Title
Playing Games to Reduce Supervision in Learning

Permalink
https://escholarship.org/uc/item/1gs1d457

Author
Balsubramani, Akshay

Publication Date
2016-01-01

Peer reviewed|Thesis/dissertation
Playing Games to Reduce Supervision in Learning

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Akshay Balsubramani

Committee in charge:

Professor Yoav Freund, Chair
Professor Sanjoy Dasgupta
Professor Patrick J. Fitzsimmons
Professor Alon Orlitsky
Professor Lawrence K. Saul

2016
The Dissertation of Akshay Balsubramani is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016
EPIGRAPH

The most important questions of life...are indeed, for the most part, really only problems of probability.

Pierre-Simon Laplace

Knowledge I possess of the game of dice, thus is my skill in numbers.

Mahabharata (Rituparna to Nala, Vana Parva)

Confer with the ignorant man as with the learned. For knowledge has no limits, and none has yet achieved perfection in it.

Ptahhotep, Maxim 1

Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under observation in life.

Marcus Aurelius, “Meditations” (Book III)

From discrimination between this and that a host of demons blazes forth!

Huàngbò Xìyùn

To see what is in front of one’s nose needs a constant struggle.

George Orwell, “In Front of Your Nose”

Picture all experts as if they were mammals.

Christopher Hitchens, “Letters to a Young Contrarian”

If life is going to exist in a Universe of this size, then the one thing it cannot afford to have is a sense of proportion.

Douglas Adams, “The Restaurant at the End of the Universe”

I’m not much but I’m all I have.

Philip K. Dick, “Martian Time-Slip”

Since that’s the way we’re playing it...let’s play it that way...

Samuel Beckett, “Endgame”
# TABLE OF CONTENTS

Signature Page .................................................................................. iii

Epigraph ......................................................................................... iv

Table of Contents ............................................................................. v

List of Figures .................................................................................. xi

List of Tables .................................................................................... xiii

List of Algorithms ........................................................................... xiv

Acknowledgements .......................................................................... xv

Vita ................................................................................................. xviii

Abstract of the Dissertation ............................................................ xix

Chapter 1 Introduction ................................................................... 1
  1.1 Outline of Part I: “Muffled” Semi-Supervised Classifier Aggregation .. 1
  1.2 Outline of Part II: Martingales, Stopping Times, and Statistical Testing 5
  1.3 Notation and Preliminaries ............................................................ 7

Part I Games over Unlabeled Data for Semi-Supervised Classification .... 8

Chapter 2 Prologue: A Formulation for Classifier Aggregation ........... 9
  2.1 A Motivating Scenario ................................................................. 9
  2.2 Some Illustrative Examples ....................................................... 11
  2.3 Playing Against An Adversary ................................................... 14
  2.4 Semi-Supervised Classifier Aggregation ..................................... 16
  2.5 Advantages of the New Formulation ......................................... 17

Chapter 3 Combining Binary Classifiers to Minimize Classification Error .. 18
  3.1 Introduction .............................................................................. 18
  3.2 Mathematical Preliminaries ...................................................... 20
  3.3 The Transductive Binary Classification Game ............................. 21
  3.4 Bounding the Correlation Vector Using Labeled Data ............... 24
  3.5 Interpretation and Discussion .................................................... 25
    3.5.1 Subgradient Conditions ....................................................... 27
    3.5.2 Approximate Learning ....................................................... 28
    3.5.3 Characteristics of the Solution .......................................... 28
    3.5.4 Independent Label Noise .................................................. 30
  3.6 Computational Considerations .................................................. 31
### Chapter 4 Optimal Binary Classifier Aggregation for General Losses

#### 4.1 Setup

- 4.1.1 Loss Functions
- 4.1.2 Minimax Formulation

#### 4.2 Results for Binary Classification

- 4.2.1 Solution of the Game
- 4.2.2 The Ensemble Aggregation Algorithm
- 4.2.3 Examples of Different Losses
- 4.2.4 Technical Discussion

#### 4.3 Related Work and Extensions

- 4.3.1 Weighted Test Sets, Covariate Shift, and Label Noise
- 4.3.2 Uniform Convergence Bounds for $b$

#### 4.4 Constraints on General Losses for Binary Classification

- 4.4.1 Matching Objective and Constraint Losses
- 4.4.2 Beating the Best Classifier and the Best Weighted Majority

#### 4.5 Supporting Results and Proofs

- 4.5.1 Proof of Theorem 4
- 4.5.2 Other Proofs

### Chapter 5 Improving Random Forests with Semi-Supervised Specialists

#### 5.1 Introduction

#### 5.2 Learning with Specialists

- 5.2.1 Creating Specialists for an Algorithm
- 5.2.2 Discussion

#### 5.3 Experimental Evaluation

#### 5.4 Related Work

#### 5.5 Additional Information on Experiments

- 5.5.1 Datasets
- 5.5.2 Algorithms

### Chapter 6 Muffled Incremental Combination of Classifiers

#### 6.1 Introduction

#### 6.2 An Algorithm for Incrementally Aggregating Classifiers

#### 6.3 Maximizing the Performance of an Ensemble of Trees

- 6.3.1 Line Search
- 6.3.2 Wilson’s Score Interval for Estimating $b$
- 6.3.3 Discussion

#### 6.4 Empirical Results

#### 6.5 Discussion

#### 6.6 Related and Future Work

#### 6.7 Derivation of the MARVIN Update Rule
6.8 Generalization and Estimating $b$ ........................................... 90
  6.8.1 Wilson’s Interval ................................................... 90
  6.8.2 Other Details ..................................................... 92
  6.9 Implementation Details ............................................... 92

Chapter 7 Optimal Classifier Aggregation for Decomposable Multiclass Losses 95
  7.1 Preliminaries .......................................................... 95
  7.2 Decomposable Losses ................................................ 97
  7.3 Main Result ........................................................... 98
    7.3.1 Example: Optimality of the Softmax Artificial Neuron .......... 100
  7.4 Proof of Theorem 15 ................................................ 101
  7.5 Proof of Theorem 16 ................................................ 106

Chapter 8 Learning to Abstain from Binary Prediction ......................... 108
  8.1 Introduction .......................................................... 108
  8.2 Abstaining with a Fixed Cost ....................................... 112
  8.3 Predicting with Constrained Abstain Rate .......................... 115
    8.3.1 Solving the Game .............................................. 116
    8.3.2 The Pareto Frontier and Discussion .......................... 118
  8.4 Discussion and Related Work ....................................... 119
  8.5 Abstaining with General Losses .................................... 121

Chapter 9 MARVIN as a Function Approximator .............................. 125
  9.1 Algorithm and Setup ............................................... 126
    9.1.1 Definitions .................................................... 127
    9.1.2 Greedy Coordinate Descent as Residual Fitting .............. 129
  9.2 Convergence of the Algorithm ..................................... 130
    9.2.1 Convergence Rate ............................................ 131
    9.2.2 Discussion .................................................... 132
  9.3 Hard Core Sets ...................................................... 134
    9.3.1 Discussion .................................................... 135
    9.3.2 Proof of Theorem 20 .......................................... 136
  9.4 Proof of Theorem 19 ................................................ 137

Chapter 10 Perspectives, Extensions, and Open Problems .................... 143

Part II Sequential Processes and Stopping Times ............................ 146

Chapter 11 Prologue: A Game with Exponential Martingales ................. 147
  11.1 Martingales Describe Fair Games ................................ 147
  11.2 A Family of Exponential Martingales ............................. 149
  11.3 Hoeffding’s Large Deviation Bound ............................... 150
  11.4 From Stopping Times to Uniform Concentration ................... 152
14.6.2 A General Independence Test ........................................ 208

Chapter 15 Preventing $P$-Value Manipulation by Stopping .......... 211
  15.1 Approach: Robustness Against Any Sampling Plan .............. 213
  15.2 Algorithms ....................................................... 216
    15.2.1 Standard $z$-Test ........................................... 217
    15.2.2 The Introductory Example: Successive Conditional Estimation 219
    15.2.3 A Robust $p$-Value from General Test Statistics ............ 220
    15.2.4 Validity and Optimality .................................... 222
  15.3 Discussion ...................................................... 225
    15.3.1 Examples of Martingale Test Statistics .............. 225
    15.3.2 Related Work .............................................. 227
    15.3.3 Proof Discussion ........................................... 228
  15.4 Proof of Tight Finite-Time LIL Concentration (Theorem 45) .... 229
    15.4.1 Proof Overview ............................................ 229
    15.4.2 Proof of Theorem 48 – Details ............................ 232

Appendix A Minimax Theorem .............................................. 236

Appendix B Proofs from Chapter 8 ........................................ 237

Appendix C Proofs from Chapter 12 ....................................... 244
  C.1 Proof of Theorem 19 (Anti-Concentration Bound) .............. 244
    C.1.1 Preliminaries and Proof Overview .................................. 245
    C.1.2 Full Proof of Theorem 19 ..................................... 246
    C.1.3 Supporting Proofs ............................................... 249
    C.1.4 Ancillary Results and Proofs .................................. 253
  C.2 Miscellaneous Results: Concentration Bounds and Extensions ... 257
    C.2.1 Proof of Lemma 28 ............................................ 257
    C.2.2 Hoeffding and Bernstein Concentration Bounds ............... 258
    C.2.3 Initial Time Conditions ....................................... 262

Appendix D Deferred Material from Chapter 14 ......................... 264
  D.1 Proof of Theorem 40 ............................................. 264
  D.2 Proof of Proposition 23 ........................................... 265
  D.3 Proof of Theorem 41 ............................................. 265
  D.4 Proportionality Constants and Guaranteed Correctness ........... 267
  D.5 Type II Error Approximation ..................................... 269
  D.6 Proof of Lemma 42 .............................................. 270
  D.7 Proof of Theorem 44 ............................................. 273
  D.8 Experimental Protocol ............................................. 274
  D.9 Supplemental Graphs ............................................... 275
LIST OF FIGURES

Figure 3.1. The optimal strategy for the predictor player on an example $x$ plotted with the score-dependent term of the slack function, as a function of the score $x^\top \sigma^*$. ................................. 23

