon, \delta > 0, \gamma = \frac{1}{t} - \varepsilon$, and f be an n-bit Boolean function s.t. $\|\mathrm{FS}(f)\|_1 \leqslant t$. Then there exists a quantum learning algorithm that returns a γ-Weak Learner learner for f with $O\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ queries to a QEX $(\mathcal{U}, f)$ oracle.

Proof. Using Lemma 1 of Bshouty and Jackson [BJ95], we can obtain an approximate Fourier sampling state using 1 call to the QEX $(\mathcal{U}, f)$ oracle as follows:

$$\frac{1}{\sqrt{2}} \sum_{S \subseteq [n]} \hat{f}(S) |S\rangle |1\rangle + \frac{1}{\sqrt{2}} |0 \ldots 0\rangle |0\rangle . \tag{4.10}$$

Applying Amplitude Amplification to the state obtained in Eq. (4.10), we can now obtain the Fourier sampling state $|\psi\rangle = \sum_{S\subseteq[n]} \hat{f}(S)|S\rangle$ using an additional $\Theta(1)$ queries to the QEX $(\mathcal{U}, f)$ oracle. Note that $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ copies of $|\psi\rangle$ allows us to estimate the mode S_{mode} of a Boolean function f with error at most ε and with probability $\geqslant 1-\delta$. From Claim 4.4, this gives us a realizable weak learner for Decision trees of size t. □

§ 4.5.2 The RCN setting

Construction of the weak learner in the realizable setting follows from the construction of the Fourier sampling state and using the argument that the mode of the Fourier spectrum of a Boolean function with a bounded ℓ_1 norm is a weak learner. In the RCN setting, the main challenge is constructing the Fourier sampling state $\sum_{S\subseteq[n]} \hat{f}(S)|S\rangle$ from the QREX oracle which is an ε-biased oracle. At this point, we recall the following result by Iwama et al. [IRY05].

Lemma 3.2 (Simulatability of Unbiased Oracles by ε-Biased Oracles)**.** For any $O(T)$ query quantum algorithm that solves a problem with high probability with query access to an unbiased oracle, there exists an $O\left(\frac{T}{\epsilon}\right)$ query quantum algorithm that solves the same problem with high probability with query access to an ϵ-biased oracle.

Combining Lemma 3.2 and Theorem 4.5 gives us the following theorem directly.

Theorem 4.6 (Quantum Weak Learner for Polynomial sized decision trees in the RCN setting)**.** Let $t, \varepsilon, \delta > 0, \gamma = \frac{1}{t}$, and f be an n-bit Boolean function s.t. $\|\text{FS}(f)\|_1 \leqslant t$. Then there exists a quantum learning algorithm that returns a γ-Weak Learner learner for f with $O\left(\frac{1}{\varepsilon^3}\log\frac{1}{\delta}\right)$ queries to a $\text{QREX}^{\varepsilon}(\mathcal{U}, f)$ oracle.

§ 4.5.3 The Agnostic setting with Uniform Marginals

In the agnostic setting, unfortunately, most of the previous techniques break down. Here, the task is to learn an unknown concept $f(x)$ (represented by a decision tree) given a QAEX oracle. We recall that

- The QAEX oracle is not a strongly biased oracle.
- Even if we somehow have access to Strongly Biased oracles, Lemma 3.2 still does not hold.

Hence, the approach taken in Sections 4.5.1 and 4.5.2 would not work since it is not entirely clear how to obtain a Fourier sampling state from the QAEX oracle. We somehow have to address both of the challenges listed above.

We start by querying the QAEX oracle to obtain copies of the state $\sum_{x,y}\sqrt{\mathcal{D}_{x,y}}\left|x\right\rangle\left|y\right\rangle$. Assuming a uniform marginal distribution over $\mathcal{X}$, this is equivalent to the quantum state $\frac{1}{\sqrt{2^n}}\sum_x\left|x\right\rangle\left(\sum_y\alpha_{y|x}\left|y\right\rangle\right)$. We now state an important quantum subroutine[2]:

Lemma 4.6 (Multidistribution Amplitude Estimation)**.** Given an oracle O that acts as $O\left|0\right\rangle = \sum_x\eta_x\left|x\right\rangle\left(\alpha_{0|x}\left|0\right\rangle+\alpha_{1|x}\left|1\right\rangle\right)$, there exists an algorithm to output the quantum state $\sum_x\eta_x\left|x\right\rangle\left(\alpha_{0|x}\left|0\right\rangle+\alpha_{1|x}\left|1\right\rangle\right)\left|\widetilde{\alpha}_{1|x}\right\rangle$ in $O\left(\delta/\varepsilon\right)$ queries with probability $\geqslant 1-\delta$, such that $\left|\widetilde{\alpha}_{1|x}-\alpha_{1|x}\right|\leqslant\varepsilon$, for any $\varepsilon,\delta>0$.

Append $k=O\left(\log\frac{1}{\gamma}\right)$ ancillas and make k independent estimations using Lemma 4.6 (M.A.E.) with parameters $(\varepsilon,\delta=1-8/\pi^2)$ to obtain the state

$$\frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{F}_2^n}\left|x\right\rangle\left(\sum_{y\in\{0,1\}}\alpha_{y|x}\left|y\right\rangle\right)\left(\beta_{gx}\left|\tilde{\alpha}_{1|x}\right\rangle+\beta_{bx}\left|\mathrm{Err}\right\rangle\right)^{\otimes k}. \tag{4.11}$$

We note here that we want to set the value of the predicted label $\tilde{f}(x)$ in the third register as the label with the larger conditional probability.

On each of the $k=O\left(\log\frac{1}{\gamma}\right)$ registers, let $\tilde{f}(x)=\mathbb{I}\left[\tilde{\alpha}_{1|x}>\frac{1}{\sqrt{2}}\right]$, i.e., perform thresholding, to obtain the state

$$\frac{1}{\sqrt{2^n}}\sum_x\left|x\right\rangle\left(\hat{\beta}_{gx}\left|\psi'(x)\right\rangle\left|\tilde{f}(x)\right\rangle+\hat{\beta}_{bx}\left|\mathrm{Err}\right\rangle\right)^{\otimes k}, \tag{4.12}$$

where $\left|\psi'(x)\right\rangle=\beta_{gx}\left(\sum_{y\in\{0,1\}}\alpha_{y|x}\left|y\right\rangle\right)\left|\tilde{\alpha}_{1|x}\right\rangle$.

Hence, we have converted the QAEX oracle into a strongly-biased oracle. Denote the unitary that maps $\left|0,0\right\rangle$ to the state in Eq. (4.12) as the strongly-biased oracle $O_f^{\varepsilon_x}$. Perform majority among the $O\left(\log\frac{1}{\gamma}\right)$ copies of $\left|\tilde{f}(x)\right\rangle$ to obtain the state

$$\frac{1}{\sqrt{2^n}}\sum_x\left|x\right\rangle\left|\xi(x)\right\rangle\left|f^*(x)\right\rangle. \tag{4.13}$$

The majority prediction $f^*(x)$ is very similar to the notion of a Bayes optimal predictor that we define below.

Definition 4.6 (Bayes Optimal Predictor)**.** The Bayes predictor $f_{\mathcal{B}}$ is the optimal

[2] See Theorem 4 of [BS22] for a detailed proof.

predictor on a joint distribution $\mathcal{D}$ over $\mathbb{F}_2^n \times \mathbb{F}_2$, and defined as

$$f_{\mathcal{B}}(x) := \underset{y \in \mathbb{F}_2}{\operatorname{argmax}} \Pr_{\mathcal{D}}[y|x] \qquad , \forall x \in \mathbb{F}_2^n.$$

Denote the unitary that maps $|0,0\rangle$ to the state in Eq. (4.13) as the Bayes-approximate oracle O_{f^*}. We note that O_{f^*} is a strongly-biased oracle. We now state the following result proved by Chatterjee et al. [CSB24]:

Lemma 4.7. Given an oracle of the form O_{f^*}, a threshold parameter $\tau > 0$, an accuracy parameter $\epsilon > 0$, and an error parameter $\delta > 0$, there exists a quantum algorithm that performs $O\left(\frac{n}{\epsilon^2\tau}\log\left(\frac{\delta\tau^2}{n}\right)\right)$ queries to O_{f^*} and either outputs a subset $\tilde{S}$ such that $\widehat{f^*}(\tilde{S}) \geqslant \tau - \epsilon$, or indicates if $\nexists \tilde{S}$, s.t. $\widehat{f^*}(\tilde{S}) \geqslant \tau$, w.p. $\geqslant 1-\delta$.

The algorithm in Lemma 4.7 is called the Quantum Goldreich Levin (QGL) algorithm[3] since it is a quantum extension of the classical Goldriech-Levin algorithm (see Lemma 4.3). The QGL algorithm tries to approximate a decision tree with a monomial, and it's operations are motivated by the classical GL algorithm. The technical difficulty was to generalize it to take as input a strongly biased oracle instead of an (error-free) oracle for a Boolean function and further enhance it to contain three kinds of errors: (a) errors from the biased oracle, (b) errors arising from amplitude estimation, and (c) errors from amplitude amplification (the state that we will amplify may contain false positives arising due to the first two errors, and those will now be incorrectly amplified).

Once we know how to construct the oracle O_{f^*}, we perform a binary search over the intervals of size $\epsilon/16$ on $(0,1]$, to find the largest threshold τ such that the QGL algorithm outputs a tuple $(l, \tilde{S})$ with $l = 1$. The search terminates if $\tau < 1/t$. If the above subroutine returns a subset S s.t. $\hat{h}(S) \geqslant \tau - \varepsilon \geqslant \frac{1}{t} - \varepsilon$, then the parity monomial corresponding to S is our desired weak learner. See Algorithm 4.1 for a consolidated view of the entire algorithm thus far.

We now state the following intermediate claims.

Claim 4.7. Algorithm 4.1 performs $\tilde{O}\left(\frac{nt}{\varepsilon^3} \cdot \log\frac{1}{\varepsilon}\right)$ queries to QAEX using the QGL algorithm (Lemma 4.7) where $\varepsilon > 0$ is the accuracy parameter. The time complexity for Algorithm 4.1 is the same as its query complexity with a logarithmic overhead.

Proof. To compute the query complexity, note that we make $O\left(\frac{n}{\varepsilon^2\tau}\log\left(\frac{\tau^2}{n}\right) \cdot \frac{1}{\varepsilon}\right)$ queries to the QAEX oracle whenever we invoke the QGL algorithm in Algorithm 4.1, with

[3] See Algorithm 3 of [CSB24].

Algorithm 4.1: Weak Quantum Agnostic Learner

Input: The QAEX oracle, threshold t.

Initialize: Set $\delta = 1/10$, $\gamma \leqslant \min\left\{\delta/4nt^2, \varepsilon^2/8\right\}$, $k = \left\lceil \log_2 \frac{1}{\varepsilon} \right\rceil + 1$, gap $g = \frac{1}{8}\left(\varepsilon - \frac{1}{2^k}\right)$, threshold $\tau = \frac{1}{2}$.

Output: A $(m, 1/t, \varepsilon)$ WL for size-t decision trees.

1 Query the QAEX oracle to obtain the state $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle \left(\sum_y \alpha_{y|x} |y\rangle\right)$.

2 Perform ℓ independent estimations using Lemma 4.6 with parameters $(\varepsilon, 1 - 8/\pi^2)$ conditioned on the second register to obtain $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle \left(\sum_y \alpha_{y|x} |y\rangle\right) \left(\beta_{gx} \left|\tilde{\alpha}_{1|x}\right\rangle + \beta_{bx} |\mathrm{Err}\rangle\right)^{\otimes \ell}$. // Let $\ell = O(\log 1/\gamma)$.

3 On each of the ℓ registers, perform thresholding to obtain the state in Eq. (4.12).

4 Perform majority on $|h(x)\rangle$ registers over all ℓ copies to obtain the state in Eq. (4.13). // Let O_{f^*} be the combined unitary from steps 1 to 4.

5 **for** $i = 1 \ldots k$ **do**

6 Invoke $\mathrm{QGL}(O_{f^*}, n, \tau, g, \frac{\delta}{k}) \to (l, \tilde{S})$. // Here, $\ell = 0$ indicates failure to find a monomial whose coefficient exceeds the threshold parameter.

7 **If** $l = 0$, set $\tau = \tau + \frac{1}{2^{i+1}}$. **Else** set $\tau = \tau - \frac{1}{2^{i+1}}$.

8 **If** $\frac{1}{2^{i+1}} < g/2$ or $\tau \leqslant \frac{1}{t}$ return $\tilde{S}$.

9 Return the parity monomial $\chi_{\tilde{S}}$.

threshold $\tau \geqslant \frac{1}{t}$. The extra $\frac{1}{\varepsilon}$ factor comes from the fact that we need to invoke the Multidistribution Amplitude Estimation subroutine every time we query the QAEX oracle. We invoke the QGL algorithm at most $O\left(\log \frac{1}{\varepsilon}\right)$ many times. Setting $\delta = \frac{1}{10}$. Combining the steps we have the final query complexity. We need an additional logarithmic overhead in running time to perform majority and thresholding operations to obtain the state in Eqs. (4.12) and (4.13). □

Claim 4.8. Let $\varepsilon > 0$ be the accuracy parameter of Algorithm 4.1. Then, Algorithm 4.1 produces $\tilde{S}$ s.t. $\left|\mathrm{corr}_{\mathcal{D},\widehat{f^*}}\left(\chi_{\tilde{S}}\right) - \max_S \mathrm{corr}_{\mathcal{D}}\left(\chi_S\right)\right| \leqslant \varepsilon$, w.h.p.

Proof. We know from an averaging argument over the Fourier spectrum of decision trees with t leaves, $\max_S \mathrm{corr}_{\mathcal{D}}\left(\chi_S\right) \geqslant \frac{1}{t}$. Setting $\tau = 1/t$ in Lemma 4.7 gives us $\widehat{f^*}(S) = \mathrm{corr}_{\mathcal{D}}\left(\chi_{\tilde{S}}, f^*\right) \geqslant 1/t - \varepsilon$. Therefore, we have

$$\max_S \mathrm{corr}_{\mathcal{D}}\left(\chi_S\right) - \mathrm{corr}_{\mathcal{D}}\left(\chi_{\tilde{S}}, f^*\right) \geqslant 1/t - \mathrm{corr}_{\mathcal{D}}\left(\chi_{\tilde{S}}, f^*\right) \tag{4.14}$$

$$\implies \max_S \mathrm{corr}_{\mathcal{D}}\left(\chi_S\right) - \mathrm{corr}_{\mathcal{D}}\left(\chi_{\tilde{S}}, f^*\right) \geqslant \varepsilon. \tag{4.15}$$

On the other hand,

$$\implies \operatorname{corr}_{\mathfrak{D}}\left(\chi_{\tilde{S}}, f^*\right) - \max_S \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right) \geqslant 1/t - \varepsilon - \max_S \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right) \tag{4.16}$$

$$\implies \operatorname{corr}_{\mathfrak{D}}\left(\chi_{\tilde{S}}, f^*\right) - \max_S \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right) \geqslant -\varepsilon \tag{4.17}$$

$$\implies \max_S \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right) - \operatorname{corr}_{\mathfrak{D}}\left(\chi_{\tilde{S}}, f^*\right) \leqslant \varepsilon. \tag{4.18}$$

Combining Eqs. (4.15) and (4.18), we have the required bound. □

Claim 4.9. The parity monomial $\chi_{\tilde{S}}$ produced by Algorithm 4.1 is a (η, κ, δ)-Weak agnostic learner for an n-bit Boolean function f that can be computed by a size-t decision trees, where $n, t > 0, \eta = \frac{1}{t}, \kappa = \varepsilon, \delta = \frac{1}{10}$.

Proof. Let $\mathcal{C}$ be a family of size-t decision trees, and let $c \in \mathcal{C}$ be the optimal classifier. Using the Fourier expansion of c and applying linearity of expectation in Definition 2.4 we have

$$\operatorname{corr}_{\mathfrak{D}}\left(c(x)\right) = \sum_{S \subseteq [n]} \hat{c}(S) \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right). \tag{4.19}$$

From Lemma 4.5, we have that $\sum_{S \subseteq [n]} |\hat{c}(S)| \leqslant t$. Using an averaging argument, we have

$$\max_S |\operatorname{corr}_{\mathfrak{D}}\left(\chi_S(x)\right)| \geqslant \frac{1}{t} \operatorname{corr}_{\mathfrak{D}}\left(c\right). \tag{4.20}$$

Let $\tilde{S}$ be a subset that estimates the mode of the Bayes-approximation of $\mathrm{FS}(c)$ such that

$$\left|\operatorname{corr}_{\mathfrak{D}, \widehat{c^*}}\left(\chi_{\tilde{S}}\right) - \max_S \operatorname{corr}_{\mathfrak{D}}\left(\chi_S\right)\right| \leqslant \kappa. \tag{4.21}$$

From Eqs. (4.20) and (4.21), we have

$$\operatorname{corr}_{\mathfrak{D}, \widehat{c^*}}\left(\chi_{\tilde{S}}\right) \geqslant \frac{1}{t} \operatorname{corr}_{\mathfrak{D}}\left(c\right) - \kappa. \tag{4.22}$$

Hence, $\chi_{\tilde{S}}$ is indeed an $\left(\frac{1}{t}, \kappa, \delta\right)$-weak quantum agnostic learner w.r.t. the optimal classifier c in the benchmark class of size-t decision trees. □

Claim 4.7 gives us the final query complexity and runtime for Algorithm 4.1. Claims 4.8 and 4.9 guarantee that Algorithm 4.1 produces a weak learner for size-t decision trees in polynomial running time. We now state the main theorem for quantum weak learning the class of size-t decision trees below.

Theorem 4.10 (Weak Agnostic Learner for size-t Decision Trees)**.** Let $\eta = 1/t$, and let $\kappa \in [0, 1/2)$. Assuming uniform marginal over instances, given access to a QAEX$(\mathfrak{D})$ oracle, Algorithm 4.1 makes $m = \tilde{O}\left(\frac{n}{\eta\kappa^3} \cdot \log\frac{1}{\kappa}\log\frac{1}{\delta}\right)$ calls to the QAEX oracle and runs for an additional $\tilde{O}\left(\frac{n}{\eta\kappa^3} \cdot \log\frac{1}{\kappa}\log\frac{1}{\delta}\right)$ time to obtain a (η, κ, δ)-Weak agnostic learner for size-t decision trees with high probability.

§ 4.6 Discussion and Conclusion

In this chapter, we show how to design quantum algorithms for constructing weak learners for the class of size-t decision trees. Our results are presented in a concise form in Table 4.1.

Setting	Uniform Marginal	Near-Uniform marginal
Realizable	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$
RCN	$O\left(\frac{1}{\varepsilon^3}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\varepsilon^3}\log\frac{1}{\delta}\right)$
Agnostic	$O\left(\frac{nt}{\varepsilon^3}\log\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$	$O\left(\frac{nt}{\varepsilon^3}\log\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$

Table 4.1: Comparing the query complexities of constructing weak PAC learners for size-t decision trees in various settings without access to MQ oracle.

We note here that the distribution oracle model considered for non-uniform learnability is an extremely strong distribution access model. One could ask whether a more natural distribution access model exists - such as an (unbounded) noisy version of $\mathcal{O}_{\mathfrak{D}}$ for which non-MQ non-uniform marginal learnability is still viable. To this end, we ask the following question.

Open Problem 2. Is there a notion of a weaker/more realistic distribution access model for which we can obtain weak agnostic learners for size-t decision trees with non-uniform marginal over the instances, *without access* to MQ?

Part II

Quantum Ensembling Learning

Coming together is the beginning. Keeping together is progress. Working together is success.

HENRY FORD

Chapter 5

Realizable Quantum Boosting for Domain Partitioning Weak Learners

Abstract

Boosting algorithms combine several inaccurate learning algorithms to construct a highly accurate learner. In this work, we address an open question posed in [IW23] on the existence of boosting algorithms for quantum weak learners with non-binary hypotheses by proposing a quantum boosting algorithm, QREALBOOST, that boosts quantum weak learners with confidence-rated hypotheses. Our algorithm achieves a polynomial speedup over known adaptive quantum boosting algorithms in terms of both the bias and the time complexity of the weak learner. We also prove that the QREALBOOST algorithm induces large margins over the training set. Finally, we empirically evaluate the convergence behavior of the QREALBOOST algorithm and contrast it against a few boosting algorithms. A significant part of this chapter is based on the following publication:

- Sagnik Chatterjee et al. "Quantum boosting using domain-partitioning hypotheses". In: *Quantum Machine Intelligence* 5.2 (July 2023), p. 33. ISSN: 2524-4914. DOI: 10.1007/s42484-023-00122-3

Contents

5.1 Introduction 74
5.1.1 Properties of Boosting Algorithms 75
5.2 The ADABOOST algorithm and its variants 77
5.2.1 Classical variants of ADABOOST 78
5.2.1.1 Real-valued weak learners with Bounded Range 78
5.2.1.2 Domain-partitioning weak learners 79

5.2.2 Quantum Boosting algorithms 80
5.3 The QREALBOOST algorithm 82
5.3.1 Overview of the QREALBOOST algorithm 82
5.3.2 Proof of Correctness and Convergence Analysis 91
5.3.3 Query and Time Complexity Analysis of QREALBOOST 94
5.4 Discussion 96

§ 5.1 Introduction

In discriminative supervised machine learning, a learning algorithm A aims to learn an unknown concept provided labeled examples drawn from an unknown distribution $\mathfrak{D}$. The learner A is then used to label (or classify) new unlabeled instances, which are assumed to be drawn independently at random from $\mathfrak{D}$. As discussed in Section 4.2, strong learners classify unseen data with arbitrarily high accuracy, while weak learners perform slightly better than random guessing. We recall the formal definitions of strong and weak learning as given earlier. Schapire [Sch90] proved that in the realizable setting, Definitions 4.1 and 4.2 imply each other.

Lemma 5.1 (Equivalence of Weak and Strong learning [Sch90])**.** An unknown concept class $\mathcal{C} = \bigcup_{n\geqslant 1} C_n$ is efficiently weakly PAC learnable if and only if C is efficiently strongly PAC learnable.

▷ Ensemble Learning.

Lemma 5.1 reconciles these two variants of learning algorithms by suggesting the existence of an entirely new paradigm of machine learning algorithms known as ensemble learning. In an informal sense, a strong learner constructed from underlying weak learners retains some advantages of its constituent weak learner(s), such as model interpretability/explainability, but also exhibits good generalization behavior at the same time. From a quantum perspective, ensemble learning is important in the NISQ regime since a theoretically accurate quantum learner can have bad accuracy (due to noise that cannot be mitigated/corrected) when implemented on near-term devices. Designing ensembling algorithms for quantum weak learners is, therefore, a very practically relevant task. The focus of this chapter is on boosting algorithms, a specific variant of ensemble learning algorithms. In this chapter, we introduce a new quantum-boosting algorithm that boosts weak quantum learners with *domain-partitioning hypotheses*.

▷ General Assumptions and Limitations.

In this chapter, we shall be limiting our discussions only to boosting algorithms for concept classes that are realizable. We will also not be placing too much emphasis on designing concrete instances of weak learners in this chapter. Instead, we will mostly assume that certain classes of base hypotheses exist and then design boosting algorithms for them.

▷ Structure and Organisation.

The chapter is arranged as follows: The original AdaBoost algorithm is introduced in Section 5.2 along with its variants in Section 5.2.1. We then give a brief overview of existing realizable quantum boosting algorithms in Section 5.2.2. We present the main contribution of this chapter - the QRealBoost algorithm in Section 5.3. Finally, we wrap up the chapter with a few open questions and discussions in Section 5.4.

§ 5.1.1 Properties of Boosting Algorithms

We outline some important properties of Boosting algorithms.

▷ The Smoothness Property.

Informally, if a boosting algorithm A is *smooth*, then the distributions $\mathfrak{D}^t$ and $\mathfrak{D}^{t+1}$ produced by A are "close" for all $t \geqslant 1$. Alternatively, a smooth boosting algorithm produces intermediate distributions $\mathfrak{D}^t$ that do not diverge rapidly from the starting distribution $\mathfrak{D}^1$.

Smooth boosting algorithms do not put too much weight on misclassified instances in successive iterations, unlike algorithms like AdaBoost, which exponentially skews the weight of outliers in subsequent iterations. The smooth boosting algorithms are, therefore, more robust toward noise. However, since the distributions only change in a bounded fashion from one iteration to another, smooth boosting algorithms are slow to converge. Examples of smooth boosting algorithms are SmoothBoost [Ser03] and MadaBoost [DW00]. Gavinsky [Gav03] gave an agnostic boosting algorithm which was optimally smooth.

Remark 5.1.1 (A Note on SmoothBoost and Noisy Boosting)**.** [Ser03] proposed smooth boosting to design PAC learners for *malicious noise*. The smoothness property of SmoothBoost makes sure that no single instance gets assigned a large weight (beyond a given threshold) going into the next iteration. Therefore we are guaranteed that the weight of noisy instances also never crosses a given threshold, allowing for eventual convergence.

On the other hand, in non-smooth adaptive algorithms such as AdaBoost or RealBoost, the weight of noisy samples might be so great in the intermediate distributions that the hypothesis given by the weak learner ceases to have any value at all.

Therefore, boosting algorithms with smoothness properties are more robust to noise. We point out in more detail when we discuss agnostic boosting algorithms in Chapter 6. We give some more details on SMOOTHBOOST in Appendix B.4.

▷ Margins induced by Boosting algorithms.

We define an important concept related to boosting algorithms - the *margin* of instances induced by various classifiers.

> **Definition 5.1** (Margin of instances induced by a combined classifier)**.** For a given labeled example (x, y), the margin of the instance is the quantity $y \cdot \tilde{H}(x) / \sum_t \alpha_t$, where $\tilde{H}(x) = \sum_t \alpha_t h_t(x)$. The magnitude of the margin is the difference between the weight of classifiers that correctly predicted the label and the weight of classifiers that incorrectly predicted the label, while the sign of the margin indicates whether the example has been correctly classified by the combined classifier.

It has been noted that adaptive boosting algorithms tend to induce large margins over the training set, while smooth boosting algorithms tend to induce smaller margins over the training set. However, this is anecdotal since adaptivity and smoothness are not mutually exclusive properties of boosting algorithms.

▷ The Adaptivity Property.

Recall that according to Definition 4.1, any γ-weak learner achieves a bias γ w.r.t. any distributions over the instance space $\mathcal{X}$. Calculating the bias of an unknown learner is, therefore, an intractable task. This poses a challenge to the design of boosting algorithms since weight is assigned to the intermediate classifiers using the bias of the weak learner that produced said hypotheses.

If a boosting algorithm does not require apriori knowledge of the bias to function, it is known as an adaptive boosting algorithm. One possible workaround for computing the bias for any weak learner is to compute the *empirical bias* of the weak learner over the training set weighted by some intermediate distribution produced by the boosting algorithm. This approach is used by adaptive boosting algorithms such as ADABOOST and its variants, which make them practical to implement.

In fact, Bartlett et al. [Bar+98] showed that when the weak classifiers are better than random guessing, i.e., have non-zero bias γ, the margins of instances induced by ADABOOST in the training set are guaranteed to be large after boosting for a sufficient number of iterations.

§ 5.2 The ADABOOST algorithm and its variants

In light of Lemma 5.1, researchers aimed to answer the following question:

Question 5.1. Given only black-box query access to weak learners $A_1, \ldots, A_k$ w.r.t. an unknown concept class $\mathcal{C}$, can we (efficiently) construct a strong learner for $\mathcal{C}$?

Various ensemble learning algorithms for converting weak learners into strong learners were proposed to answer Question 5.1. This line of research culminated in the seminal work of Freund and Schapire [FS97], where they proposed the AdaBoost algorithm, which won the Gödel prize in 2003.

The AdaBoost algorithm learns an ensemble of classifiers using black-box queries to one weak learner, which it finally combines into a strong learner using a majority of weighted sum approach[1]. We point out some interesting features of the AdaBoost algorithm now.

Initially, AdaBoost sets $\mathfrak{D}^1 = \mathcal{U}(\mathcal{S})$, i.e., we start with the uniform distribution over the training set $\mathcal{S}$. Computing the weight of the t^{th} classifier h_t is done in two steps. First, AdaBoost computes the training error ε_t of h_t over the training set $\mathcal{S}$ weighted according to the current distribution $\mathfrak{D}^t$. Then, the confidence of the hypothesis α_t is computed using the training error.

$$\varepsilon_t = \sum_{i \in \mathcal{S}} \mathfrak{D}_i^t \cdot \mathbb{1}_{[h_t(x_i) \neq y_i]}, \tag{5.1}$$

$$\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t}. \tag{5.2}$$

An interesting point to note here is that if the weak learner produces an especially bad classifier h_t s.t. $\varepsilon_t > 1/2$ (alternatively, if $\mathfrak{D}^t$ is especially bad), then the corresponding classifier has negative weight, i.e., $\alpha_t < 0$. The AdaBoost algorithm is formally presented in Appendix B.2. The following proposition fleshes out the convergence behaviour of AdaBoost. The proof is deferred to Appendix C for completeness.

Proposition 5.1. [Convergence of AdaBoost [FS97]] The training error of the final classifier obtained by AdaBoost in $O(\log m/\gamma^2)$ iterations, is zero w.r.t. $\mathfrak{D}^1$, where γ is the bias of weak learner A.

Combining Lemma 2.7 and Proposition 5.1 allows us to obtain generalization error bounds on the final classifier H produced by AdaBoost w.r.t. to the uniform distribution in terms of the VC-dimension d_A of the weak learner A as

$$\Pr\left[\mathrm{err}_{\mathcal{U}}(H) \leqslant O\left(\frac{\log m/d_A}{m/d_A}\right)\right] \geqslant 1 - \delta.$$

[1] See Algorithm B.1 for a general framework of Realizable Boosting algorithms.

We note a few important ideas in the proof of Proposition 5.1[2].

Observation 5.2.1 (Updating the weights is costly)**.** The distribution update step, i.e., computing $\mathfrak{D}_i^{t+1}$ is usually the most computationally intensive step in the AdaBoost algorithm since it adds a multiplicative factor of $|\mathcal{S}| = m$ per time step to the time complexity. The computation of $\mathfrak{D}_i^{t+1}$ critically depends on the normalization term Z_t.

Observation 5.2.2 (Greedily minimizing training error)**.** The training error $\hat{\text{err}}^H_{\mathcal{S},\mathfrak{D}^1} = \prod_t Z_t$ can be minimized by choosing the weak learner h_t in a greedy manner to minimize the normalization term Z_t at every iteration. In fact, a good strategy is to minimize an appropriate surrogate term (usually an upper bound on Z_t) instead of Z_t itself to reduce the overall running time while achieving reasonable bounds on convergence.

Observation 5.2.3 (Using surrogates for Z_t)**.** In the AdaBoost algorithm, the normalization term Z_t can be expressed in three distinct ways:

$$Z_t = \sum_{i=1}^{m} \mathfrak{D}_i^t \cdot \exp\{\alpha_t \cdot -y_i \cdot h_t(x_i)\} = e^{-\alpha_t} \cdot \varepsilon_t + e^{\alpha_t} \cdot (1-\varepsilon_t) \leqslant \sqrt{1-4\gamma^2}.$$

The expression in the middle is more desirable than the first equality since it takes less time to compute, as α_t and ε_t terms are already known from earlier in the algorithm. As discussed above, any reasonable upper bound on Z_t is also acceptable as a surrogate for Z_t.

§ 5.2.1 Classical variants of AdaBoost

The AdaBoost algorithm critically assumes that the hypothesis generated by the weak classifier and the labels assigned are binary-valued. The analysis of the convergence of AdaBoost particularly uses the fact that $h \in \mathcal{H}_A : \mathcal{X} \mapsto \{\pm 1\}$, where A is the weak learner we have query access to. In this section, we take a look at boosting algorithms for different types of weak learners.

§ 5.2.1.1 Real-valued weak learners with Bounded Range

Here, we consider weak learners that output real-valued hypotheses h_t s.t. $h_t : \mathcal{X} \mapsto [-1,+1]$. Our aim here is to prove the convergence of a boosting algorithm that mostly acts like AdaBoost but with access to bounded range real-valued weak learners. We define the error of the weak classifier h_t analogously to AdaBoost as $\varepsilon_t = \sum_{i=1}^m \mathfrak{D}_i^t \cdot -y_i \cdot h_t(x_i)$. Note that ε_t is still bounded between $[-1,+1]$. Instead of defining the confidence term α_t, we obtain it as a minimizer of the normalization term. Consider the normalization term $Z_t = \sum_{i=1}^m \mathfrak{D}_i^t \cdot \exp\{\alpha_t \cdot -y_i \cdot h_t(x_i)\}$, where the terms are exactly

[2] For details of the proof see Appendix C.

the same as given in Algorithm B.2, apart from the hypotheses $h_t : \mathcal{X} \mapsto [-1, +1]$. Since it is not possible to exactly expand Z_t as in Eq. (C.10) due to h_t not being ± 1-valued, Schapire and Singer [SS99] proved the following proposition.

Proposition 5.2. For weak learners that output bounded real-valued hypotheses $h_t : \mathcal{X} \mapsto [-1, +1]$, the normalization term Z_t can be upper-bounded as $Z_t \leqslant \prod_{t=1}^{T} \sqrt{1 - \varepsilon_t^2}$, where ε_t is the weighted training error of h_t on the training set $\mathcal{S}$ weighted by $\mathfrak{D}^t$.

Now, we can proceed to bound the error of the combined classifier as $\hat{\text{err}}^H_{\mathcal{S},\mathfrak{D}^1} = \prod_{t=1}^{T} Z_t \leqslant \prod_{t=1}^{T} \sqrt{1 - \varepsilon_t^2}$ as shown earlier.

Remark 5.2.1. We can obtain a tighter upper bound in Eq. (C.18) by using different approximation techniques. This would give us a better convergence bound in terms of the number of iterations T needed to drive $\hat{\text{err}}^H_{\mathcal{S},\mathfrak{D}^1}$ down to 0.

So far, both models of weak learners focus on greedily minimizing the normalization constant Z_t (refer Algorithm B.2) or some surrogate term at each iteration in order to upper-bound the number of iterations for convergence. The proofs in Propositions 5.1 and 5.2 also hold when the confidence and the hypothesis are combined into one quantity, i.e., weak learners output a term $\tilde{h}_t(x_i) = \alpha_t h_t(x_i)$.

§ 5.2.1.2 Domain-partitioning weak learners

A domain-partitioning hypothesis h partitions the input domain $\mathcal{X}$ into a set of mutually exclusive and exhaustive blocks, such that the hypothesis h predicts the same labels for all instances belonging to a given partition $\mathcal{X}_j$. Formally, a C domain-partitioning hypothesis h induces a disjoint partitioning of the domain of instances $\mathcal{X}$ into C sets $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_C\}$, i.e., $\mathcal{X} = \bigcup_{i=1}^{C} \mathcal{X}_i$, and $\mathcal{X}_i \cap \mathcal{X}_j = \varnothing, \forall i, j \in \{1, \ldots, C\}$, s.t.

$$h(x) = h(x'), \quad \forall x, x' \in \mathcal{X}_j, \; j \in \{1, \ldots, C\}.$$

Since the prediction is constant for all instances assigned to a specific partition, we denote the prediction h for the partition $\mathcal{X}_j$ by the partition-specific constant $\beta_j \in \mathbb{R}$. Note that $\text{sign}(\beta_j)$ and $|\beta_j|$ gives us the prediction and its confidence w.r.t. partition $\mathcal{X}_j$ respectively.

Example 5.1. Suppose $\beta_j = \log\left(W_j^+ / W_j^-\right)$, where W_j^b is the weighted fraction of examples with label $y = b$. Consider a partition $\mathcal{X}_i$ that has 100 samples with label -1 and 5 samples with label $+1$. Then, the weighted prediction $\beta_i = -1.3$ for $\mathcal{X}_i$, which means that all instances in this partition are predicted to have label

-1 with a confidence rating of 1.3.

Consider another partition $\mathcal{X}_j$ that contains 55 samples with label $+1$ and 45 samples with label -1. Then $\beta_i = 0.08$. Since the majority has a $+1$ label, we assign it to the entire partition, but with much lower confidence than in the previous case. Therefore, if there is almost an equal number of samples of both labels in a particular domain, then the confidence for predicting either class will be low.

Example 5.2. Weak learners that abstain from classifying instances are an interesting example of Domain-Partitioning weak learners. Abstaining weak learners have a ternary range $h : \mathcal{X} \mapsto \{-1, 0, +1\}$, where output 0 indicates that the learner has abstained from voting on a particular instance.

Schapire and Singer [SS99] introduced the RealBoost algorithm (see Algorithm B.3) for boosting weak learners that output discrete class partitions (or domain partitions). The RealBoost algorithm has three features that distinguish it from AdaBoost:

1. Since domain partitioning hypotheses do not output binary labeling, the predicted label of an instance is given by the majority label of the particular partition that the instance belongs to (see Line 5 in RealBoost).
2. Due to the domain-partitioning feature, unlike AdaBoost, a notion of training error over the entire training set is not useful. A predicted domain-partitioning hypothesis might be very accurate in partitioning a portion of the domain, while it may perform very poorly for the rest of the domain. We would like to keep the accurate behavior for the final ensemble while moving on to better predictors for the rest of the domain. Hence, the confidence/margin of the hypothesis is computed separately with respect to every partition (see Line 6 in RealBoost).
3. Due to the domain-partitioning nature of the hypotheses, the normalization term can be written as $Z_t = 2\sum_{k=1}^{C}\sqrt{W_+^{k,t}\cdot W_-^{k,t}}$, where $W_b^{k,t}$[3]. For the sake of completeness, we state this in Lemma 5.2, and defer the proof to Appendix C.) This is different from the surrogate used for AdaBoost as seen in Observation 5.2.3.

Lemma 5.2. [Normalization term in RealBoost [SS99]] The normalization term in RealBoost can be expressed succinctly as $Z_t = 2\sum_{k=1}^{C}\sqrt{W_+^{k,t}\cdot W_-^{k,t}}$, where $W_b^{k,t}$ is the weight of the label $b \in \{\pm 1\}$ in the k^{th} partition at the t^{th} iteration.

§ 5.2.2 Quantum Boosting algorithms

Early works on quantum boosting algorithms consisted primarily of three categories

[3] See Line 7 of RealBoost

- quantum boosting algorithms that considered heuristic perfomance [Nev+12];
- quantum algorithms that used the classical AdaBoost as a subroutine [SP18];
- quantum algorithms that only obtained speed-ups for certain parts of AdaBoost, such as the margin computation [Wan+20].

Arunachalam and Maity [AM20] gave the first purely quantum algorithm that had a quadratic speedup over the original AdaBoost algorithm in terms of the VC-dimension of the weak learner (but with a worse dependence on the bias of the learner). Subsequently, Izdebski and Wolf [IW23] proposed QSmoothBoost- a quantum variant of Servedio's classical SmoothBoost algorithm[Ser03], which retains the quadratic speedup in time/query complexity while also achieving a polynomial speedup in the bias of the weak learner as compared to the QAdaBoost algorithm. However, similar to its classical counterpart, QSmoothBoost is not adaptive and, therefore, is subject to issues like slow convergence and small induced margins over the training set.[4] See Table 5.1 for a comparison of various boosting algorithms.

Table 5.1: A comparison of AdaBoost, RealBoost, SmoothBoost, QAdaBoost, and QSmoothBoost with the QRealBoost algorithm. The range of the hypotheses produced by the weak learner can be binary or non-binary. The weak learner (classical or quantum) has bias γ, an associated hypothesis class $\mathcal{H}$ with VC-dimension d, and takes time R (classically), or Q (quantumly) to produce a hypothesis $h \in \mathcal{H}$.

Algorithm	Model	Adaptive	Range	Query Complexity
AdaBoost [FS97]	Classical	Yes	Binary	$O\left(d \cdot R \cdot \gamma^{-4}\right)$
RealBoost [SS99]	Classical	Yes	Non-Binary	$O\left(d \cdot R \cdot \gamma^{-4}\right)$
SmoothBoost [Ser03]	Classical	No	Binary	$\tilde{O}\left(d \cdot \gamma^{-4} + R \cdot \gamma^{-2}\right)$
QAdaBoost [AM20]	Quantum	Yes	Binary	$\tilde{O}\left(\sqrt{d} \cdot Q^{1.5} \cdot \gamma^{-11}\right)$
QSmoothBoost [IW23]	Quantum	No	Binary	$\tilde{O}\left(\sqrt{d} \cdot \gamma^{-5} + Q \cdot \gamma^{-4}\right)$
QRealBoost [Cha+23]	Quantum	Yes	Non-Binary	$\tilde{O}\left(\sqrt{d} \cdot Q \cdot \gamma^{-9}\right)$

▷ An important caveat.

Quantum boosting algorithms only enjoy a speedup over their classical counterparts under two specific circumstances.

[4] For a detailed discussion on "adaptiveness", induced margins, and a brief outline of the SmoothBoost algorithm [Ser03] see Chapter 7.

1. A quadratic improvement in the dependence on the VC dimension of the concept class dominates the polynomial bad dependence on the bias of the weak learner. For example, consider the concept class that consists of all Boolean functions $\mathcal{C}_{\text{all}} = \{c : \mathbb{F}_2^n \mapsto \mathbb{F}_2\}$. The VC-dimension of $\mathcal{C}_{\text{all}}$ is 2^n. For this specific concept class, quantum boosting algorithms would give a speedup over their classical counterparts. However, the resulting quantum boosting algorithms would not run in polynomial time.
2. The weak learner is truly quantum, and Q (where Q is some polynomial in all desired parameters) is smaller than R. We saw examples of such separations in Section 9.3. In this case, quantum boosting algorithms provide a genuine speedup over their classical counterparts.

§ 5.3 The QREALBOOST algorithm

In this section, we describe the QREALBOOST algorithm - the first quantum boosting algorithm for weak learners with non-binary ranges with provable convergence guarantees and generalization performance. We note at the outset that we require a few assumptions for the algorithm to work.

1. It is an interesting open question to find a natural example of a truly quantum domain-partitioning (weak) learning algorithm. For the purposes of this section, we assume that such weak learners exist.
2. The QREALBOOST algorithm requires CQRAM access, or access to a QEX oracle. Access to noiseless examples is crucial for the operation of QREALBOOST.

In Section 5.3.1, we describe the QREALBOOST algorithm in detail, which is presented in Algorithm 5.1. In Section 5.3.2, we provide proof of correctness and analyze the convergence behavior of the QREALBOOST algorithm. We then provide query complexity and time complexity bounds in Section 5.3.3.

§ 5.3.1 Overview of the QREALBOOST algorithm

This entire section walks the reader through the t^{th} iteration of the QREALBOOST algorithm. We assume that we are given oracle access to a C domain partitioning quantum weak PAC learner A that has sample complexity Q.

▷ State Preparation Subroutine.

A training set $S = \{(x_i, y_i)\}_{i\in[m]}$ is provided as input to the QREALBOOST algorithm.

Algorithm 5.1: The QREALBOOST algorithm

Input: A training set $\mathcal{S} = \{(x_i, y_i)\}_{i\in[m]}$ where $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and $\mathcal{Y} = \{\pm 1\}$. Oracle access to a C domain-partitioning weak quantum learner A with sample complexity Q.

Initialize: Set weights $\mathfrak{D}_i^1 = 1/m$, $\forall i \in [m]$. Fix T, $\varepsilon = O\left(\frac{1}{QT^2}\right)$, $\kappa = \frac{C}{(1-\varepsilon)}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$.

for $t = 1$ *to* T **do**

Prepare Q copies of $|\phi_0\rangle$ and 2C copies of $|\psi_0\rangle$ as in Eq. (5.3).

Generate $Q + 2C$ copies of a distribution of training samples

for $s = 1$ *to* $t-1$ **do**

Query each O_{h_s} to produce $Q + 2C$ copies of $|\phi_1\rangle$ as in Eq. (5.5).

Using the update rule (Eq. (5.6)) apply the unitary $U_{\mathfrak{D}}$ to obtain $Q + 2C$ copies of $|\phi_2\rangle = \frac{1}{\sqrt{m}}\sum_{i\in[m]} \left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle \left|j_i^1, j_i^2, \ldots, j_i^{t-1}\right\rangle$ as in Eq. (5.8).

For $Q + 2C$ copies of $|\phi_2\rangle$, perform conditional rotation on $\left|\tilde{\mathfrak{D}}_i^t\right\rangle$ to obtain $|\phi_3\rangle$.

Perform amplification and uncompute ancillas to obtain $|\phi_5\rangle = \sum_i \sqrt{\tilde{\mathfrak{D}}_i^t}\,|x_i, y_i\rangle + |\zeta_t'\rangle$.

Obtain the t^{th} hypothesis oracle.

$A\left(|\phi_5\rangle^{\otimes Q}\right) \to$ (followed by a measurement) oracle O_{h_t} corresponding to hypothesis h_t.

Obtain confidence-rated predictions using h_t on the last 2C copies

On 2C copies of $|\psi_2\rangle$ query $O_{h_1}, \ldots, O_{h_{t-1}}, O_{h_t}$ to create 2C copies of $|\psi_3\rangle$ as in Eq. (5.42).

for $k = 1$ *to* C *and* $b \in \{-1, +1\}$ **do**

Take the $(k, b)^{th}$ copy of $|\psi_3\rangle$ and prepare $|\psi_4\rangle = \frac{1}{\sqrt{m}}\sum \left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle \left|j_i^1, \cdot, j_i^t\right\rangle \left|\tilde{\mathfrak{D}}_i^{k,b,t}\right\rangle$.

// $\tilde{\mathfrak{D}}_i^{k,b,t} = \tilde{\mathfrak{D}}_i^t$ if $j_i^t = k$ and $y_i = b$, and 0 otherwise.

Perform conditional rotation on $\left|\tilde{\mathfrak{D}}_i^{k,b,t}\right\rangle$ to obtain states with amplitudes $\sqrt{W_b^{k,t}}$.

Perform amplitude estimation to obtain $\tilde{W}_b^{k,t}$ with relative error ε.

Perform Laplace correction on the estimated weights $\tilde{W}_b^{k,t}$.

Compute Z_t', and $\beta_{j,t}'$ for all partitions using the estimated partition-label weights.

Output: Output combined classifier $H(x) = \text{sign}\left(\sum_{t=1}^{T} \beta'_{j^t,t}\right)$, where $x \in \mathcal{X}_{j^t}^t$ for all $t \in [T]$.

We prepare Q copies of $|\phi_0\rangle$ and 2C copies of $|\psi_0\rangle$ as follows:

$$|\phi_0\rangle = |\psi_0\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \left|x_i, y_i, \mathfrak{D}_i^1\right\rangle, \tag{5.3}$$

where $\mathfrak{D}_i^1 = 1/m$, $\forall i \in [m]$. Suppose there exists a CQRAM that stores the training set $\mathcal{S}$ s.t. querying index i returns the element $|x_i, y_i\rangle$. We can now query the CQRAM with the state $\frac{1}{\sqrt{m}} \sum_{i\in[m]} |i\rangle \left|\mathfrak{D}_i^1\right\rangle$ to obtain individual copies of $|\phi_0\rangle$ and $|\psi_0\rangle$.

▷ Training the Weak Learner.

Let $\{h_1, \ldots, h_{t-1}\}$ be the hypotheses produced by the QREALBOOST algorithm in iterations 1 to $t-1$. Oracular access to all the previous hypotheses is expressed as

$$O_{h_t} |x_i, y_i\rangle |0\rangle \mapsto |x_i, y_i\rangle \left|j_i^t\right\rangle, \tag{5.4}$$

where $j_i^t = h_t(x_i)$ refers to the domain partition of the i^{th} sample at the t^{th} iteration. We append an ancilla register $|0\rangle^{\otimes t-1}$ to each copy of the states $|\phi_0\rangle$ and $|\psi_0\rangle$ and query each such hypothesis oracle in order to produce Q copies of $|\phi_1\rangle$ and 2C copies of $|\psi_1\rangle$ as follows:

$$|\phi_1\rangle = |\psi_1\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \left|x_i, y_i, \mathfrak{D}_i^1\right\rangle \left|j_i^1, j_i^2, \ldots, j_i^{t-1}\right\rangle. \tag{5.5}$$

We now describe the QREALBOOST update rule as follows:

$$\tilde{\mathfrak{D}}_i^{s+1} = \frac{\tilde{\mathfrak{D}}_i^s \cdot \exp\left(-\beta'_{j,s} \cdot y_i\right)}{\kappa \cdot Z'_s}, \tag{5.6}$$

where $\tilde{\mathfrak{D}}_i^s$ is the approximate weight of instance i at iteration $s \in [t-1]$,[5] $\beta'_{j,s}$ is the approximate confidence-rated prediction for partition j at iteration $s \in [t-1]$ s.t. $x_i \in \mathcal{X}_j^s$, κ is a normalization term, $Z'_s = 2\sum_{j=1}^{C} \sqrt{\tilde{W}_+^{j,s} \cdot \tilde{W}_-^{j,s}}$ is an approximate normalization term at iteration $s \in [t-1]$, and $\tilde{W}_b^{j,s}$ are approximate partition-label weights for partition $j \in [C]$ and label $b \in \{\pm 1\}$ at iteration $s \in [t-1]$. We note here that

$$\left|W_b^{j,t} - \tilde{W}_b^{j,t}\right| \leqslant \varepsilon \cdot W_b^{j,t}. \tag{5.7}$$

Eq. (5.7) is guaranteed by Algorithm 5.1, as we shall see later. Here, $\varepsilon = 1/QT^2$.

Using the stored confidence values $\beta'_{j,s}$, weights Z'_s, and oracles O_{h_s}, we construct a unitary mapping $U_{\mathfrak{D}}$ for updating the weight register using $t-1$ applications of Eq. (5.6) as follows:

[5] Note that $\tilde{\mathfrak{D}}_i^1 = \mathfrak{D}_i^1$ for all instances $i \in [m]$ in the training set.

$$|\phi_1\rangle \xrightarrow{U_{\mathfrak{D}}} |\phi_2\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle \left|j_i^1, j_i^2, \ldots, j_i^{t-1}\right\rangle . \tag{5.8}$$

We update all Q copies of $|\phi_1\rangle$ and 2C copies of $|\psi_1\rangle$ to $|\phi_2\rangle$ and $|\psi_2\rangle$ respectively.

▷ Obtaining domain-partitioning Weak Hypotheses.

For all Q copies of $|\phi_2\rangle$, we perform a conditional rotation on the register $\left|\tilde{\mathfrak{D}}_i^t\right\rangle$ to obtain the state

$$|\phi_3\rangle = \sum_{i\in[m]} \frac{1}{\sqrt{m}} \left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle \left|j_i^1, \ldots, j_i^{t-1}\right\rangle \left(\sqrt{\tilde{\mathfrak{D}}_i^t}\,|1\rangle + \sqrt{1-\tilde{\mathfrak{D}}_i^t}\,|0\rangle\right) \tag{5.9}$$

Let $U_{0\to3}$ be the unitary that performs $|0\rangle \to |\phi_3\rangle$. We perform Amplitude Amplification as stated in Lemma 2.8 on $|\phi_3\rangle$ to obtain the state $|\phi_4\rangle$ (using $O(\sqrt{m}\log T)$ applications of $U_{0\to3}$ and $U_{0\to3}^{-1}$) with probability at least $O(1-1/T)$. The amplified state $|\phi_4\rangle$ is as follows

$$|\phi_4\rangle = \sum_{i\in[m]} \sqrt{\tilde{\mathfrak{D}}_i^t} \left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle \left|j_i^1, \ldots, j_i^{t-1}\right\rangle + |\zeta_t\rangle . \tag{5.10}$$

The state $|\zeta_t\rangle$ is present since $\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^t \leqslant 1$ (i.e. the weights are sub-normalized). We state a claim now that shows that the sum of the weights is very close to 1, and hence, very little interference is expected from $|\zeta_t\rangle$.

Claim 5.1. For $\kappa = \frac{C}{(1-\varepsilon)}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$ and $\varepsilon = \frac{1}{QT^2}$, $\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^t \in \left[1-\frac{4\varepsilon}{1+\varepsilon}, 1\right]$, where $\tilde{\mathfrak{D}}_i^t$ is updated as in Eq. (5.6), $\forall t \in [T]$.

Proof. We start by recalling that $W_b^{j,t} = \sum_{i:x_i\in\mathfrak{X}_j \wedge y_i=b} \tilde{\mathfrak{D}}_i^t$. This implies the following equation directly.

$$\frac{\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^t \cdot e^{-\beta'_{j,t}\cdot y_i}}{\sum_{j=1}^{C} W_+^{j,t}\cdot e^{-\beta'_{j,t}} + W_-^{j,t}\cdot e^{\beta'_{j,t}}} = 1. \tag{5.11}$$

We start by obtaining a preliminary bound on the quantity $\sum_i \tilde{\mathfrak{D}}_i^{t+1}$. We expand the R.H.S. in Eq. (5.6) as

$$\tilde{\mathfrak{D}}_i^{t+1} = \frac{\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^t \cdot e^{-\beta'_{j,t}\cdot y_i}}{\sum_{j=1}^{C} W_+^{j,t}\cdot e^{-\beta'_{j,t}} + W_-^{j,t}\cdot e^{\beta'_{j,t}}} \cdot \frac{\sum_{j=1}^{C} W_+^{j,t}\cdot \exp\left(-\beta'_{j,t}\right) + W_-^{j,t}\cdot \exp\left(\beta'_{j,t}\right)}{\kappa Z'_t}$$

From Eq. (5.11), we can set the first term to 1. Then,

$$\tilde{\mathfrak{D}}_i^{t+1} = \frac{\sum_{j=1}^{C} W_+^{j,t} \cdot \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \cdot \exp\left(\beta'_{j,t}\right)}{\kappa Z'_t} \tag{5.12}$$

$$= \frac{\sum_{j=1}^{C} W_+^{j,t} \cdot \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \cdot \exp\left(\beta'_{j,t}\right)}{2\kappa \sum_{j=1}^{C} \sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}} \tag{5.13}$$

$$= \frac{1}{2\kappa} \sum_{j=1}^{C} \frac{W_+^{j,t} \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \exp\left(\beta'_{j,t}\right)}{\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}}. \tag{5.14}$$

In Eq. (5.13), we use the value of Z'_t as discussed earlier in Eq. (5.6). From Eq. (5.14) we can upper-bound the quantity $\tilde{\mathfrak{D}}_i^{t+1}$ using Eq. (5.7) as

$$\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1} \leqslant \frac{1}{2\kappa\,(1-\varepsilon)} \sum_{j=1}^{C} \frac{W_+^{j,t} \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \exp\left(\beta'_{j,t}\right)}{\sqrt{W_+^{j,t} \cdot W_-^{j,t}}}. \tag{5.15}$$

Substituting $\kappa = \frac{C}{(1-\varepsilon)}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$ in Eq. (5.15) we have

$$\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1} \leqslant \frac{1}{2\kappa\,(1-\varepsilon)} \sum_{j=1}^{C} \frac{W_+^{j,t} \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \exp\left(\beta'_{j,t}\right)}{\sqrt{W_+^{j,t} \cdot W_-^{j,t}}} \tag{5.16}$$

$$= \frac{1}{2\kappa\,(1-\varepsilon)} \sum_{j=1}^{C} \sqrt{\frac{W_+^{j,t}}{W_-^{j,t}}} \cdot \sqrt{\frac{\tilde{W}_-^{j,t}}{\tilde{W}_+^{j,t}}} + \sqrt{\frac{W_-^{j,t}}{W_+^{j,t}}} \cdot \sqrt{\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}}} \tag{5.17}$$

$$\leqslant \frac{1}{2\kappa\,(1-\varepsilon)} \sum_{j=1}^{C} \left(2\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\right) \tag{5.18}$$

$$= \frac{C}{\kappa\,(1-\varepsilon)} \sqrt{\frac{1+\varepsilon}{1-\varepsilon}} = 1. \tag{5.19}$$

Similarly, from Eq. (5.14) we can lower-bound the quantity $\tilde{\mathfrak{D}}_i^{t+1}$ using Eq. (5.7) as

$$\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1} \geqslant \frac{1}{2\kappa\,(1+\varepsilon)} \sum_{j=1}^{C} \frac{W_+^{j,t} \exp\left(-\beta'_{j,t}\right) + W_-^{j,t} \exp\left(\beta'_{j,t}\right)}{\sqrt{W_+^{j,t} \cdot W_-^{j,t}}}. \tag{5.20}$$

Substituting $\kappa = \frac{C}{(1-\varepsilon)}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$ in Eq. (5.20) we have

$$\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1} \geqslant \frac{1}{2\kappa\,(1+\varepsilon)} \sum_{j=1}^{C} \frac{W_+^{j,t}\exp\left(-\beta'_{j,t}\right) + W_-^{j,t}\exp\left(\beta'_{j,t}\right)}{\sqrt{W_+^{j,t}\cdot W_-^{j,t}}} \tag{5.21}$$

$$= \frac{1}{2\kappa\,(1+\varepsilon)} \sum_{j=1}^{C} \sqrt{\frac{W_+^{j,t}}{W_-^{j,t}}}\cdot\sqrt{\frac{\tilde{W}_-^{j,t}}{\tilde{W}_+^{j,t}}} + \sqrt{\frac{W_-^{j,t}}{W_+^{j,t}}}\cdot\sqrt{\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}}} \tag{5.22}$$

$$\geqslant \frac{C}{\kappa\,(1+\varepsilon)}\sqrt{\frac{1-\varepsilon}{1+\varepsilon}} = \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2 = \left(1-\frac{2\varepsilon}{1+\varepsilon}\right)^2 \geqslant 1-\frac{4\varepsilon}{1+\varepsilon}. \tag{5.23}$$

Combining Eqs. (5.19) and (5.23) gives us the desired bound on $\tilde{\mathfrak{D}}_i^t$ for any $t \in [T]$. □

Now, apply U_D^{-1} and $O_{h_1}^{-1}\ldots, O_{h_{t-1}}^{-1}$ to $|\phi_4\rangle$ to obtain the state $|\phi_5\rangle$.

$$|\phi_4\rangle \xrightarrow{\text{Uncompute ancillas}} |\phi_5\rangle = \sum_{i\in[m]} \sqrt{\tilde{\mathfrak{D}}_i^t}\,|x_i, y_i\rangle + \left|\zeta'_t\right\rangle. \tag{5.24}$$

We pass Q copies of $|\phi_5\rangle$ to the weak learner A. In turn, the weak learner produces a hypothesis h_t w.h.p.[6] to which we assume oracular access. The following claim shows that the learned hypothesis is a good hypothesis.

Claim 5.2. If at the t^{th} iteration, the γ-weak learner A produces a hypothesis h_t on being fed Q copies of the ideal state $\left|\phi'_5\right\rangle = \sum_{i\in[m]} \sqrt{\mathfrak{D}_i^t}\,|x_i, y_i\rangle$, then A produces the same hypothesis h_t w.h.p. when given Q copies of $|\phi_5\rangle$, for $\varepsilon = O\left(1/QT^2\right)$.

Proof. Let p be the probability that A outputs the hypothesis h_t on being fed Q copies of the ideal state $\left|\phi'_5\right\rangle = \sum_{i\in[m]} \sqrt{\mathfrak{D}_i^t}\,|x_i, y_i\rangle$, and let q be the probability that A outputs the hypothesis h_t on being fed Q copies of the state $|\phi_5\rangle$. We want to bound the quantity $|p-q|$. Let the density matrices corresponding to $|\phi_5\rangle$ and $\left|\phi'_5\right\rangle$ be ρ and σ respectively. We denote the class of POVM's on the hypothesis space $\mathcal{H}$ as $\{E_h\}_{h\in\mathcal{H}}$ such that $\sum_{h\in\mathcal{H}} E_h = I$. Therefore, we have

$$|p-q| \leqslant \max_{E_h} |\mathrm{Tr}\{E_h\rho\} - \mathrm{Tr}\{E_h\rho\}| \tag{5.25}$$

$$\leqslant D(\rho,\sigma) = \sqrt{1-\left|\left\langle\phi_5\middle|\phi'_5\right\rangle^{\otimes Q}\right|^2} \tag{5.26}$$

$$\implies |p-q| \leqslant \sqrt{1-\left|\left\langle\phi_5\middle|\phi'_5\right\rangle\right|^{2Q}} \tag{5.27}$$

[6] We assume that the probability of A not producing any hypothesis is $O(1/T)$, similar to earlier works [Gav03; AM20]. Therefore, the hypotheses in Claim 5.2 are produced with prob. $O\left(1-1/T\right)$.

Eq. (5.25) follows from the definition of POVMs, and Eq. (5.26) follows from Definition 2.9 and Eq. (2.8). To evaluate the R.H.S. of Eq. (5.27), we first lower bound the quantity $\left|\langle\phi_5|\phi'_5\rangle\right|$ as follows.

$$\left|\langle\phi_5|\phi'_5\rangle\right| = \left|\sqrt{\tilde{\mathfrak{D}}_i^t \cdot \mathfrak{D}_i^t} + \langle\zeta_t|\phi'_5\rangle\right| \tag{5.28}$$

$$\implies \left|\langle\phi_5|\phi'_5\rangle\right| \geqslant \left|\sqrt{\tilde{\mathfrak{D}}_i^t \cdot \mathfrak{D}_i^t}\right| - \left|\langle\zeta_t|\phi'_5\rangle\right|. \tag{5.29}$$

We can bound the term $\left|\langle\zeta_t|\phi'_5\rangle\right|$ as

$$\left|\langle\zeta_t|\phi'_5\rangle\right| \leqslant \|\zeta_t\|_1 \leqslant 1 - \left(1 - \frac{4\varepsilon}{1+\epsilon}\right) = \frac{4\varepsilon}{1+\epsilon} \tag{5.30}$$

using Claim 5.1. We bound the first term in Eq. (5.29) as

$$\tilde{\mathfrak{D}}_i^{t+1} \cdot \mathfrak{D}_i^{t+1} = \sum_{i\in[m]} \sqrt{\frac{\tilde{\mathfrak{D}}_i^t \cdot e^{-\beta'_{j,t}}}{\kappa \cdot Z'_t} \cdot \frac{\tilde{\mathfrak{D}}_i^t \cdot e^{-\beta'_{j,t}}}{Z_t}} = \sqrt{\frac{Z_t}{\kappa \cdot Z'_t}} \sum_{i\in[m]} \frac{\tilde{\mathfrak{D}}_i^t \cdot e^{-\beta'_{j,t}}}{Z_t} = \sqrt{\frac{Z_t}{\kappa \cdot Z'_t}} \tag{5.31}$$

We now substitute $Z_t = \sum_{j=1}^{C} W_+^{j,t} e^{-\beta'_{j,t}} + W_-^{j,t} e^{\beta'_{j,t}}$ and $Z'_t = 2\sum_{j=1}^{C} \sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}$ to obtain the following expression.

$$\frac{1}{\sqrt{\kappa}} \cdot \sqrt{\sum_{j=1}^{C} \frac{W_+^{j,t} e^{-\beta'_{j,t}} + W_-^{j,t} e^{\beta'_{j,t}}}{2\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}}} = \frac{1}{\sqrt{2\kappa}} \cdot \sqrt{\sum_{j=1}^{C} \frac{W_+^{j,t} e^{-\beta'_{j,t}}}{\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}} + \frac{W_-^{j,t} e^{\beta'_{j,t}}}{\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}}} \tag{5.32}$$

Expanding the $e^{\pm\beta'_{j,t}}$ terms, we obtain,

$$\tilde{\mathfrak{D}}_i^{t+1} \cdot \mathfrak{D}_i^{t+1} = \frac{1}{\sqrt{2\kappa}} \cdot \sqrt{\sum_{j=1}^{C} \frac{W_+^{j,t}}{\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}} \sqrt{\frac{\tilde{W}_-^{j,t}}{\tilde{W}_+^{j,t}}} + \frac{W_-^{j,t}}{\sqrt{\tilde{W}_+^{j,t} \cdot \tilde{W}_-^{j,t}}} \sqrt{\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}}}} \tag{5.33}$$

$$= \frac{1}{\sqrt{2\kappa}} \cdot \sqrt{\sum_{j=1}^{C} \frac{W_+^{j,t}}{\tilde{W}_+^{j,t}} + \frac{W_-^{j,t}}{\tilde{W}_-^{j,t}}} \tag{5.34}$$

$$\implies \tilde{\mathfrak{D}}_i^{t+1} \cdot \mathfrak{D}_i^{t+1} \geqslant \frac{1}{\sqrt{2\kappa}} \cdot \sqrt{\sum_{j=1}^{C} \frac{2}{1+\varepsilon}} = \sqrt{\frac{C}{\kappa(1+\varepsilon)}} \tag{5.35}$$

Eq. (5.35) follows from Eq. (5.7). Therefore, we have $\left|\langle\phi_5|\phi'_5\rangle\right| \geqslant \left(\frac{C}{\kappa(1+\varepsilon)}\right)^{\frac{1}{4}} - \frac{4\varepsilon}{1+\epsilon}$.

Substituting $\kappa = \frac{C}{1-\varepsilon}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$ in the above equation gives us

$$\left|\langle\phi_5|\phi_5'\rangle\right| \geqslant \left(\frac{C\cdot(1-\varepsilon)}{C\cdot(1+\varepsilon)}\frac{1-\varepsilon}{1+\varepsilon}\right)^{\frac{1}{4}} - \frac{4\varepsilon}{1+\epsilon} = \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{\frac{1}{2}} - \frac{4\varepsilon}{1+\epsilon} \tag{5.36}$$

$$\implies \left|\langle\phi_5|\phi_5'\rangle\right| \geqslant \left(1-\frac{2\varepsilon}{1+\varepsilon}\right)^{\frac{1}{2}} - \frac{4\varepsilon}{1+\epsilon} \tag{5.37}$$

$$\implies \left|\langle\phi_5|\phi_5'\rangle\right| \geqslant 1-\frac{\varepsilon}{(1+\varepsilon)} - \frac{4\varepsilon}{1+\epsilon} = 1-\frac{5\varepsilon}{(1+\varepsilon)}. \tag{5.38}$$

The first inequality in Eq. (5.38) follows from the inequality $(1-x)^t \geqslant 1-xt, \forall x \leqslant 1, t > 0$. Plugging Eq. (5.38) back into Eq. (5.27) gives us

$$|p-q| \leqslant \sqrt{1-\left|\langle\phi_5|\phi_5'\rangle\right|^{2Q}} \leqslant \sqrt{1-\left(1-\frac{5\varepsilon}{(1+\varepsilon)}\right)^{2Q}} \tag{5.39}$$

$$\implies |p-q| \leqslant \sqrt{1-1+\frac{10Q\varepsilon}{1+\varepsilon}} \leqslant \sqrt{5Q\varepsilon} \leqslant 5\sqrt{Q\varepsilon}, \tag{5.40}$$

where the first inequality in Eq. (5.40) follows from another application of $(1-x)^t \geqslant 1-xt, \forall x \leqslant 1, t > 0$. Setting $p = O(1-1/T)$ and $\varepsilon = O(1/QT^2)$ proves the lemma. □

▷ Obtaining confidence-rated predictions.

Recall the definition of $U_{\mathfrak{D}}$ from Eq. (5.8). Applying $U_{\mathfrak{D}}$ to $|\psi_1\rangle$, we can obtain 2C copies of the state

$$|\psi_1\rangle \xrightarrow{U_{\mathfrak{D}}} |\psi_2\rangle = \frac{1}{\sqrt{m}}\sum_{i\in[m]}\left|x_i, y_i, \tilde{\mathfrak{D}}_i^t\right\rangle\left|j_i^1, j_i^2, \ldots, j_i^{t-1}\right\rangle. \tag{5.41}$$

We append an ancilla to each copy of perform the following unitary transformation.

$$\left[|\psi_2\rangle|0\rangle\right]^{\otimes 2C} \xrightarrow{O_{h_t}} \left[|\psi_3\rangle\right]^{\otimes 2C} = \bigotimes_{\substack{k\in\{1,\ldots,C\}\\ b\in\{-1,+1\}}}\left[\frac{1}{\sqrt{m}}\sum_{i\in[m]}\left|x_i, y_i, \tilde{D}_i^t\right\rangle\left|j_i^1, j_i^2, \ldots, j_i^{t-1}\right\rangle\left|j_i^t\right\rangle\right]. \tag{5.42}$$

Consider the $(k,b)^{\text{th}}$ copy of $[|\psi_3\rangle]^{\otimes 2C}$ for $k \in \{1,2,\ldots,C\}$ and $b \in \{-1,+1\}$. For each copy of $|\psi_3\rangle$, perform the update $|\psi_3\rangle \to |\psi_4\rangle$ as

$$|\psi_3\rangle_{(k,b)}|0\rangle^{\otimes 2} \to |\psi_4\rangle_{(k,b)} = \frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i\rangle|y_i\rangle\left|\tilde{\mathfrak{D}}_i^t\right\rangle\underbrace{\left|j_i^1,\ldots,j_i^t\right\rangle}_{|j(i,t)\rangle}\underbrace{\left|\mathbb{1}_{[j_i^t=k]}\right\rangle}_{|\mathbb{I}_1\rangle}\underbrace{\left|\mathbb{1}_{[y_i=b]}\right\rangle}_{|\mathbb{I}_2\rangle}. \tag{5.43}$$

Note here that $|\mathbb{I}_1\rangle$ and $|\mathbb{I}_2\rangle$ are binary-valued states. Using the $|\mathbb{I}_1\rangle$ and $|\mathbb{I}_2\rangle$ registers as controls, we append an ancilla to $|\psi_4\rangle$ and obtain the state

$$|\psi_4\rangle_{(k,b)}|0\rangle \to |\psi_5\rangle_{(k,b)} = \frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i\rangle|y_i\rangle\left|\tilde{\mathfrak{D}}_i^t\right\rangle|j(i,t)\rangle|\mathbb{I}_1\rangle|\mathbb{I}_2\rangle\underbrace{\left|\tilde{\mathfrak{D}}_i^t\cdot\mathbb{I}_1\cdot\mathbb{I}_2\right\rangle}_{\tilde{\mathfrak{D}}_i^{k,b,t}}. \tag{5.44}$$

Now, we perform a conditional rotation on the $\left|\tilde{\mathfrak{D}}_i^{k,b,t}\right\rangle$ register to obtain

$$|\psi_6\rangle_{(k,b)} = \sqrt{W_b^{k,t}}\,|\chi\rangle^1_{(k,b)}|1\rangle + \sqrt{1-W_b^{k,t}}\,|\chi\rangle^0_{(k,b)}|0\rangle, \tag{5.45}$$

where,

$$W_b^{k,t} = \frac{\sum_{i\in[m]}\tilde{\mathfrak{D}}_i^{k,b,t}}{m}, \tag{5.46}$$

and,

$$|\chi\rangle^1_{(k,b)} = \frac{1}{\sqrt{m}}\sum_{i\in[m]}\frac{\sqrt{\tilde{\mathfrak{D}}_i^{k,b,t}}}{\sqrt{W_b^{k,t}}}|x_i\rangle|y_i\rangle\left|\tilde{\mathfrak{D}}_i^t\right\rangle|j(i,t)\rangle|\mathbb{I}_1\rangle|\mathbb{I}_2\rangle\left|\tilde{\mathfrak{D}}_i^{k,b,t}\right\rangle \tag{5.47}$$

$$|\chi\rangle^0_{(k,b)} = \frac{1}{\sqrt{m}}\sum_{i\in[m]}\frac{\sqrt{1-\tilde{\mathfrak{D}}_i^{k,b,t}}}{\sqrt{1-W_b^{k,t}}}|x_i\rangle|y_i\rangle\left|\tilde{\mathfrak{D}}_i^t\right\rangle|j(i,t)\rangle|\mathbb{I}_1\rangle|\mathbb{I}_2\rangle\left|\tilde{\mathfrak{D}}_i^{k,b,t}\right\rangle \tag{5.48}$$

Let $V_{0,6}^{(k,b)}$ be the unitary that performs $|0\rangle \to |\psi_6\rangle_{(k,b)}$. We perform relative-error amplitude estimation as stated in Lemma 2.9, with an expected $\tilde{O}(\sqrt{m}QT^2)$ queries to $V_{0,6}^{(k,b)}$ and $V_{0,6}^{-1\ (k,b)}$ to obtain the quantity $\tilde{W}_b^{k,t}$ that is an estimate of $W_b^{k,t}$ w.h.p.

Hence, we obtain all 2C values of $\tilde{W}_b^{j,t}$ for all $j \in \{1,2,\ldots,C\}$, $b \in \{-1,+1\}$. Note that it is possible for the values of $W_b^{j,t}$ to be very small (even zero) for some j. This would result in the quantities $\beta'_{j,t}$ becoming very large or unbounded, thus increasing the tendency of the learner to overfit. We use a general smoothing technique known as **Laplace correction**[CN89] to overcome this issue and use the smoothed values to calculate the margins as $\beta'_{j,t} = \frac{1}{2}\ln\left(\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}}\right)$ for all $j \in \{1,\ldots,C\}$ and the normalization constant as

$$Z'_t = 2\sum_{j=1}^{C}\sqrt{\tilde{W}_+^{j,t}\cdot\tilde{W}_-^{j,t}}. \tag{5.49}$$

▷ A note on Laplace Smoothing.

We give a brief overview of Laplace correction here. Let $V_b^{k,t} = \tilde{W}_b^{k,t}\cdot m$. We update the

values of $\tilde{W}^{k,t}$ to $\frac{V_b^{k,t}+1}{m+2C}$. Recall that the estimated confidence values after smoothing are

$$\beta'_{k,t} = \frac{1}{2}\ln\left(\frac{\tilde{W}_+^{k,t}}{\tilde{W}_-^{k,t}}\right) \qquad \forall k \in \{1,\ldots,C\} \tag{5.50}$$

Let us look at the corner cases now. If there exists a partition where $W_b^{k,t} = 0$ or very small, then we now have $\tilde{W}_b^{k,t} \sim \frac{1}{m+2C} \sim \frac{1}{m}$, which essentially resets the weight of every instance in that partition to $\mathfrak{D}_t^1$. On the other hand, consider a partition where $V_b^{k,t} \sim m$. This implies that $V_{-b}^{k,t} \sim 0$. By Eq. (5.50), this would give us unbounded margins $\beta'_{k,t} = \frac{1}{2}\ln\left(\frac{V_b^{k,t}}{V_{-b}^{k,t}}\right) \sim \infty$. Now, due to the smoothing, the confidence for this domain partition will still be large but bounded above by $O(\log m)$. In the next section, we prove that Algorithm 5.1 outputs a strong learner with good generalization performance.

§ 5.3.2 Proof of Correctness and Convergence Analysis

▷ Obtaining good estimates for confidences and surrogate loss.

We start this section by proving bounds on the quantities Z'_t and $\beta'_{j,t}$ as computed in Line 14 of Algorithm 5.1 using the estimated partition-label weights.

Claim 5.3. If the partition-label weights $W_b^{k,t}$ are relatively estimated using the error parameter ε as $\left|W_b^{j,t} - \tilde{W}_b^{j,t}\right| \leqslant \varepsilon \cdot W_b^{j,t}$ in Algorithm 5.1, then the difference between the actual margins $\beta_{j,t}$ and the estimated margins $\beta'_{j,t}$ is bounded as $\left|\beta'_{j,t} - \beta_{j,t}\right| \leqslant \frac{1}{2}\ln\left(\frac{1+\varepsilon}{1-\varepsilon}\right); \quad j \in [C]]$.

Proof. Recall that the actual margin as given in Algorithm B.3 is $\beta_{j,t} = \frac{1}{2}\ln\left(\frac{W_+^{j,t}}{W_-^{j,t}}\right)$ and the estimated margin as computed in Algorithm 5.1 is $\beta'_{j,t} = \frac{1}{2}\ln\left(\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}}\right)$. We upper bound the difference in margins as follows:

$$\beta'_{j,t} - \beta_{j,t} = \frac{1}{2}\left[\ln\frac{\tilde{W}_+^{j,t}}{\tilde{W}_-^{j,t}} - \ln\frac{W_+^{j,t}}{W_-^{j,t}}\right] = \frac{1}{2}\left[\ln\frac{\tilde{W}_+^{j,t}}{W_+^{j,t}} - \ln\frac{\tilde{W}_-^{j,t}}{W_-^{j,t}}\right] \tag{5.51}$$

$$\implies \beta'_{j,t} - \beta_{j,t} \leqslant \frac{1}{2}\left[\ln(1+\varepsilon) - \ln(1-\varepsilon)\right] \leqslant \frac{1}{2}\ln\left(\frac{1+\varepsilon}{1-\varepsilon}\right). \tag{5.52}$$

Similarly, we obtain the lower bound as

$$\beta'_{j,t} - \beta_{j,t} \geqslant \frac{1}{2}\ln\left(\frac{1-\varepsilon}{1+\varepsilon}\right). \tag{5.53}$$

Combining Eq. (5.52) and Eq. (5.53), we get our desired bound. □

From Claim 5.3 we see that the difference between the actual margin and the estimated margins is very small. In fact, a very simple calculation shows us that $\left|\beta'_{j,t} - \beta_{j,t}\right| \leqslant 0.1$ for $\varepsilon \leqslant 0.1$. We note that the error parameter ε is far smaller than a constant fraction, which means our estimated margins are quite close to the ideal margin values.

Claim 5.4. If the partition-label weights $W_b^{k,t}$ are relatively estimated using error parameter ε as $\left|W_b^{j,t} - \tilde{W}_b^{j,t}\right| \leqslant \varepsilon \cdot W_b^{j,t}$ in Algorithm 5.1, then $|Z'_t - Z_t| \leqslant \varepsilon \cdot Z_t$.

Proof. Recall that the normalization constant in Algorithm B.3) is calculated as $Z_t = 2\sum_{j=1}^{C}\sqrt{W_+^{j,t}\cdot W_-^{j,t}}$, while in Algorithm 5.1, we substitute the calculated weights with estimated weights to obtain the quantity $Z'_t = 2\sum_{j=1}^{C}\sqrt{\tilde{W}_+^{j,t}\cdot \tilde{W}_-^{j,t}}$. Using (5.7), we upper bound the difference between the quantities as

$$Z'_t = 2\sum_{j=1}^{C}\sqrt{\tilde{W}_+^{j,t}\cdot \tilde{W}_-^{j,t}} \leqslant 2\sum_{j=1}^{C}\sqrt{W_+^{j,t}(1+\varepsilon)\cdot W_-^{j,t}(1+\varepsilon)} \tag{5.54}$$

$$= 2(1+\varepsilon)\sum_{j=1}^{C}\sqrt{W_+^{j,t}\cdot W_-^{j,t}} = Z_t(1+\varepsilon). \tag{5.55}$$

Similarly, the lower bound is obtained as

$$Z'_t = 2\sum_{j=1}^{C}\sqrt{\tilde{W}_+^{j,t}\cdot \tilde{W}_-^{j,t}} \geqslant 2\sum_{j=1}^{C}\sqrt{W_+^{j,t}(1-\varepsilon)\cdot W_-^{j,t}(1-\varepsilon)} \tag{5.56}$$

$$= 2(1-\varepsilon)\sum_{j=1}^{C}\sqrt{W_+^{j,t}\cdot W_-^{j,t}} = Z_t(1-\varepsilon) \tag{5.57}$$

Combining Eq. (5.55) and Eq. (5.57), we obtain our desired bound. □

Claim 5.4 shows that when we minimize the normalization constant at every step using the estimated values $\tilde{W}_b^{j,t}$, these quantities are themselves relatively bounded by the actual normalization constant. This implies that the training error of the combined classifier is greedily minimized when the normalization constant is minimized at every step. Hence, our training error at every step does not blow up due to the estimation of the partition weights.

▷ Probability of Failure of Algorithm 5.1.

The probability of failure of Algorithm 5.1 stems primarily from the steps Line 6,

Line 7, and Line 13, where each step fails with a probability at most $O(1/T)$. When we take a union bound over all T iterations for all three steps, the overall failure probability dips to an arbitrary constant, which is at most $1/3$. There is an extra log factor incurred due to error reduction, which can be absorbed in the $\tilde{O}(.)$ notation.

▷ Convergence of Algorithm 5.1.

Claim 5.5. For a sufficiently large number of iterations $T \geqslant \frac{\ln m}{2\gamma^2}$, the final combined classifier H produced by Algorithm 5.1 has zero training error with respect to the uniform superposition $\tilde{\mathfrak{D}}^1$ w.h.p..

Proof. Recall from Eq. (5.6) we have the following.

$$\tilde{\mathfrak{D}}_i^{T+1} = \frac{\tilde{\mathfrak{D}}_i^1}{\prod_{t=1}^{T} \kappa \cdot Z'_t} \cdot \exp\left(-y_i \sum_{t=1}^{T} \beta'_{j,t}\right) \tag{5.58}$$

Using techniques outlined in Proposition 5.1, we can upper-bound the training error of Algorithm 5.1 as follows

$$\begin{aligned}
\Pr_{x\sim\mathcal{S}}[H(x) \neq y] &= \sum_{i=1}^{m} \tilde{\mathfrak{D}}_i^1 \cdot \mathbb{I}[H(x) \neq y] && (5.59)\\
&\leqslant \sum_{i=1}^{m} \tilde{\mathfrak{D}}_i^1 \cdot \exp\left(-y_i \sum_{t=1}^{T} \beta'_{j,t}\right) && (5.60)\\
&= \sum_{i=1}^{m} \tilde{\mathfrak{D}}_i^{T+1} \prod_{t=1}^{T} \kappa \cdot Z'_t \leqslant \prod_{t=1}^{T} \kappa \cdot Z'_t && (5.61)
\end{aligned}$$

The inequality in Eq. (5.61) follows from Claim 5.1. Now plugging in the upper-bound of Z'_t from Claim 5.4 into Eq. (5.61), we get

$$\Pr_{x\sim\mathcal{S}}[H(x) \neq y] \text{ leq} \prod_{t=1}^{T} \kappa \cdot Z_t(1+\varepsilon) = \kappa^T (1+\varepsilon)^T \prod_{t=1}^{T} Z_t. \tag{5.62}$$

Substituting $\kappa = \frac{C}{(1-\varepsilon)}\sqrt{\frac{1+\varepsilon}{1-\varepsilon}}$, and $\varepsilon = O(1/QT^2)$ we have

$$\Pr_{x\sim\tilde{\mathfrak{D}}^1}[H(x)\neq y] \leqslant C^T\left(\frac{1+1/T^2}{1-1/T^2}\right)^T\prod_{t=1}^{T}Z_t \tag{5.63}$$

$$\leqslant C^T\left(\frac{T^2+1}{T^2-1}\right)^T\prod_{t=1}^{T}Z_t \leqslant C^T\left(1+\frac{2}{T^2-1}\right)^T\prod_{t=1}^{T}Z_t \tag{5.64}$$

$$\leqslant C^T\exp\left(\frac{2T}{T^2-1}\right)\prod_{t=1}^{T}Z_t \tag{5.65}$$

For sufficiently large T, we have

$$\Pr_{x\sim\tilde{\mathfrak{D}}^1}[H(x)\neq y] \leqslant C^T e^{2/T}\prod_{t=1}^{T}Z_t \leqslant C^T e^{2/T}\prod_{t=1}^{T}\sqrt{1-4\gamma_t^2} \tag{5.66}$$

$$\leqslant C^T\exp\left(\frac{2}{T}-2\sum_{t=1}^{T}\gamma_t^2\right) \tag{5.67}$$

$$\leqslant C^T\exp\left(\frac{2}{T}-2\gamma^2T\right) \tag{5.68}$$

$$\leqslant C^T\exp\left(-2\gamma^2T+\frac{2}{T}\right). \tag{5.69}$$

We note here that the term $\frac{C^T}{e^{2\gamma^2T}}$ goes down fast depending on the constant in $T = O(\log m/\gamma^2)$, which can be as small as $O(\log C)$. This leaves us with the term $e^{2/T}$ to take care of. Substituting $T = O(\log m/\gamma^2)$ gives implies

$$\Pr_{x\sim\tilde{\mathfrak{D}}^1}[H(x)\neq y] < \frac{1}{m}. \tag{5.70}$$

We recall the fact that $\tilde{\mathfrak{D}}^1$ is the uniform distribution, which implies that Algorithm 5.1 has zero training error. □

Hence, according to Claim 5.5, if we run Algorithm 5.1 for a sufficiently large number of iterations T, then with a high probability, we output the desired hypothesis H that has zero training error and a small generalization error.

§ 5.3.3 Query and Time Complexity Analysis of QRealBoost

Theorem 5.6 (Query Complexity). Algorithm 5.1 can boost a C domain-partitioning γ-weak learner A with sample complexity Q with an associated hypothesis class $\mathcal{H}$ having VC-dimension d using at most $O\left(\frac{\sqrt{d}\cdot C\cdot Q}{\gamma^9}\right)$ queries.