Figure 3.2. A schematic illustration of the optimal $\sigma^* \geq 0^p$. The vector $n_b$ is the difference between the sums of two categories of clipped examples: those with high ensemble prediction ($x^\top \sigma^* > 1$) and low prediction ($< -1$). The effect of $B(\sigma^*)$ is neglected for simplicity. 27

Figure 4.1. At left are plots of potential wells. At right are optimal prediction functions $g$, as a function of ensemble score. Both are shown for various losses, as listed in Section 4.2.3. ....................... 40

Figure 5.1. A schematic of how the forest structure is related to the unlabeled data matrix $S$, with a given example $x$ highlighted. The two colors in the matrix represent $\pm 1$ predictions, and white cells abstentions. 68

Figure 5.2. Class-conditional “awake ensemble prediction” ($x^\top \sigma^*$) distributions, on SUSY. Rows (top to bottom): $\{1K, 10K, 100K\}$ labels. Columns (left to right): $\alpha = \{1.0, 0.3, 3.0\}$, and the base RF. .... 70

Figure 6.1. Potential well as a function of score (reproduced from Fig. 3.1)... 77

Figure 6.2. Effect of Wilson score interval (Sec. 6.3) for measuring $b$, on a decision tree with 9 internal nodes. ...................... 83

Figure 6.3. Score distributions of four different classifiers in the columns from left to right – respectively logistic regression, random forests, HEDGE MOWER-1, and HEDGE MOWER. ......................... 85

Figure 8.1. The abstain/error Pareto frontier in our setting: Minimum achievable error rate $V(\alpha)$ for an aggregated classifier with allowed abstain rate $\leq \alpha$. Formalized in Theorem 11. .................... 111

Figure 8.2. Left: $\Psi(m,c)$ for various $c$. Right: optimal predictions $g_j^*$ and abstain probabilities $1 - p_j^*$. ........................................ 114

Figure 14.1. Batch (left) and sequential (right) tests................................. 188

Figure 14.2. $Pr_{H_0}(\tau \leq n)$ for different $\alpha$, on biased coin. Dotted lines of corresponding colors are the target levels $\alpha$. ...................... 202
Figure 14.3. Power vs. $\ln(N)$ for different $\delta$, on Gaussians. Dashed lines represent power of batch test with $N$ samples. ............................... 203

Figure 14.4. Distribution of $\log_{1.25}(\tau)$ for $\delta \in \{0.5(1.25)^c : c \in \{7, 6, \ldots, 0\}\}$, so that the abscissa values $\{\log_{1.25}(\frac{1}{\delta})\}$ are a unit length apart. Dashed line has slope 2. ......................................................... 204

Figure 15.1. Plot of 1,000 trials of the reported values evolving over time up to $N = 100$, when running a $z$-test on the mean of Gaussian data from $\mathcal{N}(\mu, 1)$, under $H_0 : \mu = 0$. ....................................................... 215

Figure 15.2. Comparing $p$-value peeking with the method of this paper, using the $z$-test of Fig. 15.1. ................................................................. 219

Figure D.1. $\Pr_{H_0}(\tau \leq n)$ for different $\alpha$, on biased coin, for $C = 2$ (left) and $C = 2.2$ (right). ................................................................. 276

Figure D.2. Power vs. $\ln(N)$ for different $\delta$, on Gaussians. Dashed lines represent power of batch test with $N$ samples. ......................... 276

Figure D.3. Distribution of $\ln(\tau)$ for $\delta \in \{e^{-1} : c \in \{1, 2, \ldots, 5\}\}$, so that the abscissa values $\{\ln(\frac{1}{\delta})\}$ are a unit length apart. Dashed line has slope 2. ......................................................... 277
LIST OF TABLES

Table 2.1. Basic example of Alice’s classmates’ exams; the unweighted majority vote makes no mistakes here. .......................... 11

Table 2.2. Example with two classifier blocs. .......................... 13

Table 4.1. Some binary classification losses, as in Sec. 4.2.3. For convenience, we write \( \text{clip}(x) = \min(1, \max(-1, x)) \). .......................... 45

Table 5.1. Area under ROC curve for HEDGECLIPPER vs. supervised ensemble algorithms. .......................... 71

Table 5.2. Datasets used for HEDGECLIPPER experiments. ................. 73

Table 6.1. Area under ROC curve for MARVIN and HEDGEMOWER variants, and supervised ensemble algorithms. .......................... 87

Table 6.2. Information about the datasets used in Chapter 6. ................. 94
### LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm 1.</th>
<th>HEDGECLIPPER</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 2.</td>
<td>MARVIN</td>
<td>80</td>
</tr>
<tr>
<td>Algorithm 3.</td>
<td>HEDGEMOWER</td>
<td>93</td>
</tr>
<tr>
<td>Algorithm 4.</td>
<td>Water-filling algorithm for Theorem 16</td>
<td>99</td>
</tr>
<tr>
<td>Algorithm 5.</td>
<td>Fixed Step Size Version of MARVIN</td>
<td>126</td>
</tr>
<tr>
<td>Algorithm 6.</td>
<td>Robust (p)-Values for a Standard (z)-Test</td>
<td>218</td>
</tr>
<tr>
<td>Algorithm 7.</td>
<td>Robust (p)-Values from Conditionally Estimating a Test Statistic</td>
<td>220</td>
</tr>
<tr>
<td>Algorithm 8.</td>
<td>Robust (p)-Values from General Test Statistics</td>
<td>221</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

In a personally pleasing allegory with the formulation for aggregation that com-
prises a large part of my PhD work, the work owes its existence and contents to many
people. Their contributions can scarcely be disentangled from each other, and any quality
in the end product can be traced back to them.

My advisor, Yoav Freund, believed I could tackle many of the problems in this
thesis years before I ever believed the same, and I would like to thank him profusely
for taking these chances on me, and showing me how to channel curiosity and play the
creative game. I hope I have absorbed some small amount of his uncanny freshness and
clarity of thought, and ability to see the forest despite the trees.

I much appreciate the support of my committee throughout this process. In par-
ticular, Sanjoy Dasgupta has patiently listened to many harebrained ideas, and provided
unerring guidance both in research and beyond it. Patrick Fitzsimmons also took a chance
on a cold email from an unknown student, which led in unforeseen directions; I am very
grateful.

I have been lucky enough to work with many international-class researchers
throughout my growth so far. Rani Nelken, Ben Rubinstein, Lihong Li, and Omid
Madani exposed me to a wealth of problems; conversations with them and others, like
Jake Abernethy, John Langford (and the rest of MSR NY), Shachar Lovett, and Vineet
Bafna have been very helpful, timely, and affirming. More recently I enjoyed the privilege
of working with Robert Schapire; like many of the others above, his wise example has
taught me about the value of simplicity.

Friends met through my research community have anchored the journey for me in
many ways. On my floor were Chris, Zack, Marco, Stefanos, Chicheng, Michael et al. to
provide sounding boards for all sorts of detritus and ups and downs, and the always oddly
requested translation services. Alumni/postdocs who have given me belief to endure
when I thought myself incapable include Matus, Emanuele, Nakul, Aditya, Murali, and the ex-4142 CSE folks. There are also many other influential people to mention here, like coauthors Aaditya, Masrour, and Zohar, fellow interns, and others (Akshay, Adith, et al.). Their perspectives and friendships (and couches, in some cases) will continue to be much valued.

Outside the work community, I learned so much, occasionally work-related but significantly not. My memories here started with everyone from my earlier years in SD, including Akin and the Master’s crew (incl. Srdjan, Sohini, Rohan, Matt, Anukool, et al.) with whom I enjoyed delightful conversations and trips. Other friends who have immeasurably enriched my experience know who they are, like Jack, Marcela, Joe, Irina, Eric, and more; the improv friends I’ve enjoyed playing with; and the bioinformatics crew (Anu, Viraj, Shamim, et al., also unreasonably helpful more recently when work took a twist).

Almost all the work in this manuscript was done during my 3833 Dove period, and it deserves special mention here. I spent three memorable years there; Dimo and Panos, we’ll reminisce in years to come. I’m really appreciative of each of the past/present/to-be affiliates of the house for many memories made – keep in touch.

Amma, Appa, and Simi continue to set examples, from which I might finally start learning.

Chapter 3 contains material from the Conference on Learning Theory, 2015 (“Optimally Combining Classifiers Using Unlabeled Data,” Balsubramani and Freund). The dissertation author was the primary investigator and author of this paper.

Chapter 4 contains material to appear at the conference on Advances in Neural Information Processing Systems, 2016 (“Optimal Binary Classifier Aggregation for General Losses,” Balsubramani and Freund). The dissertation author was the primary
investigator and author of this paper.

Chapter 5 contains material from the conference on Advances in Neural Information Processing Systems, 2015 (“Scalable Semi-Supervised Classifier Aggregation,” Balsubramani and Freund). The dissertation author was the primary investigator and author of this paper.

Chapter 14 contains material from the Conference on Uncertainty in Artificial Intelligence, 2016 (“Sequential Nonparametric Testing with the Law of the Iterated Logarithm,” Balsubramani and Ramdas). The dissertation author was a primary investigator and author of this paper.

Much of the above work was supported by the NSF under grant IIS-1162581.
VITA

<table>
<thead>
<tr>
<th>Year</th>
<th>Degree</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Bachelor of Science</td>
<td>University of California, Berkeley</td>
</tr>
<tr>
<td>2013</td>
<td>Master of Science</td>
<td>University of California, San Diego</td>
</tr>
<tr>
<td>2016</td>
<td>Doctor of Philosophy</td>
<td>University of California, San Diego</td>
</tr>
</tbody>
</table>
In this dissertation, we explore two fundamental sets of inference problems arising in machine learning and statistics. We present robust, efficient, and straightforward algorithms for both, adapting sensitively to structure in data by viewing these problems as playing games against an adversary representing our uncertainty.

In the problem of classification, there is typically much more unlabeled data than labeled data, but classification algorithms are largely designed to be supervised, only taking advantage of labeled data. We explore how to aggregate the predictions of an ensemble of such classifiers as accurately as possible in a semi-supervised setting, using
both types of data. The insight is to formulate the learning problem as playing a game over the unlabeled data against an adversary, who plays the unknown true labels. This formulation uses unlabeled data to improve performance over labeled data alone in an extremely general and efficient way, without model assumptions or tuning parameters. We demonstrate this by devising and evaluating a number of practical, scalable semi-supervised learning algorithms. The theoretical contributions include a proof that the optimal aggregation rules in this semi-supervised setting are artificial neurons for many natural loss functions, with efficient convex algorithms for learning them.