Proof. We first consider the number of queries to the hypothesis oracles $\{O_{h_1},\ldots,O_{h_T}\}$ made by Algorithm 5.1 in the t^{th} iteration. We require $t-1$ queries to the oracles $O_{h_1}, O_{h_2},\ldots,O_{h_{t-1}}$ to obtain each copy of $|\psi_1\rangle$ and $|\psi_1\rangle$ as in Eq. (5.5). Using Lemma 2.8, we see that our amplitude amplification algorithm uses an expected $\Theta(p'\log T/p)$ calls to the unitaries $U_{0\to3}$ and $U_{0\to3}^{-1}$, to obtain $|\phi_4\rangle$ with a high probability, where

$$p=\sum_{i\in[m]}\sqrt{\tilde{\mathfrak{D}}_i^t/M};\quad,\quad p'=\sum_{i\in[m]}\sqrt{\tilde{\mathfrak{D}}_i^t}. \tag{5.71}$$

Hence, the Amplitude Amplification step to obtain $|\phi_4\rangle$ requires $O(\sqrt{m}\log T(t-1))$ queries to the oracles for each copy of $|\phi_3\rangle$. The uncompute step to obtain $|\phi_5\rangle$ requires a further $t-1$ queries to the oracles $O_{h_1}, O_{h_2},\ldots,O_{h_{t-1}}$ for each copy of $|\phi_4\rangle$. For estimating the partition weights w.h.p. we make an expected $\tilde{O}\left(\sqrt{m}QT^2\log T\cdot t\right)$ queries to $O_{h_1}, O_{h_2},\ldots,O_{h_t}$. We obtain this by plugging in $p=O(1/M)$, $\varepsilon=O(\frac{1}{QT^2})$, and $k=\log T$ in Lemma 2.9. Hence, the query complexity is

$$\sum_{t=1}^{T}\left(\underbrace{O(\sqrt{m}Q\log T\cdot(t-1))}_{\text{Amplitude Amplification}}+\underbrace{O((Q+C)\log T\cdot(t-1))}_{\text{weight updates and uncomputing}}+\underbrace{\tilde{O}\left(\sqrt{m}CQT^2\log T\cdot t\right)}_{\text{Amplitude Estimation}}\right). \tag{5.72}$$

Simplifying, we obtain the following query complexity

$$O(\sqrt{m}QT^2\log T)+O((Q+C)T^2\log T)+\tilde{O}\left(\sqrt{m}CQT^4\log T\right)=O\left(\frac{\sqrt{m}CQ}{\gamma^8}\right). \tag{5.73}$$

The last equality follows from setting $T=O(\log m/\gamma^2)$. From [FS97], we know that the number of training samples m required for good generalization performance is $m\geqslant\tilde{O}\left(\frac{d}{\gamma^2\eta^2}\right)$ for some $\eta>0$. Setting the parameter $\eta=0.1$, we get the query complexity as $O\left(\frac{\sqrt{d}\cdot C\cdot Q}{\gamma^9}\right)$. □

Theorem 5.7 (Time Complexity). Algorithm 5.1 can boost a C domain-partitioning γ-weak learner A with sample complexity Q with an associated hypothesis class $\mathcal{H}$ having VC-dimension d in time $O\left(\frac{n^2\sqrt{d}CQ}{\gamma^9}\right)$.

Proof. As discussed earlier, we can assume a CQRAM to prepare the uniform superposition $\frac{1}{\sqrt{m}}\sum_{i\in[m]}\left|x_i, y_i, D_i^1\right\rangle$ using $O(n\log M)$ gates. Hence the time complexity for preparing the state $|\phi_0\rangle^{\otimes Q}\otimes|\psi_0\rangle^{\otimes 2C}$ is $O(n(Q+C))$. The step from $|\phi_0\rangle$ to $|\phi_1\rangle$ and $|\psi_0\rangle$ to $|\psi_1\rangle$ requires $t-1$ queries each, which can be performed in time $O((Q+C)(t-1))$. Next we perform the distribution update which is an arithmetic operation, using the unitary U_D with the $\left|j_i^1,\ldots,j_i^{t-1}\right\rangle$ register as control. This step requires time $O(n^2(Q+C)(t-1))$. We perform amplitude amplification to obtain the state $|\phi_4\rangle$. This requires $O(\sqrt{m}(t-1)\log T)$ applications of $U_{0\to 3}$ and $U_{0\to 3}^{-1}$ as discussed in the previous section. The total time taken is therefore $O(n^2\sqrt{m}Q(t-1)\log T)$. The time taken by our weak learner to output O_{h_t} is $O(n^2Q)$. The arithmetic operations to update state $|\psi_3\rangle_{(k,b)}$ to $|\psi_4\rangle_{(k,b)}$ and perform controlled rotation use $O(n)$ gates. Finally, we make $\tilde{O}\left(\sqrt{m}CQT^2\log T\right)$ queries for the amplitude estimation part, and each query requires time $O(n^2t)$. Therefore our final time complexity is

$$\sum_{t=1}^{T}\underbrace{O\left(n^2\sqrt{m}Q(t-1)\log T\right)}_{\text{Amplitude Amplification}}+\underbrace{\tilde{O}\left(n^2\sqrt{m}CQT^2\log T\cdot t\right)}_{\text{Amplitude Estimation}}+\underbrace{O\left(n^2(Q+C)(t-1)\right)}_{\text{other operations}} \tag{5.74}$$

Simplifying Eq. (5.74), we obtain the final time complexity as $O\left(\frac{n^2\sqrt{d}CQ}{\gamma^9}\right)$. □

§ 5.4 Discussion

In this work, we designed the QRealBoost algorithm, which tackles an open question posed by Izdebski and Wolf [IW23] to boost weak quantum PAC learners that output non-binary hypotheses. QRealBoost retains the performance of RealBoost, which has superior theoretical properties (supported by empirical evidence, too) compared to the AdaBoost algorithm. We also establish that both theoretically and empirically, QRealBoost outperforms QAdaBoost, the only other known adaptive quantum boosting algorithm.

As seen earlier, there are important caveats when discussing speedups for quantum boosting algorithms w.r.t. their classical counterparts. We ask the following natural question in this regard.

> **Open Problem 3.** Is there a natural concept class where the square root dependence on d dominates the polynomial dependence on $1/\gamma$, where γ is the bias of some weak learner?

In Table 5.1, we see that there is a polynomial gap w.r.t. the $1/\gamma$ term in the running time between known quantum boosting algorithms and classical boosting algorithms.

In fact, it was proved recently that even AdaBoost is suboptimal in terms of sample (and hence time) complexity with respect to the bias parameter γ [HLR23]. This worse dependence of the query and time complexity on γ, arises from recomputing $\tilde{D}^t$ over the training samples at every iteration from scratch. We believe this computation can be avoided by maintaining a "distribution oracle" that only needs to be updated in each iteration. In this context, we pose the following questions.

Open Problem 4. What is the lower bound on the exponent of the $1/\gamma$ term in the running time of adaptive quantum boosting algorithms? What about quantum boosting algorithms that are not necessarily adaptive?

A logical continuation of this work is quantizing other variants of AdaBoost which depend on domain partitioning hypotheses such as LogitBoost [FHT00], GentleBoost [FHT00], ModestBoost [VV05], Parameterized AdaBoost [WN14], and Penalized AdaBoost [WN15]. Each variant has different generalization abilities, which make them useful in different contexts. The algorithmic framework followed in this work for estimating the partition weights may be useful to model quantum versions of these variants. For a detailed discussion of realizable boosting algorithms and other variants of AdaBoost (not listed in this chapter), we refer the reader to the excellent books by Schapire and Freund [SF12] and Ferreira and Figueiredo [FF12].

No one can whistle a symphony. It takes a whole orchestra to play it.

H.E. LUCCOCK

Chapter 6

Quantum Agnostic Boosting

Abstract

The agnostic setting is the hardest generalization of the PAC model since it is akin to learning with adversarial noise. In this chapter, we design an agnostic quantum boosting algorithm by quantizing the Potential-Based boosting algorithm of Kanade and Kalai [KK09] to answer an open question posed by [AM20; IW23]. Our quantum boosting algorithm is smooth and uses *relabeling* instead of reweighting intermediate distributions, removing the need for an extra amplitude amplification step (a staple in the previous quantum boosting algorithms) prior to training the classifier. We show that our algorithm achieves a quadratic speedup over its classical counterpart. This chapter is based on a part of the following publication:

- Sagnik Chatterjee et al. "Efficient Quantum Agnostic Improper Learning of Decision Trees". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta et al. Vol. 238. Proceedings of Machine Learning Research. PMLR, May 2024, pp. 514–522. URL: https://proceedings.mlr.press/v238/chatterjee24a.html

Contents

6.1 Introduction 100
6.2 The Quantum Agnostic Boosting Algorithm 100
6.2.1 Overview of the Quantum Boosting Algorithm 101
6.2.2 Proof of Correctness 106
6.2.3 Complexity Analysis 111
6.2.4 Application: Efficient Agnostic Learners for Decision Trees w/o MQ access 111
6.3 Discussion 112

§ 6.1 Introduction

Quantum boosting algorithms for the realizable setting have been shown to exist (see Chapter 5), but their existence in the agnostic setting was an open question [AM20; IW23]. The challenge in such algorithms is precisely estimating the margins under the presence of instance-dependent noise. Our primary goal in this chapter to quantize the Potential-Based boosting algorithm by Kanade and Kalai [KK09] whose use of *relabeling* let us avoid using the amplitude amplification subroutine (a staple in the previous quantum boosting algorithms) explicitly, thereby removing a significant source of error. In Section 6.2, we show that given a weak quantum agnostic learner A with an associated hypothesis class $\mathcal{C}$ and a set of m training examples, we can construct a poly $(m, 1/\varepsilon)$ time quantum boosting algorithm to produce a hypothesis that is ε close to the best hypothesis in $\mathcal{C}$.

§ 6.2 The Quantum Agnostic Boosting Algorithm

In this section, we describe our quantum agnostic boosting algorithm (QAGBOOST, as defined in Algorithm 6.1) that has query access to a (m, κ, η)-weak quantum agnostic learner A, and to a QAEX $(\mathfrak{D})$ oracle that returns an unknown joint distribution $\mathcal{D}$ over $\mathbb{F}_2^n \times \mathbb{F}_2$. Our algorithm is inspired directly by the classical Potential-Based boosting algorithm by Kanade and Kalai [KK09].[1] At every iteration $t \geqslant 1$, the QAGBOOST algorithm has three main steps.

1. The QAGBOOST algorithm *relabels* the training set using a *conservative* weighting function (see Definition 6.1) based on the combined classifier produced by the QAGBOOST algorithm so far. This has a two-fold impact on the algorithm:

[1] The POTENTIALBOOST algorithm is detailed in Algorithm B.5 for completeness.

(a) Due to the relabeling instead of re-weighting, the intermediate distributions over the training set S are all close to the uniform distribution. Therefore, the QAGBOOST algorithm is a smooth boosting algorithm.

(b) Since there is no reweighting of the intermediate distributions, there is no error (due to probabilistic routines) introduced in the first part of the algorithm. Therefore, we do not require the use of Quantum Amplitude Amplification to *amplify* the good part of the quantum state that would be used for training the weak learner. This helps cut down on the query and time complexity of the boosting algorithm.

2. The quantum agnostic weak learner is trained using the relabeled distribution and returns a new hypothesis h^t with high probability.
3. The QAGBOOST algorithm obtains a new combined classifier that either uses **(a)** the old combined classifier H^{t-1} and h^t, or **(b)** a negation of the old combined classifier. If the combined classifier H^t is chosen correctly, then a certain potential function is minimized, which results in convergence. Hence, the QAGBOOST algorithm is also an adaptive boosting algorithm since it does not require explicit knowledge of the weak learner for convergence.

§ 6.2.1 Overview of the Quantum Boosting Algorithm

▷ State Preparation Subroutine.

We consider two types of input to Algorithm 6.1:

- A training set $S = \{(x_i, y_i)\}_{i\in[m]}$ is provided as input.
- m independent samples are drawn from the QAEX$(\mathfrak{D})$ oracle, which is then fashioned into a training set $S = \{(x_i, y_i)\}_{i\in[m]}$.

The above methods are semantically equivalent so we will assume that there exists a QAEX$(\mathfrak{D})$ oracle from which we can sample data.

We now prepare 2 copies of $|\phi_0\rangle$ and m copies of $|\psi_0\rangle$ as follows:

$$|\phi_0\rangle = |\psi_0\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle . \tag{6.1}$$

As in Algorithm 5.1, we assume that there exists a CQRAM that stores the training set S s.t. querying index i returns the element $|x_i, y_i\rangle$. We can now query the CQRAM with the state $\frac{1}{\sqrt{m}} \sum_{i\in[m]} |i\rangle$ to obtain individual copies of $|\phi_0\rangle$ and $|\psi_0\rangle$.

▷ Training the t^{th} Weak Learner.

Algorithm 6.1: The QAGBOOST algorithm

Input: Oracle access to a (m, κ, η)-weak quantum agnostic learner A for concept class f and a QAEX$(\mathfrak{D})$ oracle.

Initialize: Set $H^0 = 0$, $\varepsilon > 0$, $T = O(1/\eta^2\varepsilon^2)$. Prepare a set S of m training samples $\{(x_i, y_i)\}_{i\in[m]}$ from m independent samples drawn from QAEX$(\mathfrak{D})$.

for $t = 1$ *to* T **do**

Prepare 2 copies of $|\psi_0\rangle$ and m copies of $|\phi_0\rangle$ as $|\psi_0\rangle = |\phi_0\rangle = \frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i, y_i\rangle$.

Query the oracle $O_{H^{t-1}}$ to obtain $2+m$ copies of the state $\frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i, y_i\rangle\left|w_i^t\right\rangle$.

▷ Here, $w_i^t = \min\left\{1, \exp\{-y_i \cdot \text{sign}\left(H^{t-1}(x_i)\right)\}\right\}$.

Generate $2+m$ copies of a distribution of training samples

On the last m copies, perform arithmetic operations to obtain $1/\sqrt{m}\sum_{i\in[m]}|x_i, y_i\rangle|z_i\rangle$.

▷ Let $z_i = (1+w_i^t)/2$, $z_i' = (1-w_i^t)/2$.

Obtain $|\phi_3\rangle = \frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i, y_i\rangle\left(\sqrt{z_i}|0\rangle + \sqrt{z_i'}|1\rangle\right)$ by a conditional rotation on $|z_i\rangle$.

Perform a CNOT operation on $|y_i\rangle$ with the last register as control to obtain m copies of $|\phi_4\rangle = \frac{1}{\sqrt{m}}\sum_{i\in[m]}|x_i\rangle\left(\sqrt{z_i}|y_i, 0\rangle + \sqrt{z_i'}|\bar{y}_i, 1\rangle\right)$.

▷ Denote the unitary for obtaining $|\phi_4\rangle$ by QAEX_t.

Obtain the t^{th} hypothesis oracle.

Obtain oracle O_{h^t} corresponding to hypothesis h^t produced by weak learner A using QAEX_t as the quantum example oracle instead of QAEX$(\mathfrak{D})$. **Obtain margins of the classifier.**

Invoke O_{h^t} on the 1st copy of $|\phi_0\rangle$ to obtain $\frac{1}{\sqrt{m}}\sum_{i\in[m]}\left|x_i, y_i, w_i^t, h^t(x_i)\right\rangle$.

▷ Let $\alpha_t = \frac{1}{m}\sum_{i\in[m]}\left(w_i^t y_i h^t(x_i)\right)$.

Prepare the state $\sqrt{1-\alpha_t}|\psi_0, 0\rangle + \sqrt{\alpha_t}|\psi_1, 1\rangle$. Estimate α_t as $\tilde{\alpha}_t$.

Invoke O_{h^t} on the 2nd copy of $|\phi_0\rangle$ to obtain $\frac{1}{\sqrt{m}}\sum_i\left|x_i, y_i, w_i^t\right\rangle\left|-H^{t-1}(x_i)\right\rangle$.

▷ Let $\beta_t = 1/m\sum_{i\in[m]}\left(w_i^t \cdot y_i \cdot -H^{t-1}(x_i)\right)$.

Prepare the state $\sqrt{1-\beta_t}|\psi_0, 0\rangle + \sqrt{\beta_t}|\psi_1, 1\rangle$. Estimate β_t as $\tilde{\beta}_t$.

if $\tilde{\alpha}_t > \tilde{\beta}_t$ **then**

$H^t = H^{t-1} + \tilde{\alpha}_t \cdot h^t$. Construct the oracle O_{H^t}.

else

$H^t = \left(1-\tilde{\beta}_t\right)H^{t-1}$. Construct the oracle O_{H^t}.

Output: Hypothesis $H^{\hat{t}}$ for some $\hat{t} \in \{1, 2, \ldots T\}$ such that $\text{err}_S(H^{\hat{t}}) = \min_t \text{err}_S(H^t)$.

We now query the $t-1^{\text{th}}$ oracle $O_{H^{t-1}}$ to obtain the following state.

$$|\phi_1\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle |0\rangle |0\rangle \xrightarrow{O_{H^{t-1}}} \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle \left| \underbrace{-y_i \cdot \operatorname{sign}\left(H^{t-1}(x_i)\right)}_{z_i} \right\rangle |0\rangle . \tag{6.2}$$

At this point, we assign conservatively weighted labels to the training set.

Definition 6.1 (Conservative weighting function)**.** A function $w : \mathcal{X} \times \{-1, 1\} \to [0, 1]$ is conservative for any function $h : \mathcal{X} \to \{-1, 1\}$ if $w(x, -h(x)) = 1$, $\forall x \in \mathcal{X}$.

The labels are reweighted according to a potential function ϕ (see Definition 6.2), which is conservatively weighted. This potential function was first used in the smooth boosting algorithm MADABOOST by Domingo and Watanabe [DW00], and later adapted to the POTENTIALBOOST algorithm [KK09].

Definition 6.2 (The QAGBOOST potential function)**.** Consider the conservatively-weighted potential function ϕ as defined in Eq. (6.3) and plotted in Fig. 6.1.

$$\phi(z) = \begin{cases} 1 - z & \text{if } z \leqslant 0 \\ e^{-z} & \text{if } z > 0 \end{cases} . \tag{6.3}$$

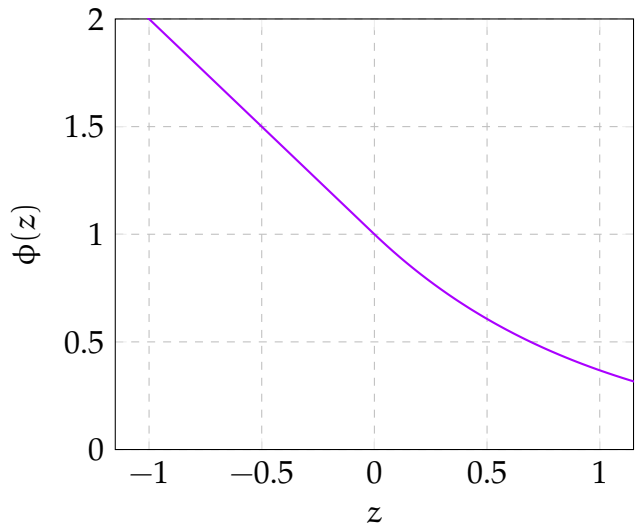


Figure 6.1: The QAGBOOST potential function

Fact 6.1 (Lemma 2 of [KK09])**.** $\phi(z) - \phi(z + \varepsilon) \geqslant -\phi'(z) \cdot \varepsilon - \frac{\varepsilon^2}{2}$.

The second step uses arithmetic operations to compute $w_i^t = \min\left\{1, e^{\{\operatorname{sign}\left(-H^{t-1}(x_i)\right)\cdot y_i\}}\right\}$. We uncompute the $|z_i\rangle$ register using one query to the $O_{H^{t-1}}$ oracle to obtain $2 + m$

copies of the state

$$|\psi_2\rangle = |\phi_2\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle \left|w_i^t\right\rangle . \tag{6.4}$$

Take the first m copies of $|\phi_3\rangle$, and perform arithmetic operations to obtain m copies of the state

$$\frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle \left|\frac{1+w_i^t}{2}\right\rangle . \tag{6.5}$$

Perform a conditional rotation on the third register to obtain the state $|\phi_3\rangle$ as shown in Line 5.

$$|\phi_3\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle \left(\sqrt{\frac{1+w_i^t}{2}}\,|0\rangle + \sqrt{\frac{1-w_i^t}{2}}\,|1\rangle\right) . \tag{6.6}$$

After we perform the C-NOT, we get Q copies of a state $|\phi_4\rangle$ with *conservatively* relabeled samples, as shown in Line 6.

$$|\phi_4\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i\rangle \left(\sqrt{\frac{1+w_i^t}{2}}\,|y_i, 0\rangle + \sqrt{\frac{1-w_i^t}{2}}\,|\bar{y}_i, 1\rangle\right) .$$

▷ Training the Weak Learner to obtain the t^{th} Hypothesis.

We denote the unitary for obtaining $|\phi_4\rangle$ as QAEX_t. Now, we pass QAEX_t to the (m, κ, η)-weak quantum agnostic learner A, to obtain query access to the t^{th} intermediate hypothesis h^t. Note that the weak learner A obtains the intermediate hypothesis using QAEX_t as the quantum example oracle instead of $\text{QAEX}(\mathfrak{D})$.

Remark 6.2.1. We note here that the quantum weak agnostic learner that the QAGBOOST algorithm actually boosts has a very specific requirement - it only produces a weak hypothesis with access to a ε-**strongly biased oracle**, instead of access to a $\text{QAEX}(\mathfrak{D})$ oracle. We shall explore this requirement in detail later in Chapter 4. This constraint can be relaxed for the purposes of the QAGBOOST algorithm, but it helps us capture the entire picture if carry this restriction along with us for the time being.

▷ Obtaining the combined classifier for the next round.

At this point, we have two copies of $|\psi_2\rangle$ left over. On the first copy, use the O_{h^t} oracle to obtain

$$\left|\psi_3^1\right\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} |x_i, y_i\rangle \left|w_i^t\right\rangle \left|w_i^t \cdot y_i \cdot h^t(x_i)\right\rangle . \tag{6.7}$$

Perform a conditional rotation on the last register to obtain

$$\left|\psi_4^1\right\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \sqrt{\kappa_i} \left|x_i, y_i\right\rangle \left|w_i^t\right\rangle \left|\kappa_i\right\rangle \left|1\right\rangle \tag{6.8}$$

$$+ \frac{1}{\sqrt{m}} \sum_{i\in[m]} \sqrt{1-\kappa_i} \left|x_i, y_i\right\rangle \left|w_i^t\right\rangle \left|\kappa_i\right\rangle \left|0\right\rangle \tag{6.9}$$

where $\kappa_i = w_i^t \cdot y_i \cdot h^t(x_i)$. We can rewrite the first part as

$$\sqrt{\alpha_t} \sum_{i\in[m]} \sqrt{\frac{\kappa_i}{\sum_{i\in[m]} \kappa_i}} \left|x_i, y_i\right\rangle \left|w_i^t, \kappa_i, 1\right\rangle. \tag{6.10}$$

We perform quantum amplitude estimation with relative error ε, conditioned on the $|1\rangle$ register, to obtain an estimate $\tilde{\alpha}_t$. On the second copy, use the $O_{H^{t-1}}$ oracle to obtain

$$\left|\psi_3^2\right\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \left|x_i, y_i\right\rangle \left|w_i^t\right\rangle \left|w_i^t \cdot y_i \cdot -H^{t-1}(x_i)\right\rangle. \tag{6.11}$$

Let $\kappa_i = w_i^t \cdot y_i \cdot -\operatorname{sign}\left(H^{t-1}(x_i)\right)$. Perform a conditional rotation on the last register to obtain the state

$$\begin{aligned}\left|\psi_4^2\right\rangle = \frac{1}{\sqrt{m}} \sum_{i\in[m]} \sqrt{\kappa_i} \left|x_i, y_i\right\rangle \left|w_i^t\right\rangle \left|\kappa_i\right\rangle \left|1\right\rangle \\ + \frac{1}{\sqrt{m}} \sum_{i\in[m]} \sqrt{1-\kappa_i} \left|x_i, y_i\right\rangle \left|w_i^t\right\rangle \left|\kappa_i\right\rangle \left|0\right\rangle.\end{aligned} \tag{6.12}$$

We can rewrite the first part as

$$\sqrt{\beta_t} \sum_{i\in[m]} \sqrt{\frac{\kappa_i}{\sum_{i\in[m]} \kappa_i}} \left|x_i, y_i\right\rangle \left|w_i^t, \kappa_i, 1\right\rangle. \tag{6.13}$$

We perform amplitude estimation with relative error ε to obtain the estimate $\tilde{\beta}_t$.

Observation 6.2.1. The quantities α_t and β_t are the margins induced by the classifiers h^t and $-\operatorname{sign}\left(H^{t-1}\right)$ respectively. The quantities $\tilde{\alpha}_t$ and $\tilde{\beta}_t$ are estimates of α_t and β_t.

If the margin of h^t is greater than the margin of $-\operatorname{sign}\left(H^{t-1}\right)$, then the combined classifier for the $(t+1)^{\text{th}}$ iteration is the weighted ensemble $H^{t-1} + \tilde{\alpha}_t h^t$. Otherwise, the combined classifier for the $(t+1)^{\text{th}}$ iteration is $1 - \tilde{\beta}_t H^{t-1}$. In the next section, we prove that such a choice leads to a β-optimal quantum agnostic learner.

Remark 6.2.2. A lot of our analysis follows from the analysis of the classical Algo-

rithm B.5 as shown by Kanade and Kalai [KK09].

§ 6.2.2 Proof of Correctness

In this section, we prove that Algorithm 6.1 actually produces a β-optimal quantum agnostic learner. First, we would like to recall a few definitions.

Definition 4.3 ((η, κ, δ)-Weak agnostic learner)**.** For any $\kappa = 1/\mathrm{poly}(n)$ and $\delta > 0$, a learner A is a (η, κ, δ)-weak agnostic PAC learner for $\mathcal{C}$, if for any choice of distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$, A outputs hypothesis h s.t.

$$\Pr\left[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant \eta \cdot \mathrm{optcorr}_{\mathfrak{D}}(\mathcal{C}) - \kappa\right] \geqslant 1 - \delta.$$

Definition 3.2 (β-optimal (ε, δ)-agnostic PAC learning)**.** A learner A β-optimally (ε, δ)-agnostic PAC learns a benchmark concept class $\mathcal{C}$ over n-bit instances, with query access to $\mathrm{AEX}(\mathfrak{D})$, using hypothesis class $\mathcal{H}_A$, if for every $\varepsilon, \delta \in (0,1)$, for some $\beta \in [0, 1/2)$, for every distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathcal{Y}$, A makes at most $m = \mathrm{poly}(1/\varepsilon, 1/\delta, n, \mathrm{size}(\mathcal{H}_A))$ queries to $\mathrm{AEX}(\mathfrak{D})$ and outputs h s.t. $\Pr\left[\mathrm{err}_{\mathfrak{D}}(h) \leqslant \mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) + \beta + \varepsilon\right] \geqslant 1 - \delta$ for some $\beta \in [0, 1/2)$. Equivalently, in terms of correlation, $\Pr\left[\mathrm{corr}_{\mathfrak{D},c}(h) \geqslant \mathrm{optcorr}_{\mathfrak{D}}(\mathcal{C}) - 2\beta - 2\varepsilon\right] \geqslant 1 - \delta$..

Definition 2.4 (Correlation)**.** The correlation of a hypothesis $h \in \mathcal{H} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ w.r.t. an unknown concept $c \in \mathcal{C} : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ and an unknown underlying distribution $\mathfrak{D}$ is defined as:

$$\mathrm{corr}_{\mathfrak{D},c}(h) := \mathop{\mathbb{E}}_{\mathfrak{D}}\left[h(x) \cdot c(x)\right] = 1 - 2\,\mathrm{err}_{\mathfrak{D},c}(h). \tag{2.5}$$

Remark 6.2.3. Let $\mathfrak{D}$ be any arbitrary joint distribution over $\mathcal{X} \times \{-1, 1\}$. We denote the resulting relabeled distribution[2] (relabeled using any weighting function $w : \mathcal{X} \times \{-1, 1\} \to [0, 1]$) by $\mathfrak{D}'_w$.

Claim 6.1. Given any arbitrary distribution $\mathfrak{D}$ over $\mathcal{X} \times \{-1, 1\}$, an optimal classifier c and a classifier h s.t. $c, h : \mathcal{X} \to [-1, 1]$, and a weighting function $w : \mathcal{X} \times \{-1, 1\} \to [0, 1]$ which is conservative for h,

$$\mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) \geqslant \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h).$$

[2] Technically, this is $\mathfrak{D}'_{w^t}$, but the usage should be apparent from the context.

Proof. From Algorithm 6.1, we can see that $\mathop{\mathbb{E}}_{\{x,y\}\in\mathfrak{D}'_w}[h(x)\cdot y] = \mathop{\mathbb{E}}_{\{x,y\}\in\mathfrak{D}}[h(x)\cdot y\cdot w(x,y)]$. We now evaluate the quantity $\mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h)$ using Definition 2.4.

$$\mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) \tag{6.14}$$
$$= \mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) + \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h) - \mathrm{corr}_{\mathfrak{D}}(c) + \mathrm{corr}_{\mathfrak{D}}(h) \tag{6.15}$$
$$= \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h) + \mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) + \mathrm{corr}_{\mathfrak{D}}(h) \tag{6.16}$$
$$= \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h) + \mathop{\mathbb{E}}_{\mathfrak{D}}[c(x)\cdot y\cdot(1-w(x,y))] - \mathop{\mathbb{E}}_{\mathfrak{D}}[h(x)\cdot y\cdot(1-w(x,y))] \tag{6.17}$$
$$= \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h) - \mathop{\mathbb{E}}_{\mathfrak{D}}[(c(x)-h(x))\cdot y\cdot(1-w(x,y))]. \tag{6.18}$$

The proof follows from the definition of conservative weighting(Definition 6.1), and the fact that $w(x,y) = -\phi'(h(x)y)$. When $h(x) = y$, we have $w(x,y) = 1/e \implies 1-w(x,y) > 0$, and $c(x)\cdot y \leqslant 1$ (true for any classifier). Therefore,

$$\mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) \geqslant \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h).$$

Alternatively, when $h(x) = -y$, we have $w(x,y) = 1 \implies 1-w(x,y) = 0$ which implies

$$\mathrm{corr}_{\mathfrak{D}'_w}(c) - \mathrm{corr}_{\mathfrak{D}'_w}(h) = \mathrm{corr}_{\mathfrak{D}}(c) - \mathrm{corr}_{\mathfrak{D}}(h).$$

□

Consider the case when $\mathrm{corr}_{\mathfrak{D}'_w}(\mathcal{C}) = 0$. In this case, the optimal classifier behaves like a random guesser under the relabeled distribution. Therefore, either the combined classifier H^{t-1} is worse than random guessing (since it was used to set the weights for relabeling), and we should use its negation as a weak agnostic learner, or the hypothesis returned by the weak learner trained on the relabeled distribution is close to optimal. Therefore, we need to pick either of these to add to the combined classifier for the next iteration. The selected hypothesis is denoted by g^t. The QAGBOOST combines the existing combined classifier and g^t (weighted by its correlation γ^t) to form the combined classifier for the t^{th} iteration. We now state a result that lower-bounds the drop in potential at every iteration t.

Claim 6.2 (Drop In Potential at every iteration)**.** Given any function $H : \mathcal{X} \to \mathbb{R}$, hypothesis $h : \mathcal{X} \to [-1,1]$, a weight $\gamma \in \mathbb{R}$, an arbitrary joint distribution $\mathfrak{D} \sim \mathcal{X}\times\{-1,1\}$, a conservative weighting function $w(x,y) = -\phi'(y\cdot H(x))$, and a

relabeled distribution $\mathfrak{D}'_w$

$$\mathop{\mathbb{E}}_{\{x,y\}\sim\mathfrak{D}}\left[\phi(y\cdot H(x))\right]-\mathop{\mathbb{E}}_{\{x,y\}\sim\mathfrak{D}}\left[\phi\left(y\cdot(H+\gamma h)(x)\right)\right]\geqslant \mathrm{corr}_{\mathfrak{D}'_w}(h)-\frac{\gamma^2}{2}.$$

The proof follows directly by plugging in appropriate values for z and ε in Fact 6.1 and taking an expectation over both sides. If any algorithm achieves the bounds set by Claim 6.2, then the number of iterations needed by that algorithm to converge is immediately upper-bounded.

Since the QAGBOOST algorithm satisfies Claim 6.2, it produces a combined classifier H^t on round t, which has a lower potential than H^{t-1} until the potential eventually drops from 1 in iteration $t=1$ to (or gets arbitrarily close to) 0 for some iteration $\hat{t}$. Since there is a lower bound on how much the potential can drop every round, this gives us an upper bound on the number of iterations until the QAGBOOST algorithm converges. Finally, we see that when the potential drops to its lowest value, the combined classifier $H^{\hat{t}}$ qualifies as an agnostic learner.

For a large enough training set size m, we can give a tight enough estimate for the correlation of the new classifier g^t, which is an (η,κ,δ)-weak agnostic learner. We also see from Claim 6.2 that a confidence-based weighted combination drops the potential, and we can lower bound this drop in potential. Therefore, we can obtain an upper bound on the number of iterations of Algorithm 6.1, such that the potential function eventually reaches the minimum possible value. The proof follows from the fact that when the potential function reaches the minimum possible value, the corresponding combined classifier is a (κ/η)-optimal agnostic learner.

Claim 6.3. Either the weak hypothesis produced by the (η,κ,δ)-quantum agnostic weak learner in Algorithm 6.1 on the t^{th} iteration, or the negation of the combined hypotheses up to the $t-1^{\text{th}}$ step has a correlation greater than $\frac{\eta\varepsilon}{3}$.

Proof. Consider the optimal hypothesis $c\in\mathcal{C}$, and the combined hypothesis produced by Algorithm 6.1 at iteration $t-1$ to be H^{t-1}. Let $\beta=\frac{\kappa}{\eta}$. We now consider two cases:

1. $\mathrm{corr}_{\mathfrak{D}'_w}(c)\geqslant\beta+\frac{\varepsilon}{2}$. Consider the hypothesis h^t produced by the weak learner at the t^{th} iteration in Algorithm 6.1. By the weak learning assumption (see Definition 4.3), we have $\mathrm{corr}_{\mathfrak{D}'_w}(h^t)\geqslant\eta\cdot\mathrm{corr}_{\mathfrak{D}'_w}(c)-\kappa$. Hence, $\mathrm{corr}_{\mathfrak{D}'_w}(h^t)\geqslant\frac{\eta\varepsilon}{2}$.
2. $\mathrm{corr}_{\mathfrak{D}'_w}(c)<\beta+\frac{\varepsilon}{2}$. If H^{t-1} is not a β-optimal agnostic learner, then by Definition 3.2, we have

$$\mathrm{corr}_{\mathfrak{D}}(c)>\mathrm{corr}_{\mathfrak{D}}\left(H^{t-1}\right)+\beta+\varepsilon. \tag{6.19}$$

From Claim 6.1 and Eq. (6.19), we have

$$\operatorname{corr}_{\mathcal{D}'_w}(c) > \operatorname{corr}_{\mathcal{D}'_w}\left(H^{t-1}\right) + \beta + \varepsilon. \tag{6.20}$$

From the condition on $\operatorname{corr}_{\mathcal{D}'_w}(c)$ and Eq. (6.20), we have for any $\eta \in (0,1)$,

$$\begin{aligned}
\beta + \frac{\varepsilon}{2} > \operatorname{corr}_{\mathcal{D}'_w}(c) > \operatorname{corr}_{\mathcal{D}'_w}\left(H^{t-1}\right) + \beta + \varepsilon & \tag{6.21}\\
\implies \operatorname{corr}_{\mathcal{D}'_w}\left(H^{t-1}\right) < -\frac{\varepsilon}{2} & \tag{6.22}\\
\implies -\operatorname{corr}_{\mathcal{D}'_w}\left(H^{t-1}\right) > \frac{\varepsilon}{2} & \tag{6.23}\\
\implies \operatorname{corr}_{\mathcal{D}'_w}\left(-H^{t-1}\right) > \frac{\varepsilon}{2} > \frac{\eta\varepsilon}{2}. & \tag{6.24}
\end{aligned}$$

The first inequality in Eq. (6.24) follows from Definition 2.4 and the linearity of expectation. Combining both cases gives us the desired bound. □

We now state and prove the following claim which shows that we can estimate the correlation of the best classifier g^t at every step $t > 0$ with a high probability.

Claim 6.4. Algorithm 6.1 computes estimates of margins $\tilde{\alpha}_t$ and $\tilde{\beta}_t$ s.t. $\tilde{\gamma}_t = \max(\tilde{\alpha}_t, \tilde{\beta}_t)$ s.t. $\left|\tilde{\gamma}^t - \operatorname{corr}_{\mathcal{D}'_{w^t}}(g^t)\right| \leqslant \eta\varepsilon/10$ with probability $\geqslant 1 - 3\delta T$.

Proof. Let $g^t : \mathbb{F}_2^n \mapsto \{\pm 1\}$ be the classifier chosen by Algorithm 6.1 at the t^{th} iteration. We denote the correlation of g^t w.r.t. the relabeled distribution as $\operatorname{corr}_{\mathcal{D}'_w}(g^t)$. From Definition 2.4, we can restate this as

$$\operatorname{corr}_{\mathcal{D}'_{w^t}}(g^t) = \mathop{\mathbb{E}}_{x_i, y_i \sim \mathcal{D}}\left[w_i^t \cdot y_i \cdot g^t(x_i)\right]. \tag{6.25}$$

Let $\mathbf{X}_i = w_i^t \cdot y_i \cdot g^t(x_i)$ be a random variable. From the definition of conservative weighting (see Definition 6.1), we obtain $\mathbf{X}_i \in \left[-\frac{1}{e}, 1\right]$. Let $\gamma^t = \frac{1}{m}\sum_{i\in[m]} \mathbf{X}_i$. Applying Chernoff-Hoeffding bounds (see Lemma 2.1) and an appropriate value of δ gives us

$$\Pr\left[\left|\gamma^t - \operatorname{corr}_{\mathcal{D}'_{w^t}}(g^t)\right| \geqslant \frac{\eta\varepsilon}{20}\right] \leqslant 2 \cdot \exp\left(\frac{-2\frac{\eta^2\varepsilon^2}{400}}{\sum_{i=1}^{m}\left(1+\frac{1}{e}\right)^2}\right). \tag{6.26}$$

By setting $m = \frac{200}{\eta^2\varepsilon^2}\log\frac{1}{\delta}$ in Eq. (6.26), we can obtain with probability at least $1 - 2\delta$,

$$\left|\gamma^t - \operatorname{corr}_{\mathcal{D}'_w}(g^t)\right| \leqslant \frac{\eta\varepsilon}{20}. \tag{6.27}$$

Recall from Claim 6.3 that $\gamma^t > \frac{\eta\varepsilon}{3}$. In Algorithm 6.1, we obtain an estimate $\tilde{\gamma}^t$ of γ^t

using Lemma 2.9 with probability at least $1-\delta$, s.t.

$$\left|\tilde{\gamma}^t - \gamma^t\right| \leqslant \varepsilon \cdot \gamma^t. \tag{6.28}$$

From Eq. (6.28) and Claim 6.3, we observe that even the estimated correlation is tightly coupled to the actual correlation. Hence, we always end up choosing the classifier with the better correlation. Using triangle inequality on Eq. (6.27) and Eq. (6.28), we obtain with probability at least $1-3\delta$,

$$\begin{aligned}\left|\tilde{\gamma}^t - \operatorname{corr}_{\mathfrak{D}'_{w^t}}\left(g^t\right)\right| &\leqslant \left|\tilde{\gamma}^t - \gamma^t + \gamma^t - \operatorname{corr}_{g^t}\left(\mathfrak{D}'_{w^t}\right)\right| &(6.29)\\ &\leqslant \left|\tilde{\gamma}^t - \gamma^t\right| + \left|\gamma^t - \operatorname{corr}_{g^t}\left(\mathfrak{D}'_{w^t}\right)\right| \leqslant \epsilon \cdot \gamma^t + \frac{\eta\varepsilon}{20}. &(6.30)\end{aligned}$$

In the last step we observe that γ^t can be at most 1. Setting $\epsilon = \frac{\eta\varepsilon}{20}$ gives us the required upper-bound on $\left|\tilde{\gamma}^t - \operatorname{corr}_{\mathfrak{D}'_{w^t}}\left(g^t\right)\right|$. □

Theorem 6.5. Algorithm 6.1 takes as input an (η, κ, δ)-weak quantum agnostic learner and outputs a (κ/η)-quantum agnostic learner with a probability of failure of at most $5\delta T$.

Proof. From Claim 6.2 and Claim 6.4, we obtain that the drop in potential for Algorithm 6.1 at every iteration is bounded by at most $O\left(\eta^2\varepsilon^2\right)$. Since $\eta \in \left[0, \frac{1}{2}\right)$, we have from Claim 6.3 that $\operatorname{corr}_{\mathfrak{D}'_w}\left(g^t\right) \geqslant \frac{\eta\varepsilon}{3}$, where g^t is the better of the two candidate hypotheses at iteration t. Now, consider the margin $\gamma^t t = \frac{1}{m}\sum_{i\in[m]} g^t(x_i)\cdot y_i \cdot w^t(x_i, y_i)$ of the best classifier g^t at iteration t obtained using m training samples. This margin is simply the estimated correlation of g^t. Algorithm 6.1 further estimates γ^t as $\tilde{\gamma}^t$. From Claim 6.4, we have $\left|\tilde{\gamma}^t - \operatorname{corr}_{\mathfrak{D}'_w}\left(g^t\right)\right| \leqslant O\left(\eta\varepsilon\right)$ with high probability. Setting the appropriate values for $\operatorname{corr}_{\mathfrak{D}'_w}\left(g^t\right)$ allows us to lower bound the potential drop to at least $O\left(\eta^2\varepsilon^2\right)$ for any iteration $t > 0$ using Claim 6.2.

Since the potential function is bounded in the range $[0, 1]$, and the potential drops by at least $O\left(\eta^2\varepsilon^2\right)$, in $O\left(\frac{1}{\eta^2\varepsilon^2}\right)$ iterations, Algorithm 6.1 must produce a hypothesis such that the potential function drops to its lowest value. Let us denote the iteration τ in which the potential drops to its lowest as τ. From Claim 6.1 we have

$$\begin{aligned}&\operatorname{corr}_{\mathfrak{D}'_w}(c) - \operatorname{corr}_{\mathfrak{D}'_w}\left(g^\tau\right) \geqslant \operatorname{corr}_{\mathfrak{D}}(c) - \operatorname{corr}_{\mathfrak{D}}\left(g^\tau\right)\\ \implies &\operatorname{corr}_{\mathfrak{D}}\left(g^\tau\right) \geqslant \operatorname{corr}_{\mathfrak{D}}(c) - \left[\operatorname{corr}_{\mathfrak{D}'_w}(c) - \operatorname{corr}_{\mathfrak{D}'_w}\left(g^\tau\right)\right].\end{aligned}$$

Substituting $\operatorname{corr}_{\mathfrak{D}'_w}(c) > \frac{\kappa}{\eta} + \frac{\varepsilon}{2}$ (since the potential is lowest at this iteration) and $\operatorname{corr}_{\mathfrak{D}'_w}\left(g^\tau\right) \geqslant \frac{\eta\varepsilon}{3}$, we have $\operatorname{corr}_{\mathfrak{D}}\left(H^\tau\right) \geqslant \operatorname{corr}_{\mathfrak{D}}(c) - \frac{\kappa}{\eta} - \varepsilon$. Therefore, we have that in

$O\left(\frac{1}{\eta^2\varepsilon^2}\right)$ iterations, Algorithm 6.1 produces a $\left(\frac{\kappa}{\eta}\right)$-optimal agnostic learner.

We allow the algorithm to fail with probability 3δ during estimation of $\tilde{\gamma}^t$ (see Claim 6.4). We allow the algorithm to fail with another δ probability while invoking the weak learner to produce a hypothesis h^t at the t^{th} iteration. Finally, estimating the correlation of the constructed hypothesis g^t can fail with an additional probability of δ at every iteration. □

§ 6.2.3 Complexity Analysis

Claim 6.6. Algorithm 6.1 computes estimates of margins $\tilde{\alpha}_t$ and $\tilde{\beta}_t$ s.t. $\tilde{\gamma}_t = \max\left(\tilde{\alpha}_t, \tilde{\beta}_t\right)$ using $\tilde{O}\left(\frac{1}{\eta\varepsilon}\sqrt{m}\log\frac{1}{\delta}\right)$ queries.

Proof. We get the required query complexity by plugging the terms of Eq. (6.28) into Lemma 2.9. □

Theorem 6.7. Given a quantum (η, κ, δ)-Weak agnostic learner A with an associated Hypothesis class having a VC dimension of at most $d > 0$, Algorithm 6.1 makes at most $\tilde{O}\left(\frac{1}{\eta^4\varepsilon^3}\sqrt{d}\log\frac{1}{\delta}\right)$ queries to A and takes an additional running time of $\tilde{O}\left(\frac{n^2}{\eta^4\varepsilon^3}\sqrt{d}\log\frac{1}{\delta}\right)$.

Proof. The quantum algorithm runs for $O\left(\frac{1}{\eta^2\varepsilon^2}\right)$ iterations (see Theorem 6.5). From Claim 6.6, we see that each iteration makes $\tilde{O}\left(\frac{\sqrt{m}}{\eta\varepsilon}\log\frac{1}{\delta}\right)$ queries. Plugging in sample complexity bounds from Lemma 9.1, we have $m = \tilde{\Theta}\left(\frac{d}{\eta^2}\right)$, where d is the VC-dimension of the $\left(\frac{\kappa}{\eta}\right)$-optimal agnostic learner. This gives us a total of $\tilde{O}\left(\frac{\sqrt{d}}{\eta^4\varepsilon^3}\log\frac{1}{\delta}\right)$ queries made by Algorithm 6.1. Arithmetic operations in the algorithm add, at most, a multiplicative factor of n^2 to give us our required time complexity. □

§ 6.2.4 Application: Efficient Agnostic Learners for Decision Trees w/o MQ access

Ehrenfeucht and Haussler [EH89] gave the first *weakly* proper learning algorithm with quasi-polynomial running time and sample-complexity in the realizable setting using random examples. Subsequent works on properly learning decision trees[MR02; BLT20; Bla+20; Bla+22] either have quasi-polynomial dependence on error parameters and intensive memory requirements or require the use of Membership Query

(MQ) oracles (see Table 9.1). Recently, it was shown by Koch et al. [KST23] that efficient proper learning of decision trees has a superpolynomial lower bound. Bshouty [Bsh23] showed that the superpolynomial lower bound also holds for proper learning of monotone decision trees.

Kushilevitz and Mansour [KM91] gave the first polynomial time improper decision tree learning algorithm (we henceforth refer to this as the KM algorithm) using MQ in the realizable setting. Their approach was later extended to the agnostic setting by Gopalan et al. [GKK08a], Kanade and Kalai [KK09], and Feldman [Fel10] who constructed weak learners for size-t decision trees using MQ oracles. In this section, we construct quantum learners for size-t decision trees using QEX and QAEX oracles, which we have previously shown to be weaker than MQ oracles. We now state the following application of Algorithm 6.1.

Theorem 6.8 (Agnostic Learning polynomial sized Decision Trees)**.** For any $\delta > 0$, $\varepsilon \in (0, 1/2)$, there exists a quantum learning algorithm with VC dimension d that makes $\tilde{O}\left(\frac{nt^5\sqrt{d}}{\varepsilon^6}\log(1/\delta)\right)$ queries to a QAEX oracle and takes an additional $\tilde{O}\left(\frac{n^3t^5\sqrt{d}}{\varepsilon^6}\log(1/\delta)\right)$ time for (εt)-optimal agnostic PAC learning size-t decision trees on n-bits.

Proof. We use the weak quantum agnostic learner for size-t decision trees constructed in Algorithm 4.1 (set $\kappa = \epsilon$ and $\eta = \frac{1}{t}$ in Theorem 4.10) as a weak learner for the quantum agnostic boosting algorithm as described in Algorithm 6.1. By Theorem 6.5, the output of Algorithm 6.1 is a $(t\varepsilon)$-optimal agnostic learner for size-t decision trees. We get the required query and time complexity from Theorems 4.10 and 6.7. □

See Fig. 6.2 for an overview of our algorithm.

§ 6.3 Discussion

We end our discussion on quantum agnostic boosting algorithms with a very interesting open question for the reader:

Open Problem 5. What is the lower bound for boosting a (η, κ, δ) classical or quantum weak agnostic learner to a β-optimal classical or quantum agnostic learner (respectively)?

Although such lower bounds for boosting have been investigated recently for the classical realizable case (see the results by Høgsgaard et al. [HLR23]), to the best of our knowledge, no such explicit lower bounds are known for agnostic boosting. The relabeling step in Algorithm 6.1 removes the need for amplitude amplification and also results in a smooth boosting algorithm, leading us to the following question.

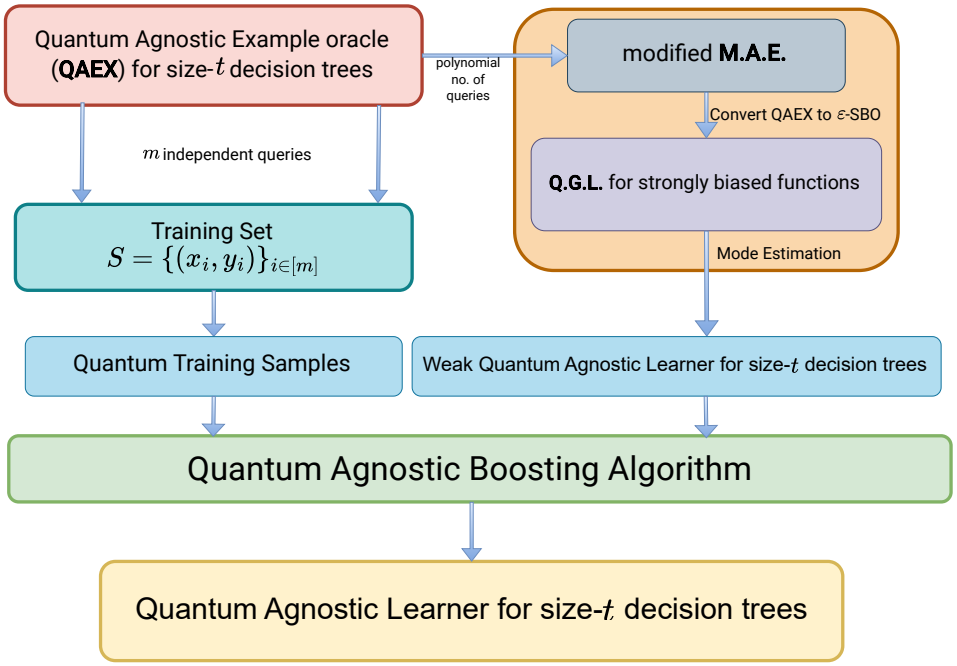


Figure 6.2: Schematic for agnostically learning polynomial-sized decision trees without MQ.

Open Problem 6. Can we use ideas similar to relabeling to obtain a smooth adaptive quantum boosting algorithm in the realizable, hopefully with better query and time complexity than QREALBOOST?

Part III

Generalization Error of Statistical Learners

In the margin for error lies all our room for maneuver.

JAMES GEARY

Chapter 7

Generalization Error and Empirical Behaviour of QREALBOOST

Abstract

In this chapter, we give an alternate analysis of Algorithm 5.1 introduced in Chapter 5 from the point of view of margins. This allows us to obtain a deeper understanding of the empirical behavior of Algorithm 5.1. We supplement our theoretical results by conducting experiments in which we contrast the behavior of QREALBOOST against various realizable classical and quantum boosting algorithms on the `Breast Cancer Wisconsin` [WSM95], and `MNIST` [Den12]. A significant part of this chapter is based on the following publication:

- Sagnik Chatterjee et al. "Quantum boosting using domain-partitioning hypotheses". In: *Quantum Machine Intelligence* 5.2 (July 2023), p. 33. ISSN: 2524-4914. DOI: 10.1007/s42484-023-00122-3

Contents

7.1 Generalization Bounds on QREALBOOST 118
7.1.1 Generalization Error w.r.t. Induced Margins 118
7.1.2 QREALBOOST induces large margins 119
7.2 Empirical evaluations of QREALBOOST 120
7.2.1 Implementation Details 120
7.2.2 Methodology 121
7.2.3 Experimental Details 122
7.2.3.1 The Breast Cancer Wisconsin Dataset 122
7.2.3.2 The MNIST dataset 122

7.3 Discussion and Future Work 128

§ 7.1 Generalization Bounds on QREALBOOST

The ADABOOST algorithm and many of its variants are *adaptive* in nature, which allows these boosting algorithms to converge in training error (and hence exhibit good generalization performance) without understanding the underlying weak learners in a white-box manner. Algorithms that are not adaptive, however, require information about the underlying weak learners to obtain convergence guarantees.

Earlier, we noted in Proposition 5.1 that the training error of ADABOOST drops to zero after roughly $O(\log m/\gamma^2)$ iterations. Interestingly, it was observed in practice [DC95; Qui96; Bre98] that the test error continues to drop with further iterations without exhibiting the pitfalls of overfitting. This semi-paradoxical behavior was explained partially in [Bar+98], where the authors showed that even when the training error drops to zero, the confidence of the combined (final) classifier produced by ADABOOST still improves with more iterations. In this section, we demonstrate the QREALBOOST also exhibits similar behavior.

§ 7.1.1 Generalization Error w.r.t. Induced Margins

In Lemma 2.7, we presented a bound for the generalization error of a hypothesis in terms of the training error of the learner A over the training set and the VC dimension of the corresponding hypothesis space. Bartlett et al. [Bar+98] later proved that a similar bound could be obtained for the generalization error of the learner A in terms of the margin induced over S.

Lemma 7.1 (Margin induced Generalization Error Bounds [Bar+98])**.** Let $\delta, \theta > 0$ be constants, and $\mathcal{H}$ be a hypothesis class with VC-dimension $d_{\mathcal{H}}$. Then for any distribution $\mathcal{D}$ over the instances, any concept $c \in \mathcal{C}$, and any $h \in \mathcal{H}$, we have

$$\Pr\left[\operatorname{err}_{\mathcal{D},c}(h) \leqslant \Pr_{(x,y)\sim S}[y \cdot h(x) \leqslant \theta] + O\left(\sqrt{\frac{\log(m/d_{\mathcal{H}})}{m/d_{\mathcal{H}}}}\right)\right] \geqslant 1-\delta.$$

Bartlett et al. [Bar+98] also initialized their generalization error bounds by showing that ADABOOST induces large margins on the training set as follows:

Lemma 7.2. [Margins induced by ADABOOST [Bar+98]] Let θ be a desired margin parameter. Let ν_i be the margin of an instance $(x_i, y_i) \in S$ induced by the combined classifier produced by ADABOOST. The number of instances such that

$\nu_i < \theta$ drops exponentially fast with the no. of iterations T. Also, $\forall(x_i, y_i) \in \mathcal{S}$, $\nu_i \geqslant \gamma$, where γ is the bias of the weak learner.

§ 7.1.2 QREALBOOST induces large margins

Before we prove that QREALBOOST induces large margins, we generalize the definition of margins (as introduced in Definition 5.1) as follows:

Definition 7.1 (Margins for Confidence-Rated Predictions)**.** For boosting algorithms that output ensembles of confidence-rated predictions, the margin of an instance (x_i, y_i) is defined as $\nu_i = y \cdot \beta_{j^t,t}$, where $x \in \mathcal{X}^t_{j^t}$ for all $t \in [T]$.

We now prove the following theorem:

Theorem 7.1. Let θ be a desired margin parameter. Let ν_i be the margin of an instance (x_i, y_i) in the training set $\mathcal{S}$ induced by the combined classifier produced by QREALBOOST. The number of instances such that $\nu_i < \theta$ drops exponentially fast with the no. of iterations T.

Proof. Let $\nu_i = y \cdot \beta'_{k,t}$, where $x \in \mathcal{X}^t_k$ for all $t \in [T]$. Now,

$$\nu_i \leqslant \theta \iff y_i \cdot \cdot \beta'_{k,t} \leqslant \theta \implies y_i \cdot \sum_t \beta'_{k,t} \leqslant \theta \tag{7.1}$$

$$\implies \nu_i \leqslant \theta \iff 0 \leqslant \theta - y_i \cdot \sum_t \beta'_{k,t} \tag{7.2}$$

$$\implies \nu_i \leqslant \theta \iff 1 \leqslant \exp\left\{\theta - y_i \cdot \sum_t \beta'_{k,t}\right\} \tag{7.3}$$

$$\implies \nu_i \leqslant \theta \iff \mathbb{1}_{[\nu_i \leqslant \theta]} \leqslant \exp\left\{\theta - y_i \cdot \sum_t \beta'_{k,t}\right\}. \tag{7.4}$$

Recall the QREALBOOST update rule as stated in Eq. (5.6).

$$\tilde{\mathfrak{D}}_i^{t+1} = \frac{\tilde{\mathfrak{D}}_i^t \cdot \exp\left(-\beta'_{k,t} \cdot y_i\right)}{\kappa \cdot Z'_t}, \tag{7.5}$$

We can unroll the recurrence relation to express $\tilde{\mathfrak{D}}_i^{t+1}$ in terms of $\tilde{\mathfrak{D}}_i^1$ as

$$\tilde{\mathfrak{D}}_i^{t+1} = \frac{\tilde{\mathfrak{D}}_i^1 \cdot \exp\left(-y_i \cdot \sum_{t=1}^T \beta'_{k,t}\right)}{\kappa^T \cdot \prod_{t=1}^T Z'_t} \tag{7.6}$$

$$\tilde{\mathfrak{D}}_i^{t+1} \kappa^t \prod_{t=1}^T Z'_s = \mathfrak{D}_i^1 \cdot \exp\left(-y_i \cdot \sum_{t=1}^T \beta'_{k,t}\right) \tag{7.7}$$

In Eq. (7.7) we use the fact that $\tilde{\mathfrak{D}}_i^1 = \mathfrak{D}_i^1$. We now bound the probability of the number of instances in the training set s.t. their margins are less than θ.

$$\Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{[\nu_i\leqslant\theta]} \leqslant \frac{e^{\theta}}{m}\sum_{i=1}^{m}\exp\left\{-y_i\cdot\sum_t \beta'_{k,t}\right\}. \tag{7.8}$$

Substituting the value of $\mathfrak{D}_i^1 = \frac{1}{m}$, we get

$$\Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant e^{\theta}\sum_{i=1}^{m}\mathfrak{D}_i^1\exp\left\{-y_i\cdot\sum_t \beta'_{k,t}\right\} \tag{7.9}$$

$$\implies \Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant e^{\theta}k^T\prod_{t=1}^{T} Z'_t \leqslant e^{\theta}(4+\kappa)^T \leqslant e^{\theta}(4+4C)^T. \tag{7.10}$$

In the second inequality of Eq. (7.10), we use the fact that if each partition j contains $\ell \in (0,1)$ fraction of instances, then $Z_t \leqslant 2$, and from Claim 5.4 we have $Z'_t \leqslant Z_t(1+\varepsilon) \leqslant 4$. The third inequality comes by setting $\varepsilon < 0.5$. For a C-domain partitioning learner, we note that the terms e^{θ} and $(4+4C)$ are constants. Hence, the probability that instances in $\mathcal{S}$ have an induced margin less than θ exponentially decreases with T. □

§ 7.2 Empirical evaluations of QREALBOOST

We evaluate the generalization ability and convergence of QREALBOOST algorithm on two datasets: `Breast Cancer Wisconsin` [WSM95], and `MNIST` [Den12]. We compare the performance of QREALBOOST against four alternatives: The classical ADABOOST algorithm [FS97], the classical REALBOOST algorithm [SS99], the classical SMOOTHBOOST algorithm [Ser03], and the QADABOOST algorithm [AM20]. We discuss our design decisions in Section 7.2.1 and focus on our methodology in Section 7.2.2. We discuss our findings on the `Breast Cancer Wisconsin` Dataset and `MNIST` datasets in Section 7.2.3.1 and Section 7.2.3.2 respectively. Our code is freely available at `https://github.com/braqiiit/QRealBoost`.

§ 7.2.1 Implementation Details

Due to the unavailability of quantum simulators or quantum backends large enough to carry out experiments on the QREALBOOST algorithm, we make some design and implementation choices detailed below.

▷ Convergence behaviour instead of Quantum Speedups.

We focus on the qualitative analysis behavior (training and test convergence) of the

algorithms in these experiments rather than their efficiency due to the lack of efficient quantum simulators and quantum backends supporting a sufficient number of qubits. Due to the lack of fault-tolerant physical qubits needed to store the weights up to a reasonable degree of precision, we store the updated weights after every iteration instead of computing the distribution weights from scratch. Even though this choice sacrifices the quantum speedup, it has no bearing on the convergence behavior of QREALBOOST.

We used a weak classical learner (k-means) since off-the-shelf quantum weak learners [Pra14; RML14] are not readily available right now, and implementing one is out of the scope of this thesis. The implementation is modular and can be easily modified for any learner implemented as a quantum circuit. We measure the $|\phi_5\rangle$ state and pass the top Q training samples to the k-means algorithm. We use the Iterative Quantum Amplitude Estimation (IQAE) algorithm [Gri+21] due to its ready availability on `QISKIT` as a library function. IQAE replaces quantum phase estimation with Grover iterations, essentially performing additive estimation. Our choice was motivated by the performance of the IQAE subroutine, which helped us decrease the number of qubits needed for the implementation.

Remark 7.2.1. Even though the experiments were conducted with *additive estimation* instead of *relative estimation*, we still managed to boost the weak learner. Such a trade-off is possible since we are storing the distribution weights explicitly in every iteration, which offsets a lot of errors arising from the probabilistic subroutines. This gives us some insight into efficient implementations of quantum-inspired classical algorithms.

§ 7.2.2 Methodology

We carry out two sets of experiments on both datasets. First, we fix the size of the training set to $M = 32$ samples and evaluate the generalization ability and convergence behavior for different sample complexities $Q = 4, 6, 8$ for both QADABOOST and QREALBOOST. We note here that in our implementation (which uses a weak classical learner), the sample complexity constraint is enforced by having the weak learner sample from the quantum state $|\phi_5\rangle^{\otimes Q}$ produced by the quantum boosting algorithms.

For the classical algorithms, we do not enforce any sample complexity restriction on the weak learner and therefore do not enforce sampling inside the weak learner for the classical boosting algorithms. Hence, K-means samples Q examples from the M samples (with replacement) in every iteration for the quantum examples, while K-means uses all M samples for clustering in the case of the classical algorithms. Therefore, the worst-case guess for Q is M for the classical algorithms. If the ratio Q/M is too low, the underlying learner will underfit to the training set, resulting in high bias

γ and higher training error.

In the next set of experiments, we fix the sample complexity Q of the quantum boosting algorithms to 8 and vary the size of the training set $M = 16, 32, 64$. Then we increase the size of the training set that increases the variance of the underlying learner, which further increases the generalization error. We investigate the generalization performance of our boosting algorithm through these experiments.

In Fig. 7.1, Fig. 7.2, Fig. 7.3, Fig. 7.4, the lines for QADABOOST and QREALBOOST represent a mean accuracy over 5 independent experiments, and the hue bands represent the standard deviation across all experiments. Due to quantum resource limitations, the QADABOOST and QREALBOOST algorithms are tested on quantum simulators (instead of actual quantum backends).

§ 7.2.3 Experimental Details

§ 7.2.3.1 The Breast Cancer Wisconsin Dataset

In Fig. 7.1, we see that the training performance of QREALBOOST matches or exceeds QADABOOST in all 3 cases. Both the quantum algorithms converge fastest for $Q = 8$. As we increase the sample complexity from $Q = 4$ to $Q = 8$, QREALBOOST increases its generalization accuracy, unlike QADABOOST. In Fig. 7.2, we see that for $M = 16$, QREALBOOST has higher training accuracy than QADABOOST and similar generalization performance (QAdaBoost outperforms QREALBOOST by a hair). For $M = 16$, the classical ADABOOST algorithm does not converge, but generally has a good test accuracy. The classical REALBOOST algorithm overfits during training, and has poor test accuracy. The classical SMOOTHBOOST algorithm does not converge and has extremely poor training and test accuracy for $M = 16$. For $M = 32$, QREALBOOST has faster convergence and higher training accuracy than QADABOOST but suffers a drop in generalization performance. ADABOOST and REALBOOST converge faster than the quantum algorithms and perform better generalization. The classical SMOOTHBOOST algorithm converges faster than the quantum algorithms but has the worst generalization performance out of all the algorithms. Finally, for $M = 64$, both QREALBOOST and QADABOOST converge fast to a high training accuracy and with good generalization performance. ADABOOST and SMOOTHBOOST do not converge for $M = 64$, while REALBOOST has poor training and test performance.

§ 7.2.3.2 The MNIST dataset

On the `MNIST` dataset, we see (refer Fig. 7.3) that for all three cases of sample complexity $Q = \{4, 6, 8\}$, quantum boosting algorithms outperform classical boosting algorithms

in generalization accuracy. As expected, for the cases of $Q = \{4, 6\}$, the convergence is slow for both QREALBOOST and QADABOOST, compared to the case where $Q = 8$. We found that for `MNIST`, QADABOOST and QREALBOOST behaved similarly in every sample complexity variation. The above trend continues even under variations on the training set sizes (refer Fig. 7.4). Quantum algorithms converge faster, have higher training accuracy, and have better generalization performance than classical boosting algorithms for all cases. Once again, QADABOOST and QREALBOOST have similar performances.

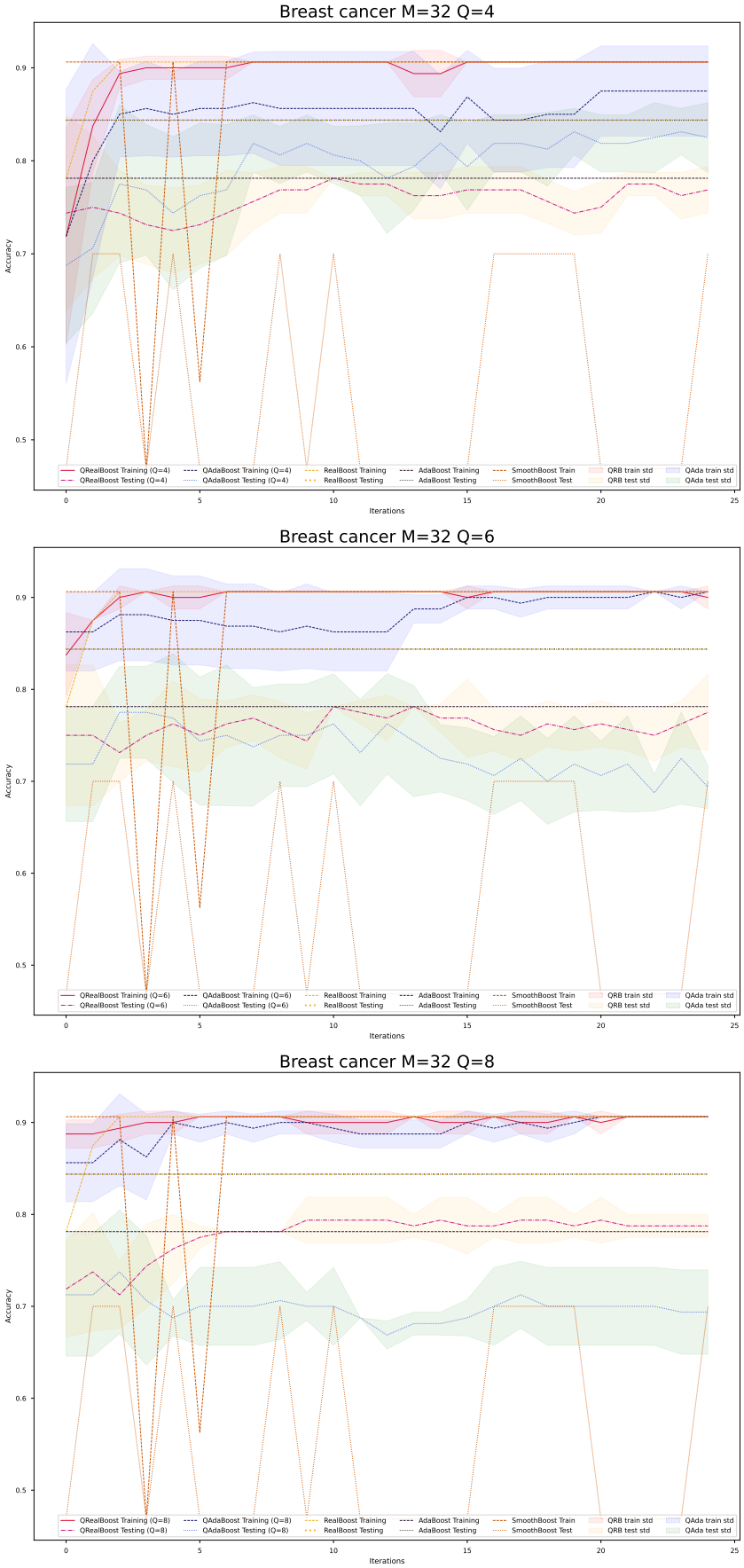


Figure 7.1: Performance of QREALBOOST, QADABOOST, REALBOOST, ADABOOST, and SMOOTHBOOST on the `Breast Cancer Wisconsin` Dataset with different sample complexities.

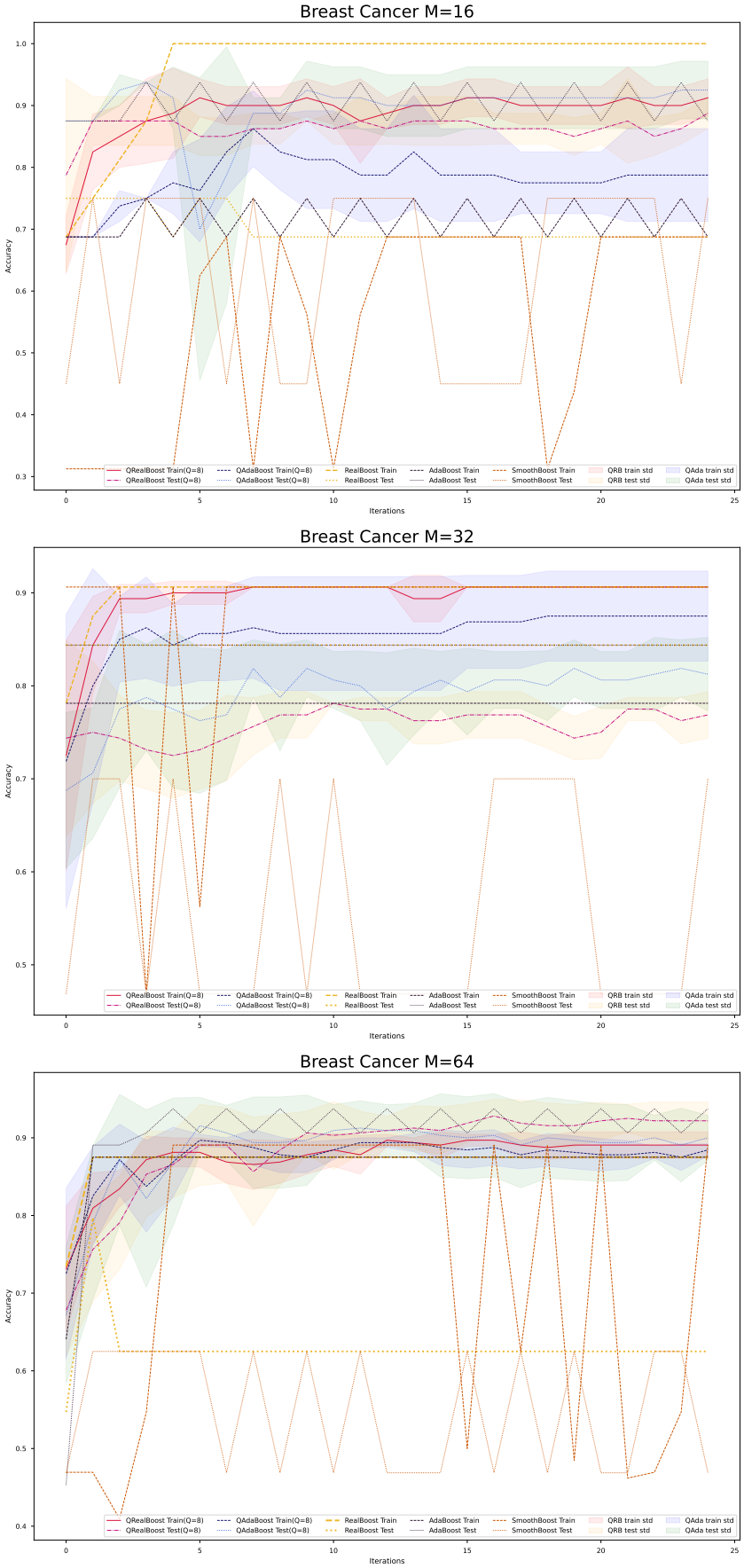


Figure 7.2: Performance of QREALBOOST, QADABOOST, REALBOOST, ADABOOST, and SMOOTHBOOST on the `Breast Cancer Wisconsin` Dataset with different training set sizes.

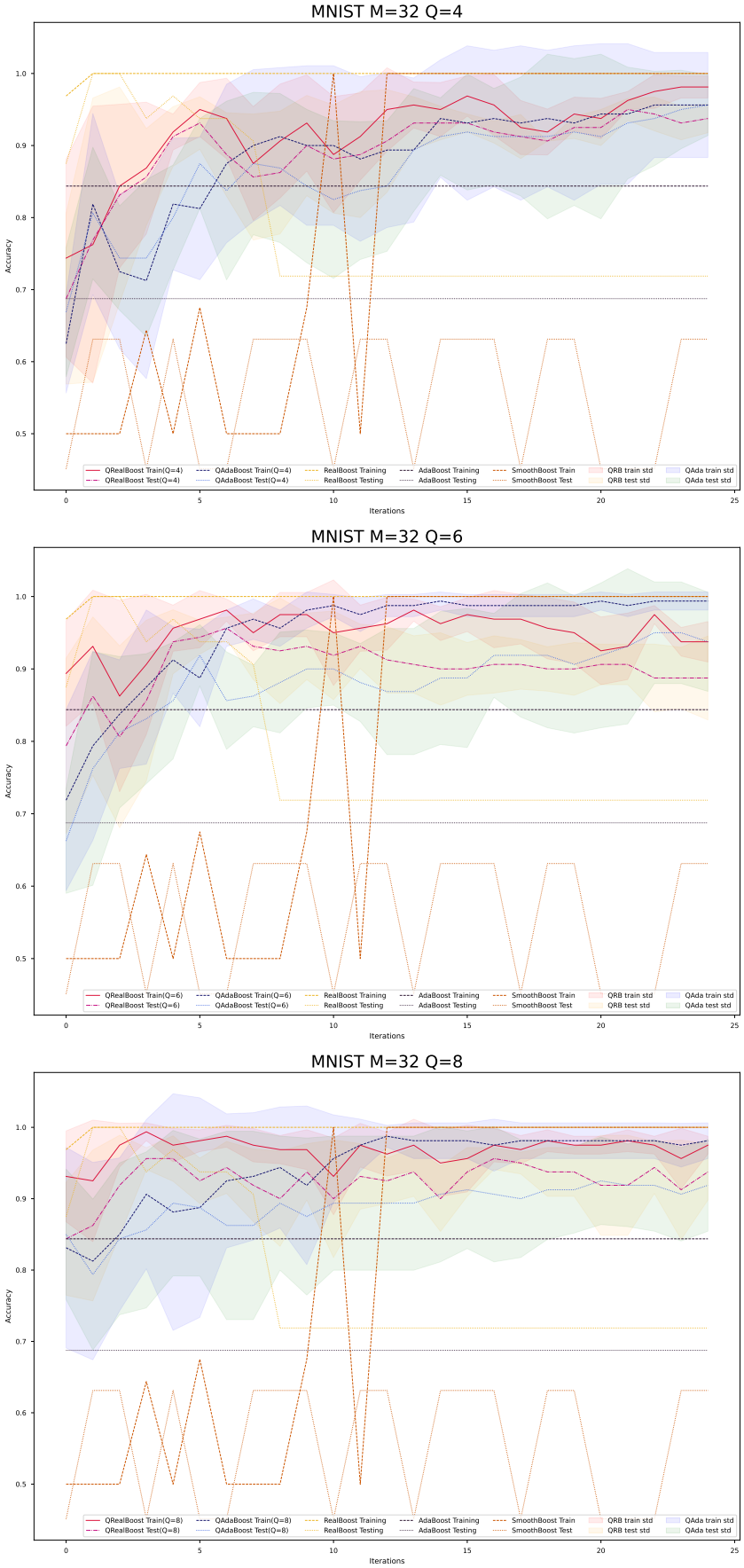


Figure 7.3: Performance of QREALBOOST, QADABOOST, REALBOOST, ADABOOST, and SMOOTHBOOST on the `MNIST` Dataset with different sample complexities.

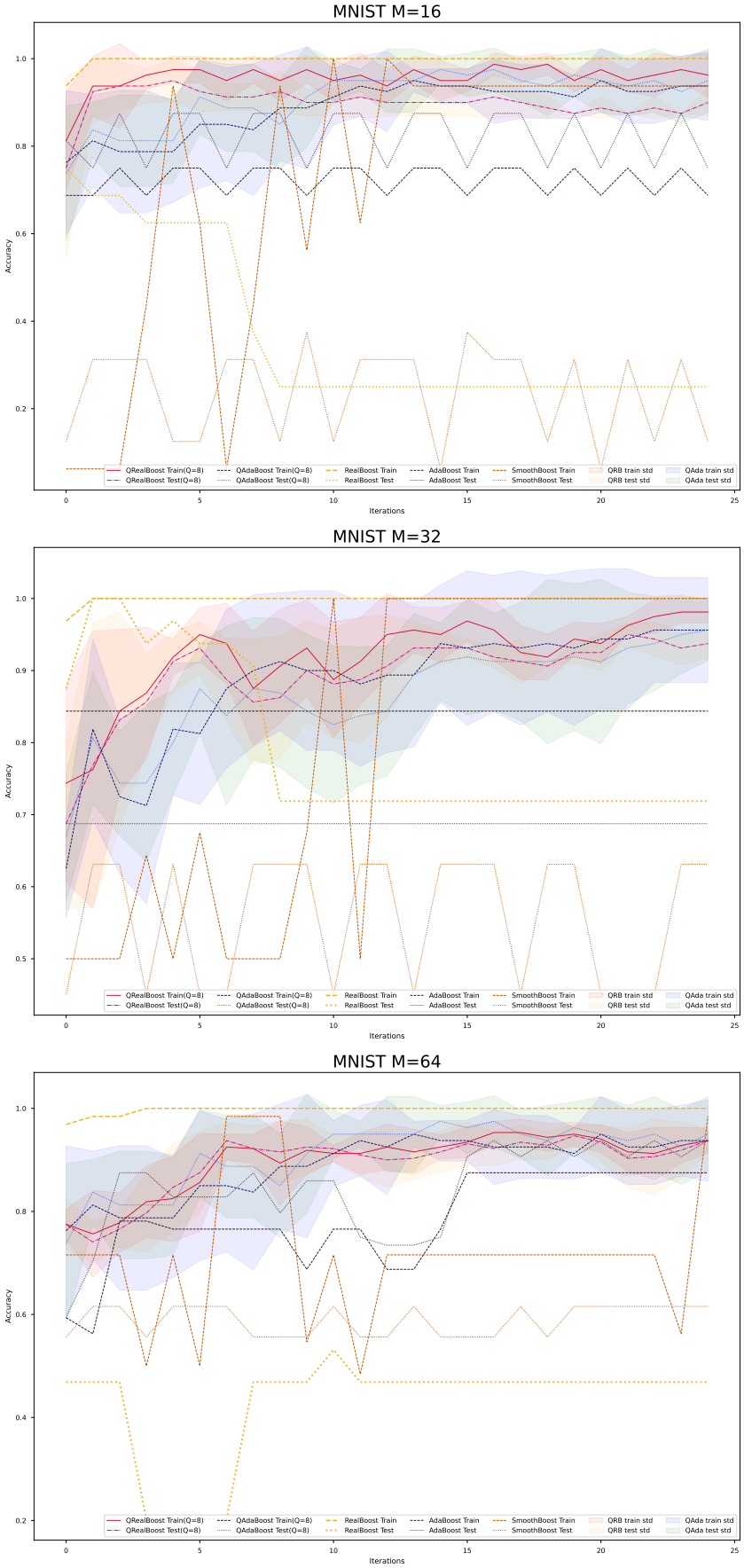


Figure 7.4: Performance of QREALBOOST, QADABOOST, REALBOOST, ADABOOST, and SMOOTHBOOST on the `MNIST` Dataset with different training set sizes.

§ 7.3 Discussion and Future Work

In this chapter, we also explored an interesting property of the QREALBOOST algorithm introduced in Chapter 5. As we have noted earlier in this chapter, the margin maximization behavior of adaptive boosting algorithms can be exploited to obtain boosting algorithms that greedily maximize the margin at every iteration instead of greedily minimizing the training error at every iteration. Mason et al. [MBB98] proposed the DOOM (Direct Optimization of Margins) boosting algorithm that explicitly carries out margin optimization at every step. This brings the following question to mind.

> **Open Problem 7.** Is it possible to obtain a margin optimizing quantum adaptive boosting algorithm with a better dependence on the bias of the weak learner compared to current known adaptive boosting algorithms?

One of the reasons Open problem 7 might be a possibility is because of the connection of margin optimization to gradient descent [Mas+99; Ben+05]. Mason et al. [Mas+99] generalized the DOOM algorithm by proposing ANYBOOST. ANYBOOST is a family of gradient descent style algorithms that choose a linear combination of elements of an inner product function space to minimize a margin cost functional. Given the existence of quantum algorithms for gradient descent that obtain speedup over existing classical techniques [Jor05; GAW19], we ask the following question:

> **Open Problem 8.** Is there a straightforward generalization of the ANYBOOST algorithm to the quantum setting that has better dependence on the bias of the weak learner compared to current known adaptive boosting algorithms?

From the errors of others, a wise man corrects his own.

PUBLILIUS SYRUS

Chapter 8

Generalization Error Bounds for Dependent Data

Abstract

In this work, we give generalization bounds of statistical learning algorithms trained on samples drawn from a dependent data source both in expectation and with high probability, using the Online-to-Batch conversion paradigm. We show that the generalization error of statistical learners in the dependent data setting is equivalent to the generalization error of statistical learners in the i.i.d. setting up to a term that depends on the decay rate of the underlying mixing stochastic process. Our proof techniques involve defining a new notion of stability of online learning algorithms based on Wasserstein distances and employing "near-martingale" concentration bounds for dependent random variables to arrive at appropriate upper bounds for the generalization error of statistical learners trained on dependent data. A significant part of this chapter is based on the following manuscript:

- Sagnik Chatterjee et al. "Generalization Bounds for Dependent Data using Online-to-Batch Conversion". In: *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*. accepted. 2025

Contents

8.1 Introduction . . . 130
8.1.1 Overview of Main results . . . 131
8.1.2 Overview of our Proof Techniques . . . 131
8.1.3 Related Works . . . 132
8.1.4 Useful Mathematical Notions . . . 136
8.1.4.1 Mixing Processes . . . 136
8.1.4.2 Wasserstein distances . . . 137

8.1.5 Notations, Assumptions, and Definitions 138
8.1.6 Generalization bounds via the OTB framework 140
8.1.6.1 Overview of Online Learning 140
8.1.6.2 The Online-to-Batch game 141
8.2 Generalization Error Bounds 142
8.3 Discussion and Future Work 148

§ 8.1 Introduction

Classically, the generalization error of an offline learner A has been characterized in terms of the VC-dimension or the Rademacher complexity of the associated hypothesis class $\mathcal{H}_A$. We saw an example of this in Chapter 7 where we expressed the generalization error of the QREALBOOST algorithm in terms of the margin induced over the training instances and the VC-dimension of the concept class.

However, recently, with the advent of the over-parameterized regime, it has been noted that the traditional models of generalization error are not tight enough to explain the small generalization behavior of massively large neural networks that have billions of tunable parameters [Zha+17]. In an effort to explain this discrepancy, instead of focusing solely on complexity measures of the learner's hypothesis space, researchers have proposed algorithm-dependent generalization error bounds, such as bounds due to stability [BE02], information-theoretic properties [RZ16; XR17], or PAC-Bayesian bounds [Hel+24; Alq24].

Most of these bounds, however, assume that the training and test samples are drawn i.i.d. from the same (unknown) underlying distribution. In many real-world applications, such as in learning from time-series dependent data such as stock prediction tasks, the i.i.d. assumption does not hold [Vid13]. A more "modern" example of a learning setting in which the i.i.d. assumption becomes untenable is Federated Learning, where the presence of non-i.i.d. data has been shown to be a crucial barrier in real-world deployment of architectures [Ami+22; Xio+22; Iye24; Li+24].

In this chapter, we derive bounds on the generalization performance of statistical learners by extending the Online-to-Batch paradigm [CBCG04; LN23] to the non-i.i.d. setting. We assume that the statistical learners are trained on data sampled from a stochastic process that is "mixing" to a stationary distribution. Mixing (see Definition 8.2) is a fairly natural assumption in which the dependence between data samples weakens over time and has been the setting of choice for analysing the performance of optimization and learning algorithms trained on non i.i.d. data [Yu94; Mei00; LKS05; MR07; MR10; AD11; Duc+12; KM17; Zha+19; Fu+23].

§ 8.1.1 Overview of Main results

Lugosi and Neu [LN23] introduced the Online-to-Batch framework[1] for obtaining generalization bounds for statistical learning algorithms trained on i.i.d. data via a reduction to a suitable online learning problem, referred to as the 'Online-to-Batch game.' In this chapter, we extend the Online-to-Batch framework to the non-i.i.d. setting, particularly when the batch learner is trained on data sampled from a *mixing* process.