We also provide fundamental results for a second set of problems relating to sequential learning and testing. Random variation in such situations can typically be described by a martingale, a generalization of a random walk that describes any repeated fair game. We describe the concentration behavior of a martingale’s sample path, extending to finite times the law of the iterated logarithm, a classic result of probability theory. With this powerful tool, we are able to show how to design simple sequential tests that use as few samples as possible to detect an effect, provably adapting to the unknown effect size. We also apply our results to optimally correct the p-values of many common statistical hypothesis tests, making them robust to the common practice of ‘peeking’ at test results and waiting for a significant one to report.
Chapter 1

Introduction

In this dissertation, we explore two fundamental sets of problems arising in machine learning and statistics. In each of them, we make basic theoretical contributions with demonstrably practical consequences, leading to scalable and simple algorithms for some basic problems in these areas.

Though the two sets of problems seem quite different, they both involve learning to optimally cope with uncertainty, and so we take related approaches to both. The key conceptual idea is to represent the uncertainty in the problem by an adversary, which attempts to foil our algorithm in a two-player zero-sum game. In both sets of problems we study, this type of thinking underlies our contributions. Viewing these problems as games also allows us to prove some notion of optimality for all our solutions.

The dissertation has two parts, one for each topic area. These are outlined here.

1.1 Outline of Part I: “Muffled” Semi-Supervised Classifier Aggregation

In Part I, we study the learning problem of classification, in which data belonging to different classes (categories) are presented to an algorithm that learns how to discriminate between the classes. This is typically done by presenting data to the algorithm with accompanying class labels, as a child learns how to identify certain animals as dogs with
some initial supervision. Learning is not memorization, however, and a child can also generalize, correctly classifying as a dog even a breed they have never previously seen.

Machine learning has long explored methods to accomplish these objectives using labeled data. Classification algorithms such as the decision tree, perceptron, support vector machine (SVM), and artificial neural network are central to the field. They have enjoyed great practical success, especially when their predictions are aggregated, as with boosting and other ensemble learning methods. However, these algorithms are designed to utilize only labeled data, and providing reliably correct labels is often difficult, expensive, or slow – indeed, this typically motivates the adoption of machine learning classifiers in the first place.

An analogous situation is faced by the curious child eliciting animal labels by asking incessant questions of their exasperated parents. But the child’s discriminative capability also depends on an abundant number of unlabeled observations. These unlabeled data are often relatively easy to obtain in machine learning applications as well, and many semi-supervised learning methods attempt to harness them in concert with the labeled data for better prediction.

Part I of the dissertation develops a formulation describing how to combine an ensemble of classifiers of varying competences using unlabeled data. The formulation leads to a basic algorithm with notable practical and theoretical advantages, achieved without making assumptions on the data or the ensemble predictions: it attains the best possible performance guarantee using the information it is given, very efficiently with no parameters to tune. We explain how this is done in detail over the course of Part I, and devise a number of algorithms for related problems with similar guarantees. Most of the chapters concern binary classification, i.e. with two categories of data, an important problem in applications and theory.

The chapters can be read in order. Chapters 3-4, and the “specialist” part of
Chapter 5, constitute the core of our formulation, which we call “muffled” learning for reasons discussed in Chapter 6. More details follow.

Chapter 2 contains a gentle informal introduction to our formulation, with minimal mathematics but retaining its core concepts fairly precisely.

Chapter 3 fully formalizes the main idea in the most basic case. We derive an efficient algorithm that combines an arbitrary finite set of binary classifiers in a way that achieves an optimal worst-case classification error bound, with the optimality guarantees on performance we have mentioned. This optimal algorithm’s decision rule resembles a weighted majority vote over the ensemble predictions, and is an artificial neuron. The unlabeled data serve to regularize the learning procedure, ensuring good generalization especially when unlabeled data are abundant. **This chapter should be read before all the subsequent chapters in Part I.**

Chapter 4 further extends the “muffled” formulation of Chapter 3 to other binary classification problems, in which the algorithm is judged using a general loss function instead of only “0-1” classification error. The loss functions we consider include not only convex functions as typically studied in machine learning, but also a large set of non-convex losses, for which we derive efficient and worst-case-optimal learning algorithms. The optimal decision rules for these general losses in our framework are all artificial neurons – sigmoid functions of linear combinations of their inputs.

Chapter 5 demonstrates the practical viability of the formulation with an algorithm. Here we extend the formulation to handle arbitrary “specialist” classifiers, so that the classifiers in the ensemble can abstain from predicting on any subset of the data. This allows us to devise an algorithm that aggregates the decision trees in a random forest, using our formulation with specialists that take advantage of the forest’s leaf structure, and achieving superior empirical performance to supervised ensemble algorithms with unlabeled data.
Chapter 6 presents more algorithms devised using our muffled formulation. Here we focus primarily on learning a good aggregate of classifiers one at a time from a possibly very large ensemble, enabling for example algorithms that sequentially build an ensemble of decision trees. The scenario is superficially similar to supervised algorithms for boosting ([SF12]), with the ensemble being accessed by an “oracle” that is given a dataset and returns a classifier performing well on it, and the data repeatedly reweighted by the algorithm to improve performance. This similarity is somewhat misleading, as the new algorithm operates on the very different principles of our formulation, which crucially uses unlabeled data. Also included in this chapter is an improvement to the algorithm of Chapter 5 that utilizes the complex internal structure of the trees in the random forest to improve performance. This chapter evaluates the most empirically successful algorithms devised so far with the muffled formulation, at the time of writing.

Chapter 7 theoretically extends the formulation even further, to cases when the data fall into $K > 2$ classes. Subsuming the results of Chapter 4, we derive the minimax optimal learning algorithms for a large class of multiclass losses, including many non-convex losses. The optimal decision rule (and learning algorithm) takes a notably concise form for multiclass logarithmic loss – it is precisely a “softmax” artificial neuron of its inputs.

Chapter 8 describes how to learn a classifier capable of abstaining from making a label prediction, using our formulation for aggregating binary classifiers. Such a classifier hopes to abstain where it would be most inaccurate if forced to predict, so it has two goals in tension with each other: minimizing errors, and avoiding abstaining unnecessarily often. We exactly characterize the best achievable tradeoff between these two goals in the general semi-supervised setting of the earlier chapters, giving an algorithm for learning a classifier which trades off its errors with abstentions in a Pareto optimal manner. This algorithm is as efficient as linear learning and prediction, and comes with strong
and robust theoretical guarantees. Our analysis extends to the general loss functions of Chapter 4.

Chapter 9 contains a theoretical exploration of the main sequential aggregation algorithm of Chapter 6. In particular, this chapter focuses on analyzing convergence to the optimum, despite the possibly very large base ensemble being aggregated, and the lack of a fixed-dimensional optimization space. The guarantees in this chapter are proved with elementary techniques, but are very general, handle ensembles of any complexity, and illuminate the existence of a “hard core” of difficult-to-classify examples which parallels the structure of boosting.

Chapter 10 summarizes the muffled learning formulation and lists some of the bevy of open problems arising from it.

1.2 Outline of Part II: Martingales, Stopping Times, and Statistical Testing

In Part II of the dissertation, we revisit our basic theme of playing games against an adversary representing uncertainty. Here, the adversary represents our uncertainty about the time at which to evaluate (and stop) a sequential process. We investigate this using a fundamental model of a repeated fair game called a martingale, a powerful generalization of a random walk.

Chapter 11 motivates and discusses martingales for the unfamiliar reader, providing an informal introduction to them and the related problems of stopping and bounding them. It also outlines a particular game-theoretic perspective on concentration that underlies all our technical contributions in the subsequent chapters.

Chapter 12 is a general application of these ideas to develop a characterization of the concentration behavior of martingales, showing that with high probability, they never stray outside a tight envelope around their mean at any time. This pathwise
characterization of how a martingale concentrates uniformly over all times is the best possible, proved with matching anti-concentration bounds. It generalizes a classic result of probability, the law of the iterated logarithm (LIL), to incorporate finite times and failure probabilities. The proofs in this chapter are not the shortest possible, for reasons of illustration – they repeatedly use different versions of a stopped martingale mixing technique that underlies all the technical contributions of Part II.

Chapter 13 further extends these results to hold for mixtures of martingales, uniformly over the mixing distribution in an optimal “PAC-Bayes” sense. These bounds are the most powerful proved in Part II.

Chapter 14 develops sequential statistical tests using this finite-time LIL, which deal well with the common problem of detecting an effect when its size is unknown – here, a difference between two given sample distributions (two-sample testing). These allow the tester to run a single simple algorithm that stops with almost as few samples as possible under any effect of unknown size, while simultaneously enjoying favorable statistical guarantees with respect to all these possible alternatives. Further, this behavior is a direct consequence of the general concentration of measure phenomenon, so more such sequential tests can be similarly derived for other situations besides two-sample testing, including more provided in Chapter 14.

Chapter 15 takes a related perspective on these ideas, resulting in a widely applicable generalization of $p$-values in statistical testing. Our contribution is a $p$-value correction to protect against “data peeking”: the practice of using test results as the data is being collected to influence when to stop collecting data. This problem is a common and serious one – a seemingly significant peek is often used to justify stopping data collection, yet such a result is likely to be seen eventually after peeking long enough. Our solution is very non-invasive, exploiting the relationship between the finite LIL and the fixed-time concentration bounds in prevalent use for statistical tests. This chapter
contains a tighter and more concise, but less instructive, proof of finite LIL concentration than Chapter 12.

As in the rest of this dissertation, the results in Part II have a minimax character, and characterize provably near-optimal behavior in the situations of their respective chapters. An implicitly game-theoretic outlook described in Chapter 11 allows us to be robust to “supervision” in the form of stopping times and effect sizes.