In order to extend the Online-to-Batch framework of [CBCG04; LN23] to the non-i.i.d. setting, we require two additional assumptions. Firstly, we require a new notion of algorithmic stability based on Wasserstein distances of order one, that we denote *Wasserstein stability* (see Definition 8.6). The Wasserstein stability notion is similar to the notion of *one-step differential stability* criteria of Abernethy et al. [Abe+19]. We would like to point out, however, that this assumption essentially comes for free since the textbook example of EWA online learning algorithms turns out to be Wasserstein stable (see Chatterjee et al. [CMS25]). Secondly, we need to assume that the loss function is Lipschitz (see Assumption 8.3) and that the observable and hypothesis spaces are bounded. However, similar to [LN23], our techniques do not require convexity assumptions on the loss function, unlike in [KT08; LN22]. Furthermore, we would like to comment that even in [LN23], instantiation of their generalization bound does require implicitly assuming that the loss function is bounded – see, for example, Corollaries 4-8 of [LN23].

Our contributions are twofold:

- Our bounds hold both in expectation and with high probability.
- In cases where the mixing rate decays exponentially fast (see Definition 8.3), we can almost recover the bounds in the i.i.d. scenario .[2]

Remark 8.1.1. We emphasize that our aim is *not* designing low-regret online algorithms. Our focus, instead, is obtaining generalization bounds for the statistical learner A via reduction to a synthetic online learning problem.

§ 8.1.2 Overview of our Proof Techniques

The generalization error of statistical learning algorithms can be upper bounded by

1. the regret of an online learning algorithm for the corresponding Online-to-Batch game, and

[1] The Online-to-Batch framework was first introduced by [CBCG04] in order to bound the expected loss of a batch learner via a corresponding online learning task.

[2] Contrast with Corollary 4 of [LN23]. We incur an extra multiplicative factor $O\left(\sqrt{\log n}\right)$ due to sacrificing the i.i.d. assumption.

2. the normalized sum of the expected costs incurred by the online learner.

In the i.i.d. setting, it has been long observed that the costs incurred by the online learner form a martingale difference sequence [CBCG04; LN23]. This simple but powerful observation allows one to upper bound the second term above using standard concentration inequalities. However, in the dependent data setting, the expected costs incurred by the online learner do not form a martingale difference sequence since the expectation of the expected costs with respect to past training samples is not necessarily zero. Consequently, it is no longer straightforward to apply concentration inequalities to bound the generalization error of the batch learning algorithms as shown in [LN23]. To circumvent this issue, we rewrite the generalization error of statistical learner as the sum of three terms:

1. the regret of an online learner for the Online-to-Batch game,
2. the expected cost incurred by this online learner at time step $t + \tau$ with respect to its decision at step t,
3. and a few remainder terms, which depend on the underlying stochastic process from which the data is sampled.

The second term is a so-called "near-martingale" [MR10; AD11; Duc+12]. More precisely, this term consists of a sum of random variables forming a martingale difference sequence and an additional expectation term that can be bounded using the mixing coefficients of the random process from which the training data is drawn.

Finally, the remainder terms also consist of differences of expectations of costs incurred by the online learner at one time step with respect to either its output at a different time step or by the output of the batch learner. In order to bound these terms, we impose a doubly-Lipschitz condition on the loss function for the batch learner, boundedness of the observation and hypothesis spaces of the learning problem, and *stability* of the online learner according to a new Wasserstein distance based criterion (see Definition 8.6). The choice of a Wasserstein distance based stability criterion is motivated by its dual form (see Lemma 8.1), which is used to bound the difference in expected cost incurred by the online learner in successive steps.

§ 8.1.3 Related Works

▷ Learning with Dependent Data.

In learning with non-i.i.d. data, we usually assume that the data is drawn from a mixing random process. Initially, techniques based on uniform convergence over complexity measures of the hypothesis space [Yu94; Mei00; MR08] were used to bound

the generalization error of learning algorithms. Later, techniques based on algorithmic stability applied to different notions of mixing processes [MR10; AD11; Duc+12; Zha+19] were used to study the generalization performance of learning algorithms. We note here that the strong notion of ψ-mixing used in [Fu+23] is not used by us.

▷ Online-to-Batch Conversion..

There is a substantial body of literature in the Online-to-Batch paradigm starting from the seminal work of [CBCG04], which was later extended to give sharp generalization bounds for constrained linear classes based on complexity measures of the hypothesis classes [KT08]. Recently, [LN23] gave a framework that consolidates a vast quantity of the Online-to-Batch literature under its umbrella to obtain generalization bounds for statistical learning algorithms without putting unnecessarily restrictive assumptions on the loss functions and the associated hypothesis spaces of the learning algorithms. All of the previous works, however, necessarily require the i.i.d. assumption. In our work, we extend the Online-to-Batch paradigm by using the framework of [LN23] to obtain generalization bounds for statistical learning algorithms that are trained on data drawn from mixing stochastic processes.

▷ Algorithmic Stability.

The notion of uniform stability was first introduced in the context of statistical learning algorithms by [BE02] to study the generalization of regularization schemes in the i.i.d. setting. Mohri and Rostamizadeh [MR07] and Fu et al. [Fu+23] later used uniform stability to derive bounds on the generalization bounds of batch learners that are trained on data drawn from mixing processes. Agarwal and Duchi [AD11] uses a different (and incomparable) notion of stability to bound the performance of online learners. In our work, we introduce a new notion of algorithmic stability based on Wasserstein distances, relevant to our setting, that is incomparable to the notions of algorithmic stability in the previously referenced works. Our definition of stability is similar to the notion of **one-step differential stability** introduced by [Abe+19]. Unlike the notions of stability referenced above, which are incompatible with differential privacy (DP), our notion of Wasserstein stability holds promise for bridging the two seemingly disparate fields.

▷ A discussion on the **OtB** technique and Wasserstein stability.

We start by recalling that the task at hand is to upper bound the generalization error of any hypothesis h_{off} produced by a fixed offline learner trained on some training set S. Consider an artificially constructed class of online learners, s.t. any online learner in this class, when provided the elements of S in an online manner, uses the offline

hypothesis h_{off} as the comparator. The **OtB** technique is used to show that the regret incurred by any such online learner upper-bounds the generalization error of the offline learner. Here, we note that the online learners in the **OtB** technique are simply an artificial construct that only serve to bound the generalization error of the offline learner.

▷ Extending the **OtB** framework with a new notion of stability.

Earlier works in the **OtB** setting assume that elements in $\mathcal{S}$ are sampled i.i.d. from an unknown distribution. In this work, we show that the **OtB** technique can be extended to analyze the generalization error for batch learners, even when elements in $\mathcal{S}$ are drawn from a mixing stochastic process, i.e., in a non-i.i.d. setting. We now discuss some of the strengths and potential issues with the extension of the **OtB** technique to the non-i.i.d. setting.

1. Using the **OtB** technique allows us to derive generalization error bounds on the offline learner without requiring stability assumptions on the offline learner itself (contrary to the works of [MR10; Fu+23]). Instead, the stability assumption gets shifted to the class of online learners, which is an artificial construct used for analysis only and not a part of the problem setup.
2. We would like to point out that a straightforward application of the standard **OtB** framework is not applicable when the batch learner is trained on a mixing process, and we need to introduce a suitable notion of algorithmic stability to integrate the **OtB** technique with 'near-martingale' concentration bounds (see the next section for a brief discussion). We refer to this notion of algorithmic stability as Wasserstein stability.
3. **The apparent tradeoff:** Since our setup requires the class of online learners to be Wasserstein-stable, this seemingly constrains the choice of class of online learners that can be used for the **OtB** technique. Chatterjee et al. [CMS25] resolves the above tradeoff by proving that one of the most celebrated and widely used classes of online learning algorithms - the class of EWA learners, conforms to the notion of Wasserstein stability. This result allows us to instantiate our generalization error bounds.

We note that our contributions extend merely beyond simply integrating different paradigms and carry a deep semantic contribution as follows - our techniques allow one to shift the assumption of stability from the problem setup, i.e., the batch learner, to the analysis, i.e., the intermediate online learner (which is an artificial construct) used in the **OtB** technique. This allows us to derive generalization error bounds for a wider class of statistical learners compared to earlier works like [MR10; Fu+23].

▷ A Short Note on different notions of Algorithmic Stability.

The notion of uniform stability was first introduced in the context of statistical learning algorithms by [BE02] to study the generalization of regularization schemes in the i.i.d. setting. [MR07] and [Fu+23], later used uniform stability to derive bounds on the generalization bounds of batch learners that are train on data drawn from mixing processes. [AD11] uses a different (and incomparable) notion of stability to upper bound the generalization error of online learners. In our work, we introduce a new notion of algorithmic stability based on Wasserstein distances, relevant to our setting, that is algebraically incomparable to the notions of algorithmic stability in the previously referenced works.

The fundamental difference in the different notions of algorithmic stability lies in the settings in which they are defined. Uniform stability (as in [BE02]) is defined for batch learners, while our notion of Wasserstein stability (similar to [AD11]) is defined strictly for online learning algorithms. In more detail, uniform stability requires that the loss incurred by the batch learner does not change too much when trained on neighboring datasets (for instance, datasets that differ at one point). On the other hand, our notion of Wasserstein stability is not defined via the data presented to the online learner but rather in terms of the output distributions produced by the online learner at consecutive time steps, i.e., we require that for a Wasserstein-stable online learner, the successive output distributions are "not too far" from each other, in the sense of Wasserstein distance.

▷ Comparisons with the Concurrent Work [ACN24].

Concurrent with the results presented in this chapter, Abeles et al. [ACN24] independently addressed the same problem in a draft updated on ArXiv in June 2024. Both works derive generalization bounds of the same order, i.e., $\frac{\text{regret}}{n} + O(\frac{1}{\sqrt{n}})$, but involve different assumptions, and hence vastly different techniques, and also differ in the formulation of the Online-to-Batch framework.

- We use the standard Online-to-Batch framework, whereas Abeles et al. [ACN24] uses a different Online-to-Batch framework, using a *delayed variant of the online learning problem* introduced by Weinberger and Ordentlich [WO02], where the learner starts seeing the cost of its choices only after a finite number of plays.
- This delayed Online-to-Batch framework allows Abeles et al. [ACN24] to avoid the stability requirement on the online learner, which is imposed by us. However, as noted earlier, this stability assumption essentially doesn't limit our choice of potential online learners to instantiate the bounds, as the canonical choice of the EWA algorithm turns out to be stable.

- To circumvent the issue of working with non-i.i.d. data, both works require assumptions on the mixing properties of the random process. We use a standard variant of the β and the ϕ mixing assumptions (see, for example, Section 2 of [AD11]) on the random process from which the dataset is drawn. On the other hand, Abeles et al. [ACN24] uses a much stronger mixing assumption, applied directly on the associated loss function – see Assumption 1 of [ACN24]. Since our mixing assumption is weaker, our proofs, therefore, require extra technical steps – see Lemmas 8.7 and 8.8.
- We, however, require an additional Lipschitz assumption on the loss function for the above proofs to go through, which is not required by Abeles et al. [ACN24] thanks to their already strong mixing assumption on the loss function.
- Finally, both works require the loss function to be bounded. In our case, we assume both the observation and hypothesis spaces to be bounded, which then directly implies that the loss function is bounded thanks to our Lipschitz assumption. Instead, in [ACN24], a boundedness assumption is explicitly applied to the loss function during the instantiation of their bounds – see Corollary 3 of [ACN24]. Also, while not explicitly stated, a boundedness assumption on the loss is implicitly assumed in deriving the general upper bounding framework of [ACN24], through the use of Hoeffding's Lemma in the proof of [ACN24, Lemma 2].

See Table 8.1 for a comparison.

Work	Samples	OTB setting	Loss function	Stability (Online)
[LN23]	i.i.d.	Normal	Bounded in $[0,1]$	N/A
[ACN24]	mixing	Delayed	Bounded in $[0,1]$	N/A
This Work	mixing	Normal	Doubly-Lipschitz	Wasserstein-stable

Table 8.1: An overview of the three frameworks for deriving generalization bounds using Online-to-Batch techniques.

§ 8.1.4 Useful Mathematical Notions

§ 8.1.4.1 Mixing Processes

Consider a random process $(Z_t)_{t\in\mathbb{N}}$ with the probability distribution P. Let $\mathcal{F}_t = \sigma(Z_1,\dots,Z_t)$ denotes the smallest sigma-algebra generated by the set $\{Z_s\}_{s\in[t]}$. We denote by $\mathcal{P}^t_{[s]} = \mathcal{P}^t(\cdot \mid \mathcal{F}_s)$ the conditional probability distribution of Z_t given the sigma-algebra $\mathcal{F}_s$. In this work, we consider the case where the distribution of Z_t con-

verges (w.r.t. two different notions of convergence) to a stationary distribution $\mathcal{D}$ as $t \to \infty$, as defined below.

Definition 8.1 (β and ϕ coefficients)**.** The β and ϕ mixing coefficients for the distribution P are defined as

$$\beta(k) := \sup_{t \in \mathbb{N}} \left\{ 2 \mathop{\mathbb{E}}_{\mathcal{P}_{[t]}} \left[d_{TV}\left(\mathcal{P}^{t+k}_{[t]}, \mathcal{D} \right) \right] \right\}, \tag{8.1}$$

$$\phi(k) := \sup_{t \in \mathbb{N}, B \in \mathcal{F}_t} \left\{ 2\, d_{TV}\left(\mathcal{P}^{t+k}(\cdot|B), \mathcal{D} \right) \right\}, \tag{8.2}$$

where the supremum in the definition of $\phi(k)$ is over elements of $\mathcal{F}_t$ having non-zero measure, and $\mathcal{P}_{[t]}$ is the joint distribution of $Z_1, \ldots, Z_t$.

Definition 8.2 (β and ϕ mixing)**.** A stochastic process $Z_1, Z_2, \ldots$ is β-mixing (or ϕ-mixing) if its distribution P satisfies $\lim_{k\to\infty} \beta(k) = 0$ (resp., if $\lim_{k\to\infty} \phi(k) = 0$).

Remark 8.1.2. It is trivial to see that for i.i.d. random processes, with $\mathcal{D}$ being the per-letter marginal, the mixing coefficients satisfy $\beta(k) = \phi(k) = 0$ for all $k \geqslant 1$. Hence, i.i.d. processes are both β and ϕ mixing.

Definition 8.3 (Geometric ϕ-mixing)**.** Let $K, r > 0$. A stochastic process is geometrically ϕ-mixing with rate K if $\phi(k) \leqslant K \cdot \exp\{-k^r\}$, for all $k > 0$.

We note at this point that there are no practical approaches to finding the decay rate of an unknown mixing process or even determining whether a stochastic process is mixing [Yu94; Mei00] unless other properties (such as Gaussianity or Markovity) of the mixing process are known beforehand. There are, however, known examples of stochastic processes that are exponentially mixing (see [Mok88; Mei00] for examples).

§ 8.1.4.2 Wasserstein distances

Let $(\mathcal{X}, d)$ be any Polish space[3], and let P and Q be any pair of probability measures on $\mathcal{X}$. We denote by $\Pi(P, Q)$ the set of joint measures on $\mathcal{X}$ whose marginals are respectively P and Q.

Definition 8.4 (Wasserstein distance of order one)**.** The *Wasserstein distance* of or-

[3] A complete metric space is *Polish* if it has a countable dense subset.

der one between P and Q is defined as

$$W(P,Q) \triangleq \inf_{\pi \in \Pi(P,Q)} \int_{\mathcal{X}} d(x,y) d\pi(x,y). \tag{8.3}$$

We now state the *Kantorovich-Rubinstein duality formula* for Wasserstein distances of order one as given in Remark 6.5 of [Vil08].

Lemma 8.1. Let P and Q be any pair of probability measures on a Polish space $(\mathcal{X}, d)$. Then, $W(P,Q) = \sup_{\substack{\phi:\mathcal{X}\to\mathbb{R} \\ \phi \text{ is 1-Lipschitz}}} \left\{ \int_{\mathcal{X}} \phi \, dP - \int_{\mathcal{X}} \phi \, dQ \right\}$.

As an immediate consequence of Lemma 8.1, we have the following corollary.

Corollary 8.1. Let P and Q be any pair of probability measures on a Polish space $(\mathcal{X}, d)$. Let $\phi : \mathcal{X} \to \mathbb{R}$ be G-Lipschitz. Then, $W(P,Q) \geqslant \frac{1}{G}\left[\int_{\mathcal{X}} \phi \, dP - \int_{\mathcal{X}} \phi \, dQ \right]$.

Proof. Define a new function $\psi : \mathcal{X} \to \mathbb{R}$ as $\psi(x) = \frac{1}{G}\phi(x)$, and note that ψ is 1-Lipschitz by definition. The result now follows using Lemma 8.1. □

Next, we present an inequality that relates Wasserstein distances of order 1 to the Total Variation distance.[4]

Lemma 8.2. Let P and Q be two probability measures on a Polish space $(\mathcal{X}, d)$. If the diameter of the underlying metric space is bounded by $M \geqslant 0$, then $W(P,Q) \leqslant M \cdot d_{TV}(P,Q)$.

§ 8.1.5 Notations, Assumptions, and Definitions

Consider a measurable instance space $\mathcal{Z}$. Let the set $S_n = (Z_1, \ldots, Z_n)$ denoted as a *training set*, be a tuple of n random variables (not necessarily independent), drawn from some random process $Z_1, Z_2, \cdots$ over $\mathcal{Z}$, with a probability distribution P which mixes to a stationary distribution $\mathcal{D}$, as defined in Definition 8.2.

Assumption 8.1. We assume that $\mathcal{Z}$ is equipped with a norm $\|\cdot\|_{\mathcal{Z}}$, and the diameter of the instance space is $R_{\mathcal{Z}}$.

We recall (and slightly) modify the definition of a learning algorithm as introduced in Chapter 2. A *learning algorithm* $A : \mathcal{Z}^n \mapsto \mathcal{H}$ maps (in a randomized fashion) any such n tuple to an element $H^* = A(S_n)$ in a measurable set $\mathcal{H}$ known as the *hypothesis class*. One metric of quantifying the performance of the learning algorithm A is a loss

[4] See Particular Case 6.16 of [Vil08].

function $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$. We now state our assumptions with respect to the hypothesis space $\mathcal{H}$ and the loss function ℓ below.

Assumption 8.2 (Assumptions on the Hypothesis Space $\mathcal{H}$)**.** We assume that the space $\mathcal{H}$ is a *Banach space* equipped with the norm $\|\cdot\|_{\mathcal{H}}$. The diameter of $\mathcal{H}$ is $R_{\mathcal{H}}$. Furthermore, we assume that $\mathcal{H}$ is a *Polish space* with respect to the metric induced by the norm $\|\cdot\|_{\mathcal{H}}$.

Assumption 8.3 (Doubly Lipschitz Loss)**.** Let $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$, be a loss function. We assume that ℓ is *doubly Lipschitz*. More precisely, ℓ is $G_{\mathcal{H}}$*-Lipschitz* w.r.t the first argument h, and it is $G_{\mathcal{Z}}$-Lipschitz w.r.t. the second argument z. Since ℓ is doubly-Lipschitz and both of its inputs are from bounded domains, the range of ℓ is also bounded, as shown in the following lemma.

Lemma 8.3. For any $(h, z) \in \mathcal{H} \times \mathcal{Z}$, the loss function ℓ satisfies $|\ell(h, z)| \leqslant B_\ell$, where $B_\ell := \inf_{(h',z')\in\mathcal{H}\times\mathcal{Z}} |\ell(h', z')| + G_{\mathcal{H}} R_{\mathcal{H}} + G_{\mathcal{Z}} R_{\mathcal{Z}}$.

Proof of Lemma 8.3. Fix any $(h', z') \in \mathcal{H} \times \mathcal{Z}$. On applying triangle inequality, we have $\Big||\ell(h, z)| - |\ell(h', z')|\Big| \leqslant |\ell(h, z) - \ell(h', z')|$. Now, noting that ℓ is doubly-Lipschitz, and $\mathcal{H}$ and $\mathcal{Z}$ have diameters $R_{\mathcal{H}}$ and $R_{\mathcal{Z}}$ respectively, we have

$$\begin{aligned} |\ell(h, z) - \ell(h', z')| &= |\ell(h, z) - \ell(h, z') + \ell(h, z') - \ell(h', z')| \\ &\leqslant |\ell(h, z) - \ell(h, z')| + |\ell(h, z') - \ell(h', z')| \\ &\leqslant G_{\mathcal{Z}} R_{\mathcal{Z}} + G_{\mathcal{H}} R_{\mathcal{H}}. \end{aligned}$$

Therefore, we have $|\ell(h, z)| \leqslant |\ell(h', z')| + G_{\mathcal{Z}} R_{\mathcal{Z}} + G_{\mathcal{H}} R_{\mathcal{H}}$. The lemma is then proved by taking an infimum over $(h', z') \in \mathcal{H} \times \mathcal{Z}$. □

The *training error* of the learning algorithm A is the cumulative loss conceded by A over its training set S_n: $\sum_{i=1}^{n} \ell(H^*, Z_i)$, where $H^* = A(S_n)$. The *test error* of A is defined as the expected loss of the learning algorithm over any instance from the instance space: $\mathop{\mathbb{E}}_{Z'\sim\mathcal{D}} [\, \ell(H^*, Z') \,]$.

Remark 8.1.3. In Chapters 2 to 7, we assumed that the instance set had the form $Z \in \mathcal{X} \times \{\pm 1\}$, the hypothesis was a Boolean function, and the loss function was defined as the indicator loss $\ell(H, Z) = \mathbb{1}_{[(H(x)\neq c(y))]}$, where $x \in \mathcal{X}$ and $y \in \{\pm 1\}$.

We now add the following error-quantifying metrics to the definitions of training error, test error, and correlation as given in Chapter 2.

Definition 8.5 (Generalization Error of Offline Learners)**.** The *overfitting error* of A is defined as the difference between the test error and the mean training error of A, as

$$\mathrm{gen}(A, S_n) = \mathop{\mathbb{E}}_{Z' \sim \mathcal{D}} \left[\ell(H^*, Z') \right] - \frac{1}{n} \sum_{t=1}^{n} \ell(H^*, Z_t).$$

The generalization error of a fixed statistical learning algorithm A is defined as

$$\overline{\mathrm{gen}}(A, S_n) := \mathop{\mathbb{E}}_{H \sim P_{H^*}} \left[\mathrm{gen}(H, S_n) \mid S_n \right],$$

where $P_{H^*} := P_{A(S_n)}$, i.e., the conditional distribution of the output H^* produced by A given training set S_n.

§ 8.1.6 Generalization bounds via the OTB framework

An interesting paradigm for evaluating the generalization ability of statistical learning algorithms that has been studied recently is Online-to-Batch conversions [CBCG04]. In the Online-to-Batch setting, a connection is established between the performance of batch learners (on unknown instances) and the performance of online learning algorithms (on known instances). We first introduce the online learning setting and subsequently describe the Online-to-Batch paradigm.

§ 8.1.6.1 Overview of Online Learning

The online learning setting can be modeled as the following two-player game, henceforth referred to as the Online-to-Batch game, between a learning algorithm $\mathcal{L}$ and an adversary. The learner $\mathcal{L}$ has sequential access to a stream of data generated from an arbitrary source, and at every time step t, based on decisions taken up to the $t-1^{\text{th}}$ time-step, the learner tries to predict the correct label of the next data point and incurs a loss that is decided by an adversarial cost function c_t. The goal of the online learning setup is to minimize some notion of *regret*, i.e., the loss incurred at the t^{th} time step by the online learner should be reasonably close to the loss incurred by the best possible offline learner that has access to the data points up to the $(t-1)^{\text{th}}$ time-step.

An example of an online learning game is the Hedging algorithm or the Exponential-Weighted Average (EWA) algorithm [LW94; FS97; Vov98]. In EWA, we first fix a data-free prior distribution $P_1 \in \Delta_{\mathcal{H}}$ and a learning rate parameter $\eta > 0$. At every iteration $t > 0$, we perform the following updates:

$$P_{t+1} := \mathop{\mathrm{argmin}}_{P \in \Delta_{\mathcal{H}}} \left\{ \langle P, c_t \rangle + \frac{1}{\eta} D(P \| P_t) \right\}. \tag{8.4}$$

§ 8.1.6.2 The Online-to-Batch game

In the Online-to-Batch conversion game, we assume that the instances $Z_t, t \in [n]$ in the offline setting are provided to the online learner $\mathcal{L}_n$. We now describe the generalization game from Lugosi and Neu [LN23], played over n rounds below.

Game 1 (Generalization game for Online-to-Batch conversion)**.**

1. *At the* t^{th} *iteration, the online learning algorithm* $\mathcal{L}_n$ *chooses a distribution* $P_t \in \Delta_{\mathcal{H}}$ *over the hypothesis space, with knowledge of only* $Z_1, \ldots, Z_{t-1}$.
2. *The adversary picks a cost function* $c_t : \mathcal{H} \to \mathbb{R}$ *for each hypothesis* $h \in \mathcal{H}$*, which is defined as* $c_t(h) := \ell(h, Z_t) - \mathop{\mathbb{E}}_{Z' \sim \mathcal{D}} [\, \ell(h, Z') \,]$.
3. *The online learning algorithm* $\mathcal{L}_n$ *incurs a cost* $\langle P_t, c_t \rangle := \mathop{\mathbb{E}}_{P_t} [\, c_t(H_t) \,]$.
4. *The adversary reveals to the online learner the sample* Z_t*. Now the online learner can compute the cost function.*

Recall that $P_{H^*} := P_{A(S_n)}$. Then the regret of the learning algorithm $\mathcal{L}_n$ with respect to the comparator distribution P_{H^*} over the hypothesis space is:

$$\text{regret}_{\mathcal{L}_n, A}(P_{H^*}) := \sum_{t=1}^{n} \langle P_t - P_{H^*}, c_t \rangle, \tag{8.5}$$

Henceforth, we shall drop the comparator distribution P_{H^*} from the notation of regret for the sake of brevity. Lugosi and Neu [LN23] proved the following result connecting the overfitting error of A to the regret of the online learner $\mathcal{L}_n$ obtained from Game 1.[5]

Proposition 8.1. When $Z_1, Z_2, \ldots, Z_n$ is sampled i.i.d. from an unknown distribution $\mathcal{D}$,

$$\text{gen}(A, S_n) = \frac{\text{regret}_{\mathcal{L}_n, A}}{n} - \frac{1}{n} \langle P_t, c_t \rangle .$$

Lugosi and Neu [LN23] instantiated Proposition 8.1 using the EWA algorithm. Unfortunately, Proposition 8.1 does not hold when the samples are non i.i.d. In order to bound the generalization error of A using the regret of the online learner $\mathcal{L}_n$ for Game 1, we shall require the $\mathcal{L}_n$ to be *Wasserstein-stable* as defined below.

[5] See thm 1 of [LN23] for a detailed proof.

Definition 8.6 (Wasserstein Stable)**.** Given a non-increasing sequence $\kappa(t), t \geqslant 1$, an online learning algorithm is said to be $\kappa(t)$-Wasserstein-stable if for any $t \in [n]$, the online learner $\mathcal{L}_n$ satisfies

$$W(P_t, P_{t+1}) \leqslant \kappa(t). \tag{8.6}$$

We refer to $\kappa(t)$ as the stability parameter at round t.

§ 8.2 Generalization Error Bounds

In this section, we state and prove our main results on the generalization error of statistical learning algorithms trained on training samples drawn from a mixing process. Our first result is an upper bound on the expected generalization error in terms of the expected regret of an online learner $\mathcal{L}_n$ for Game 1.

Theorem 8.2 (Expected generalization error)**.** For any arbitrary Wasserstein-stable online learner $\mathcal{L}_n$ for Game 1, and any $\tau = o(n)$, the expected generalization error $\mathbb{E}\left[\overline{\text{gen}}(A, S_n)\right]$ of the learning algorithm A with input $S_n = (Z_1, \ldots, Z_n)$ drawn from the mixing random process $Z_1, Z_2, \ldots$ is upper bounded by

$$\frac{1}{n}\mathbb{E}\left[\text{regret}_{\mathcal{L}_n, A}\right] + \frac{2\tau G_{\mathcal{H}}}{n}\left(\sum_{t=1}^{n}\kappa(t) + 2R_{\mathcal{H}}\right) + \frac{\tau G_{\mathcal{Z}} R_{\mathcal{Z}}}{n} + B_\ell \cdot \beta(\tau+1).$$

The next thm complements thm 8.2 by providing a high probability upper bound on the generalization error of the learning algorithm A in terms of the regret of any learner $\mathcal{L}_n$ for Game 1.

Theorem 8.3 (Generalization Error)**.** For any arbitrary Wasserstein-stable online learner $\mathcal{L}_n$ for Game 1, and any $\tau = o(n)$, $\delta > 0$, the generalization error $\overline{\text{gen}}(A, S_n)$ of the learning algorithm A with input $S_n = (Z_1, \ldots, Z_n)$ drawn from the mixing random process $Z_1, Z_2, \ldots$ is upper bounded, with probability at least $1 - \delta$, by

$$\frac{\text{regret}_{\mathcal{L}_n, A}}{n} + \frac{2\tau G_{\mathcal{H}}}{n}\left(\sum_{t=1}^{n}\kappa(t) + 2R_{\mathcal{H}}\right) + \frac{\tau G_{\mathcal{Z}} R_{\mathcal{Z}}}{n}$$
$$+ 2G_{\mathcal{H}} R_{\mathcal{H}}\sqrt{\frac{2\tau \log(\tau/\delta)}{n}} + B_\ell \cdot \phi(\tau+1).$$

In order to establish Theorem 8.2 and Theorem 8.3, we first prove a series of technical lemmas. The first of these lemmas establish that the cost functions c_t in Game 1 are Lipschitz.

Lemma 8.4. For any fixed instance of Z_t, and any $h_1, h_2 \in \mathcal{H}$, the cost function $c_t(\cdot)$ picked by the adversary in the generalization game satisfies $|c_t(h_1) - c_t(h_2)| \leqslant 2G_{\mathcal{H}}R_{\mathcal{H}}$. On the other hand, for any fixed $h \in \mathcal{H}$, and any t, t' and a fixed realization of $Z_t, Z_{t'}$, we have $|c_t(h) - c_{t'}(h)| \leqslant G_{\mathcal{Z}}R_{\mathcal{Z}}$.

Proof. Let $h_1, h_2 \in \mathcal{H}$. Then, for any fixed instance of Z_t, we have

$$\begin{aligned}
|c_t(h_1) - c_t(h_2)| &= |\ell(h_1, Z_t) - \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h_1, Z')\right] - \ell(h_2, Z_t) + \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h_2, Z')\right]| \\
&\overset{(a)}{\leqslant} |\ell(h_1, Z_t) - \ell(h_2, Z_t)| + \left|\mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h_1, Z')\right] - \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h_2, Z')\right]\right| \\
&\overset{(b)}{\leqslant} G_{\mathcal{H}}\|h_1 - h_2\|_{\mathcal{H}} + \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[|\ell(h_1, Z') - \ell(h_2, Z')|\right] \\
&\overset{(c)}{\leqslant} 2G_{\mathcal{H}}\|h_1 - h_2\|_{\mathcal{H}} \overset{(d)}{\leqslant} 2G_{\mathcal{H}}R_{\mathcal{H}},
\end{aligned}$$

where (a), (b) use the triangle inequality, and (b) and (c) use the fact that $\ell(\cdot,\cdot)$ is $G_{\mathcal{H}}$-Lipschitz in the first argument, and (d) uses the bounded diameter of the hypothesis space $\mathcal{H}$.

On the other hand, for any $h \in \mathcal{H}$ and any t, t' and any instance of $Z_t, Z_{t'}$,

$$\begin{aligned}
|c_t(h) - c_{t'}(h)| &= \left|\ell(h, Z_t) - \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h, Z')\right] - \ell(h, Z_{t'}) + \mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(h, Z')\right]\right| \\
&\overset{(a)}{\leqslant} G_{\mathcal{Z}}\|Z_t - Z_{t'}\|_{\mathcal{Z}} \overset{(b)}{\leqslant} G_{\mathcal{Z}}R_{\mathcal{Z}},
\end{aligned}$$

where (a) follows from the fact that $\ell(\cdot,\cdot)$ is $G_{\mathcal{Z}}$-Lipschitz in the second argument, and (b) follows by noting that the diameter of $\mathcal{Z}$ is $R_{\mathcal{Z}}$. □

To relate the generalization error of the offline learner A with the online learner L_n, we need the following technical lemma.

Lemma 8.5. For any Wasserstein-stable online learning algorithm L_n for Game 1 and any $\tau = o(n)$, the following bound holds with probability one.

$$\sum_{t=1}^{n}\left[\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_{t+\tau}(H)\right]\right] \leqslant \mathrm{regret}_{L_n, A} + 2G_{\mathcal{H}}\tau\sum_{t=1}^{n}\kappa(t) + 4\tau G_{\mathcal{H}}R_{\mathcal{H}}.$$

Proof. We first rearrange the terms of the summation in the LHS of the hypothesis of Lemma 8.5, and proceed to bound each term individually as follows.

$$\begin{aligned}
&\sum_{t=1}^{n}\left[\mathop{\mathbb{E}}_{H\sim P_t}[c_{t+\tau}(H)] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}[c_{t+\tau}(H)]\right] \\
&\quad= \underbrace{\sum_{t=1}^{n}\left[\mathop{\mathbb{E}}_{H\sim P_t}[c_t(H)] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}[c_t(H)]\right]}_{T_1} \\
&\quad+ \underbrace{\sum_{t=1}^{n-\tau}\left[\mathop{\mathbb{E}}_{H\sim P_t}[c_{t+\tau}(H)] - \mathop{\mathbb{E}}_{H\sim P_{t+\tau}}[c_{t+\tau}(H)]\right]}_{T_2} \\
&\quad+ \underbrace{\sum_{t=n-\tau+1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}[c_{t+\tau}(H)] - \sum_{t=n+1}^{n+\tau}\mathop{\mathbb{E}}_{H\sim P_{H^*}}[c_t(H)]}_{T_3} \\
&\quad+ \underbrace{\sum_{t=1}^{\tau}\mathop{\mathbb{E}}_{H\sim P_{H^*}}[c_t(H)] - \sum_{t=1}^{\tau}\mathop{\mathbb{E}}_{H\sim P_t}[c_t(H)]}_{T_4}.
\end{aligned} \tag{8.7}$$

By definition, we have $T_1 \leqslant \text{regret}_{L_n,\mathcal{A}}$. Next, we bound T_2 as follows:

$$\begin{aligned}
T_2 &\overset{(a)}{\leqslant} \sum_{t=1}^{n-\tau} 2G_{\mathcal{H}} W(P_t, P_{H_{t+\tau}}) \overset{(b)}{\leqslant} \sum_{t=1}^{n-\tau} 2G_{\mathcal{H}} \sum_{r=0}^{\tau-1} W(P_{t+r}, P_{t+r+1}) \\
&\overset{(c)}{\leqslant} \sum_{t=1}^{n-\tau} 2G_{\mathcal{H}} \sum_{r=0}^{\tau-1} \kappa(t+r) \overset{(d)}{\leqslant} 2G_{\mathcal{H}}\tau \sum_{t=1}^{n-\tau} \kappa(t) \overset{(e)}{\leqslant} 2G_{\mathcal{H}}\tau \sum_{t=1}^{n} \kappa(t),
\end{aligned}$$

where (a) follows from Corollary 8.1 and the fact that c_t is $2G_{\mathcal{H}}$-Lipschitz (see Lemma 8.4), (b) follows using the triangle inequality, (c) uses the stability assumption of the learner L_n, (d) uses the fact that $\kappa(\tau)$ is non-increasing, and (e) uses the non-negativity of $\kappa(\tau)$ which follows from Eq. (8.6). Next, we bound T_3 as follows.

$$T_3 = \sum_{t=n+1}^{n+\tau}\left[\mathop{\mathbb{E}}_{H\sim P_{H_{t-\tau}}}[c_t(H)] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}[c_t(H)]\right] \tag{8.8}$$

$$= \sum_{t=n+1}^{n+\tau} \mathop{\mathbb{E}}_{\substack{H_1\sim P_{H_{t-\tau}} \\ H_2\sim P_{H^*}}}[c_t(H_1) - c_t(H_2)] \leqslant \sum_{t=n+1}^{n+\tau} 2G_{\mathcal{H}}R_{\mathcal{H}} = 2\tau G_{\mathcal{H}}R_{\mathcal{H}}, \tag{8.9}$$

where the penultimate step uses the fact that c_t is $2G_{\mathcal{H}}$-Lipshitz via Lemma 8.4, and that the diameter of $\mathcal{H}$ is $R_{\mathcal{H}}$. Similarly, one can bound $T_4 \leqslant 2\tau G_{\mathcal{H}}R_{\mathcal{H}}$. Plugging in all

the bounds in Eq. (8.7), we have the result. □

Next, we relate the generalization error of the offline learner A to the regret of the online learner L_n.

Lemma 8.6. For any $\tau = o(n)$, and any Wasserstein-stable online learner L_n for Game 1, with probability one, $\overline{\text{gen}}(A, S_n)$ is at most

$$\frac{1}{n}\text{regret}_{L_n,A} + \frac{\tau}{n}\left[2G_{\mathcal{H}}\sum_{t=1}^{n}\kappa(t) + 4G_{\mathcal{H}}R_{\mathcal{H}} + G_{\mathcal{Z}}R_{\mathcal{Z}}\right] - \frac{1}{n}\sum_{t=1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right].$$

Proof. Recall that $\overline{\text{gen}}(A, S_n) := \mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[\text{gen}(H, S_n) \mid S_n\right]$. Expand the R.H.S. as

$$\begin{aligned}
&\mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[\mathop{\mathbb{E}}_{Z'\sim\mathcal{D}}\left[\ell(H, Z')\right] - \frac{1}{n}\sum_{t=1}^{n}\ell(H, Z_t) \mid S_n\right] \\
&= -\frac{1}{n}\sum_{i=1}^{n}\mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_t(H)\right] \\
&= \frac{1}{n}\sum_{t=1}^{n}\left[\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_{t+\tau}(H)\right]\right] - \frac{1}{n}\sum_{t=1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right] \\
&\quad + \frac{1}{n}\sum_{t=1}^{n}\left[\mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_{t+\tau}(H)\right] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_t(H)\right]\right]
\end{aligned}$$

Plugging in the definition of $\frac{1}{n}\text{regret}_{L_n,A}$, we can upper bound the above term as

$$\begin{aligned}
&\overset{(a)}{\leqslant} \frac{1}{n}\text{regret}_{L_n,A} + \frac{\tau}{n}\left[2G_{\mathcal{H}}\sum_{t=1}^{n}\kappa(t) + 4G_{\mathcal{H}}R_{\mathcal{H}}\right] - \frac{1}{n}\sum_{t=1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right] \\
&+ \frac{1}{n}\sum_{t=1}^{\tau}\left[\mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_{n+t}(H)\right] - \mathop{\mathbb{E}}_{H\sim P_{H^*}}\left[c_t(H)\right]\right] \\
&\overset{(b)}{\leqslant} \frac{1}{n}\text{regret}_{L_n,A} + \frac{\tau}{n}\left[2G_{\mathcal{H}}\sum_{t=1}^{n}\kappa(t) + 4G_{\mathcal{H}}R_{\mathcal{H}} + G_{\mathcal{Z}}R_{\mathcal{Z}}\right] - \frac{1}{n}\sum_{t=1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right],
\end{aligned}$$

where (a) uses Lemma 8.5 and (b) uses Lemma 8.4. □

To complete the proofs of Theorem 8.2 and Theorem 8.3, we upper bound the final term in Lemma 8.6, respectively, in expectation and with high probability. This is accomplished in the following pair of lemmas which rearranges the terms inside of $\sum_{t=1}^{n}\mathop{\mathbb{E}}_{H\sim P_t}\left[c_{t+\tau}(H)\right]$ as sums of random variables forming a martingale difference sequence, and a remainder term which can be bounded using the mixing coefficients

for the random process $Z_1, Z_2, \ldots$. First, we bound the final term in Lemma 8.6 in expectation.

Lemma 8.7. For any $\tau = o(n)$, $-\sum_{t=1}^{n} \mathop{\mathbb{E}}_{\mathcal{P}} \left[\mathop{\mathbb{E}}_{H \sim P_t} [c_{t+\tau}(H)] \right] \leqslant n \cdot B_\ell \cdot \beta(\tau+1)$.

Proof. First, we rearrange the terms in $\sum_{t=1}^{n} \mathop{\mathbb{E}}_{H \sim P_t} [c_{t+\tau}(H)]$ as follows. Consider the indices $a \in \{1, \ldots, \tau\}$, and $b \in \{1, \ldots, i_a\}$, where $i_a := \min\{b' : (b'-1)\tau + a \leqslant n\}$, and note that $i_a \leqslant \lceil n/\tau \rceil$ for any $1 \leqslant a \leqslant \tau$. Now, let $\mathcal{F}_{(b-1)\tau+a-1} := \sigma(Z_1, \ldots, Z_{(b-1)\tau+a-1})$, and define $X_b^a := -\mathop{\mathbb{E}}_{H \sim P_{(b-1)\tau+a}} [c_{b\tau+a}(H)]$. We rewrite the term $Y := -\sum_{t=1}^{n} \mathop{\mathbb{E}}_{H \sim P_t} [c_{t+\tau}(H)]$ as follows:

$$Y = \sum_{a=1}^{\tau} \underbrace{\sum_{b=1}^{i_a} \left(X_b^a - \mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right)}_{\mathcal{M}_a} + \sum_{a=1}^{\tau} \sum_{b=1}^{i_a} \left(\mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right). \tag{8.10}$$

Firstly, note that $\mathop{\mathbb{E}}_{\mathcal{P}} [\mathcal{M}_a] = 0$ for all $1 \leqslant a \leqslant \tau$. Therefore,

$$\mathop{\mathbb{E}}_{\mathcal{P}} [Y] = \sum_{a=1}^{\tau} \sum_{b=1}^{i_a} \mathop{\mathbb{E}}_{\mathcal{P}} \left[\mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right]. \tag{8.11}$$

Now, let $p^t_{[s]}$ and d be the densities with respect to some measure μ.[6] Then, the second term in Eq. (8.10) can be rewritten as follows:

$$\mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \tag{8.12}$$

$$= \mathop{\mathbb{E}}_{P^{b\tau+a}} \left[-\mathop{\mathbb{E}}_{P_{(b-1)\tau+a}} [c_{b\tau+a}(H)] \,\middle|\, \mathcal{F}_{(b-1)\tau+a-1} \right] \tag{8.13}$$

$$\leqslant \mathop{\mathbb{E}}_{P^{b\tau+a}} \left[\mathop{\mathbb{E}}_{P_{(b-1)\tau+a}} [|c_{b\tau+a}(H)|] \,\middle|\, \mathcal{F}_{(b-1)\tau+a-1} \right] \tag{8.14}$$

$$\leqslant \mathop{\mathbb{E}}_{P^{(b-1)\tau+a}_{[(b-1)\tau+a-1]}} \left[\int_{\mathcal{Z}} \ell(H, Z) \cdot \left| p^{b\tau+a}_{[(b-1)\tau+a-1]} - d \right| d\mu \right] \tag{8.15}$$

$$\overset{(b)}{\leqslant} 2 \cdot B_\ell \cdot d_{TV} \left(P^{b\tau+a}_{[(b-1)\tau+a-1]}, \mathcal{D} \right) \tag{8.16}$$

$$\leqslant B_\ell \cdot \beta(\tau+1). \tag{8.17}$$

[6] For example, μ can be chosen as $\frac{P^t_{[s]} + \mathcal{D}}{2}$.

where Eq. (8.15) follows via Fubini's theorem[7] and by noting the fact that the distribution $P_{(b-1)\tau+a}$ returned by the online-learner is independent of $Z_{b\tau+a}$ conditioned on $Z_1, \ldots, Z_{(b-1)\tau+a-1}$, Eq. (8.16) uses Lemma 8.3, and Eq. (8.17) follows from Definition 8.1. Plugging Eq. (8.17) in (8.11) completes the proof. □

The next lemma bounds the final term of Lemma 8.6 with high probability.