1.3 Notation and Preliminaries

In this dissertation, we use standard concepts, notation, and abbreviations from probability theory and basic linear algebra, including the notion of “independently and identically distributed” (i.i.d.) random variables. Only some chapters contain proofs, and they assume a level of mathematical maturity essentially equivalent to the machine learning textbook [SF12]. Each chapter can be read standalone, borrowing from the notation of previous chapters only when indicated.

We make a few further notes on mathematical notation and terminology. We call a continuous function \( f(x) \) decreasing if it is monotonically non-increasing, i.e. \( f(x) > f(y) \Rightarrow x < y \); a decreasing function has a pseudoinverse \( f^{-1}(y) = \min\{y : f(x) \leq y\} \).

We use the notation \([n] := \{1, 2, \ldots, n\}\). Vectors are almost always written in \textbf{bold}. All vector inequalities, as well as functions like \( \text{sgn}(v) \), are componentwise. We also use the abbreviations \( \text{w.(h.)p.} \) – “with (high) probability,” \( \text{w.l.o.g.} \) – “without loss of generality,” and \( \text{w.r.t.} \) – “with respect to.”
Part I

Games over Unlabeled Data for
Semi-Supervised Classification
Chapter 2

Prologue: A Formulation for Classifier Aggregation

In the spirit of this dissertation, we begin with a game that involves predicting binary labels associated with data.

The prediction game and its formalization constitute a new “muffled” formulation of semi-supervised classification, a main contribution of this thesis which underpins the algorithms and theory in Part I. In this chapter, we describe and build intuition for the game in an informal but fairly precise way, highlighting the salient features and advantages of the formulation, which we will theoretically and practically demonstrate in the subsequent chapters.

2.1 A Motivating Scenario

Suppose a student Alice is sick on a school day, during which her class takes a multiple-choice exam. To her consternation, the teacher of the class requires her to make up the same exam in a couple of days’ time.

Alice’s health recovers in time for her to attend school the next day, only to see her classmates receive their graded exams. The teacher Bob has marked the score on each exam (fraction of questions correct), but no other feedback like which answers were
wrong. Alice, who has not prepared for the exam, sees this as a silver lining, giving her a chance to do well on her make-up test by gleaning information about the correct answers from her classmates. She opportunistically obtains their graded exams for this purpose.

The central question we ask is: how can Alice answer the exam to ensure the highest possible score, using her classmates’ graded exams? As described in this chapter, this is a learning problem. Alice is trying to use the information from her classmates’ imperfect approximations of the unknown correct answers, in order to most accurately predict the answer key.

If her classmates’ exams were marked with the particular questions each erred on along with their correct answers, Alice could trivially use any classmate’s exam to completely determine the answer key, and score perfectly. However, each exam is only marked with the aggregate score, so Alice cannot find out the answer to any particular question by looking at the exam of one of her classmates.

Alice’s potential score intuitively depends on the overall difficulty of the exam; if her classmates do well, so can she, with access to their answers. For instance, if one of her classmates gets a perfect score, Alice will see this grade on their exam, and can simply copy their answers. In general, Alice can just copy the answers of the highest-scoring classmate, and guarantee herself a tie for the top score in her class. Can she do even better? How high can she potentially score with this information?

We answer these and related questions in the rest of Part I, showing that Alice can score higher than any of her classmates, by making basic logical inferences about the correct answers using the predictions and grades of all her classmates jointly. This is possible because there can be significant variation in the competencies of her classmates; they may disagree on different questions, having mastered the range of tested topics to varying degrees. Alice can actually use such diversity to her advantage to reduce her uncertainty about the true answers.
2.2 Some Illustrative Examples

To straightforwardly illustrate how she can do this, we consider the test to have two choices for each answer, one correct and one incorrect. Denote the possible answers as “+” and “-” (e.g. corresponding to a true-false test).

Let us start with a simplified example. Suppose Alice has three classmates and the exam has four questions, with the graded exams shown in Table 2.1 as a matrix. Each of her classmates has made one mistake out of four, and Alice does not know on which questions.

However, the information she has is still helpful. Notice that on the first question, there is one minority answer, with the other two in the majority; so regardless of the true answer, at least one (and possibly two) of the classmates has made a mistake. The same can be said about questions 2 and 3, so that at least three mistakes total must be made on these questions. Since each classmate only makes one mistake total, Alice can conclude that all the minority predictions are wrong, and that the actual answer key is (+, +, +, −) on the four questions respectively. So she can guarantee that she answers the exam perfectly!

Table 2.1. Basic example of Alice’s classmates’ exams; the unweighted majority vote makes no mistakes here.

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Mistakes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>1</td>
</tr>
<tr>
<td>C2</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>1</td>
</tr>
<tr>
<td>C3</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>1</td>
</tr>
</tbody>
</table>

This toy example demonstrates several features of the formulation that will be inherited by the machine learning algorithms devised in Part I of this dissertation.
Alice is able to score higher than any of her classmates, even though she has no more information than they provide. The joint patterns of disagreement among her classmates’ predictions allow Alice to correct out the mistakes that they make. Intuitively, more diverse classmate predictions can potentially help more; if all the classmates had exactly the same answers, Alice clearly could not guarantee better performance than any of them.

Also, she does not need to make modeling assumptions about the content of each question, or about the way her classmates predict. This makes the setting very general and the formulation suitable for machine learning, in which Alice’s situation corresponds to a binary classification problem.

In this setting, the task is to learn a classifier that most accurately predicts the label, or category, of data falling into two categories. The matrix of predictions of Table 2.1 represents a dataset. The four questions – the columns of the matrix – are data points. Alice’s classmates – the rows of the matrix – are then other classifiers, so the learning problem facing her is to aggregate their predictions on the data. The argument Alice uses does not assume any restrictions on the structure of the matrix of classifiers’ predictions (Table 2.1), and correspondingly the algorithms of the subsequent chapters are able to combine the predictions of any arbitrary set, or ensemble, of classifiers.

Another salient feature of Table 2.1 and Alice’s argument is that the correct answer key is a majority vote over the imperfect classmates. This may appear to result from the specific argument chosen for Alice above, which budgets the classmates’ errors according to majority and minority predictions. Such a view calls the argument’s usefulness into question, as majority voting is a very common way to combine predictions and correct out noise, and can be done without resorting to reasoning like Alice’s.

However, this style of reasoning can produce better classifiers than a simple majority vote. Table 2.2 demonstrates this, written in a machine learning setting with six
data (questions on the exam) \(x_1, \ldots, x_6\), and six classifiers (classmates) \(h_1, \ldots, h_p\) split into two blocs. One bloc \(\{h_1, h_2, h_3\}\) is composed of classifiers which err less on data but all agree with each other, while in the other bloc the remaining classifiers err more and also disagree more.

**Table 2.2. Example with two classifier blocs.**

<table>
<thead>
<tr>
<th></th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(x_5)</th>
<th>(x_6)</th>
<th>Mistakes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h_1)</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>2</td>
</tr>
<tr>
<td>(h_2)</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>2</td>
</tr>
<tr>
<td>(h_3)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>(h_4)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>(h_5)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>(h_6)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

We apply Alice’s reasoning from the earlier example to Table 2.2. Observe that \(x_1, x_2, \ldots, x_5\) all have one classifier in the minority, so there are at least five mistakes made among those data in total. If the majority prediction were wrong on any of these data (say \(x_1\)), then there would be five mistakes occurring on \(x_1\), at least one each on \(x_2, \ldots, x_5\), and at least two on \(x_6\), for a total of 11. This is impossible, since we know there are only \(2+2+2+1+1+1 = 9\) mistakes made total. Therefore, the correct labeling of the dataset (the teacher Bob’s answer key) must be ‘+’ for all six data points.

Consequently, we conclude that the majority vote over \(\{h_1, h_2, h_3\}\) has zero error, performing significantly better than any of the base classifiers. In contrast, a simple majority vote over all classifiers would err on \(x_6\), as would the best single classifier, and the seemingly natural strategy of taking a vote over the “good” classifiers \(\{h_4, h_5, h_6\}\).

So Alice’s budgeting-type reasoning does result in interesting data-dependent
behavior, not just a flat majority vote. The example also shows that it can be better to combine less accurate classifiers that disagree with each other than to combine more accurate classifiers that tend to agree.

### 2.3 Playing Against An Adversary

A shortcoming of these simplified examples is that in general, we might expect there to be many possible answer keys consistent with Alice’s classmates’ exams – which one to choose? Can Alice guarantee good performance with such reasoning, as she could by just copying the best classmate?

A related, “dual” set of questions also arises. How well can Alice potentially score with whatever approach she uses? Is she making the best use possible of the information in her classmates’ marked exams?

All these questions can be answered with our formulation, whose central idea we state in somewhat more detail. Each marked exam gives an approximate idea of the actual answer key, which constitutes a vector \( z \) with one coordinate corresponding to each of the questions. As we formalize subsequently, each classmate’s graded exam can be seen as a constraint on \( z \) – a classmate who scores 99% guarantees that \( z \) will be close to the vector of their predictions. The joint effect of these constraints is what leads to the interesting behavior seen in the examples, as the constraint sets associated with the ensemble classifiers intersect to form a smaller constraint set \( S \). \( S \) contains exactly the information about \( z \) given by the ensemble, and represents the possible answer keys given the information in Alice’s classmates’ exams.

As Alice, we aim for the best possible guarantee on our score (i.e. on our error), minimizing our worst-case error on the exam. Let us think of this equivalently as playing a game against the teacher Bob, who must mark Alice’s make-up exam when she has completed it. Bob suspects Alice’s chicanery with other students’ exams, so he wishes to
give her the lowest score in his power, but he has no proof she is cheating. Let us say that as a professional, he cannot confront her and just give her a zero; Bob is constrained to grade the exam consistently with the marks given to Alice’s classmates. But he otherwise has free rein to choose how to mark Alice’s exam, because the students do not know the answer key but only the exam scores.

With these intentions, Bob is Alice’s worst possible adversary, and effectively controls $z$. In the game, Bob plays $z$ with knowledge of Alice’s answers on the test, a vector $g$. He plays to maximize her error given $g$:

$$\max_{z \in S} \text{error}(z, g)$$

(We have informally omitted any constraints on $g$ for this exposition.)