Lemma 8.8. With probability at least $1-\delta$, for any $\tau = o(n), \delta > 0$,

$$-\sum_{t=1}^{n} \mathop{\mathbb{E}}_{H\sim P_t} [\, c_{t+\tau}(H) \,] \leqslant 2G_{\mathcal{H}} R_{\mathcal{H}} \sqrt{2n\tau \log(\tau\delta)} + n \cdot B_\ell \cdot \phi(\tau+1).$$

Proof. Let $\mathcal{F}_{(b-1)\tau+a-1} := \sigma(Z_1, \ldots, Z_{(b-1)\tau+a-1})$. Then we proceed exactly as in the proof of Lemma 8.7, by defining

$$X_b^a := -\mathop{\mathbb{E}}_{H\sim P_{(b-1)\tau+a}} [\, c_{b\tau+a}(H) \,], \quad \text{and,} \quad Y := -\sum_{t=1}^{n} \mathop{\mathbb{E}}_{H\sim P_t} [\, c_{t+\tau}(H) \,],$$

and observing that

$$Y = \sum_{a=1}^{\tau} \underbrace{\sum_{b=1}^{i_a} \left(X_b^a - \mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right)}_{\mathcal{M}_a} + \sum_{a=1}^{\tau} \sum_{b=1}^{i_a} \left(\mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right). \tag{8.18}$$

Now, note that for each $1 \leqslant a \leqslant \tau$, the term $\mathcal{M}_a$ is a sum of random variables forming a martingale difference sequence w.r.t. the filtration $\mathcal{F}_{(b-1)\tau+a-1}, 1 \leqslant b \leqslant i_a$. Furthermore, we also observe that $\left| X_b^a - \mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \right| \leqslant 2G_{\mathcal{H}} R_{\mathcal{H}}$ via Lemma 8.4. From Lemma 2.2, using the fact that $i_a \leqslant \lceil \frac{n}{\tau} \rceil$, we can now obtain for any $\gamma > 0$

$$\Pr[\mathcal{M}_a \geqslant \gamma] \leqslant \exp\left(-\frac{\tau\gamma^2}{8(n+\tau)G_{\mathcal{H}}^2 R_{\mathcal{H}}^2} \right). \tag{8.19}$$

Again, as in the proof of Lemma 8.7, let $p_{[s]}^t$ and d be the densities with respect to some measure μ. Then, by applying Definition 8.1 in Eq. (8.17) we get

$$\mathop{\mathbb{E}}_{P^{b\tau+a}} \left[X_b^a \mid \mathcal{F}_{(b-1)\tau+a-1} \right] \leqslant B_\ell \cdot \phi(\tau+1). \tag{8.20}$$

[7] See Theorem 14.19 of [Kle07].

Combining Eq. (8.19) and Eq. (8.20), we can now write

$$\Pr\left[Y > \gamma + n \cdot B_\ell \cdot \phi(\tau+1)\right] \leqslant \sum_{a=1}^{\tau} \Pr\left[\mathcal{M}_a \geqslant \gamma/\tau\right] \leqslant \tau \cdot \exp\left(-\frac{\gamma^2}{8(n+\tau)\tau G_{\mathcal{H}}^2 R_{\mathcal{H}}^2}\right) \tag{8.21}$$

where the first inequality follows from a union-bound argument. Therefore, by setting $\gamma = O\left(\sqrt{n \cdot \tau \log(\tau/\delta)}\right)$, we have $Y \leqslant 2G_{\mathcal{H}} R_{\mathcal{H}} \sqrt{2n\tau \log(\tau/\delta)} + n \cdot B_\ell \cdot \phi(\tau+1)$ with probability $\geqslant 1-\delta$. □

We now complete the proofs of thms 8.2 and 8.3 below.

Proofs of Theorems 8.2 and 8.3. Theorem 8.2 now follows by plugging Lemma 8.7 in Lemma 8.6. Similarly, Theorem 8.3 follows by plugging Lemma 8.8 in Lemma 8.6. □

§ 8.3 Discussion and Future Work

In this work, we extend the Online-to-Batch framework to give generalization bounds for statistical learning algorithms that are trained on data sampled from mixing processes. An immediate avenue of future work is to obtain generalization bounds for non-i.i.d. data in various settings by using different choices of online learners to instantiate our framework as presented in Theorem 8.2 and Theorem 8.3. One such example is provided in Chatterjee et al. [CMS25], where the class of EWA learners was shown to be Wasserstein-stable. We ask the following question in this regard.

> **Open Problem 9.** Can we get better generalization bounds compared to [CMS25] by instantiating Theorem 8.3 with different Wasserstein-stable online learners?

To compensate for considering the weaker non-i.i.d. assumption on our dataset, our Online-to-Batch framework requires online learners that are Wasserstein-stable. As noted earlier, the notion of Wasserstein stability is quite similar to the differential privacy-inspired notion of stability proposed by Abernethy et al. [Abe+19]. Abernethy et al. [Abe+19] used their notion of stability to develop regret bounds for a variety of online learning problems, such as follow-the-perturbed-leader algorithms, thereby cementing a connection between differentially private learning algorithms and online learners. In this light, we ask the following question.

> **Open Problem 10.** Can we use the techniques introduced in this paper to analyze generalization error bounds for differentially private (DP) learners, especially in the non-i.i.d. setting?

Algorithmic stability and its relation to tight generalization bounds have been recently studied by Gastpar et al. [Gas+24a; Gas+24b] in the i.i.d. setting. Due to the dearth of notions of algorithmic stability in the non-i.i.d setting (see Section 8.1.3), our Wasserstein stability criteria is a potential candidate to extend the results of [Gas+24b; Gas+24a] to the non-i.i.d. setting.

Open Problem 11. Is Wasserstein stability a necessary and sufficient condition to obtain tight generalization bounds in the non-i.i.d. setting?

Part IV

Epilogue

Welcome to the jungle, we got fun and games.
We got everything you want, honey, we know the names.

GUNS N' ROSES

Chapter 9

The Quantum Learning Zoo

Abstract

In this chapter, we aim to survey some interesting results focused on sample and time complexity separations between quantum learning algorithms and classical learning algorithms with respect to Boolean-valued concept classes. The results in this chapter are written from the perspective of justifying the difficulty of the central problem addressed in this thesis - learning decision trees in the agnostic setting, and are not intended to be a comprehensive survey of existing quantum vs classical learning separations. For more details on the various topics we touched upon in this chapter, we encourage the reader to peruse the various resources mentioned in the chapter. An excellent starting point in the field of quantum learning theory is the survey by [AW17a], which covers some of the results in Sections 9.2 and 9.3. We also highlight the surveys [DB18; AA24] for a broader look at the field of Quantum Learning. This chapter is intended to capture the spirit of the various zoos and jungles out there.[1]

Contents

9.1 Introduction 154
9.2 Sample Complexity separations 154
9.2.1 Sample Complexity in the Multi-class setting 156
9.2.2 Sample Complexity separations via Unitary Access 157
9.3 Time Complexity separations 158
9.3.1 Separations via the Hidden Subgroup Problem 158
9.3.2 Separations based on Lattice-Based Problems 162
9.3.3 Learning separations for Halfspaces 165

[1] See for example the Hamiltonian Jungle, the Complexity Zoo, the Quantum Algorithm Zoo, and the Error Correction Zoo.

9.3.4 Learning separations for Decision Trees 165
9.3.5 Learning separations for DNFs 166
9.3.6 Learning separations for Juntas 167
9.3.7 Learning algorithms for Shallow Circuits 168
9.4 Discussion and Conclusion 170

§ 9.1 Introduction

Quantum Learning can be broadly categorized into two specific sub-classes based on the type of objects being learned:

1. **Quantum Learning for Quantum objects (QLQ).** This sub-field of quantum learning theory aligns with Feynman's original vision of using quantum computers to figure out quantum phenomena. Examples include *learning* arbitrary quantum states, quantum channels, and quantum processes. We refer the reader to [DB18; AA24; Hua24] for more details on this area of quantum learning.
2. **Quantum Learning for Classical objects (QLC).** In this area of quantum learning, we care about quantum advantage in the context of classical learning tasks - such as learning Boolean functions. The results of this thesis and the contents of this chapter fall into this category. Gyurik and Dunjko [GD23] considered various sub-categories in the *realizable* QLC setup and their relative advantages compared to a purely classical setup.

We outline results on separations (or lack thereof) between the sample complexity and time complexity of quantum learning algorithms and classical learning algorithms in Sections 9.2 and 9.3, respectively.

§ 9.2 Sample Complexity separations

In the noiseless setting, Blumer et al. [Blu+89] and Hanneke [Han16] showed that the classical (ε, δ)-PAC sample complexity of learning a concept class $\mathcal{C}$ with VC-dimension d is $\Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$. In the agnostic setting, Vapnik and Chervonenkis [VC74] and Talagrand [Tal94] showed that the (ε, δ)-agnostic PAC sample complexity of learning a concept class $\mathcal{C}$ with VC-dimension d is $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$.

We recall that having access to k classical samples (or quantum samples) is analogous to making k queries to the EX oracle (respectively, the QEX oracle) in the realizable case, and making k queries to the AEX oracle (respectively, the QAEX oracle) in the agnostic case. The first results on quantum vs classical sample complexity were given by Servedio and Gortler [SG01] who showed that quantum and classical sample

complexities were polynomially related in the noiseless setting as $D = O(nQ)$, for any concept class $\mathcal{C}$ that is PAC learnable using D queries to the EX oracle and Q queries to the QEX oracle.

This result was later improved by Arunachalam and Wolf [AW17b], who showed that the quantum sample complexity in both the noiseless and agnostic settings is equal to the respective classical sample complexities up to a constant factor, thereby proving that quantum samples are not inherently more powerful than classical samples. The proof techniques of [AW17b] included a state identification argument using "Pretty Good Measurement" and Fourier analysis. Hadiashar et al. [HNS24] later reproved the results of [AW17b] using information-theoretic arguments.

Lemma 9.1 (Quantum Sample Complexity [AW17b; HNS24])**.** Given a benchmark concept class $\mathcal{C}$ with VC-dimension d, the (ε, δ)-quantum PAC sample complexity of learning $\mathcal{C}$ is $\Theta\left(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$, while the (ε, δ)-quantum agnostic PAC sample complexity of learning $\mathcal{C}$ is $\Theta\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$.

When marginal distributions over the instances are assumed, however, Lemma 9.1 may not hold. For example, Caro [Car20] gives quantum sample complexity speedups for Boolean linear functions under bounded noise settings when the samples are drawn from a product distribution. This raises the following questions:

Open Problem 12. Does there exist any quantum sample complexity speedup for learning DNFs in the noiseless/RCN settings when the marginal distribution over the instances is a product distribution?

Open Problem 13. Does there exist a quantum sample complexity speedup for DNFs in the agnostic noise setting if the marginal distribution over the instances is assumed to be a smoothed product distribution?

Open Problem 14. Does there exist a quantum sample complexity speedup for halfspaces in the agnostic noise setting if the marginal distribution over the instances is assumed to be an isotropic log-concave distribution?

▷ Generalized Quantum Example Oracles.

Recently, Caro et al. [Car+24] proposed a new type of quantum agnostic example oracle, that they term mixture-of-superpositions (MOS) oracle $\mathcal{O}_{\mathfrak{D}}$. For any arbitrary

distribution $\mathfrak{D}$ over $\mathcal{X} \times \mathbb{F}_2$, querying $\mathcal{O}_{\mathfrak{D}}$ yields the mixed state

$$\rho_{\mathfrak{D}} = \mathop{\mathbb{E}}_{f \sim \mathcal{F}_{\mathfrak{D}}} \left[\left| \psi_{\mathfrak{D}_{\mathcal{X}}, f} \right\rangle \left\langle \psi_{\mathfrak{D}_{\mathcal{X}}, f} \right| \right], \tag{9.1}$$

where $\mathfrak{D}_{\mathcal{X}}$ is the marginal distribution over $\mathcal{X}$, and $\mathcal{F}_{\mathfrak{D}}$ is the probability distribution over all labeling functions $f : \mathcal{X} \mapsto \{0, 1\}$ induced by $\mathcal{D}$. Measuring all $n + 1$ qubits of $\rho_{\mathfrak{D}}$ produces a sample from $\mathfrak{D}$ as follows:

$$\langle x, y | \rho_{\mathfrak{D}} | x, y \rangle = \mathop{\mathbb{E}}_{f \sim \mathcal{F}_{\mathfrak{D}}} \left[\left| \langle x, y | \psi_{\mathfrak{D}_{\mathcal{X}}, f} \rangle \right|^2 \right] = \mathop{\mathbb{E}}_{f \sim \mathcal{F}_{\mathfrak{D}}} \left[\mathfrak{D}_{\mathcal{X}}(x) \cdot \mathbb{1}_{[f(x)=y]} \right] \tag{9.2}$$

$$\implies \langle x, y | \rho_{\mathfrak{D}} | x, y \rangle = \mathfrak{D}_{\mathcal{X}}(x) \cdot \Pr_{f \sim \mathcal{F}_{\mathfrak{D}}} [\mathbb{1}_{[f(x)=y]}] = \mathfrak{D}(x, y). \tag{9.3}$$

The MOS oracle straightforwardly generalizes the QEX, QREX, and QAEX oracles, and it is admittedly more general than the QAEX oracle since it allows us to model correlated label noise. Later in Section 9.3.2, we shall see results that indicate the power of the MOS oracle. However, with respect to the sample complexity, [Car+24] prove that in the agnostic setting, quantum (MOS) sample complexity is equivalent to classical sample complexity up to logarithmic factors. Concretely, the MOS quantum agnostic sample complexity is $\Omega\left(\frac{d}{\varepsilon^2 \log d} + \frac{1/\delta}{\varepsilon^2}\right)$. This naturally leads us to the following question.

Open Problem 15. Is the $\frac{d}{\varepsilon^2 \log d}$ term in the mixture-of-superpositions sample complexity tight?

If there is a positive answer to Open problem 15, it would indicate that the MOS model provides a slight (but non-trivial) advantage over the standard quantum example models even in terms of sample complexity.

§ 9.2.1 Sample Complexity in the Multi-class setting

Mohan and Tewari [MT23] investigated a host of different learning settings, such as noisy and agnostic offline and online quantum PAC learning, where $|\mathcal{Y}| < \infty$, and showed that quantum and classical sample complexities are equivalent up to constant factors in all but one of these cases (the bounds in the online multi-class agnostic setting are not tight and differ from the best-known classical bounds by a log factor[2]). The results of Mohan and Tewari [MT23] naturally lead to the following questions.

[2] See section 5.4 of [MT23] for a detailed discussion on this setting

Open Problem 16. Are the classical and quantum query/sample complexities equivalent (up to constant factors) in the online multi-class agnostic setting?

Open Problem 17. Are the classical and quantum query/sample complexities equivalent (up to constant factors) in the unbounded label case, even in the realizable offline PAC setting?

Open Problem 18. Are the classical and mixture-of-superposition sample complexities equivalent (up to constant factors) in the multi-class realizable and agnostic settings?

Finally, we may ask the above questions in terms of other recently introduced quantum online learning frameworks as well.

Open Problem 19. Does the online learning framework in [MT23] extend to the more general quantum online-learning framework introduced in [Ban+24]?

§ 9.2.2 Sample Complexity separations via Unitary Access

In Definitions 3.3 and 3.4, instead of assuming query access to the QEX and QAEX oracles, the learner can instead be provided with access to copies of the corresponding quantum states in the RHS of Eqs. (3.1) and (3.3). The results of [AW17b; HNS24; MT23] also hold in this equivalent setting. The learning algorithms throughout this thesis assume this setup.

On the other hand, if we assume a white-box setting, where the quantum learner has unitary access to the quantum oracles generating the quantum samples directly (instead of only being able to access the quantum samples), then Salmon et al. [SSG24] showed that the sample complexity of (ϵ, δ)- (noiseless) PAC learning a concept class $\mathcal{C}$ with VC-dimension d is $\tilde{O}\left(\frac{d}{\sqrt{\epsilon}} + \frac{\log(1/\delta)}{\sqrt{\epsilon}}\right)$. In the light of the results of Mohan and Tewari [MT23], the following questions arise.

Open Problem 20. Does the quantum sample complexity advantage in [SSG24] extend to the agnostic setting?

Open Problem 21. Does the quantum sample complexity advantage in [SSG24] extend to the multi-class setting as explored in [MT23]?

The proof of Salmon et al. [SSG24] critically uses the fact that having unitary access implies having access to the inverses of the unitaries. One might ask the following question, in terms of a more grey-box setting.

Open Problem 22. What is the quantum (realizable/agnostic) sample complexity of learning when the quantum learner has access to a quantum channel that generates quantum examples?

Here, we assume that having access to a quantum channel does not necessarily imply access to its inverse.

§ 9.3 Time Complexity separations

The results in Section 9.2 are information-theoretic in nature and do not disallow quantum vs classical separations in the computational sense. In other words, the results in Section 9.2 indicate that every concept class $\mathcal{C}$ that is learnable by quantum learners using a polynomial number of samples/queries is also learnable by classical learners using a polynomial number of samples/queries. However, Lemma 9.1 does not imply that if a concept class $\mathcal{C}$ is efficiently learnable by a quantum learner, it must be efficiently learnable by a classical learner. In this section, we investigate separations in the running times between quantum and classical learners.

§ 9.3.1 Separations via the Hidden Subgroup Problem

Servedio and Gortler [SG01] first provided strong evidence that there is a separation between quantum and classical learning by using a concept class $\mathcal{C}$ based on factoring Blum integers[3] [KV94a]. Factoring Blum integers is believed to be computationally intractable classically [KV94a; AK95], but can be efficiently solved by quantum algorithms for the Abelian Hidden Subgroup Problem (Abelian Hidden Subgroup Problem (HSP)) [Sho94].

A similar separation between classical and quantum was later demonstrated by Liu et al. [LAT21], using a concept class $\mathcal{C}_{\text{DLP}}$ based on the Discrete Logarithm Problem (DLP).

Definition 9.1 (DLP)**.** Given a large prime p, a generator g of $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$, and an input $x \in \mathbb{Z}_p^*$, compute $\log_g x$ in time poly $\left(\lceil \log_2 p \rceil\right)$.

It is widely believed that the DLP problem is classically intractable [BM84], but we can efficiently solve the DLP problem using Shor's algorithm [Sho94]. Using this technique, [LAT21] obtained a quantum kernel estimation procedure to learn $\mathcal{C}_{\text{DLP}}$.

[3] A Blum integer is an integer $N = pq$ where p, q are m-bit primes congruent to $3 \mod 4$.

▷ The Hidden Subgroup Problem over finite Abelian groups.

The results obtained by [SG01; LAT21] are not surprising, especially in light of Simon's algorithm and Shor's algorithm, which are textbook examples of quantum speedups. Both Simon's algorithm and Shor's algorithm solve restricted instances of a more general problem known as the HIDDEN SUBGROUP PROBLEM (HSP), defined below.

Definition 9.2 (The Hidden Subgroup Problem)**.** Let $\mathcal{G}$ be a known finite group, $\mathcal{S}$ be a finite set, and a black-box function $f : \mathcal{G} \mapsto \mathcal{S}$ s.t. $f(x) = f(y) \iff x\mathcal{H} = y\mathcal{H}$, where $\mathcal{H} \leqslant \mathcal{G}$ is some unknown subgroup, and $h \in \mathcal{H}$. Determine a generating set for $\mathcal{H}$.

In Definition 9.2, the black-box function f is said to *hide* the subgroup $\mathcal{H}$. Alternatively, f can be thought of as separating the cosets of $\mathcal{H}$. Most exponential separations between quantum and classical computing are a consequence of quantum speedups w.r.t. HSP over Abelian groups.

Example 9.1. Restricted versions of the Abelian-HSP problem amenable to quantum speedups include

- **Deutsch's problem:** $\mathcal{G} = \mathbb{Z}_2$ and $\mathcal{H}$ is either 0 (balanced) or $\mathbb{Z}_2$ (constant),
- **Simon's problem:** $\mathcal{G} = \mathbb{Z}_2^n$ and $\mathcal{H} = \{0, s\}$, where $s \in \mathbb{Z}_2^n$ is some secret, and
- DLP: $\mathcal{G} = \mathbb{Z}_p \times \mathbb{Z}_p$, and $\mathcal{H} = \{(0,0), (1, \log_g x), \ldots, (p-1, (p-1)\log_g x)\}$.

In fact, we know how to efficiently solve any Abelian HSP problem.

Lemma 9.2 (Efficiently solving Abelian HSP, Theorem 2.2 of [EH00])**.** Let $\mathcal{G}$ be a known finite group, $\mathcal{S}$ be a finite set, $f : \mathcal{G} \mapsto \mathcal{S}$ be a black-box function that hides a subgroup $\mathcal{H} \leqslant \mathcal{G}$. There exists a quantum algorithm that has size $O(\text{polylog}(|\mathcal{G}|))$ and runs in time $\log|\mathcal{G}|$ to output a generating set of $\mathcal{H}$ w.p. at least $1 - 1/|\mathcal{G}|$.

A critical component in efficiently solving the HSP problem over finite Abelian groups is the existence of an efficient Quantum Fourier Transform (QFT) over $\mathcal{G}$. Cleve and Watrous [CW00] showed the existence of a $O(\log n + \text{polylog}(1/\varepsilon))$ depth and $O(n \log n/\varepsilon)$ size circuit to approximate QFT over the group $\mathbb{Z}_{2^n}$. Cheung and Mosca [CM01] proposed an efficient quantum decomposition algorithm, where given a generating set $\{g_1, \ldots, g_s\}$ of a finite Abelian group $\mathcal{G}$, we can output a set of elements $d_1, \ldots, d_\ell$ s.t. $\mathcal{G} = \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_\ell}$ in time $\text{polylog}(|G|)$. Combining the results of Cleve and Watrous [CW00] and Cheung and Mosca [CM01] gives us Shor's factoring algorithm (via Lemma A.1).

▷ Decision Problems on the Hidden Subgroup as Concept classes.

We now define a new concept class induced by the HSP problem.

Definition 9.3 (HSP induced Concept class)**.** Any decision problem over an HSP instance $P = (\mathcal{G}, \mathcal{H})$ that can be posed as a membership problem over a set of instances $\mathcal{X}$ is said to be a concept class $\mathcal{C} : \mathcal{X} \mapsto \{0,1\}$ induced by HSP_P.

Example 9.2. Let $P = (\mathcal{G}, \mathcal{H})$ be an HSP instance s.t. $|\mathcal{G}| = 2^n$. The following decision problems $\mathcal{C} : \mathbb{F}_2^n \mapsto \{0,1\}$ are concept classes induced by HSP_P:

- Given a string $x \in \mathbb{F}_2^n$, is $|\mathcal{H}| \geqslant |x|$?
- Given a string $x \in \mathbb{F}_2^n$, does $\mathcal{H}$ have a generating set of size at least $|x|$?

Remark 9.3.1. Definition 9.3 is not restricted to the Abelian HSP case and can be defined for HSP over arbitrary finite groups.

Classically, obtaining efficient learners for concept classes induced by HSP over arbitrary finite Abelian groups is subject to various hardness assumptions, while efficient quantum learning algorithms can be obtained for any concept class induced by HSP over finite Abelian groups. This leads us to the following proposition.

Proposition 9.1 (Quantum Learning for Abelian HSP-induced concept classes)**.** If $\mathcal{C} : \mathbb{F}_2^n \mapsto \{0,1\}$ is a concept class induced by some HSP over a finite Abelian group $\mathcal{G}$, then there exists an efficient quantum learning algorithm for $\mathcal{C}$.

Observation 9.3.1. Assuming the hardness of DLP [KV94a; AK95], Proposition 9.1 implies the separation between quantum and classical learning for several HSP-induced concept classes (as seen in [SG01; LAT21]).

▷ The Hidden Subgroup Problem over Arbitrary Groups.

Even though known efficient algorithms for HSP over Abelian groups require efficient subroutines for QFT, it is unknown if the ability to efficiently perform QFT over a group G is either a necessary or a sufficient condition for efficiently solving HSP over $\mathcal{G}$ in the non-Abelian case. Ettinger et al. [EHK04] show that there exist QFT based quantum algorithms to solve HSP over arbitrary finite groups $\mathcal{G}$, which make only $\log|G|$ queries to O_f but requires exponential running time.

Efficient algorithms exist for exactly calculating or approximating QFT over classes of *well-behaved* non-Abelian groups, such as metacyclic groups[4], or even metabelian groups[5] [MRR06]. The reader is encouraged to peruse the surveys by Lomont [Lom04] and Wang [Wan10] for a comprehensive overview of HSP over arbitrary finite groups, which is an important problem in its own right, even outside the context of learning

[4] A metacyclic group $\mathcal{G}$ is a group having a cyclic normal subgroup $\mathcal{N}$ s.t. $\mathcal{G}/\mathcal{N}$ is also cyclic.
[5] A metabelian group $\mathcal{G}$ is a group having a normal subgroup $\mathcal{N}$ s.t. $\mathcal{N}$ and $\mathcal{G}/\mathcal{N}$ are both Abelian.

theory. Efficient quantum algorithms for HSP over certain classes of finite non-abelian groups imply very interesting results in theoretical computer science, such as efficient solvability of GRAPH ISOMORPHISM (GI) and SHORTEST VECTOR PROBLEM (Shortest Vector Problem (SVP)). SVP, in particular, is central to the field of post-quantum cryptography, and its connection to learning is discussed in the next section.

▷ Open Questions.

Consider the symmetric group S_n. Since S_n is a group with $n!$ elements, defining any HSP-induced concept class that displays a separation between quantum and classical learnability would be very interesting. It might also be worthwhile looking at concept classes induced by subgroups of S_n with more structure in pursuit of efficient quantum learning algorithms.

Open Problem 23. Does there exist a concept class induced by HSP on S_n, such that there is a super-polynomial separation between quantum and classical learning algorithms?

Open Problem 24. Does there exist a concept class induced by HSP on a subgroup of S_n (such as a permutation group or an alternating group), such that there is a super-polynomial separation between quantum and classical learning algorithms?

We note here that the answer to both Open problems 23 and 24 is likely to be negative when we require the additional notion of efficient learnability for quantum algorithms. We specifically highlight the important case of dihedral groups due to their close connection to Lattice-based cryptography.

Definition 9.4 (Dihedral Groups)**.** A dihedral group D_{2N} is a non-abelian group of order 2N that is generated by two elements x and y s.t.

$$x^N = 1, \quad y^2 = 1, \quad yxy = -x.$$

D_{2N} is group of symmetries of a regular polygon of 2N sides. The elements x and y are rotation and reflection operations about the vertical axis.

See Fig. 9.1 for a geometric interpretation of Dihedral groups. Informally, an efficient quantum algorithm for solving dihedral HSP implies that SVP $\in$ BQP [Reg09]. This would essentially imply that NP $\subseteq$ BQP since SVP has been shown to be NP-hard under restricted settings [Ben23]. The best-known algorithm for solving the dihedral

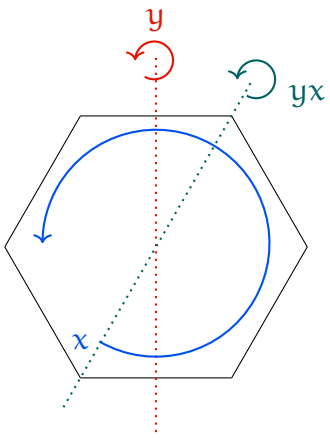


Figure 9.1: Geometric interpretation of Dihedral group D_6 as a group of symmetries of a regular hexagon. Here y denotes reflection across the vertical axis; x denotes a counter-clockwise rotation; and yx is reflection across the slanted line.

HSP is a subexponential time algorithm by Kuperberg [Kup05]. We now ask the following question.

Open Problem 25. Does there exist a concept class induced by HSP on an arbitrary dihedral group $\mathcal{G}$ for $N \geqslant 3$, which is efficiently quantumly learnable?

§ 9.3.2 Separations based on Lattice-Based Problems

In computational hardness, we usually care about problems with known *worst-case hardness guarantees* since we want to design algorithms that run efficiently even on the worst possible input. However, for cryptographic schemes to be secure, we require hardness guarantees to hold even for random keys, i.e., we require *average-case hardness guarantees*. This is a challenging task since it is not immediately apparent how to create hard instances of problems even when it has worst-case hardness guarantees. In fact, for many NP-hard problems like Graph coloring (see [DF89]), such a guarantee does not exist.

▷ Introducing the Learning with Errors Problem.

Ajtai [Ajt96] showed that for certain lattice-based problems, such a reduction from worst-case to average-case hardness exists. Following this, Regev [Reg09] introduced the LEARNING WITH ERRORS (Learning with Errors (LWE)) problem and showed that an efficient algorithm to solve the LWE problem gives us an efficient algorithm[6] for SVP. We define the decision version of the LWE problem now.

[6] Efficiently solving LWE actually implies an efficient solution to the Gap-SVP problem.

Definition 9.5 (Decision Version of LWE)**.** We are given m independent samples $(x, y) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, for some prime $q \geqslant 2$, with the promise that they are sampled according to one of the following distributions:

1. The uniform distribution: $(x, y) \sim \mathfrak{U}\left(\mathbb{Z}_q^n \times \mathbb{Z}_q\right)$.
2. The LWE distribution: The marginal distribution of $x \in \mathbb{Z}_q^n$ is uniform and $y = \langle s, x\rangle + \varepsilon \mod q$, where $s \in \mathbb{Z}_q^n$ is a secret string, and $\varepsilon \sim \mathfrak{D}$ is some noise parameter.

Determine if the samples are drawn from the LWE distribution.

The LWE problem is also a generalization of another important learning problem - Learning Parities with Noise (Learning Parity with Noise (LPN)) where $q = 2$ and $\mathfrak{D}$ is a Bernoulli distribution over $\mathbb{Z}_2$. LPN is an average-case version of the NP-hard problem decoding a linear code [Lyu05]. Without the presence of the random noise term ε, LWE and LPN can be efficiently solved using Gaussian elimination over the samples.

Remark 9.3.2. Unless explicitly specified, the LWE and LPN problems are assumed to be in the RCN setting, i.e., ε is RCN noise.

In the previous section, we saw that SVP reduces to the Dihedral HSP, similar to LWE. We make the connection between these two classes of problems explicit below.

Fact 9.1 (Connection between LWE and the Dihedral HSP)**.** LWE can be reduced (average-worst case reduction) to a stronger variant of the Dihedral HSP, known as the robust Dihedral HSP. Since known subexponential algorithms for solving dihedral HSP [Kup05] do not quite work for robust Dihedral HSP, there does not exist any subexponential learning algorithm for LWE.

▷ Learning algorithms for LWE and LPN.

Blum et al. [BKW03] provided the first classical sub-exponential time and sample learning algorithm for LWE (and hence LPN) that requires $2^{O(n/\log n)}$ time and samples. Interestingly, Feldman et al. [Fel+06] showed that LPN w.r.t. uniform marginals in the *agnostic setting* reduces to LPN w.r.t. uniform marginals in the RCN setting. Hence, the algorithm by [BKW03] also implies a sub-exponential time/sample learning algorithm for LPN in the agnostic setting (under uniform marginals).

In the quantum setting, however, there exist efficient quantum learning algorithms for LWE [GKZ19] and LPN [CSS15], respectively). The algorithms of [CSS15; GKZ19]

assume access to the following $\mathrm{QREX}_{\mathrm{LWE}}^{s,\varepsilon}$ oracle:

$$\mathrm{QREX}_{\mathrm{LWE}}^{s,\varepsilon}|0\rangle|0\rangle \mapsto \frac{1}{\sqrt{q^n}}\sum_{x\in\mathbb{F}_q^n}|x\rangle|\langle s,x\rangle + \varepsilon \mod q\rangle \quad , \varepsilon \sim \mathfrak{D}.$$

It is important to note that the results of [GKZ19] only hold when the quantum learner is provided oracle access to the $\mathrm{QREX}_{\mathrm{LWE}}^{s,\varepsilon}$ oracle, unlike in the classical case, where we only have access to m classical labeled examples.

Remark 9.3.3. The results of [GKZ19] also follow straightforwardly from the MOS example model of Caro et al. [Car+24].

Despite the presence of powerful oracle models providing separations, it is important to understand the exact avenue of speedup for quantum learners. The following question therefore naturally comes to mind:

Open Problem 26 (Quantum learning for LWE under random examples)**.** Given access to quantum examples of the form: $\frac{1}{\sqrt{|\mathcal{S}|}}\sum_{x\in\mathcal{S}\subseteq[m]}|x\rangle|\langle s,x\rangle + \varepsilon \mod q\rangle$, where $\mathcal{S}$ is unknown and $x \sim \mathfrak{U}\left(\mathbb{Z}_q^n\right)$, does there exist an efficient quantum learning algorithm for LWE?

We note that it is likely that the answer Open problem 26 to is a negative one, since LWE is believed to be hard for many reasons. If efficient quantum algorithms exist for LWE under random examples, then there would exist efficient quantum algorithms for breaking the LWE cryptosystem. This is highly unlikely since such a quantum algorithm would also imply an efficient algorithm for SVP. If SVP is believed to be NP-complete, a positive answer to Open problem 26 would imply $\mathsf{NP} \subseteq \mathsf{BQP}$.

Unlike LWE, cryptographic applications for LPN have been limited in scope [Pie12]. Recently, a series of works [Bra+19; YZ21] showed that the worst-case hardness of the nearest codeword problem (NCP) under very low noise rates is implied by the hardness of LPN at very high noise rates. However, unlike the hardness guarantees of SVP, NCP is not believed to be NP-hard [Bra+19]. Therefore, it is important to question the existence of an efficient quantum algorithm for LPN as a separate question from LWE.

Open Problem 27 (Quantum learning for LPN under random examples)**.** Given access to quantum examples of the form:

$$\frac{1}{\sqrt{|\mathcal{S}|}}\sum_{x\in\mathcal{S}\subseteq[m]}|x\rangle|\langle s,x\rangle + \varepsilon \mod 2\rangle,$$

where s is unknown, $x \sim \mathcal{U}\left(\mathbb{Z}_2^n\right)$, and $\varepsilon \sim \mathrm{Ber}(\mathbb{Z}_2)$, does there exist an efficient quantum learning algorithm for LPN?

§ 9.3.3 Learning separations for Halfspaces

Halfspaces are a class of Boolean functions that are extremely important to the field of learning algorithms. Boosting algorithms, SVMs, and neural networks all employ some variant of the halfspace learning problem under the hood. A halfspace is a Boolean function $c \in \mathcal{C} : \mathbb{R}^d \mapsto \mathbb{F}_2$ defined as follows:

$$c(x) = \mathrm{sign}\left(\langle a, x\rangle - \theta\right) = \mathrm{sign}\left(\sum_{i\in[m]} a_i x_i - \theta\right), \quad a, x, \theta \in \mathbb{R}^d.$$

Any learner that takes labeled samples drawn according to a distribution $\mathfrak{D}$ over $\mathbb{R}^d \times \mathbb{F}_2$ and outputs a hypothesis h s.t. $\mathrm{err}_{\mathfrak{D},\mathcal{C}}(h) \leqslant \mathrm{opterr}_{\mathfrak{D}}(\mathcal{C}) + \beta + \varepsilon$ is a beta-optimal learner. The learner is considered to be efficient if it runs in time $\mathrm{poly}\left(d, \frac{1}{\varepsilon}\right)$.

It is easy to see that in the realizable case/noiseless setting, halfspaces can be efficiently learnt using linear programming techniques. Classical efficient learners for halfspaces also exist for bounded noise models such as RCN [BF96], Massart noise [Dia+20] and Tsybakov noise [Dia+21a] under log-concave marginal distributions.

The most interesting case of halfspace learning seems to be beyond bounded noise models. Learning halfspaces in the agnostic setting under the uniform marginal assumption directly implies learning algorithms for LPN. Hence, halfspace learning seems to be a much harder problem than LPN. Various hardness results are known for improper [Dan15; Tie23] as well as proper [GR09; Fel+06; Dia+21b] classical learning of halfspaces in the agnostic setting.

It is important to note that the hardness results are in the random classical example setting (without queries). Earlier, we have seen evidence of quantum and classical learning separations in the case of LWE [GKZ19] or LPN [CSS15] when the quantum learner has query access to the QREX oracle. It is worthwhile to ask if such a separation is possible even for halfspace learning.

Open Problem 28. Does there exist a sample/query/time efficient quantum agnostic learner with access to QAEX queries for halfspaces assuming uniform marginal over the instances?

§ 9.3.4 Learning separations for Decision Trees

Table 9.1: Comparing different algorithms for learning size-t decision trees on n-bit Boolean functions. Note here that t and $1/\varepsilon$ can be as large as $\text{poly}(n)$, which renders the running time of many of the algorithms given below as super-polynomial. The algorithms of Chatterjee et al. [CSB24] are strictly polynomial in all parameters while being the only algorithm to work in the agnostic and realizable settings and not use membership queries (denoted by MQ). Here we note that the number of training samples m required for learning is $\text{poly}(n)$.

Work	Setting	Type	Setting	MQ	Runtime
EH 1989	Classical	Proper	Realizable	No	$\text{poly}\left(n^{\log t}, 1/\varepsilon\right)$
KM 1991	Classical	Improper	Realizable	Yes	$\text{poly}\left(n, t, 1/\varepsilon\right)$
LMN 1993	Classical	Proper	Realizable	No	$\text{poly}\left(n^{\log(t/\varepsilon)}\right)$
MR 2002	Classical	Proper	Agnostic	No	$\text{poly}\left(n^{\log(t/\varepsilon)}\right)$
GKK 2008 KK 2009 Feldman 2010	Classical	Improper	Agnostic	Yes	$\text{poly}\left(n, t, 1/\varepsilon\right)$
BLT 2020	Classical	Proper	Agnostic	No	$\text{poly}\left(n^{\log t}, 1/\varepsilon\right)$
CSB 2024	Quantum	Improper	Realizable	No	$\text{poly}\left(n, t, 1/\varepsilon\right)$
	Quantum	Improper	Agnostic	No	$\text{poly}\left(n, t, 1/\varepsilon\right)$

§ 9.3.5 Learning separations for DNFs

A DNF function f is a disjunction of DNF clauses, which in turn are conjunctions of Boolean literals $\tilde{x}$. Mathematically,

$$f(x) = C_1(x) \vee C_2(x) \vee \ldots \vee C_m(x),$$

where each clause $C(x) = \wedge_{i \in S} \tilde{x}_i$ is defined w.r.t. some $S \in [n]$. The width of DNF f is the maximum number of literals in any clause of f, and the size of DNF f is the total number of clauses of f.

When the learner only has access to random examples, Daniely and Shalev-Shwartz

[DSS16] showed that even improperly learning DNFs would imply an efficient algorithm for refuting random k-SAT formulas.[7] Since learning DNFs (in the realizable/RCN settings) reduces to learning LPN [Fel+06], the results of [DSS16] also give hardness results for classically learning LPN with random examples. Classically, we know of efficient improper learning algorithms for DNF (depth-2 AC^0 circuits) in the noiseless/RCN settings under the uniform (or "near"-uniform) marginal assumption, only when the learner has access to membership queries [Jac97]. In the quantum learning regime, however, Bshouty and Jackson [BJ95] gave efficient improper quantum learning algorithms for DNFs in the realizable and RCN settings using quantum learners with access to QEX or QREX oracles.

In the non-uniform marginal setting, Feldman [Fel12] showed that an efficient classical learning algorithm for DNFs exists under the membership query model when the instances are sampled from any product distribution. They also show that a DNF learning algorithm exists for random examples only when the instances are sampled from a smoothed product distribution [ST04]. Kanade et al. [KRS19] showed that DNFs can be efficient PAC learned by quantum learners with access to the QEX when the samples are drawn from a product distribution. We end this section with the following "difficult" open question:

Open Problem 29. Does there exist an efficient agnostic quantum learning algorithm (with access to the QAEX oracle) for learning DNF formulas when the marginal distribution over the instances is a smoothed product distribution?

§ 9.3.6 Learning separations for Juntas

A junta is a Boolean function that only depends on a subset of its input variables. More formally, a k-junta is a Boolean function on n variables that depends only on an unknown subset of k variables and is invariant under the remaining $n - k$ variables.

Fact 9.2. Every k-junta can be expressed as a decision tree of size 2^k or a DNF formula of size k. Every decision tree of size k can be expressed as a DNF formula of size k.

Although the problem of learning k-juntas straightforwardly reduces to that of learning DNF functions of width k, we mention the results related to learning k-juntas separately here. Trivially, learning a k-junta from random classical examples can be done using $2^k \log n$ samples and $O(n^k)$ time (by solving linear equations). Mossel et al. [MOS04] gave the first non-trivial algorithm for learning k-juntas which takes time $O\left(n^{\omega k/\omega+1} \operatorname{poly}(n)\right)$, where matrix multiplication can be solved in time $O(n^\omega)$.[8]

[7] If $P \neq NP$, it is hard to refute k-SAT formulas [Hås01].

[8] Here, $\omega < 2.376$ is the matrix multiplication exponent. Hence [MOS04] takes time $O\left(n^{2k/3}\right)$.

This was later improved by [Val15] to $O\left(n^{0.6k}\,\mathrm{poly}\,(n)\right)$. Valiant [Val15] also gave a $O\left(n^{0.8k}\,\mathrm{poly}\,(n, 1/\varepsilon)\right)$ for learning k-juntas in the RCN setting with noise rate ε.

In the quantum setting, the DNF learning algorithm of [BJ95] implies a $O\left(2^{6k}\,\mathrm{poly}\,(n, 1/\varepsilon)\right)$ time and sample learning algorithm for k-juntas using QEX queries. Atıcı and Servedio [AS07] improved upon this result by giving a $O\left(\frac{k}{\varepsilon}\log k\right)$ time and $2^k \log 1/\varepsilon$ sample learning algorithm using QEX queries. Both of these results also hold in the RCN setting with noise rate η with an additional multiplicative factor of $\mathrm{poly}\,(1/\eta)$. In Chapters 4 and 6, we design the first quantum algorithm that learns k-juntas in the agnostic setting. The running time of our algorithm depends on n and has far worse dependence on k and ε. This brings us to the following question:

Open Problem 30. What is the best time and sample complexity for (quantum) learning k-juntas in the agnostic setting without using membership queries?