Alice, in trying to guarantee herself a good score, must play to minimize error, assuming the worst possible from Bob and playing

$$g^* = \arg \min_g \max_{z \in S} \text{error}(z, g)$$  \hspace{1cm} (2.1)

In these terms, our learning algorithm for this scenario (in Chapter 3) solves for $g^*$. This is equivalent to solving a constrained optimization problem, which encodes Alice’s budgeting arguments of Sec. 2.2. Remarkably, the optimal prediction rule specified by (2.1) will turn out to be a “soft” weighted majority vote over the members of the class. Another suggestive interpretation of it is as an artificial neuron.

Note that this implies an extremely strong and robust error guarantee. For any predictions $g$ played by Alice, the error suffered against the adversarial Bob is $\max_{z \in S} \text{error}(z, g)$, so Eq. (2.1) implies that no $g$ can guarantee a better error bound. We stress this because of the lack of assumptions on $g$ – even if Alice uses a complex learning
algorithm like a deep network to predict the answers on the exam, she cannot do better than Eq. (2.1) in general, given only the other students’ exams marked in the specified manner. This is a very distinctive theoretical feature of our formulation, discussed further in Sec. 2.5 and later chapters.

2.4 Semi-Supervised Classifier Aggregation

As mentioned previously, the formulation of Alice’s situation corresponds precisely to the machine learning problem of binary classifier aggregation.

This problem of aggregating an arbitrary set of classifiers is very practical, due to the abundance of classification algorithms in common use, because the problem of (binary) classification is one of the most fundamental in machine learning. It has a vast range of applications, from spam filtering and advertisement matching to handwriting recognition, disease prediction, and more ([FHT01, Bis06]). Most general-purpose binary classification algorithms for these applications are supervised, learning to classify correctly by training on a given set of data points with known labels (labeled data). But obtaining correct labels for data is typically expensive, often requiring direct human intervention. In contrast, unlabeled data are abundant and freely obtained ([ZG09]).

In Alice’s case, a supervised approach would use only the grades of her classmates. The best approach in this case is clearly the simple one of copying her best classmate, as suggested earlier; this corresponds to the canonical supervised technique of empirical risk minimization (ERM).

Crucially, the disagreement patterns that Alice relies on to improve her performance are all visible in the matrix of her classmates’ predictions, which is unlabeled data – it does not depend on the actual answer key. The teacher Bob does use the answer key to mark the exams. This information – on classifier errors – can be estimated from labeled data in a standard machine learning setting in which labeled and unlabeled data are drawn
from a distribution. So our formulation and the resulting algorithms are *semi-supervised*, using both labeled and unlabeled data for learning.

## 2.5 Advantages of the New Formulation

Though the semi-supervised classification algorithms of Part I have still to be specified, we have already seen enough of the core formulation to foreshadow a few of their advantages.

- **Strongest possible error guarantee, without dependence on model parameters**: As discussed earlier, this guarantee does not depend on a choice of model class – it applies to any classifier that uses the given semi-supervised information. As a corollary, we show that our method guarantees accuracy at least that of the best single classifier, corresponding to ERM, or Alice copying her best classmate; so using our method “cannot hurt” the error guarantee. Meanwhile, the lack of explicit model fitting means our formulation introduces no parameters to tune.

- **Applies to arbitrary classifiers, specialists**: As mentioned earlier, we assume nothing about the structure of the matrix of data (the class’s predictions). They may form voting blocs and collude in complex ways, but our formulation is sensitive to this because it sees these collusions in the unlabeled data, as in the toy examples of this chapter. In the example with Alice, the types of arguments she uses are also capable of handling classmates who leave exam questions blank, because Bob’s marks for such tests still give useful information about the answers that can be seen in the matrix of data.

We will see these characteristics in the semi-supervised learning algorithms that we develop with the “muffled” formulation.
Chapter 3

Combining Binary Classifiers to Minimize Classification Error

The intuition outlined in Chapter 2 can be formalized in a machine learning setting, which we address in this chapter. Subsequent chapters in Part I build on this chapter’s formulation. However, this chapter’s theoretical results are subsumed by those proved later in Chapter 4, so we defer proofs to the later chapter.

3.1 Introduction

Suppose that we have a finite set, or ensemble, of binary classifiers \( \mathcal{H} = \{h_1, h_2, \ldots, h_p\} \), with each \( h_i \) mapping data in some space \( \mathcal{X} \) to a binary prediction \( \{-1, +1\} \). Examples \((x, y) \in \mathcal{X} \times \{-1, +1\} \) are generated i.i.d. according to some fixed but unknown distribution \( \mathcal{D} \), where \( y \in \{-1, +1\} \) is the class label of the example. We write the expectation with respect to \( \mathcal{D} \) or one of its marginal distributions as \( \mathbb{E}_{\mathcal{D}}[\cdot] \).

Consider a statistical learning setting, in which we assume access to two types of i.i.d. data: a small set of labeled training examples \( S = \{(x'_1, y'_1), \ldots, (x'_m, y'_m)\} \) drawn from \( \mathcal{D} \) and a much larger set of unlabeled test examples \( U = \{x_1, \ldots, x_n\} \) drawn i.i.d. according to the marginal distribution over \( \mathcal{X} \) induced by \( \mathcal{D} \). A typical use of the labeled set is to find an upper bound on the expected error rate of each of the classifiers in the
ensemble. Accordingly, we assume a set of lower bounds \(\{b_i\}_{i=1}^p\) such that the correlation \(\text{corr}(h_i) := \mathbb{E}_\mathcal{D}[y h_i(x)]\) satisfies \(\text{corr}(h_i) \geq b_i\); this is equivalent to assuming an upper bound of \(\frac{1}{2}(1 - b_i)\) on the error rate of classifier \(i\).

If we ignore the test set, then the best achievable error guarantee, in the worst case, is achieved by predicting with the classifier with the largest correlation (smallest error). This corresponds to the common practice of empirical risk minimization (ERM). However, in many cases we can glean useful information from the distribution of the test set that will allow us to greatly improve over ERM.

We motivate this statement by contrasting two simple prediction scenarios, A and B. In both cases there are \(p = 3\) classifiers and \(n = 3\) unlabeled test examples. The correlation vector is \(\mathbf{b} = (1/3, 1/3, 1/3)\); equivalently, the classifier error rates are 33%. Based on that information, the predictor knows that each classifier makes two correct predictions and one incorrect prediction.

So far, both cases are the same. The difference is in the relations between different predictions on the same example. In case A, each example has two predictions that are the same, and a third that is different. In this case it is apparent that the majority vote over the three classifiers has to be correct on all 3 examples, i.e. we can reduce the error from \(\frac{1}{3}\) to 0. In case B, all three predictions are equal for all examples. In other words, the three classification rules are exactly the same on the three examples, so there is no way to improve over any single rule.

These cases show that there is information in the unlabeled test examples that can be used to reduce error – indeed, cases A and B can be distinguished without using any labeled examples. In this chapter, we give a complete characterization of the optimal worst-case (minimax) predictions given the correlation vector \(\mathbf{b}\) and the unlabeled test examples.

Our development does not consider the feature space \(\mathcal{X}\) directly, but instead
represents the knowledge of the $p$ ensemble predictions on $U$ with a $p \times n$ matrix that we denote by $\mathbf{F}$. Our focus is on how to use the matrix $\mathbf{F}$ in conjunction with the correlation vector $\mathbf{b}$, to make minimax optimal predictions on the test examples.

The rest of the chapter is organized as follows. In Section 3.2 we introduce some additional notation. In Section 3.3 we formalize the above intuition as a zero-sum game between the predictor and an adversary, and solve it, characterizing the minimax strategies for both sides by minimizing a convex objective we call the slack function. This solution is then linked to a statistical learning algorithm in Section 3.4.

In Section 3.5, we interpret the slack function and the minimax strategies, providing more toy examples following the one given above to build intuition. In Section 3.6 we focus on computational issues in running the statistical learning algorithm, and discuss related work in Section 3.7.

### 3.2 Mathematical Preliminaries

We use a combination of linear algebra notation and the probabilistic notation given in the previous section. The algorithm is first described in a deterministic context where some inequalities are assumed to hold; probabilistic arguments are used to show that these assumptions are correct with high probability.

The ensemble’s predictions on the unlabeled data are denoted by $\mathbf{F}$:

$$
\mathbf{F} = \begin{pmatrix}
    h_1(x_1) & h_1(x_2) & \cdots & h_1(x_n) \\
    \vdots & \vdots & \ddots & \vdots \\
    h_p(x_1) & h_p(x_2) & \cdots & h_p(x_n)
\end{pmatrix} \in [-1,1]^{p \times n}
$$

The true labels on the test data are nominally denoted by the binary values $y_1, \ldots, y_n \in \{-1,+1\}$, but we use the term to refer to a real vector $\mathbf{z} = (z_1; \ldots; z_n) \in \mathbb{R}^n$. 
$[-1, 1]^n$. This represents the expected values of randomized binary labels $y_i$. For example, a value of $\frac{1}{2}$ indicates the corresponding binary label is $\{+1 \text{ w.p. } \frac{3}{4}, -1 \text{ w.p. } \frac{1}{4}\}$. (The same argument justifies why we take $F$ to be in the continuous interval.) This interpretation extends to our definition of the correlation on the test set, $\hat{\text{corr}}_U(h_i) := \frac{1}{n} \sum_{j=1}^n h_i(x_j)z_j$. \footnote{We are slightly abusing the term “correlation” here. Strictly speaking this is just the expected value of the product, without standardizing by mean-centering and rescaling for unit variance. We prefer this to inventing a new term.}

The labels $z$ are hidden from the predictor, but we assume the predictor has knowledge of a correlation vector $b \geq 0^p$ such that

$$\forall i \in [p], \quad \hat{\text{corr}}_U(h_i) := \frac{1}{n} \sum_{j=1}^n h_i(x_j)z_j \geq b_i$$ \hspace{1cm} (3.2)

i.e. $\frac{1}{n}Fz \geq b$. These $p$ inequalities represent upper bounds on individual classifier error rates.

Define $1^n = (1; 1; \ldots; 1) \in \mathbb{R}^n$, and $0^n$ similarly. Also, write $I_n$ as the $n \times n$ identity matrix, and define \( \text{clip}(x) = \max(\min(1, x), -1). \) Finally, we use vector notation for the rows and columns of $F$: $h_i = (h_i(x_1), h_i(x_2), \ldots, h_i(x_n))^\top$ and $x_j = (h_1(x_j), h_2(x_j), \ldots, h_p(x_j))^\top$.

### 3.3 The Transductive Binary Classification Game

We now describe our prediction problem, and formulate it as a zero-sum game between two players: a predictor and an adversary.

In this game, the predictor plays first with $g = (g_1; g_2; \ldots; g_n)$, a potentially randomized label prediction $g_i \in [-1, 1]$ for each example $\{x_i\}_{i=1}^n$. The adversary then plays, setting the labels $z \in [-1, 1]^n$ under ensemble test error constraints defined by $b$.

The predictor’s goal is to minimize (and the adversary’s to maximize) the worst-case expected classification error on the test data (w.r.t. the randomized labelings $z$ and $g$):
\[ \frac{1}{2} \left( 1 - \frac{1}{n} z^\top g \right). \]

To summarize concretely, we study the following game:

\[
V := \min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n} \quad \frac{1}{2} \left( 1 - \frac{1}{n} z^\top g \right) \tag{3.3}
\]

It is important to note that we are only performing “test time” prediction, and represent the information gleaned from the labeled data by the parameter \( b \). Inferring the vector \( b \) from training data is a standard application of Occam’s Razor ([BEHW87]), which we provide in Section 3.4.

The minimax theorem (Theorem 25) applies to the game (3.3), since the constraint sets are convex and compact and the payoff linear. Therefore, it has a minimax equilibrium and associated optimal strategies \( g^*, z^* \) for the two sides of the game, i.e.

\[
\min_z \frac{1}{2} \left( 1 - \frac{1}{n} z^\top g^* \right) = V = \max_g \frac{1}{2} \left( 1 - \frac{1}{n} z^{*\top} g \right).
\]

As we will show, optimal play depends on a particular weighting over the \( p \) hypotheses – a nonnegative \( p \)-vector. Define this weighting as follows.

**Definition 1 (Slack Function and Optimal Weighting).** Let \( \sigma \geq 0^p \) be a weight vector over \( \mathcal{H} \) (not necessarily a distribution). The **slack function** is

\[
\gamma(\sigma, b) = \gamma(\sigma) := -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \max \left( \left| x_j^\top \sigma \right|, 1 \right)
\]

An **optimal weight vector** \( \sigma^* \) is any minimizer of the slack function: \( \sigma^* \in \arg\min_{\sigma \geq 0^p} [\gamma(\sigma)] \).

The vector of ensemble **scores** w.r.t. \( \sigma \) is \( F^\top \sigma = (x_1^\top \sigma, \ldots, x_n^\top \sigma) \), whose elements’ magnitudes are the **margins**. We call the convex function \( \max \left( \left| x_j^\top \sigma \right|, 1 \right) \) the **potential well**.

The main result of this chapter uses these definitions to describe the solution of the game (3.3).
The optimal strategy for the predictor player on an example $\mathbf{x}$ plotted with the score-dependent term of the slack function, as a function of the score $\mathbf{x}^\top \mathbf{\sigma}^*$. 

**Theorem 2 (Minimax Optimal Strategy for (3.3)).** The minimax value of the game (3.3) is

$$V = \frac{1}{2} \gamma(\mathbf{\sigma}^*)$$

The minimax optimal predictor strategy\(^2\) is defined as follows: for all $i \in [n]$,

$$g_i^* = g_i(\mathbf{\sigma}^*) = \begin{cases} \frac{\mathbf{x}_i^\top \mathbf{\sigma}^*}{\frac{\mathbf{x}_i^\top |\mathbf{\sigma}^*|}{< 1}} & \mathbf{x}_i^\top |\mathbf{\sigma}^*| < 1 \\ \text{sgn}(\mathbf{x}_i^\top \mathbf{\sigma}^*) & \text{otherwise} \end{cases}$$ (3.5)

The proof of this theorem is a standard application of Lagrange duality and the minimax theorem.

Theorem 2 illuminates the importance of the optimal weighting $\mathbf{\sigma}^*$ over hypotheses. This weighting $\mathbf{\sigma}^* \in \arg\min_{\mathbf{\sigma} \geq 0^p} \gamma(\mathbf{\sigma})$ is the solution to a convex optimization problem, and therefore we can efficiently compute it and $\mathbf{g}^*$ to any desired accuracy. The

\(^2\)For completeness,

$$z_i^* = \begin{cases} 0 & \mathbf{x}_i^\top |\mathbf{\sigma}^*| < 1 \\ \text{sgn}(\mathbf{x}_i^\top \mathbf{\sigma}^*) & \mathbf{x}_i^\top |\mathbf{\sigma}^*| > 1 \end{cases}$$

For $\mathbf{x}_i$ with margin 1, $z_i^* = c_i \text{sgn}(\mathbf{x}_i^\top \mathbf{\sigma}^*)$ for any $c_i \in [0, 1]$, with the proviso that $\frac{1}{n} \mathbf{Fz} \geq \mathbf{b}$. Our paper [BF15a] contains further discussion.
ensemble scores (w.r.t. this weighting) on the test set are represented by $\mathbf{F}^\top \sigma^*$, which is the only dependence of the solution on $\mathbf{F}$.

More specifically, the minimax optimal prediction in (3.5), for any test set example $\mathbf{x}_j$, can be expressed as a function of the score $\mathbf{x}_j^\top \sigma^*$ on that test point alone without considering the others. The $\mathbf{F}$-dependent part of the slack function also depends separately on each test point’s ensemble prediction (Figure 3.1). Even though we consider all the data and the ensemble’s interdependencies jointly to add power to the formulation, and must therefore search over a space of exponential size in $n$, the data are completely decoupled from each other given $\sigma^*$, so we can solve the problem efficiently (see Sec. 3.6).

### 3.4 Bounding the Correlation Vector Using Labeled Data

In the analysis presented above, we assumed that a correlation vector $\mathbf{b}$ is given, and that each component is guaranteed to be a lower bound on the test correlation of the corresponding hypothesis. This section justifies that assumption, showing how $\mathbf{b}$ can be calculated from a labeled training set.

The algorithm that we use is a natural one using uniform convergence: we compute the empirical correlations for each of the $p$ classifiers, and add a uniform penalty term to guarantee that the $\mathbf{b}$ is a lower bound on the correlation of the test data. For each classifier, we consider three quantities:

- The true correlation: $\text{corr}(h) = \mathbb{E}_{\mathcal{D}}[yh(x)]$
- The correlation on the training set of labeled data: $\widehat{\text{corr}}_S(h) = \frac{1}{m} \sum_{i=1}^{m} h(x'_i) y'_i$
- The correlation on the test set of unlabeled data: $\widehat{\text{corr}}_U(h) = \frac{1}{n} \sum_{i=1}^{n} h(x_i) z_i$
Using Chernoff bounds, we can show that the training and test correlations are concentrated near the true correlation. Specifically, for each individual classifier $h$ we have the two inequalities

$$\Pr(\hat{\text{corr}}_S(h) > \text{corr}(h) + \epsilon_S) \leq e^{-2m\epsilon_S^2}$$

$$\Pr(\hat{\text{corr}}_U(h) < \text{corr}(h) - \epsilon_U) \leq e^{-2n\epsilon_U^2}$$

Let $\delta$ denote the probability we allow for failure of uniform convergence. If we set $\epsilon_S = \sqrt{\ln(2p/\delta)/2m}$ and $\epsilon_U = \sqrt{\ln(2p/\delta)/2n}$, we are guaranteed that all the $2p$ inequalities hold concurrently with probability at least $1 - \delta$.

We thus set the correlation bound to:

$$b_i := \hat{\text{corr}}_S(h_i) - \epsilon_S - \epsilon_U$$

and have with probability $\geq 1 - \delta$ that $b$ is a good correlation vector, i.e. $\hat{\text{corr}}_U(h_i) \geq b_i \quad \forall i \in [p]$.

This argument, using uniform convergence bounds on the classifiers to compute $b$, can be extended to more nuanced bounds than the union bound (Occam’s Razor) employed here, including VC and Rademacher inequalities ([DGL13]).

### 3.5 Interpretation and Discussion

Notions of margin are well-studied in the literature on learning algorithms (see Sec. 3.7), which generally view it as something to be maximized. In contrast, the slack function encourages us to minimize the margin that emerges from our formulation, tempering how aggressively we rely on the labeled data through $b$, which improves generalization by avoiding weaknesses which the adversary can exploit. So the unlabeled
data has a regularizing effect, encouraging the minority label and thereby “muffling” the
guidance given by labeled examples. This will recur in the algorithms developed later in
Chapters 5 and 6.

Given a weighting $\sigma$, we partition the examples $x$ into three subsets, depending
on the value of the margin: the **hedged set** $H(\sigma) := \{x : |x^\top \sigma| < 1\}$, the **clipped set**
$C(\sigma) := \{x : |x^\top \sigma| > 1\}$, and the **borderline set** $B(\sigma) := \{x : |x^\top \sigma| = 1\}$. Using these
sets, we now give some intuition regarding the optimal choice of $g$ and $z$ given in (3.5),
for some fixed $\sigma$.

Consider first examples $x_i$ in $H(\sigma)$. Here the optimal $g_i$ is to predict with the
score $x_i^\top \sigma$, a number in $(-1, 1)$. Making such an intermediate prediction might seem
to be a type of calibration, but this view is misleading. The optimal strategy for the
adversary in this case is to set $z_i = 0$, equivalent to predicting $\pm 1$ with probability $1/2$
each. The reason that the learner hedges is because if $g_i < x_i^\top \sigma$, the adversary would
respond with $z_i = 1$, and with $z_i = -1$ if $g_i > x_i^\top \sigma$. In either case, the loss of the predictor
would increase. In other words, our ultimate rationale for hedging is not calibration, but
rather “defensive forecasting” in the spirit of [VTS05].