§ 9.3.7 Learning algorithms for Shallow Circuits

We conclude the discussion on separations between quantum and classical learning algorithms for the AC^0 and TC^0 circuit classes, which are ubiquitous in complexity theory, cryptography, and learning theory. For the sake of completeness, we formally define these circuit classes now.

Definition 9.6 (AC^0_d circuits and TC^0_d circuits)**.** An AC^0 circuit is a constant-depth, polynomial-size Boolean circuit on n-bits that consists of NOT, and unbounded fan-in AND and OR gates. A TC^0 circuit is a constant-depth, polynomial-size Boolean circuit on n-bits that consists of NOT, and unbounded fan-in AND, OR, and MAJ gates. When the circuit has depth $d = O(1)$, we use the notation AC^0_d and TC^0_d respectively.

Fact 9.3. For any $d > 0$, $\mathsf{AC}^0_d \subset \mathsf{TC}^0_d$.

In the context of Boolean functions and learning theory, AC^0 and TC^0 circuit classes are quite important. We know that DNF $\in \mathsf{AC}^0_2$ and intersections/majority of halfspaces can be computed by TC^0 circuits. Hence, efficient learning algorithms for AC^0 and TC^0 circuits would imply efficient learning algorithms for DNF and intersections of halfspaces. However, Arunachalam et al. [AGS21] showed the following impossibility results on learning AC_0 and TC_0.

Lemma 9.3 (Impossibility of learning Shallow classical circuits [AGS21])**.** The following impossibility results hold for learning TC_0 circuits.

1. If there does not exist a BQP algorithm to solve the LWE problem with ran-

dom examples, then no quantum learner can efficiently learn TC^0_2 even with access to membership queries.

2. If there does not exist a BQP algorithm to solve the Ring-LWE problem[9] with random examples, then no quantum learner can efficiently learn TC^0 even with access to membership queries.

Further, if there does not exist a strongly subexponential time quantum algorithm to solve the Ring-LWE problem with random examples, then there is no quasi-polynomial time quantum learner for AC^0 even with access to membership queries.

We note that Lemma 9.3 implies that there are no efficient quantum learning algorithms for learning intersections of halfspaces (this class of Boolean function is in TC^0_2) even with membership queries, assuming the hardness of LWE.

Unlike the strong hardness of learning TC^0 circuits, all hope is not lost in the case of AC^0 circuits. We have seen previously the class AC^0_2 is much easy to learn quantumly using just QEX and QREX queries [BJ95], while efficient algorithms are known for classical learners only with access to membership queries. The best-known classical learning algorithm for learning AC^0_3 circuits (AND of DNFs) with bounded top fan-in stems due to Ding et al. [DRG17] in a distribution-free setting. This leads us to the following questions:

Open Problem 31. Does there exist an efficient quantum learner (with access to queries) for AC^0_3 circuits (with bounded top fan-in) under uniform marginals?

Open Problem 32. What is the smallest $d > 1$ for which there does not exist an efficient quantum learner (with access to queries) for AC^0_d circuits (with bounded top fan-in) under uniform marginals?

We end this section by asking the following open questions regarding the learnability of shallow *quantum* circuit classes.

Open Problem 33. Does there exist an efficient quantum learner (with access to queries) for the QNC^0 class (with/without ancillas) as defined in [Wat+19]?

While QNC^0 is the simplest class of quantum circuits, Watts et al. [Wat+19] showed an exponential separation between QNC^0 and AC^0 with respect to search problems. As a first step towards solving Open problem 33, we can ponder upon the learnability

[9] See [LPR13] for a formal definition of the RLWE problem and its applications to cryptography.

of classical circuits enhanced by composition with shallow quantum circuits. This model was introduced by Slote [Slo24], who showed that the class $\mathrm{QNC}^0 \odot \mathrm{AC}^0_2$ cannot contain parity.

Open Problem 34. Does there exist an efficient quantum learner (with access to queries) for the $\mathrm{QNC}^0 \odot \mathrm{AC}^0_2$ class as defined in [Slo24]?

§ 9.4 Discussion and Conclusion

In this section, we place the results of Chapter 4, and by extension, the central contribution of this thesis in the context of the results highlighted in this chapter.

In this thesis, we design efficient quantum learning algorithms for size k decision trees in the agnostic setting. While efficient classical algorithms [KK09; GKK08b; Fel10] are also known for this setting, all of these classical solutions invoke the use of membership queries. In contrast, our algorithm only uses QAEX queries to learn decision trees. Our results also imply the learnability of all Boolean functions with bounded ℓ_1 Fourier Spectrum norm, such as k-juntas, Parity Decision trees, Affine subspaces, etc. We hope that the techniques in this thesis can be useful in answering Open problem 29, which in turn might provide clues about Open problems 31 and 32.

Appendix

Chapter A

Omitted Definitions

Contents

A.1 Group Theory 172

§ A.1 Group Theory

A group is a non-empty set $\mathcal{G}$ is a set with an associative binary operation $(\cdot) : \mathcal{G} \times \mathcal{G} \mapsto \mathcal{G}$, such that there exists an identity element in $\mathcal{G}$ w.r.t. $(\cdot)$, and every element of $\mathcal{G}$ has an inverse element in $\mathcal{G}$ w.r.t $(\cdot)$. If for all $x, y \in \mathcal{G}$, $x(\cdot)y = y(\cdot)x$, then $\mathcal{G}$ is an Abelian group. A generating subset $\mathcal{H}$ of $\mathcal{G}$ is a subset s.t. every element of $\mathcal{G}$ can be expressed as a combination of finitely many elements of $\mathcal{H}$ and their inverses. If $\mathcal{G}$ is a cyclic group, it has an element x that serves as the generator of the group. Any arbitrary finite Abelian group can be decomposed into cyclic groups as follows:

Lemma A.1 (Fundamental Structure Theorem for Finite Abelian Groups)**.** Every Abelian group $\mathcal{G}$ is isomorphic to a group of the form $\mathbb{Z}_{p_1^{a_1}} \times \mathbb{Z}_{p_2^{a_2}} \times \ldots \times \mathbb{Z}_{p_k^{a_k}}$, where $p_1, \ldots, p_k$ are prime numbers (not necessarily unique), and the decomposition is unique up to the order in which the factors $\{p_1^{a_1}, \ldots, p_k^{a_k}\}$ are written.

Remark A.1.1. $\mathbb{Z}_k$ is a finite Abelian group of order k w.r.t. the $+_{\bmod k}$ operation. $\mathbb{Z}_k$ is also a cyclic group if $k = 1, 2, 4$ or some power of an odd prime.

Chapter B

Omitted Algorithms

Contents

B.1 A General Framework for Realizable Boosting 174
B.2 The AdaBoost algorithm 176
B.3 The RealBoost algorithm 177
B.4 The SmoothBoost algorithm 178
B.5 The PotentialBoost algorithm 180

§ B.1 A General Framework for Realizable Boosting

Algorithm B.1: GENBOOST or General Boosting Algorithm

Input: ▷ A training set $\mathcal{S} = \{(x_i, y_i)\}_{i\in[m]}$ where $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and $\mathcal{Y} = \{\pm 1\}$.
▷ A bound on the number of iterations T, and a loss functional ℓ.
▷ Oracle access to T weak PAC learners $A_1, \ldots, A_T$ where $A_i : \mathcal{S} \times \mathfrak{D} \mapsto \mathcal{H}_i \subseteq \mathcal{H}$.
Initialize: Set initial weights $\mathfrak{D}_i^1$ for all $i \in [m]$ s.t. they add up to 1.

1 **for** $t = 1$ **to** T **do**
2 Obtain the weak classifier $h_t = A_t(\mathcal{S}, \mathfrak{D}^t)$. // $\mathcal{S}$ is weighted by $\mathfrak{D}^t$.
3 Compute the weight α_t of h_t. // This is known as the confidence of h_t.
4 **for** $i \in [m]$ **do**
5 Compute weight of x_i for $(t+1)^{th}$ iteration: $\tilde{\mathfrak{D}}_i^{t+1} = \alpha_t \cdot \mathfrak{D}_i^t \cdot \ell\left(h_t(x_i)\right)$.
6 Obtain distribution $\mathfrak{D}^{t+1}$ by normalizing $\tilde{\mathfrak{D}}^{t+1}$.
7 Compute the intermediate weighted ensemble $H^{(t)} = \operatorname{sign}\left(\sum_{j=1}^{t} \alpha_j h^j\right)$.

Output: Output $H = \underset{H^{(1)},\ldots,H^{(T)}}{\operatorname{argmin}} \ell(H^{(t)})$ as the strong learner.

GENBOOST (see Algorithm B.1) takes as input a training set $\mathcal{S}$ labeled according to an unknown concept class and several binary-valued weak learners. The output of GENBOOST is an ensemble of weighted weak classifiers that serves as a strong learner for the unknown concept class. The strong learner minimizes a loss functional $\ell : \operatorname{lin}(\mathcal{H}) \mapsto \mathbb{R}$, where $\mathcal{H} : \mathcal{X} \mapsto \mathcal{Y}$ is a class of hypothesis functions, and $\operatorname{lin}(\mathcal{H})$ denotes the set of all linear combinations of functions in $\mathcal{H}$. There are three main phases.

1. **Obtaining the t^{th} weak classifier.** At every iteration t, the algorithm assigns a weight (probability mass) $\mathfrak{D}_i^t$ to each instance x_i in the training set $\mathcal{S}$. This weighted training set is then passed to the t^{th} weak learner A_t, which returns a classifier $h_t \in \mathcal{H}_t$.
2. **Assigning weight to the weak classifier.** After obtaining h_t, GENBOOST assigns a weight to h_t, that indicates how confident h_t is in its predictions. This confidence score is used as the weight of h_t when it is used in the final ensemble.
3. **Assigning weight to the instances.** If some instance was "hard" to classify, i.e., $\ell(h_t(x_i))$ was large during the t^{th} iteration, then GENBOOST pays more attention (increases its weight) during the next iteration. Otherwise, GENBOOST pays less attention (decreases its weight) during the next iteration. In Algorithm B.1,

GENBOOST multiplies the current weight $\mathfrak{D}_i^t$ by the loss incurred $\ell(h_t(x_i))$, and then again multiplies the resulting term by the hypothesis weight α_t. Finally, the weights are normalized to form a probability distribution for the next iteration.

Example B.1. Suppose the loss functional ℓ takes values in the range $[0,1]$, and GENBOOST assigns confidence scores to hypotheses in the range $[0,1]$. Consider the following update scenarios for an instance x_i that has weight 0.3 in iteration t, and h_t incurs a large loss on x_i (say $\ell(h_t(x_i)) = 0.9$).

1. If $\alpha_t = 0.9$, the un-normalized weight of x_i for iteration $t+1$ is $\tilde{\mathfrak{D}}^{t+1} = 0.243$.
2. If $\alpha_t = 0.4$, the un-normalized weight of x_i for iteration $t+1$ is $\tilde{\mathfrak{D}}^{t+1} = 0.108$.

We can further contrast the above scenario with the case where h_t incurs a small loss on x_i (for example, $\ell(h_t(x_i)) = 0.4$). The un-normalized weight of x_i for iteration $t+1$ is $\tilde{\mathfrak{D}}^{t+1} = 0.108$.

§ B.2 The AdaBoost algorithm

Algorithm B.2: The AdaBoost algorithm [FS97]

Input: ▷ A training set $\mathcal{S} = \{(x_i, y_i)\}_{i\in[m]}$ where $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and $\mathcal{Y} = \{\pm 1\}$.
▷ Oracle access to a weak PAC learner A.

Initialize: Set weights $\mathfrak{D}_i^1 = 1/m$, $\forall i \in [m]$. Set no. of iterations T.

1 **for** $t = 1$ **to** T **do**

2 Obtain the weak classifier $h_t = A_t(\mathcal{S}, \mathfrak{D}^t)$. // $\mathcal{S}$ is weighted by $\mathfrak{D}^t$.

3 Compute the weight α_t of h_t as shown in Eq. (5.2).

4 **for** $i \in [m]$ **do**

5 Compute weight of x_i for $(t+1)^{\text{th}}$ iteration:

$$\tilde{\mathfrak{D}}_i^{t+1} = \mathfrak{D}_i^t \cdot \exp\{\alpha_t \cdot -y_i \cdot h_t(x_i)\},$$

6 $$\mathfrak{D}_i^{t+1} = \tilde{\mathfrak{D}}_i^{t+1}/Z_t = \frac{\tilde{\mathfrak{D}}_i^{t+1}}{\sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1}}.$$ // Z_t is a normalization term.

Output: Final classifier $H = \operatorname{sign}\left(\sum_{i=1}^{T} \alpha_t h_t\right)$.

§ B.3 The REALBOOST algorithm

Algorithm B.3: The REALBOOST algorithm [SS99; FHT00]

Input: ▷ A training set $\mathcal{S} = \{(x_i, y_i)\}_{i\in[m]}$ where $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and $\mathcal{Y} = \{\pm 1\}$.
▷ Oracle access to a C domain-partitioning weak PAC learner A.
Initialize: Set weights $\mathcal{D}_i^1 = 1/m$, $\forall i \in [m]$. Set no. of iterations T.

for $t = 1$ **to** T **do**

  Obtain the weak classifier $h_t = A_t(\mathcal{S}, \mathcal{D}^t)$ that induces a partitioning $\mathcal{X}^t = \{\mathcal{X}_1^t, \ldots, \mathcal{X}_C^t\}$ of the domain $\mathcal{X}$. // $\mathcal{S}$ is weighted by $\mathcal{D}^t$.

  for $k = 1$ *to* C **do** // Iterate over every partition.

    for $b \in \{-1, +1\}$ **do** // Iterate over every label.

      Compute $W_b^{k,t} = \sum_{i\in\mathcal{I}_{k,b}} \mathcal{D}_i^t$. // $\mathcal{I}_{k,b} = \{i \mid (x_i, y_i) \in \mathcal{S} \wedge x_i \in \mathcal{X}_k^t \wedge y_i = b\}$

    Compute $\beta_{k,t} = \frac{1}{2} \ln\left(\frac{W_+^{k,t}}{W_-^{k,t}}\right)$. // Z_t is a normalization term.

  Compute $Z_t = 2\sum_{k=1}^{C} \sqrt{W_+^{k,t} \cdot W_-^{k,t}}$. // Partition margins.

  for $i \in [m]$ **do**

    Compute weight of x_i for $(t+1)^{\text{th}}$ iteration, where $x_i \in \mathcal{X}_k^t$:

$$\mathcal{D}_i^{t+1} = \frac{\mathcal{D}_i^t \cdot \exp\{-y_i \cdot \beta_{k,t}\}}{Z_t}.$$

Output: Final classifier $H(x) = \operatorname{sign}\left(\sum_{i=1}^{T} \beta_{k,t}\right)$, where $x \in \mathcal{X}_k^t$.

§ B.4 The SMOOTHBOOST algorithm

Algorithm B.4: The SMOOTHBOOST Algorithm [Ser03]

Input: Classical weak learner A with sample complexity R, M Training Samples $S = \{(x_i, y_i)\}_{i\in[M]}$, and paramters $\kappa \in (0,1)$, $\theta \in [0, 1/2)$.

Initialize: Set $t = 0$, and for all $i \in [M]$, $N_0(i) = 0$, $m(N_1(i)) = 1$

Compute $s = \sum_{i\in[M]} m(N_0(i))$

while $s < \kappa M$ **do**

  We set the weights as $D_t(i) = \frac{m(N_t(i))}{\sum_{i\in[M]} m(N_t(i))}$.

  Prepare R i.i.d. examples with respect to D_t. Pass these to A which returns a weak hypothesis h_t.

  Set $N_t(i)$ and $m(N_{t+1}(i))$ for all $i \in [M]$.

  Set $t = t + 1$.

Set $h_{fin}(i) = \text{sign}\left(\sum_{j=0}^{t-1} h_j(i)\right)$.

Output: Hypothesis h_{fin}.

The parameter κ controls the error rate of the final hypothesis h_{fin}. The parameter θ is the desired margin of the final hypothesis, and the quantity $N_t(i) = N_{t-1}(i) + y_i \cdot h_t(i) - \theta$ is representative of the amount by which the cumulative sum of the hypotheses up to the t-th iteration beat the desired margin. The quantity $m(N_t)$ re-weights the samples by assigning more weight to samples which have smaller values of N_t.

$$m(N_{t+1}(i)) = \begin{cases} 1 & ; N_t < 0 \\ (1-\gamma)^{N_t/2} & ; N_t \geqslant 0 \end{cases} \tag{B.1}$$

This works similarly to ADABOOST and REALBOOST, as the weak learner is forced to evaluate samples which were performing poorly till this point with more priority. The difference here is that, there is a strict upper bound on the amount of weight that can be assigned to any particular instance - therefore making the intermediate distributions smooth. A few key features of SMOOTHBOOST are as follows.

1. SMOOTHBOOST uses multiplicative updates on weights, and never allows weights to exceed 1. This gives us a faster convergence rate than most other smooth boosting algorithms.
2. SMOOTHBOOST covers both binary-valued and real-valued hypothesis unlike MadaBoost [DW00]. Thus SMOOTHBOOST can also be retrofitted to be useful in the domain-partitioning case.
3. SMOOTHBOOST generates a large margin final hypothesis unlike other boosting

algorithms with smooth intermediate distributions.

4. Boosting algorithms that produce smooth intermediate distributions tend to converge much slower than non-smooth algorithms. For example, SMOOTHBOOST converges in poly $(1/\gamma^2)$ iterations, while AdaBoost converges in exp $(1/\gamma^2)$ iterations. This was also observed earlier in Table 5.1.

§ B.5 The POTENTIALBOOST algorithm

Algorithm B.5: The Potential-Boost algorithm

Input: (m, κ, η)-weak agnostic learner A with complexity R, and m labeled training samples $S = \{(x_i, y_i)\}_{i\in[m]}$.

Output: (κ/η)-optimal hypothesis $H^{\hat{t}}$ for $1 \leqslant \hat{t} \leqslant T$ such that $\mathrm{err}_S(H^{\hat{t}}) = \mathbf{argmin}_t \mathrm{err}_S(H^t)$.

Data: Initialize $H^0 = 0$, and a worst-case guess for T.

for $t = 1$ *to* T **do**

- Define $w_i^t = -\phi'(z_i) = \min\left\{1, e^{\{-H^{t-1}(x_i)\cdot y_i\}}\right\}$.
- **Relabeling Step:** Set $\tilde{y}_i = y_i$ w.p. $(1 + w_i^t)/2$, and w.p. $(1 - w_i^t)/2$ set $\tilde{y}_i = \bar{y}_i$.
- Pass the set of relabeled samples $\tilde{S} = \{(x_i, \tilde{y}_i)\}_{i\in[m]}$ to A to obtain intermediate hypothesis h^t.
- Let $\alpha_t = \frac{1}{m}\sum_{i\in[m]}\left(w_i^t \cdot y_i \cdot h^t(x_i)\right)$, and $\beta_t = \frac{1}{m}\sum_{i\in[m]}\left(w_i^t \cdot y_i \cdot -H^{t-1}(x_i)\right)$.
- If $\alpha_t > \beta_t$, set $H^t = H^{t-1} + \alpha_t \cdot h^t$. Otherwise, set $H^t = (1 - \beta_t)\, H^{t-1}$.

Chapter C

Omitted Proofs

Contents

C.1 Proofs from Chapter 5 181
C.2 Proofs from Chapter 7 184

§ C.1 Proofs from Chapter 5

Proposition 5.1. [Convergence of AdaBoost [FS97]] The training error of the final classifier obtained by AdaBoost in $O(\log m/\gamma^2)$ iterations, is zero w.r.t. $\mathfrak{D}^1$, where γ is the bias of weak learner A.

Proof. We want to compute the training error of H w.r.t. $\mathfrak{D}^1$. Let $\tilde{H}(x) = \sum_{i=1}^{T} \alpha_t h_t$. Recall from Definition 2.2 that this can be defined as follows:

$$\varepsilon_H^{\mathfrak{D}^1} = \hat{\mathrm{err}}_{S,\mathfrak{D}^1}^H = \sum_{i\in[m]} \mathfrak{D}_i^1 \cdot \mathbb{1}_{[H(x_i)\neq y_i]} \leqslant \sum_{i\in[m]} \mathfrak{D}_i^1 \cdot \exp\{-y_i \cdot \tilde{H}(x_i)\} \tag{C.1}$$

$$= \sum_{i\in[m]} \mathfrak{D}_i^{T+1} \cdot \prod_{t=1}^{T} Z_t \tag{C.2}$$

$$\implies \hat{\mathrm{err}}_{S,\mathfrak{D}^1}^H = \prod_{t=1}^{T} Z_t. \tag{C.3}$$

The inequality in Eq. (C.1) follows from the fact that

$$\mathbb{1}_{[H(x)\neq y]} = 1 \implies y \cdot H(x) \leqslant 0 \implies y \cdot \tilde{H}(x) \leqslant 0 \implies \exp\{-y \cdot \tilde{H}(x)\} \geqslant 1 \tag{C.4}$$

$$\implies \mathbb{1}_{[H(x)\neq y]} \leqslant \exp\{-y \cdot \tilde{H}(x)\}. \tag{C.5}$$

To see how we obtain Eq. (C.2), we first unroll the recurrence relation on the probability weights $\mathfrak{D}^t$, and obtain an expression for $\mathfrak{D}^{T+1}$ in terms of $\mathfrak{D}^1$ as follows:

$$\mathfrak{D}_i^{T+1} = \mathfrak{D}_i^1 \cdot \frac{\exp\{\alpha_1 \cdot -y_i \cdot h^1(x_i)\}}{Z^1} \cdot \ldots \cdot \frac{\exp\{\alpha_T \cdot -y_i \cdot h_t(x_i)\}}{Z_t} \tag{C.6}$$

$$\implies \mathfrak{D}_i^1 \cdot \exp\left\{-y_i \cdot \sum_{i=1}^{T} \alpha_t h_t\right\} = \mathfrak{D}_i^{T+1} \cdot \prod_{t=1}^{T} Z_t \tag{C.7}$$

$$\implies \mathfrak{D}_i^1 \cdot \exp\{-y_i \cdot \tilde{H}(x_i)\} = \mathfrak{D}_i^{T+1} \cdot \prod_{t=1}^{T} Z_t. \tag{C.8}$$

We now express Z_t in terms of the bias γ of the weak learner as follows:

$$Z_t = \sum_{i\in[m]} \tilde{\mathfrak{D}}_i^{t+1} = \sum_{i\in[m]} \mathfrak{D}_i^{t+1} \cdot \exp\{\alpha_t \cdot -y_i \cdot h_t(x_i)\} \tag{C.9}$$

$$= \sum_{i:y_i=h_t(x_i)} \mathfrak{D}_i^{t+1}\mathbb{1}_{[h_t(x_i)=y_i]} \cdot \exp\{-\alpha_t\} + \sum_{i:y_i\neq h_t(x_i)} \mathfrak{D}_i^{t+1}\mathbb{1}_{[h_t(x_i)\neq y_i]} \cdot \exp\{\alpha_t\} \tag{C.10}$$

$$= e^{-\alpha_t} \cdot \varepsilon_t + e^{\alpha_t} \cdot (1-\varepsilon_t) \tag{C.11}$$

$$= e^{-\alpha_t} \cdot (1/2+\gamma) + e^{\alpha_t} \cdot (1/2-\gamma) \tag{C.12}$$

In Eq. (C.12), we used the weak PAC learning assumption (see Definition 4.1) that the weak learner A always produces a classifier h_t s.t. $\varepsilon_t \leqslant 1/2-\gamma$, where γ is the bias of A. We plug the expression of α_t from Eq. (5.2) into Eq. (C.3) to obtain:

$$\hat{\text{err}}_{S,\mathfrak{D}^1}^H = \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} e^{-\alpha_t} \cdot (1/2+\gamma) + e^{\alpha_t} \cdot (1/2-\gamma) \tag{C.13}$$

$$= \prod_{t=1}^{T} \sqrt{\varepsilon_t/1-\varepsilon_t} \cdot (1/2+\gamma) + \sqrt{1-\varepsilon_t/\varepsilon_t}\,(1/2-\gamma) \tag{C.14}$$

$$\leqslant \prod_{t=1}^{T} \sqrt{\frac{1/2-\gamma}{1/2+\gamma}} \cdot (1/2+\gamma) + \sqrt{\frac{1/2+\gamma}{1/2-\gamma}} \cdot (1/2-\gamma) = \prod_{t=1}^{T} 2\sqrt{(1/2-\gamma)\,(1/2+\gamma)} \tag{C.15}$$

$$\implies \hat{\text{err}}_{S,\mathfrak{D}^1}^H \leqslant \prod_{t=1}^{T} \sqrt{1-4\gamma^2} = \left(\sqrt{1-4\gamma^2}\right)^T \leqslant \exp\left\{-2\gamma^2 T\right\}. \tag{C.16}$$

We obtain Eq. (C.16) by the inequality $1 + x \leqslant e^x$, for all x. Now, plugging in the value of $T = O(1/\gamma^2 \log m)$, we obtain that $\hat{\text{err}}^H_{S,\mathfrak{D}^1} < 1/m$. Since we started with m elements in the training set, the training error actually goes down to 0. □

Proposition 5.2. For weak learners that output bounded real-valued hypotheses $h_t : \mathcal{X} \mapsto [-1, +1]$, the normalization term Z_t can be upper-bounded as $Z_t \leqslant \prod_{t=1}^{T} \sqrt{1 - \varepsilon_t^2}$, where ε_t is the weighted training error of h_t on the training set S weighted by $\mathfrak{D}^t$.

Proof.

$$Z_t = \sum_{i=1}^{m} \mathfrak{D}_i^t \cdot \exp\left\{-\alpha_t \frac{1 + y_i \cdot h_t(x_i)}{2} + \alpha_t \frac{1 - y_i \cdot h_t(x_i)}{2}\right\} \tag{C.17}$$

$$\leqslant \sum_{i=1}^{m} \mathfrak{D}_i^t \left[\frac{1 + y_i \cdot h_t(x_i)}{2} e^{-\alpha_t} + \frac{1 - y_i \cdot h_t(x_i)}{2} e^{\alpha_t}\right] \tag{C.18}$$

$$= \frac{e^{\alpha_t} + e^{-\alpha_t}}{2} - \frac{e^{\alpha_t} - e^{-\alpha_t}}{2} \left[\sum_{i=1}^{m} \mathfrak{D}_i^t \cdot y_i \cdot h_t(x_i)\right] \tag{C.19}$$

Eq. (C.17) follows from expanding the term $\exp\{-\alpha_t \cdot -y_i \cdot h_t(x_i)\}$. Eq. (C.18) follows from the fact that $y_i \cdot h_t(x_i) \in [-1, +1]$ and the convexity of the exponential function. We note that Eq. (C.19) is minimized when

$$\alpha_t = \frac{1 + \sum_{i=1}^{m} \mathfrak{D}_i^t \cdot y_i \cdot h_t(x_i)}{1 - \sum_{i=1}^{m} \mathfrak{D}_i^t \cdot y_i \cdot h_t(x_i)} = \frac{1 - \varepsilon_t}{1 + \varepsilon_t}.$$

Plugging this value of α_t into Eq. (C.19) gives us $Z_t \leqslant \sqrt{1 - \varepsilon_t^2}$. □

Lemma 5.2. [Normalization term in REALBOOST [SS99]] The normalization term in REALBOOST can be expressed succinctly as $Z_t = 2 \sum_{k=1}^{C} \sqrt{W_+^{k,t} \cdot W_-^{k,t}}$, where $W_b^{k,t}$ is the weight of the label $b \in \{\pm 1\}$ in the k^{th} partition at the t^{th} iteration.

Proof. We first recall the partition-label weights $W_b^{k,t} = \sum_{i \in \mathcal{I}_{k,b}^t} \mathfrak{D}_i^t$ at the t^{th} iteration, where $\mathcal{I}_{k,b}^t = \{i \mid (x_i, y_i) \in S \wedge x_i \in \mathcal{X}_k^t \wedge y_i = b\}$. Now, we can rewrite the normalization term Z_t in terms of $W_b^{k,t}$ as follows:

$$Z_t = \sum_{k=1}^{C} \sum_{i \mid x_i \in \mathcal{X}_j^t} \mathfrak{D}_i^t \cdot \exp\{-y_i \cdot \beta_{k,t}\} = \sum_{k=1}^{C} \left(W_+^{k,t} \cdot \exp\{-\beta_{k,t}\} + W_-^{k,t} \cdot \exp\{\beta_{k,t}\}\right) \tag{C.20}$$

Setting $\beta_{k,t} = \frac{1}{2}\ln\left(\frac{W_+^{k,t}}{W_-^{k,t}}\right)$ as given in Line 6 of Algorithm B.3 minimizes Eq. (C.20). Plugging the value of $\beta_{k,t}$ back into Eq. (C.20) gives us the desired form of Z_t. □

§ C.2 Proofs from Chapter 7

Lemma 7.2. [Margins induced by ADABOOST [Bar+98]] Let θ be a desired margin parameter. Let ν_i be the margin of an instance $(x_i, y_i) \in \mathcal{S}$ induced by the combined classifier produced by ADABOOST. The number of instances such that $\nu_i < \theta$ drops exponentially fast with the no. of iterations T. Also, $\forall (x_i, y_i) \in \mathcal{S}$, $\nu_i \geqslant \gamma$, where γ is the bias of the weak learner.

Proof. The margin of an instance (x_i, y_i) is $\nu_i = {}^{y\cdot\tilde{H}(x)}/_{\sum_t \alpha_t}$. Now,

$$\nu_i \leqslant \theta \iff y_i \cdot \tilde{H}(x_i) \leqslant \theta \sum_t \alpha_t \implies y_i \cdot \sum_t \alpha_t h_t(x_i) \leqslant \theta \sum_t \alpha_t \tag{C.21}$$

$$\implies \nu_i \leqslant \theta \iff 0 \leqslant \theta \sum_t \alpha_t - y_i \cdot \sum_t \alpha_t h_t(x_i) \tag{C.22}$$

$$\implies \nu_i \leqslant \theta \iff 1 \leqslant \exp\left\{\theta \sum_t \alpha_t - y_i \cdot \sum_t \alpha_t h_t(x_i)\right\} \tag{C.23}$$

$$\implies \nu_i \leqslant \theta \iff \mathbb{1}_{[\nu_i \leqslant \theta]} \leqslant \exp\left\{\theta \sum_t \alpha_t - y_i \cdot \sum_t \alpha_t h_t(x_i)\right\}. \tag{C.24}$$

We now bound the probability of the number of instances in the training set s.t. their margins are less than θ, i.e., bound the term $\Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta]$.

$$\Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{[\nu_i \leqslant \theta]} \leqslant \frac{e^{\theta\sum_t \alpha_t}}{m} \sum_{i=1}^{m} \exp\left\{-y_i \cdot \sum_t \alpha_t h_t(x_i)\right\} \tag{C.25}$$

$$\implies \Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant e^{\theta\sum_t \alpha_t} \sum_{i=1}^{m} \mathcal{D}_m^1 \exp\left\{-y_i \cdot \sum_t \alpha_t h_t(x_i)\right\} \tag{C.26}$$

$$\implies \Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant e^{\theta\sum_t \alpha_t} \cdot \prod_{t=1}^{T} Z_t \tag{C.27}$$

Eq. (C.27) follows from Eq. (C.2). Now, substituting the value of α_t from Eq. (5.2) and

an upperbound for Z_t from Eq. (C.15), we have

$$\Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant e^{\theta\sum_t \alpha_t} \cdot ((1-2\gamma)(1+2\gamma))^{T/2} \tag{C.28}$$

$$\leqslant \sqrt{\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right)^{\theta} \dots \left(\frac{1-\varepsilon_T}{\varepsilon_T}\right)^{\theta}} \cdot (1-2\gamma)^{T/2} \cdot (1+2\gamma)^{T/2} \tag{C.29}$$

$$\implies \Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant \frac{1+2\gamma}{1-2\gamma}^{T\theta/2} \cdot (1-2\gamma)^{T/2} \cdot (1+2\gamma)^{T/2} \tag{C.30}$$

$$\implies \Pr_{(x_i,y_i)\in\mathcal{S}}[\nu_i \leqslant \theta] \leqslant \left(\sqrt{(1-2\gamma)^{1-\theta} \cdot (1+2\gamma)^{1+\theta}}\right)^{T}. \tag{C.31}$$

We note that the term inside the exponent in the R.H.S. of Eq. (C.31) is a constant. Hence, the probability that instances in the training set have an induced margin less than θ exponentially decreases with T.

From Eq. (C.31), we have $\theta < \frac{-\ln(1-4\gamma^2)}{\ln(1+2\gamma/1-2\gamma)}$. Since $\gamma \in [0, 1/2)$, this implies that $\gamma \leqslant \theta \leqslant 2\gamma$. Combining this with the fact that $\nu_i \geqslant \theta$ with probability 1 for a sufficiently large T gives us the desired bound. □

Chapter D

List of Figures

1.1 Learning with queries. 5
1.2 A visual representation of a statistical learning algorithm. 10

2.1 Different representations of the same Boolean function f. 20
2.2 Broad classifications of PAC learning algorithms. 29
2.3 Important 1-qubit gates 33
2.4 Important 2-qubit unitaries 33
2.5 Change of Basis in Measurement 34
2.6 Some common oracles used in quantum algorithms 35
2.7 The Quantum Query Model 35

4.1 A decision tree computing a Boolean function on 4 bits. 61

6.1 The QAGBOOST potential function 103
6.2 Schematic for agnostically learning polynomial-sized decision trees without MQ. . . . 113

7.1 Experiments on the `Breast Cancer` Dataset for different Q. 124
7.2 Experiments on the `Breast Cancer` Dataset for different $|\mathcal{S}|$. 125
7.3 Experiments on the `MNIST` Dataset for different Q. 126
7.4 Experiments on the `MNIST` Dataset for different $|\mathcal{S}|$. 127

9.1 Geometric interpretation of Dihedral groups. 162

Chapter E

List of Tables

4.1 Comparing the query complexities of constructing weak PAC learners for size-t decision trees in various settings without access to MQ oracle. 69

5.1 A comparison of various Boosting algorithms. 81

8.1 Comparing our work against existing OTB frameworks 136

9.1 Comparing different algorithms for learning size-t decision trees on n-bit Boolean functions. 166

Chapter F

List of definitions

2.1 Definition (Inner Product of Boolean functions) . . . 22
2.2 Definition (Training Error) . . . 24
2.3 Definition (Test Error) . . . 24
2.4 Definition (Correlation) . . . 25
2.5 Definition (Polynomial Evaluatibility) . . . 25
2.6 Definition ((ε, δ)-PAC learnability) . . . 26
2.7 Definition (VC dimension of a Hypothesis Space [VC71]) . . . 27
2.8 Definition (PAC learning with Oracle access) . . . 28
2.9 Definition (Trace Distance) . . . 32

3.1 Definition ((ε, δ)-PAC learning with access to EX oracle) . . . 44
3.2 Definition (β-optimal (ε, δ)-agnostic PAC learning) . . . 45
3.3 Definition (Quantum PAC learning with QEX (resp. QREX) oracle) . . . 46
3.4 Definition (Quantum agnostic PAC learning with QAEX oracle) . . . 46
3.5 Definition (ε-Biased oracles) . . . 49
3.6 Definition (Strongly Biased oracles) . . . 50
3.7 Definition (Exact Simulatibility of MQ) . . . 50
3.8 Definition (Approximate Simulatability) . . . 52

4.1 Definition (γ-Weak Learner) . . . 57
4.2 Definition (Strong Learner) . . . 57
4.3 Definition ((η, κ, δ)-Weak agnostic learner) . . . 57
4.4 Definition (Fourier Weight at degree k) . . . 58
4.5 Definition (Spectral concentration up to degree k) . . . 58
4.6 Definition (Bayes Optimal Predictor) . . . 65

5.1 Definition (Margin of instances induced by a combined classifier) . . . 76

6.1 Definition (Conservative weighting function) . . . 103
6.2 Definition (The QAGBOOST potential function) . . . 103
4.3 Definition ((η, κ, δ)-Weak agnostic learner) . . . 106
3.2 Definition (β-optimal (ε, δ)-agnostic PAC learning) . . . 106

2.4 Definition (Correlation) . . . 106

7.1 Definition (Margins for Confidence-Rated Predictions) . . . 119

8.1 Definition (β and ϕ coefficients) . . . 137
8.2 Definition (β and ϕ mixing) . . . 137
8.3 Definition (Geometric ϕ-mixing) . . . 137
8.4 Definition (Wasserstein distance of order one) . . . 137
8.5 Definition (Generalization Error of Offline Learners) . . . 140
8.6 Definition (Wasserstein Stable) . . . 142

9.1 Definition (DLP) . . . 158
9.2 Definition (The Hidden Subgroup Problem) . . . 159
9.3 Definition (HSP induced Concept class) . . . 160
9.4 Definition (Dihedral Groups) . . . 161
9.5 Definition (Decision Version of LWE) . . . 163
9.6 Definition (AC^0_d circuits and TC^0_d circuits) . . . 168

Bibliography

[AA24] Anurag Anshu and Srinivasan Arunachalam. "A survey on the complexity of learning quantum states". In: *Nature Reviews Physics* 6.1 (Jan. 2024), pp. 59–69. ISSN: 2522-5820. DOI: 10.1038/s42254-023-00662-4 (cit. on pp. 153, 154).

[Abe+19] Jacob Abernethy, Young Hun Jung, Chansoo Lee, Audra McMillan, and Ambuj Tewari. "Online learning via the differential privacy lens". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019 (cit. on pp. 131, 133, 148).

[ACN24] Baptiste Abeles, Eugenio Clerico, and Gergely Neu. *Generalization bounds for mixing processes via delayed online-to-PAC conversions*. 2024. arXiv: 2406.12600 [cs.LG]. URL: https://arxiv.org/abs/2406.12600 (cit. on pp. 11, 12, 135, 136).

[AD11] Alekh Agarwal and John C. Duchi. "The Generalization Ability of Online Algorithms for Dependent Data". In: *IEEE Transactions on Information Theory* 59 (2011), pp. 573–587. DOI: 10.1109/TIT.2012.2212414 (cit. on pp. 12, 130, 132, 133, 135, 136).

[AFK13] Pranjal Awasthi, Vitaly Feldman, and Varun Kanade. "Learning Using Local Membership Queries". In: *Proceedings of the 26th Annual Conference on Learning Theory*. Ed. by Shai Shalev-Shwartz and Ingo Steinwart. Vol. 30. Proceedings of Machine Learning Research. Princeton, NJ, USA: PMLR, June 2013, pp. 398–431. URL: https://proceedings.mlr.press/v30/Awasthi13.html (cit. on p. 49).

[AG98] Bruno Apolloni and Claudio Gentile. "Sample size lower bounds in PAC learning by algorithmic complexity theory". In: *Theoretical Computer Science* 209.1-2 (1998), pp. 141–162. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(97)00102-3 (cit. on p. 47).

[AGS21] Srinivasan Arunachalam, Alex Bredariol Grilo, and Aarthi Sundaram. "Quantum Hardness of Learning Shallow Classical Circuits". In: *SIAM Journal on Computing* 50.3 (2021), pp. 972–1013. DOI: 10.1137/20M1344202 (cit. on p. 168).

[Ajt96] M. Ajtai. "Generating hard instances of lattice problems (extended abstract)". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Philadelphia, Pennsylvania, USA). STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, 99–108. ISBN: 0897917855. DOI: 10.1145/237814.237838. URL: https://doi.org/10.1145/237814.237838 (cit. on p. 162).

[AK95] D. Angluin and M. Kharitonov. "When Won't Membership Queries Help?" In: *Journal of Computer and System Sciences* 50.2 (1995), pp. 336–355. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1995.1026 (cit. on pp. 158, 160).

[AL88] Dana Angluin and Philip Laird. "Learning From Noisy Examples". In: *Machine Learning* 2.4 (Apr. 1988), pp. 343–370. ISSN: 1573-0565. DOI: 10.1023/A:1022873112823 (cit. on pp. 29, 44).

[Alq24] Pierre Alquier. "User-friendly Introduction to PAC-Bayes Bounds". In: *Foundations and Trends® in Machine Learning* 17.2 (2024), pp. 174–303. ISSN: 1935-8237. DOI: 10.1561/2200000100. URL: http://dx.doi.org/10.1561/2200000100 (cit. on pp. 11, 130).

[AM20] Srinivasan Arunachalam and Reevu Maity. "Quantum boosting". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020 (cit. on pp. 9, 81, 87, 99, 100, 120).

[Amb08] Andris Ambainis. "Quantum search with variable times". In: *25th International Symposium on Theoretical Aspects of Computer Science*. Ed. by Susanne Albers and Pascal Weil. Vol. 1. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008, pp. 49–60. ISBN: 978-3-939897-06-4. DOI: 10.4230/LIPIcs.STACS.2008.1333. eprint: https://arxiv.org/abs/quant-ph/0609168 (cit. on p. 38).

[Ami+22] Saba Amiri, Adam Belloum, Eric Nalisnick, Sander Klous, and Leon Gommans. "On the impact of non-IID data on the performance and fairness of differentially private federated learning". In: *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. 2022, pp. 52–58. DOI: 10.1109/DSN-W54100.2022.00018 (cit. on pp. 11, 130).

[AS07] Alp Atıcı and Rocco A. Servedio. "Quantum Algorithms for Learning and Testing Juntas". In: *Quantum Information Processing* 6.5 (Oct. 2007), pp. 323–348. ISSN: 1573-1332. DOI: 10.1007/s11128-007-0061-6 (cit. on p. 168).

[AW17a] Srinivasan Arunachalam and Ronald de Wolf. "Guest Column: A Survey of Quantum Learning Theory". In: *SIGACT News* 48.2 (June 2017), 41–67. ISSN: 0163-5700. DOI: 10.1145/3106700.3106710 (cit. on pp. 46, 153).

[AW17b] Srinivasan Arunachalam and Ronald de Wolf. "Optimal quantum sample complexity of learning algorithms". In: *Proceedings of the 32nd Computational Complexity Conference* (Riga, Latvia). CCC '17. Dagstuhl, DEU: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. ISBN: 9783959770408 (cit. on pp. 47, 48, 155, 157).

[Ban+24] Akshay Bansal, Ian George, Soumik Ghosh, Jamie Sikora, and Alice Zheng. *Online learning of a panoply of quantum objects*. 2024. arXiv: 2406.04245 [quant-ph]. URL: https://arxiv.org/abs/2406.04245 (cit. on p. 157).

[Bar+98] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. "Boosting the margin: a new explanation for the effectiveness of voting methods". In: *The Annals of Statistics* 26.5 (1998), pp. 1651 –1686. DOI: 10.1214/aos/1024691352 (cit. on pp. 76, 118, 184).

[BE02] Olivier Bousquet and André Elisseeff. "Stability and generalization". In: *J. Mach. Learn. Res.* 2 (Mar. 2002), 499–526. ISSN: 1532-4435. DOI: 10.1162/153244302760200704 (cit. on pp. 11, 130, 133, 135).