Next we consider the clipped set $x_j \in C(\sigma)$. In this case, the adversary’s optimal
strategy is to predict deterministically, and so the learner matches the adversary here. It
is interesting to note that with all else held equal, increasing the margin $|x_j^\top \sigma|$ beyond
1 is suboptimal for the learner. Qualitatively, the reason is that while $x_j^\top \sigma$ continues to
increase, the prediction for the learner is clipped, and so the value for the learner does
not increase with the ensemble prediction. The intuition here is that the learner should
not be too aggressive with its predictions, as this would leave a weakness that an optimal
adversary could exploit to increase error.
Figure 3.2. A schematic illustration of the optimal $\sigma^* \geq 0^p$. The vector $nb$ is the difference between the sums of two categories of clipped examples: those with high ensemble prediction ($x^T \sigma^* > 1$) and low prediction ($< -1$). The effect of $B(\sigma^*)$ is neglected for simplicity.

3.5.1 Subgradient Conditions

For another perspective on the result of Theorem 2, consider the subdifferential set of the slack function $\gamma$ at an arbitrary weighting $\sigma$:

$$
\partial \gamma(\sigma) = \left\{ \frac{1}{n} \left( \sum_{x_j \in C(\sigma)} x_j \text{sgn}(x_j^T \sigma) + \sum_{x_j \in B(\sigma)} c_j x_j \text{sgn}(x_j^T \sigma) \right) - b, \quad \forall c_j \in [0, 1] \right\}
$$

(3.6)

Note that the hedged set plays no role in $\partial \gamma(\sigma)$. Since the slack function $\gamma(\cdot)$ can be seen to be convex, the sub-differential set (3.6) at any optimal weighting $\sigma^*$ contains $\vec{0}$, i.e.,

$$
\exists c_j \in [0, 1] \quad s.t. \quad nb - \sum_{j: x_j^T \sigma^* > 1} x_j + \sum_{j: x_j^T \sigma^* < -1} x_j = \sum_{j: |x_j^T \sigma^*| = 1} c_j x_j \text{sgn}(x_j^T \sigma^*)
$$

(3.7)

The geometric interpretation of this equation is given in Figure 3.2. The optimal weighting $\sigma^*$ partitions the examples into five sets: hedged, positive borderline and
positive clipped, and negative borderline and negative clipped. Taking the difference between the scaled sums of the positive clipped and the negative clipped examples gives a vector that is approximately $b$. By adding a weighted sum of the borderline examples, $b$ can be obtained exactly.

### 3.5.2 Approximate Learning

Another consequence of our formulation is that predictions of the form $g(\sigma)$ are closely related to dual optima and the slack function. Indeed, by definition of $g(\sigma)$, the slack function can be rewritten in terms of $g(\sigma)$ as

$$\gamma(\sigma) = -b^T \sigma + \frac{1}{n} \|F^T \sigma - g(\sigma)\|_1 + 1 \geq \min_{\sigma' \geq 0^p} \left[ -b^T \sigma' + \frac{1}{n} \|F^T \sigma' - g(\sigma)\|_1 \right] + 1$$

The minimization is simply the dual problem of the worst-case correlation suffered by $g(\sigma)$, i.e. $\max_{z \in [-1,1]^n} -\frac{1}{n}z^T [g(\sigma)]$ (this is proved in Lemma 11). We now state this formally.

*Observation 3.* For any weight vector $\sigma \geq 0^p$, the worst-case error after playing $g(\sigma)$ is bounded by

$$\max_{z \in [-1,1]^n, \frac{1}{n}Fz \geq b} \frac{1}{2} \left( 1 - \frac{1}{n}z^T [g(\sigma)] \right) \leq \frac{1}{2} \gamma(\sigma)$$

Observation 3 shows that convergence guarantees for optimizing the slack function directly imply error guarantees on predictors of the form $g(\sigma)$, i.e. prediction rules of the form in Fig. 3.1.

### 3.5.3 Characteristics of the Solution

We now make some brief observations about the minimax solution, which formalize some of the specific intuitions discussed in Chapter 2.
First, note that no $\sigma$ such that $\|\sigma\|_1 < 1$ can be optimal, because in such a case $\gamma(\sigma) < \gamma\left(\frac{\sigma}{\|\sigma\|_1}\right)$; therefore, $\|\sigma^*\|_1 \geq 1$.

Next, suppose we do not know the matrix $F$. Then $\|\sigma^*\|_1 = 1$. This can be shown by proving the contrapositive. Assuming the negation $1 < \|\sigma^*\|_1 := a$, there exists a vector $x \in [-1, 1]^p$ such that $x^T \sigma^* = \|\sigma^*\|_1 > 1$. If each of the columns of $F$ is equal to $x$, then by definition of the slack function, $\gamma\left(\frac{\sigma^*}{\sigma}\right) < \gamma(\sigma^*)$, so $\sigma^*$ cannot be optimal.

In other words, if we want to protect ourselves against the worst case $F$, then we have to set $\|\sigma\|_1 = 1$ so as to ensure that $C(\sigma)$ is empty. In this case, the slack function simplifies to $\gamma(\sigma) = -b^T \sigma$, over the probability simplex. Minimizing this is achieved by setting $\sigma_i$ to be 1 at $\arg\max_i b_i$ and zero elsewhere. So as we stated without proof earlier, in the case that $F$ is unknown, the optimal strategy is simply the ERM strategy – to use the classifier with the best error guarantee.

A couple of examples illustrate the behavior of our formulation, with the first generalizing the example mentioned in Sec. 3.1. Take $p$ to be odd and suppose that $n = p$. Then set $F$ to be a matrix where each row (classifier) and each column (example) contains $(p + 1)/2$ entries equal to +1 and $(p - 1)/2$ entries equal to −1 (for instance, by setting $F_{ij} = 1$ if $(i + j)$ is even, and −1 otherwise). Finally choose an arbitrary subset of the columns (to have true label −1), and invert all their entries.

In this setup, all classifiers (rows) have the same error: $b = \frac{1}{p} 1^p$. The optimal weight vector in this case is $\sigma^* = 1^p$, we observe $|x^T \sigma^*| = 1 \forall x$, and the minimax value is $V = 1$, which corresponds to zero error. Any single rule has an error of $\frac{1}{2} - \frac{1}{p}$, so using $F$ with $p$ classifiers leads to a $p$-fold improvement over random guessing!

Of course, this particular case is extremal in some ways; since no point is clipped, there must be many cancellations between ensemble predictions resulting in the scores $F^T \sigma^*$. This echoes the common heuristic belief that, when combining an ensemble of classifiers, we want the classifiers to be “diverse” (e.g. [KW03]). The above example in
fact has the maximal average disagreement between pairs of classifiers for a fixed $p$.

So our formulation recovers ERM without knowledge of $F$, and can recover an (unweighted) majority vote in cases where this provides dramatic performance improvements. It is also clear (from Obs. 3) that $V$ is always less than the error of the best single classifier, which corresponds to $\sigma$ being the coordinate basis vector in this classifier’s direction (see also Prop. 9 of Ch. 4). The real algorithmic benefit of our unified formulation is in automatically interpolating between these extremes in a manner that minimizes worst-case error.

To illustrate, suppose $F$ is given in Table 2.2 from Chapter 2, in which there are six classifiers partitioned into two blocs, and six equiprobable test examples. In the earlier chapter, we argued that the true labeling must be $+$ on all examples. It can easily be verified that our algorithm recovers this perfectly with a weighting of $\sigma^* = (1;1;1;0;0;0)$.

3.5.4 Independent Label Noise

An interesting variation on the game is to limit the adversary to $z_i \in [-\alpha_i, \alpha_i]^n$ for some $\bar{\alpha} = (\alpha_1; \ldots; \alpha_n) \in [0,1)^n$. This corresponds to assuming a level $1 - \alpha_i$ of independent label noise on example $i$: the adversary is not allowed to set the label deterministically, but is forced to flip example $i$’s label independently with probability $\frac{1}{2}(1 - \alpha_i)$.

Solving the game in this case gives the result that if we know some of the ensemble’s errors to be through random noise, then we can find a weight vector $\sigma$ that would give us better performance than without such information.
Proposition 4 (Independent Label Noise).

$$\min_{g \in [-1,1]^n} \max_{\frac{1}{n} \sum z \geq b, \alpha \leq z \leq \alpha} \frac{1}{2} \left( 1 - \frac{1}{n} z^T g \right) = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \frac{1}{n} \sum_{j=1}^{n} \alpha_j \max \left( |x_j^T \sigma|, 1 \right) \right]$$

$$\leq \frac{1}{2} \min_{\sigma \geq 0^p} [\gamma(\sigma)] = V$$

Our prediction tends to clip – predict with the majority vote – more on examples with more known random noise, because it gains in minimax correlation by doing so. This mimics the Bayes-optimal classifier, which is always a majority vote. A similar result holds for the asymmetric-noise case.

3.6 Computational Considerations

The learning algorithm we presented has two steps. The first, calculating $b$, is efficient and straightforward: Section 3.4 shows that this can be done as efficiently as simply computing the average error over training examples.

So our ability to produce $g^*$ is dependent on our ability to find the optimal weighting $\sigma^*$ by minimizing the slack function $\gamma(\sigma)$ over $\sigma \geq 0^p$, with $b$ taken as given. Note that typically $p \ll n$, and so it is a great computational benefit in this case that the optimization is over the dual variables.

The slack function is a sum of $n$ i.i.d. random variables, and has a natural limiting object $-b^T \sigma + \mathbb{E}_{x \sim \mathcal{D}} \left[ \max (|x^T \sigma|, 1) \right]$; it is an ideal candidate for stochastic optimization algorithms. We suggest stochastic gradient descent (SGD) as a constant-memory algorithm, since the function is convex. We discuss this more in the next chapter, and all the novel practical algorithms that we experiment with later in Part I use some variation of SGD to optimize the slack function or similar.
3.7 Related Work

Our work relates to the literature on boosting for forming ensembles, in which the noteworthy work of [SFBL98] shows general bounds on the error of a weighted majority vote $\varepsilon_{WMV}(\hat{\sigma})$ under any distribution $\hat{\sigma}$, based purely on the distribution of a version of the margin on labeled data. As mentioned earlier, such margins recur in many classification algorithms like support vector machines ([CV95]) and boosting ([SF12]), but they give rise to max-margin methods and analysis ([SFBL98]).