[Bea+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. "Quantum lower bounds by polynomials". In: *J. ACM* 48.4 (July 2001), 778–797. ISSN: 0004-5411. DOI: 10.1145/502090.502097 (cit. on p. 35).

[Ben+05] Yoshua Bengio, Nicolas Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. "Convex Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Weiss, B. Schölkopf, and J. Platt. Vol. 18. MIT Press, 2005. URL: https://proceedings.neurips.cc/paper_files/paper/2005/file/0fc170ecbb8ff1afb2c6de48ea5343e7-Paper.pdf (cit. on p. 128).

[Ben23] Huck Bennett. "The Complexity of the Shortest Vector Problem". In: *SIGACT News* 54.1 (Mar. 2023), 37–61. ISSN: 0163-5700. DOI: 10.1145/3586165.3586172. URL: https://doi.org/10.1145/3586165.3586172 (cit. on p. 161).

[BF96] A. Blum and A. Frieze. "A polynomial-time algorithm for learning noisy linear threshold functions". In: *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. FOCS '96. USA: IEEE Computer Society, 1996, p. 330 (cit. on p. 165).

[BJ95] Nader H. Bshouty and Jeffrey C. Jackson. "Learning DNF over the uniform distribution using a quantum example oracle". In: *Proceedings of the Eighth Annual Conference on Computational Learning Theory* (Santa Cruz, California, USA). COLT '95. New York, NY, USA: Association for Computing Machinery, 1995, 118–127. ISBN: 0897917235. DOI: 10.1145/225298.225312 (cit. on pp. 3, 6, 8, 43, 45, 50, 63, 167–169).

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *J. ACM* 50.4 (July 2003), 506–519. ISSN: 0004-5411. DOI: 10.1145/792538.792543 (cit. on p. 163).

[BL92] Eric B Baum and Kenneth Lang. "Query learning can work poorly when a human oracle is used". In: *International joint conference on neural networks*. Vol. 8. Beijing China. 1992, p. 8 (cit. on p. 48).

[Bla+20] Guy Blanc, Neha Gupta, Jane Lange, and Li-Yang Tan. "Universal guarantees for decision tree induction via a higher-order splitting criterion". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 9475–9484 (cit. on p. 111).

[Bla+22] Guy Blanc, Jane Lange, Mingda Qiao, and Li-Yang Tan. "Properly Learning Decision Trees in almost Polynomial Time". In: *J. ACM* 69.6 (Nov. 2022). ISSN: 0004-5411. DOI: 10.1145/3561047. URL: https://doi.org/10.1145/3561047 (cit. on p. 111).

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, Feb. 2013. ISBN: 9780199535255. DOI: 10.1093/acprof:oso/9780199535255.001.0001. URL: http://dx.doi.org/10.1093/acprof:oso/9780199535255.001.0001 (cit. on p. 19).

[BLT20] Guy Blanc, Jane Lange, and Li-Yang Tan. "Provable guarantees for decision tree induction: the agnostic setting". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 941–949 (cit. on pp. 111, 166).

[Blu+89] Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. "Learnability and the Vapnik-Chervonenkis dimension". In: *J. ACM* 36.4 (Oct. 1989), 929–965. ISSN: 0004-5411. DOI: 10.1145/76359.76371 (cit. on p. 154).

[BM84] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudorandom Bits". In: *SIAM Journal on Computing* 13.4 (1984), pp. 850–864. DOI: 10.1137/0213053 (cit. on p. 158).

[Bra+02] Gilles Brassard, Peter Hø yer, Michele Mosca, and Alain Tapp. "Quantum amplitude amplification and estimation". In: *Quantum computation and information (Washington, DC, 2000)*. Vol. 305. Contemp. Math. Amer. Math. Soc., Providence, RI, 2002, pp. 53–74. ISBN: 0-8218-2140-7. DOI: 10.1090/conm/305/05215. URL: https://doi.org/10.1090/conm/305/05215 (cit. on p. 38).

[Bra+19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. "Worst-Case Hardness for LPN and Cryptographic Hashing via Code Smoothing". In: *Advances in Cryptology – EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III* (Darmstadt, Germany). Berlin, Heidelberg: Springer-Verlag, 2019, 619–635. ISBN: 978-3-030-17658-7. DOI: 10.1007/978-3-030-17659-4_21 (cit. on p. 164).

[Bre98] Leo Breiman. "Arcing classifier (with discussion and a rejoinder by the author)". In: *The Annals of Statistics* 26.3 (1998), pp. 801 –849. DOI: 10.1214/aos/1024691079 (cit. on p. 118).

[BS22] Debajyoti Bera and Tharrmashastha SAPV. "Few Quantum Algorithms on Amplitude Distribution". In: *arXiv preprint arXiv:2208.00162* (2022) (cit. on p. 65).

[Bsh+03] N. Bshouty, E. Mossel, R. O'Donnell, and R.A. Servedio. "Learning DNF from random walks". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* 2003, pp. 189–198. DOI: 10.1109/SFCS.2003.1238193 (cit. on pp. 49, 53).

[Bsh23] Nader H. Bshouty. *Superpolynomial Lower Bounds for Learning Monotone Classes*. 2023. arXiv: 2301.08486 [cs.DS] (cit. on p. 112).

[Bsh95] N.H. Bshouty. "Exact Learning Boolean Functions via the Monotone Theory". In: *Information and Computation* 123.1 (1995), pp. 146–153. ISSN: 0890-5401. DOI: https://doi.org/10.1006/inco.1995.1164. URL: https://www.sciencedirect.com/science/article/pii/S0890540185711649 (cit. on pp. 6, 44, 47).

[BV97] Ethan Bernstein and Umesh Vazirani. "Quantum Complexity Theory". In: *SIAM J. Comput.* 26.5 (Oct. 1997), 1411–1473. ISSN: 0097-5397. DOI: 10.1137/S0097539796300921 (cit. on pp. 9, 39).

[BWDSS20] Galit Bary-Weisberg, Amit Daniely, and Shai Shalev-Shwartz. "Distribution Free Learning with Local Queries". In: *Proceedings of the 31st International Conference on Algorithmic Learning Theory*. Ed. by Aryeh Kontorovich and Gergely Neu. Vol. 117. Proceedings of Machine Learning Research. PMLR, Feb. 2020, pp. 133–147. URL: https://proceedings.mlr.press/v117/bary-weisberg20a.html (cit. on p. 49).

[Car20] Matthias C. Caro. "Quantum learning Boolean linear functions w.r.t. product distributions". In: *Quantum Information Processing* 19.6 (Apr. 2020), p. 172. ISSN: 1573-1332. DOI: 10.1007/s11128-020-02661-1. URL: https://doi.org/10.1007/s11128-020-02661-1 (cit. on p. 155).

[Car+24] Matthias C. Caro, Marcel Hinsche, Marios Ioannou, Alexander Nietner, and Ryan Sweke. "Classical Verification of Quantum Learning". In: *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Ed. by Venkatesan Guruswami. Vol. 287. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 24:1–24:23. ISBN: 978-3-95977-309-6. DOI: 10.4230/LIPIcs.ITCS.2024.24. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2024.24 (cit. on pp. 155, 156, 164).

[CBCG04] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. "On the generalization ability of on-line learning algorithms". In: *IEEE Transactions on Information Theory* 50.9 (2004), pp. 2050–2057 (cit. on pp. 130–133, 140).

[Cha+23] Sagnik Chatterjee, Rohan Bhatia, Parmeet Singh Chani, and Debajyoti Bera. "Quantum boosting using domain-partitioning hypotheses". In: *Quantum Machine Intelligence* 5.2 (July 2023), p. 33. ISSN: 2524-4914. DOI: 10.1007/s42484-023-00122-3 (cit. on pp. 73, 81, 117).

[Chi+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. "Exponential algorithmic speedup by a quantum walk". In: *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing* (San Diego, CA, USA). STOC '03. New York, NY, USA: Association for Computing Machinery, 2003, 59–68. ISBN: 1581136749. DOI: 10.1145/780542.780552. URL: https://doi.org/10.1145/780542.780552 (cit. on p. 53).

[CM01] Kevin K. H. Cheung and Michele Mosca. "Decomposing finite Abelian groups". In: *Quantum Info. Comput.* 1.3 (Oct. 2001), 26–32. ISSN: 1533-7146 (cit. on p. 159).

[CMS25] Sagnik Chatterjee, Manuj Mukherjee, and Alhad Sethi. "Generalization Bounds for Dependent Data using Online-to-Batch Conversion". In: *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*. accepted. 2025 (cit. on pp. 12, 129, 131, 134, 148).

[CN89] Peter Clark and Tim Niblett. "The CN2 induction algorithm". In: *Machine Learning* 3.4 (Mar. 1989), pp. 261–283. ISSN: 1573-0565. DOI: 10.1007/BF00116835 (cit. on p. 90).

[CSB24] Sagnik Chatterjee, Tharrmashastha SAPV, and Debajyoti Bera. "Efficient Quantum Agnostic Improper Learning of Decision Trees". In: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*. Ed. by Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li. Vol. 238. Proceedings of Machine Learning Research. PMLR, May 2024, pp. 514–522. URL: https://proceedings.mlr.press/v238/chatterjee24a.html (cit. on pp. 55, 66, 99, 166).

[CSS15] Andrew W. Cross, Graeme Smith, and John A. Smolin. "Quantum learning robust against noise". In: *Phys. Rev. A* 92 (1 July 2015), p. 012327. DOI: 10.1103/PhysRevA.92.012327. URL: https://link.aps.org/doi/10.1103/PhysRevA.92.012327 (cit. on pp. 163, 165).

[CW00] R. Cleve and J. Watrous. "Fast parallel circuits for the quantum Fourier transform". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 526–536. DOI: 10.1109/SFCS.2000.892140 (cit. on p. 159).

[Dan15] Amit Daniely. "A PTAS for Agnostically Learning Halfspaces". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grünwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, July 2015, pp. 484–502. URL: https://proceedings.mlr.press/v40/Daniely15.html (cit. on p. 165).

[DB18] Vedran Dunjko and Hans J Briegel. "Machine learning & artificial intelligence in the quantum domain: a review of recent progress". In: *Reports on Progress in Physics* 81.7 (June 2018), p. 074001. DOI: 10.1088/1361-6633/aab406 (cit. on pp. 153, 154).

[DC95] Harris Drucker and Corinna Cortes. "Boosting decision trees". In: *Proceedings of the 8th International Conference on Neural Information Processing Systems* (Denver, Colorado). NIPS'95. Cambridge, MA, USA: MIT Press, 1995, 479–485. DOI: 10.5555/2998828.2998896 (cit. on p. 118).

[Den12] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142 (cit. on pp. 117, 120).

[Deu85] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London A* 400 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070 (cit. on p. 3).

[DF89] M.E Dyer and A.M Frieze. "The solution of some random NP-hard problems in polynomial expected time". In: *Journal of Algorithms* 10.4 (1989), pp. 451–489. ISSN: 0196-6774. DOI: https://doi.org/10.1016/0196-6774(89)90001-1. URL: https://www.sciencedirect.com/science/article/pii/0196677489900011 (cit. on p. 162).

[Dia+20] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. "Learning Halfspaces with Massart Noise Under Structured Distributions". In: *Proceedings of Thirty Third Conference on Learning Theory*. Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1486–1513. URL: https://proceedings.mlr.press/v125/diakonikolas20c.html (cit. on p. 165).

[Dia+21a] Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. "Efficiently learning halfspaces with Tsybakov noise". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy). STOC 2021. New York, NY, USA: Association for Computing Machinery, 2021, 88–101. ISBN: 9781450380539. DOI: 10.1145/3406325.3450998 (cit. on p. 165).

[Dia+21b] Ilias Diakonikolas, Daniel M. Kane, Thanasis Pittas, and Nikos Zarifis. "The Optimality of Polynomial Regression for Agnostic Learning under Gaussian Marginals in the SQ Model". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Ed. by Mikhail Belkin and Samory Kpotufe. Vol. 134. Proceedings of Machine Learning Research. PMLR, Aug. 2021, pp. 1552–1584. URL: https://proceedings.mlr.press/v134/diakonikolas21c.html (cit. on p. 165).

[DRG17] Ning Ding, Yanli Ren, and Dawu Gu. "PAC Learning Depth-3 AC^0 Circuits of Bounded Top Fanin". In: *Proceedings of the 28th International Conference on Algorithmic Learning Theory*. Ed. by Steve Hanneke and Lev Reyzin. Vol. 76. Proceedings of Machine Learning Research. PMLR, Oct.

2017, pp. 667–680. URL: https://proceedings.mlr.press/v76/ding17a.html (cit. on p. 169).

[DSS16] Amit Daniely and Shai Shalev-Shwartz. "Complexity Theoretic Limitations on Learning DNF's". In: *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. PMLR, June 2016, pp. 815–830. URL: https://proceedings.mlr.press/v49/daniely16.html (cit. on pp. 166, 167).

[Duc+12] John C. Duchi, Alekh Agarwal, Mikael Johansson, and Michael I. Jordan. "Ergodic Mirror Descent". In: *SIAM Journal on Optimization* 22.4 (2012), pp. 1549–1578. DOI: 10.1137/110836043 (cit. on pp. 12, 130, 132, 133).

[DW00] Carlos Domingo and Osamu Watanabe. "MadaBoost: A Modification of AdaBoost". In: *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*. COLT '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, 180–189. ISBN: 155860703X (cit. on pp. 75, 103, 178).

[EH00] Mark Ettinger and Peter Høyer. "On Quantum Algorithms for Noncommutative Hidden Subgroups". In: *Advances in Applied Mathematics* 25.3 (2000), pp. 239–251. ISSN: 0196-8858. DOI: https://doi.org/10.1006/aama.2000.0699 (cit. on p. 159).

[EH89] Andrzej Ehrenfeucht and David Haussler. "Learning decision trees from random examples". In: *Information and Computation* 82.3 (1989), pp. 231–246 (cit. on pp. vi, 7, 111, 166).

[EHK04] Mark Ettinger, Peter Høyer, and Emanuel Knill. "The quantum query complexity of the hidden subgroup problem is polynomial". In: *Information Processing Letters* 91.1 (2004), pp. 43–48. ISSN: 0020-0190. DOI: https://doi.org/10.1016/j.ipl.2004.01.024. URL: https://www.sciencedirect.com/science/article/pii/S0020019004000845 (cit. on p. 160).

[Fel+06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. "New Results for Learning Noisy Parities and Halfspaces". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 563–574. DOI: 10.1109/FOCS.2006.51 (cit. on pp. 163, 165, 167).

[Fel10] Vitaly Feldman. "Distribution-Specific Agnostic Boosting". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. Ed. by Andrew Chi-Chih Yao. Tsinghua University Press, 2010, pp. 241–250 (cit. on pp. 112, 166, 170).

[Fel12] Vitaly Feldman. "Learning DNF Expressions from Fourier Spectrum". In: *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*. Ed. by Shie Mannor, Nathan Srebro, and Robert C. Williamson. Vol. 23. JMLR Proceedings. JMLR.org, 2012, pp. 17.1–

17.19. URL: http://proceedings.mlr.press/v23/feldman12b/feldman12b.pdf (cit. on p. 167).

[Fey82] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179. URL: https://doi.org/10.1007/BF02650179 (cit. on p. 3).

[Fey85] Richard P. Feynman. "Quantum Mechanical Computers". In: *Optics News* 11.2 (Feb. 1985), pp. 11–20. DOI: 10.1364/ON.11.2.000011. URL: https://www.optica-opn.org/abstract.cfm?URI=on-11-2-11 (cit. on p. 3).

[FF12] Artur J. Ferreira and Mário A. T. Figueiredo. "Boosting Algorithms: A Review of Methods, Theory, and Applications". In: *Ensemble Machine Learning: Methods and Applications*. Ed. by Cha Zhang and Yunqian Ma. New York, NY: Springer New York, 2012, pp. 35–85. ISBN: 978-1-4419-9326-7. DOI: 10.1007/978-1-4419-9326-7_2 (cit. on pp. 15, 97).

[FHT00] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)". In: *The Annals of Statistics* 28.2 (2000), pp. 337–407. DOI: 10.1214/aos/1016218223 (cit. on pp. 97, 177).

[FS97] Yoav Freund and Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. ISSN: 0022-0000. DOI: 10.1006/jcss.1997.1504 (cit. on pp. 77, 81, 95, 120, 140, 176, 181).

[Fu+23] Shi Fu, Yunwen Lei, Qiong Cao, Xinmei Tian, and Dacheng Tao. "Sharper Bounds for Uniformly Stable Algorithms with Stationary Mixing Process". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=8E5Yazboyh (cit. on pp. 11, 12, 130, 133–135).

[Gas+24a] Michael Gastpar, Ido Nachum, Jonathan Shafer, and Thomas Weinberger. "Fantastic Generalization Measures are Nowhere to be Found". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=NkmJotfL42 (cit. on p. 149).

[Gas+24b] Michael Gastpar, Ido Nachum, Jonathan Shafer, and Thomas Weinberger. *Which Algorithms Have Tight Generalization Bounds?* 2024. arXiv: 2410.01969 [cs.LG]. URL: https://arxiv.org/abs/2410.01969 (cit. on p. 149).

[Gav03] Dmitry Gavinsky. "Optimally-smooth adaptive boosting and application to agnostic learning". In: 4.null (Dec. 2003), 101–117. ISSN: 1532-4435. DOI: 10.1162/153244304322765667 (cit. on pp. 75, 87).

[GAW19] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. "Optimizing quantum optimization algorithms via faster quantum gradient computation". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, California). SODA '19. USA: Society for Industrial and Applied Mathematics, 2019, 1425–1444 (cit. on p. 128).

[GD23] Casper Gyurik and Vedran Dunjko. "Exponential separations between classical and quantum learners". In: *arXiv preprint arXiv:2306.16028* (2023) (cit. on p. 154).

[GKK08a] Parikshit Gopalan, Adam Tauman Kalai, and Adam R Klivans. "Agnostically learning decision trees". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 527–536 (cit. on pp. 112, 166).

[GKK08b] Parikshit Gopalan, Adam Tauman Kalai, and Adam R. Klivans. "Agnostically learning decision trees". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* (Victoria, British Columbia, Canada). STOC '08. New York, NY, USA: Association for Computing Machinery, 2008, 527–536. ISBN: 9781605580470. DOI: 10.1145/1374376.1374451 (cit. on p. 170).

[GKZ19] Alex B. Grilo, Iordanis Kerenidis, and Timo Zijlstra. "Learning-with-errors problem is easy with quantum samples". In: *Phys. Rev. A* 99 (3 Mar. 2019), p. 032314. DOI: 10.1103/PhysRevA.99.032314. URL: https://link.aps.org/doi/10.1103/PhysRevA.99.032314 (cit. on pp. 163–165).

[GL89] O. Goldreich and L. A. Levin. "A hard-core predicate for all one-way functions". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing* (Seattle, Washington, USA). STOC '89. New York, NY, USA: Association for Computing Machinery, 1989, 25–32. ISBN: 0897913078. DOI: 10.1145/73007.73010 (cit. on p. 58).

[GR09] Venkatesan Guruswami and Prasad Raghavendra. "Hardness of learning halfspaces with noise". In: *SIAM Journal on Computing* 39.2 (2009), pp. 742–765 (cit. on p. 165).

[Gri+21] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. "Iterative quantum amplitude estimation". In: *npj Quantum Information* 7.1 (2021), pp. 1–6 (cit. on p. 121).

[Gro96] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219 (cit. on p. 3).

[Han16] Steve Hanneke. "The Optimal Sample Complexity of PAC Learning". In: *Journal of Machine Learning Research* 17.38 (2016), pp. 1–15. URL: http://jmlr.org/papers/v17/15-389.html (cit. on p. 154).

[Hås01] Johan Håstad. "Some optimal inapproximability results". In: *Journal of the ACM (JACM)* 48.4 (2001), pp. 798–859 (cit. on p. 167).

[Hau92] David Haussler. "Decision theoretic generalizations of the PAC model for neural net and other learning applications". In: *Information and Computation* 100.1 (1992), pp. 78–150. ISSN: 0890-5401. DOI: https://doi.org/10.1016/0890-5401(92)90010-D (cit. on p. 30).

[Hel+24] Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. *Generalization Bounds: Perspectives from Information Theory and PAC-Bayes*. 2024. arXiv: 2309.04381 [cs.LG]. URL: https://arxiv.org/abs/2309.04381 (cit. on pp. 11, 130).

[HLR23] Mikael Møller Høgsgaard, Kasper Green Larsen, and Martin Ritzert. "AdaBoost is not an optimal weak to strong learner". In: *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA). ICML'23. JMLR.org, 2023 (cit. on pp. 97, 112).

[HNS24] Shima Bab Hadiashar, Ashwin Nayak, and Pulkit Sinha. "Optimal Lower Bounds for Quantum Learning via Information Theory". In: *IEEE Transactions on Information Theory* 70.3 (2024), pp. 1876–1896. DOI: 10.1109/TIT.2023.3324527 (cit. on pp. 155, 157).

[HR76] Laurent Hyafil and Ronald L. Rivest. "Constructing optimal binary decision trees is NP-complete". In: *Information Processing Letters* 5.1 (1976), pp. 15–17. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(76)90095-8 (cit. on pp. 7, 62).

[Hua24] Hsin-Yuan Huang. "Learning in the Quantum Universe". PhD thesis. 2024. DOI: 10.7907/fgpv-3112. URL: https://resolver.caltech.edu/CaltechTHESIS:05032024-044352582 (cit. on p. 154).

[IRY05] Kazuo Iwama, Rudy Raymond, and Shigeru Yamashita. "General bounds for quantum biased oracles". In: *IPSJ Digital Courier* 1 (2005), pp. 415–425 (cit. on pp. 49, 64).

[IW23] Adam Izdebski and Ronald de Wolf. "Improved Quantum Boosting". In: *31st Annual European Symposium on Algorithms (ESA 2023)*. Ed. by Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman. Vol. 274. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 64:1–64:16. ISBN: 978-3-95977-295-2. DOI: 10.4230/LIPIcs.ESA.2023.64. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2023.64 (cit. on pp. 9, 73, 81, 96, 99, 100).

[Iye24] Venkataraman Natarajan Iyer. *A review on different techniques used to combat the non-IID and heterogeneous nature of data in FL*. 2024. arXiv: 2401.00809 [cs.LG] (cit. on pp. 11, 130).

[Jac97] Jeffrey C Jackson. "An Efficient Membership-Query Algorithm for Learning DNF with Respect to the Uniform Distribution". In: *Journal of Computer and System Sciences* 55.3 (1997), pp. 414–440. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1533. URL: https://www.sciencedirect.com/science/article/pii/S0022000097915336 (cit. on p. 167).

[Jor05] Stephen P. Jordan. "Fast Quantum Algorithm for Numerical Gradient Estimation". In: *Phys. Rev. Lett.* 95 (5 May 2005), p. 050501. DOI: 10.1103/PhysRevLett.95.050501. URL: https://link.aps.org/doi/10.1103/PhysRevLett.95.050501 (cit. on p. 128).

[JR23] Samuel Jaques and Arthur G. Rattew. *QRAM: A Survey and Critique*. 2023. arXiv: 2305.10310 [quant-ph]. URL: https://arxiv.org/abs/2305.10310 (cit. on pp. 36, 37).

[KK09] Varun Kanade and Adam Kalai. "Potential-Based Agnostic Boosting". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta. Vol. 22. Curran Associates, Inc., 2009. URL: https://proceedings.neurips.cc/paper_files/paper/2009/file/13f9896df61279c928f19721878fac41-Paper.pdf (cit. on pp. 99, 100, 103, 106, 112, 166, 170).

[Kle07] A. Klenke. *Probability Theory: A Comprehensive Course*. Universitext. Springer London, 2007. ISBN: 9781848000483. URL: https://books.google.co.in/books?id=tcm3y5UJxDsC (cit. on p. 147).

[KM17] Vitaly Kuznetsov and Mehryar Mohri. "Generalization bounds for non-stationary mixing processes". In: *Machine Learning* 106.1 (2017), pp. 93–117 (cit. on pp. 12, 130).

[KM91] Eyal Kushilevitz and Yishay Mansour. "Learning decision trees using the Fourier spectrum". In: *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing* (New Orleans, Louisiana, USA). STOC '91. New York, NY, USA: Association for Computing Machinery, 1991, 455–464. ISBN: 0897913973. DOI: 10.1145/103418.103466. URL: https://doi.org/10.1145/103418.103466 (cit. on pp. 7, 58, 62, 112, 166).

[KMV08] Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. "On agnostic boosting and parity learning". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* (Victoria, British Columbia, Canada). STOC '08. New York, NY, USA: Association for Computing Machinery, 2008, 629–638. ISBN: 9781605580470. DOI: 10.1145/1374376.1374466 (cit. on p. 45).

[KRS19] Varun Kanade, Andrea Rocchetto, and Simone Severini. "mu - biased quantum Fourier sampling". In: *Quantum Inf. Comput.* 19.15&16 (2019), pp. 1261–1278. DOI: 10.26421/QIC19.15-16-1. URL: https://arxiv.org/abs/1802.05690 (cit. on p. 167).

[KSS92] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. "Toward efficient agnostic learning". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Pittsburgh, Pennsylvania, USA). COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, 341–352. ISBN: 089791497X. DOI: 10.1145/130385.130424 (cit. on p. 30).

[KST23] Caleb Koch, Carmen Strassle, and Li-Yang Tan. "Superpolynomial lower bounds for decision tree learning and testing". In: *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2023, pp. 1962–1994 (cit. on pp. 7, 112).

[KT08] Sham M. Kakade and Ambuj Tewari. "On the Generalization Ability of Online Strongly Convex Programming Algorithms". In: *Neural Information Processing Systems*. 2008 (cit. on pp. 131, 133).

[Kup05] Greg Kuperberg. "A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem". In: *SIAM Journal on Computing* 35.1 (2005), pp. 170–188. DOI: 10.1137/S0097539703436345 (cit. on pp. 162, 163).

[KV94a] Michael Kearns and Leslie Valiant. "Cryptographic limitations on learning Boolean formulae and finite automata". In: *J. ACM* 41.1 (Jan. 1994), 67–95. ISSN: 0004-5411. DOI: 10.1145/174644.174647 (cit. on pp. 158, 160).

[KV94b] M.J. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press. MIT Press, 1994. ISBN: 9780262111935 (cit. on p. 15).

[LAT21] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. "A rigorous and robust quantum speed-up in supervised machine learning". In: *Nature Physics* 17.9 (Sept. 2021), pp. 1013–1017. ISSN: 1745-2481. DOI: 10.1038/s41567-021-01287-z (cit. on pp. 158–160).

[Li+24] Yiwei Li, Shuai Wang, Chong-Yung Chi, and Tony Q. S. Quek. "Differentially Private Federated Clustering Over Non-IID Data". In: *IEEE Internet of Things Journal* 11.4 (2024), pp. 6705–6721. DOI: 10.1109/JIOT.2023.3312852 (cit. on pp. 11, 130).

[LKS05] Aurelie C Lozano, Sanjeev Kulkarni, and Robert E Schapire. "Convergence and Consistency of Regularized Boosting Algorithms with Stationary B-Mixing Observations". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Weiss, B. Schölkopf, and J. Platt. Vol. 18. MIT Press, 2005 (cit. on pp. 12, 130).

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. "Constant depth circuits, Fourier transform, and learnability". In: *J. ACM* 40.3 (July 1993), 607–620. ISSN: 0004-5411. DOI: 10.1145/174130.174138 (cit. on pp. 21, 58, 166).

[LN22] Gábor Lugosi and Gergely Neu. "Generalization bounds via convex analysis". In: *Conference on Learning Theory*. PMLR. 2022, pp. 3524–3546 (cit. on p. 131).

[LN23] Gábor Lugosi and Gergely Neu. "Online-to-PAC Conversions: Generalization Bounds via Regret Analysis". In: *ArXiv* abs/2305.19674 (2023) (cit. on pp. vi, 12, 130–133, 136, 141).

[Lom04] Chris Lomont. *The Hidden Subgroup Problem - Review and Open Problems*. 2004. arXiv: quant-ph/0411037 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0411037 (cit. on p. 160).

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *J. ACM* 60.6 (Nov. 2013). ISSN: 0004-5411. DOI: 10.1145/2535925 (cit. on p. 169).

[LW94] N. Littlestone and M.K. Warmuth. "The Weighted Majority Algorithm". In: *Information and Computation* 108.2 (1994), pp. 212–261. ISSN: 0890-5401. DOI: https://doi.org/10.1006/inco.1994.1009. URL: https://www.sciencedirect.com/science/article/pii/S0890540184710091 (cit. on p. 140).

[Lyu05] Vadim Lyubashevsky. "The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem". In: *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer. 2005, pp. 378–389 (cit. on p. 163).

[Man94] Yishay Mansour. "Learning Boolean Functions via the Fourier Transform". In: *Theoretical Advances in Neural Computation and Learning*. Ed. by Vwani Roychowdhury, Kai-Yeung Siu, and Alon Orlitsky. Boston, MA: Springer US, 1994, pp. 391–424. ISBN: 978-1-4615-2696-4. DOI: 10.1007/978-1-4615-2696-4_11 (cit. on p. 15).

[Mas+99] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. "Boosting Algorithms as Gradient Descent". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf (cit. on p. 128).

[MBB98] Llew Mason, Peter Bartlett, and Jonathan Baxter. "Direct Optimization of Margins Improves Generalization in Combined Classifiers". In: *Advances in Neural Information Processing Systems*. Ed. by M. Kearns, S. Solla, and D. Cohn. Vol. 11. MIT Press, 1998. URL: https://proceedings.neurips.cc/paper_files/paper/1998/file/18ead4c77c3f40dabf9735432ac9d97a-Paper.pdf (cit. on p. 128).

[Mei00] Ron Meir. "Nonparametric time series prediction through adaptive model selection". In: *Machine learning* 39 (2000), pp. 5–34 (cit. on pp. 11, 12, 130, 132, 137).

[Mok88] Abdelkader Mokkadem. "Mixing properties of ARMA processes". In: *Stochastic Processes and their Applications* 29.2 (1988), pp. 309–315. ISSN: 0304-4149. DOI: https://doi.org/10.1016/0304-4149(88)90045-2 (cit. on p. 137).

[Mol20] C. Molnar. *Interpretable Machine Learning*. Leanpub, 2020. ISBN: 9780244768522. URL: https://books.google.co.in/books?id=jBm3DwAAQBAJ (cit. on p. 56).

[MOS04] Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. "Learning functions of k relevant variables". In: *Journal of Computer and System Sciences* 69.3 (2004). Special Issue on STOC 2003, pp. 421–434. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2004.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0022000004000418 (cit. on p. 167).

[MR02] Dinesh Mehta and Vijay Raghavan. "Decision tree approximations of Boolean functions". In: *Theoretical Computer Science* 270.1 (2002), pp. 609–623. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(01)00011-1 (cit. on pp. 111, 166).

[MR07] Mehryar Mohri and Afshin Rostamizadeh. "Stability Bounds for Non-i.i.d. Processes". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2007 (cit. on pp. 12, 130, 133, 135).

[MR08] Mehryar Mohri and Afshin Rostamizadeh. "Rademacher Complexity Bounds for Non-I.I.D. Processes". In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Vol. 21. Curran Associates, Inc., 2008. URL: https://proceedings.neurips.cc/paper_files/paper/2008/file/7eacb532570ff6858afd2723755ff790-Paper.pdf (cit. on pp. 11, 132).

[MR10] Mehryar Mohri and Afshin Rostamizadeh. "Stability Bounds for Stationary ϕ-mixing and β-mixing Processes". In: *J. Mach. Learn. Res.* 11 (Mar. 2010), 789–814. ISSN: 1532-4435 (cit. on pp. 11, 12, 130, 132–134).

[MRR06] Cristopher Moore, Daniel Rockmore, and Alexander Russell. "Generic quantum Fourier transforms". In: *ACM Trans. Algorithms* 2.4 (Oct. 2006), 707–723. ISSN: 1549-6325. DOI: 10.1145/1198513.1198525 (cit. on p. 160).

[MRT12] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2012. ISBN: 9780262018258 (cit. on p. 15).

[MT23] Preetham Mohan and Ambuj Tewari. *Quantum Learning Theory Beyond Batch Binary Classification*. 2023. arXiv: 2302.07409 [cs.LG]. URL: https://arxiv.org/abs/2302.07409 (cit. on pp. 156, 157).

[NC10] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. ISBN: 9781139495486 (cit. on p. 15).

[Nev+12] Hartmut Neven, Vasil S. Denchev, Geordie Rose, and William G. Macready. "QBoost: Large Scale Classifier Training withAdiabatic Quantum Optimization". In: *Proceedings of the Asian Conference on Machine Learning*. Ed. by Steven C. H. Hoi and Wray Buntine. Vol. 25. Proceedings of Machine Learning Research. Singapore Management University, Singapore: PMLR, Nov. 2012, pp. 333–348. URL: https://proceedings.mlr.press/v25/neven12.html (cit. on p. 81).

[O'D14] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. ISBN: 9781107038325 (cit. on p. 15).

[PCG23] Koustubh Phalak, Avimita Chatterjee, and Swaroop Ghosh. "Quantum Random Access Memory for Dummies". English (US). In: *Sensors* 23.17 (Sept. 2023). ISSN: 1424-8220. DOI: 10.3390/s23177462 (cit. on p. 37).

[Pie12] Krzysztof Pietrzak. "Cryptography from Learning Parity with Noise". In: *SOFSEM 2012: Theory and Practice of Computer Science*. Ed. by Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 99–114. ISBN: 978-3-642-27660-6 (cit. on p. 164).

[Pra14] Anupam Prakash. "Quantum algorithms for linear algebra and machine learning". PhD thesis. University of California, Berkeley, 2014 (cit. on p. 121).

[Qui96] J. R. Quinlan. "Bagging, boosting, and C4.S". In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1* (Portland, Oregon). AAAI'96. AAAI Press, 1996, 725–730. ISBN: 026251091X. DOI: 10.5555/1892875.1892983 (cit. on p. 118).

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (Sept. 2009). ISSN: 0004-5411. DOI: 10.1145/1568318.1568324. URL: https://doi.org/10.1145/1568318.1568324 (cit. on pp. 161, 162).

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. "Quantum Support Vector Machine for Big Data Classification". In: *Phys. Rev. Lett.* 113 (13 Sept. 2014), p. 130503. DOI: 10.1103/PhysRevLett.113.130503 (cit. on p. 121).

[RZ16] Daniel Russo and James Zou. "Controlling Bias in Adaptive Data Analysis Using Information Theory". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, May 2016, pp. 1232–1240 (cit. on pp. 11, 130).

[Sch90] Robert E Schapire. "The strength of weak learnability". In: *Machine learning* 5 (1990), pp. 197–227. DOI: 10.1007/BF00116037 (cit. on p. 74).

[Ser03] Rocco A. Servedio. "Smooth boosting and learning with malicious noise". In: *J. Mach. Learn. Res.* 4.null (Dec. 2003), 633–648. ISSN: 1532-4435. DOI: 10.1162/153244304773936072 (cit. on pp. 75, 81, 120, 178).

[SF12] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012. ISBN: 0262017180 (cit. on pp. 15, 97).

[SG01] R.A. Servedio and S.J. Gortler. "Quantum versus classical learnability". In: *Proceedings 16th Annual IEEE Conference on Computational Complexity*. 2001, pp. 138–148. DOI: 10.1109/CCC.2001.933881 (cit. on pp. 154, 158–160).

[Sho94] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700 (cit. on pp. 3, 158).

[Slo24] Joseph Slote. "Parity vs. AC0 with Simple Quantum Preprocessing". In: *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Ed. by Venkatesan Guruswami. Vol. 287. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 92:1–92:21. ISBN: 978-3-95977-309-6. DOI: 10.4230/LIPIcs.ITCS.2024.92 (cit. on p. 170).

[SP18] Maria Schuld and Francesco Petruccione. "Quantum ensembles of quantum classifiers". In: *Scientific Reports* 8.1 (Feb. 2018), p. 2772. DOI: 10.1038/s41598-018-20403-3 (cit. on p. 81).

[SS99] Robert E. Schapire and Yoram Singer. "Improved Boosting Algorithms Using Confidence-rated Predictions". In: *Machine Learning* 37.3 (Dec. 1999), pp. 297–336. ISSN: 1573-0565. DOI: 10.1023/A:1007614523901. URL: https://doi.org/10.1023/A:1007614523901 (cit. on pp. 79–81, 120, 177, 183).

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. ISBN: 9781107298019. DOI: 10.1017/CBO9781107298019 (cit. on pp. 15, 26).

[SSG24] Wilfred Salmon, Sergii Strelchuk, and Tom Gur. "Provable Advantage in Quantum PAC Learning". In: *Proceedings of Thirty Seventh Conference on Learning Theory*. Ed. by Shipra Agrawal and Aaron Roth. Vol. 247. Proceedings of Machine Learning Research. PMLR, 2024, pp. 4487–4510. URL: https://proceedings.mlr.press/v247/salmon24a.html (cit. on pp. 157, 158).

[ST04] Daniel A. Spielman and Shang-Hua Teng. "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time". In: *J. ACM* 51.3 (May 2004), 385–463. ISSN: 0004-5411. DOI: 10.1145/990308.990310. URL: https://doi.org/10.1145/990308.990310 (cit. on p. 167).

[Tal94] M. Talagrand. "Sharper Bounds for Gaussian and Empirical Processes". In: *The Annals of Probability* 22.1 (1994), pp. 28–76. ISSN: 00911798, 2168894X. URL: http://www.jstor.org/stable/2244494 (visited on 07/15/2024) (cit. on p. 154).

[Tie23] Stefan Tiegel. "Hardness of Agnostically Learning Halfspaces from Worst-Case Lattice Problems". In: *Proceedings of Thirty Sixth Conference on Learning Theory*. Ed. by Gergely Neu and Lorenzo Rosasco. Vol. 195. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 3029–3064. URL: https://proceedings.mlr.press/v195/tiegel23a.html (cit. on p. 165).

[Val15] Gregory Valiant. "Finding Correlations in Subquadratic Time, with Applications to Learning Parities and the Closest Pair Problem". In: *J. ACM* 62.2 (May 2015). ISSN: 0004-5411. DOI: 10.1145/2728167. URL: https://doi.org/10.1145/2728167 (cit. on p. 168).

[Val84] L. G. Valiant. "A theory of the learnable". In: *Commun. ACM* 27.11 (Nov. 1984), 1134–1142. ISSN: 0001-0782. DOI: 10.1145/1968.1972. URL: https://doi.org/10.1145/1968.1972 (cit. on pp. vi, 3, 5, 26, 44).

[Vap82] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer New York, 1982. ISBN: 10: 0387907335 (cit. on pp. vi, 3).

[VC71] V. N. Vapnik and A. Ya. Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability & Its Applications* 16.2 (1971), pp. 264–280. DOI: 10.1137/1116025 (cit. on pp. vi, 3, 27).

[VC74] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979). Moscow: Nauka, 1974 (cit. on pp. vi, 3, 154).

[Vid13] Mathukumalli Vidyasagar. *Learning and generalisation: with applications to neural networks*. Springer Science & Business Media, 2013 (cit. on pp. 11, 130).

[Vil08] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008 (cit. on p. 138).

[Vov98] V Vovk. "A Game of Prediction with Expert Advice". In: *Journal of Computer and System Sciences* 56.2 (1998), pp. 153–173. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1997.1556. URL: https://www.sciencedirect.com/science/article/pii/S0022000097915567 (cit. on p. 140).

[VV05] Alexander Vezhnevets and Vladimir Vezhnevets. "Modest AdaBoost-teaching AdaBoost to generalize better". In: *Graphicon*. Vol. 12. 5. 2005, pp. 987–997 (cit. on p. 97).

[Wan10] Frédéric Wang. *The Hidden Subgroup Problem*. 2010. arXiv: 1008.0010 [quant-ph]. URL: https://arxiv.org/abs/1008.0010 (cit. on p. 160).

[Wan+20] XiMing Wang, YueChi Ma, Min-Hsiu Hsieh, and Man-Hong Yung. "Quantum speedup in adaptive boosting of binary classification". In: *Science China Physics, Mechanics & Astronomy* 64.2 (Dec. 2020), p. 220311. ISSN: 1869-1927. DOI: 10.1007/s11433-020-1638-5. URL: https://doi.org/10.1007/s11433-020-1638-5 (cit. on p. 81).

[Wat+19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. "Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2019. New York, NY, USA: Association for Computing Machinery, 2019, 515–526. ISBN: 9781450367059. DOI: 10.1145/3313276.3316404 (cit. on p. 169).

[WN14] Shuqiong Wu and Hiroshi Nagahashi. "Parameterized adaboost: introducing a parameter to speed up the training of real adaboost". In: *IEEE Signal Processing Letters* 21.6 (2014), pp. 687–691 (cit. on p. 97).

[WN15] Shuqiong Wu and Hiroshi Nagahashi. "Penalized AdaBoost: improving the generalization error of gentle AdaBoost through a margin distribution". In: *IEICE TRANSACTIONS on Information and Systems* 98.11 (2015), pp. 1906–1915 (cit. on p. 97).

[WO02] M.J. Weinberger and E. Ordentlich. "On delayed prediction of individual sequences". In: *IEEE Transactions on Information Theory* 48.7 (2002), pp. 1959–1976. DOI: 10.1109/TIT.2002.1013136 (cit. on p. 135).

[Wol23] Ronald de Wolf. *Quantum Computing: Lecture Notes*. 2023. arXiv: 1907.09415 (cit. on p. 15).

[WSM95] Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian. *UCI Machine Learning Repository*. 1995. URL: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic) (cit. on pp. 117, 120).

[Xio+22] Zuobin Xiong, Zhipeng Cai, Daniel Takabi, and Wei Li. "Privacy Threat and Defense for Federated Learning With Non-i.i.d. Data in AIoT". In: *IEEE Transactions on Industrial Informatics* 18.2 (2022), pp. 1310–1321. DOI: 10.1109/TII.2021.3073925 (cit. on pp. 11, 130).

[XR17] Aolin Xu and Maxim Raginsky. "Information-theoretic analysis of generalization capability of learning algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017 (cit. on pp. 11, 130).

[Yu94] Bin Yu. "Rates of convergence for empirical processes of stationary mixing sequences". In: *The Annals of Probability* (1994), pp. 94–116 (cit. on pp. 11, 12, 130, 132, 137).

[YZ21] Yu Yu and Jiang Zhang. "Smoothing Out Binary Linear Codes and Worst-Case Sub-exponential Hardness for LPN". In: *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2021, 473–501. ISBN: 978-3-030-84251-2. DOI: 10.1007/978-3-030-84252-9_16 (cit. on p. 164).

[Zha+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Sy8gdB9xx (cit. on pp. 11, 130).

[Zha+19] Rui (Ray) Zhang, Xingwu Liu, Yuyi Wang, and Liwei Wang. "McDiarmid-Type Inequalities for Graph-Dependent Variables and Stability Bounds". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019 (cit. on pp. 11, 12, 130, 133).