Interestingly, [AUL09] take a related approach to prove bounds on $\varepsilon_{WMV}(\hat{\sigma})$ in a transductive setting, as a function of the average ensemble error $b^\top \hat{\sigma}$ and the test data margin distribution; but their budgeting is looser and purely deals with majority votes, in contrast to our $g$ in a hypercube. The transductive setting has general benefits for averaging-based bounds also ([BL03]).

One class of philosophically related methods to ours uses moments of labeled data in the statistical learning setting to find a minimax optimal classifier; notably among linear separators ([LGBJ01]) and conditional label distributions under log loss ([LZ14]). Our formulation instead uses only one such moment and focuses on unlabeled data, and is thereby more efficiently able to handle a rich class of dependence structure among classifier predictions, not just low-order moments.

There is also a long tradition of analyzing worst-case binary prediction of online sequences, from which we highlight work of Feder et al. ([FMG92]), which shows universal optimality for bit prediction of a piecewise linear “clipped” decision function very similar to ours in Fig. 3.1. The work of [CBFH+93] demonstrates this idea to result in optimal error when predicting with expert advice as well, and similar results have been shown in related settings ([Vov90, AP13]).

Our emphasis on the benefit of considering global effects (our transductive setting)
even when data are i.i.d. is in the spirit of the idea of shrinkage, well known in statistical literature as exemplified by the James-Stein estimator ([EM77]).

**Acknowledgements**

This chapter contains material from the Conference on Learning Theory, 2015 ("Optimally Combining Classifiers Using Unlabeled Data," Balsubramani and Freund). The dissertation author was the primary investigator and author of this paper.
Chapter 4

Optimal Binary Classifier Aggregation for General Losses

4.1 Setup

In Chapter 3, we described how to optimally aggregate binary classifiers to minimize classification error, known in machine learning and decision theory as zero-one loss. However, the zero-one loss is inappropriate for other common binary classification tasks, such as estimating label probabilities, and handling false positives and false negatives differently. Such goals motivate the use of different losses, like the logarithmic loss and cost-weighted misclassification loss respectively.

In this chapter, we generalize the setup of Chapter 3 to these loss functions and a large class of others. Like the earlier work, we show that the choice of loss function completely governs an ensemble aggregation algorithm that is minimax optimal in our setting, and is very efficient and scalable to boot.

The algorithm learns a weighting over ensemble classifiers by solving a convex optimization problem. The optimal prediction on each example in the test set turns out to be a sigmoid-like function of a linear combination of the ensemble predictions, using the learned weighting. The minimax structure ensures that this prediction function and the training algorithm are completely data-dependent without parameter choices, relying
merely on the structure of the loss function. It also establishes the minimax optimal prediction to have structure reminiscent of a weighted majority vote over the ensemble, and exactly paralleling the prediction function of a generalized linear model ([MN89]).

In short, the optimal prediction function we derive takes the form of an artificial neuron.

4.1.1 Loss Functions

Recall the notation of Chapter 3, defining the classifier ensemble $\mathcal{H}$, unlabeled data $F$, and true labels $z$. In that setting, we suffer loss on each test point according to our expected classification error on it, if our prediction $g_j \in [-1, 1]$ is interpreted as a randomized binary value in the same way as $z_j$. So if the true label for example $j$ is $y_j = 1$, then the loss of predicting $g_j$ on it is $\ell_+(g_j) = \frac{1}{2}(1 - g_j)$; this is $\{0, 1\}$ when $g_j = \{1, -1\}$ respectively, and a convex combination of the two when $g_j \in (-1, 1)$.

Similarly, if $y_j = -1$, then the loss is $\ell_-(g_j) = \frac{1}{2}(1 + g_j)$.

We call $\ell_\pm$ the partial losses here, following earlier work (e.g. [RW10]). Since the true label can only be $\pm 1$, the partial losses fully specify the decision-theoretic problem we face, and changing them is tantamount to altering the prediction task.

To guide intuition as to what such partial losses could conceivably look like, observe that they intuitively measure discrepancy to the true label $\pm 1$. As a result it makes sense for e.g. $\ell_+(g)$ to be decreasing, as $g$ increases toward the notional true label $+1$. This suggests that both partial losses $\ell_+(\cdot)$ and $\ell_-(\cdot)$ should be monotonic, which we assume in this chapter.

Assumption 1. Over the interval $(-1, 1)$, $\ell_+(\cdot)$ is decreasing and $\ell_-(\cdot)$ is increasing, and both are twice differentiable.

We view Assumption 1 as natural, as we have motivated (differentiability is convenient for our proofs, but most of our arguments do not require it; see Section 4.3 for
details). Notably, we do not require convexity or symmetry of the losses. We refer to loss functions whose partial losses satisfying Assumption 1 as “general losses” to contrast them with convex losses or other less broad subclasses – our main learnability result holds for all such losses.

The expected loss we suffer with respect to the randomized true label \( z_j \) is a linear combination of the partial losses:

\[
\ell(z_j, g_j) := \left( \frac{1 + z_j}{2} \right) \ell_+(g_j) + \left( \frac{1 - z_j}{2} \right) \ell_-(g_j)
\] (4.1)

In decision theory and learning theory, there has been much investigation into the nature of the loss \( \ell \) and its partial losses, particularly on how to estimate the “conditional label probability” \( z_j \) using \( \ell(z_j, g_j) \). A natural operation to do this is to minimize the loss over \( g_j \); accordingly, a loss \( \ell \) such that \( \arg\min_{g \in [-1, 1]} \ell(z_j, g) = z_j \) (for all \( z_j \in [-1, 1] \)) is called a proper loss ([BSS05, RW10]), which will be used in later discussions.

### 4.1.2 Minimax Formulation

The predictor’s goal is to minimize the worst-case expected loss on the test data (w.r.t. the randomized labeling \( z \)), using the loss function \( \ell(z, g) \) as defined earlier in Equation (4.1):

\[
\ell(z, g) := \frac{1}{n} \sum_{j=1}^{n} \ell(z_j, g_j)
\]
This goal can be written as the following optimization problem, a two-player zero-sum game:

\[
V := \min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} \ell(z, g) \tag{4.2}
\]

\[
= \min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} \frac{1}{n} \sum_{j=1}^{n} \left[ \left( \frac{1 + z_j}{2} \right) \ell_+(g_j) + \left( \frac{1 - z_j}{2} \right) \ell_-(g_j) \right] \tag{4.3}
\]

In this chapter, we solve the learning problem faced by the predictor, finding an optimal strategy \(g^*\) realizing the minimum in (4.2) for any given “general loss” \(\ell\). This strategy guarantees good worst-case performance on the unlabeled dataset, with an upper bound of \(V\) on the loss. This bound is perfectly tight, by virtue of the argument above – for all \(z_0\) and \(g_0\) obeying the constraints, our definitions give the tight inequalities

\[
\min_{g \in [-1,1]^n} \ell(z_0, g) \leq V \leq \max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} \ell(z, g_0) \tag{4.4}
\]

which are met with equality, respectively by \(g^*\) and some \(z^*(g_0)\).

In our formulation of the problem, the constraints on the adversary play a central role. As discussed previously, these constraints encode the information we have about the true labels. Without them, the adversary would find it optimal to trivially guarantee error (very close to) \(\frac{1}{2}\) by simply setting all labels uniformly at random \((z = 0^n)\). It is easy to see that adding more information through constraints will never raise the error bound \(V\).  

So far, we have given no assumption about the characteristics of \(\ell(z, g)\) other than Assumption 1. Many of our results will require only this, holding for these “general losses.”

\[\text{1 Though it may pose statistical difficulties to do with uniform convergence over the ensemble ([BF15b]).}\]
This brings us to our contributions:

1. We give the exact minimax $g^* \in [-1, 1]^n$ for general losses (in Section 4.2.1). The optimal prediction on each example $j$ is a sigmoid function of a fixed linear combination of the ensemble’s $p$ predictions on it, so $g^*$ is a non-convex function of the ensemble predictions on $x_j$. This is also a constructive proof of a bound on the worst-case loss of any predictor constructed from the ensemble, by Equation (4.4).

2. We derive an efficient algorithm for finding $g^*$, by solving a $p$-dimensional convex optimization problem (Section 4.2.2). We prove this for a broad subclass of losses (the conditions of Lemma 2), including all convex ERM surrogate losses. Extensions to weighted problems and others are in Section 4.3.

3. The optimal $g^*$ and an efficient algorithm for it, as above, extended to the case when the constraints can arise from general loss bounds on ensemble classifiers (see appendices), rather than classifier error rates.

### 4.2 Results for Binary Classification

A few more quantities will be convenient to specify before discussing our main results.

Based on the loss, define the **score function** $\Gamma : [-1, 1] \mapsto \mathbb{R}$ is

$$\Gamma(g) := \ell_-(g) - \ell_+(g)$$

(We will also write the vector $\Gamma(g)$ componentwise with $[\Gamma(g)]_j = \Gamma(g_j)$ for convenience, so that $\Gamma(h_j) \in \mathbb{R}^n$ and $\Gamma(x_j) \in \mathbb{R}^p$.) Observe that by our assumptions, $\Gamma(g)$ is increasing on its domain, so we can discuss its inverse $\Gamma^{-1}(m)$, which is typically sigmoid-shaped.
With these we will set up the solution to the game (4.2). The solution depends on the optimum of a convex function, defined here for further use.

**Definition 1** (Potential Well). Define the potential well 

\[
\Psi(m) := \begin{cases} 
-m + 2\ell_-(1) & \text{if } m \leq \Gamma(-1) \\
\ell_+(\Gamma^{-1}(m)) + \ell_-(\Gamma^{-1}(m)) & \text{if } m \in (\Gamma(-1), \Gamma(1)) \\
m + 2\ell_+(1) & \text{if } m \geq \Gamma(1)
\end{cases}
\]  

(4.5)

**Lemma 2.** The potential well \(\Psi(m)\) is continuous and 1-Lipschitz. It is also convex under any of the following conditions:

(A) The partial losses \(\ell_\pm(\cdot)\) are convex over \((-1, 1)\).

(B) The loss function \(\ell(\cdot, \cdot)\) is a proper loss.

(C) \(\ell_-(x)\ell_+(x) \geq \ell_-(x)\ell_+(x)\) for all \(x \in (-1, 1)\).

(Indeed, the proof shows that the last condition is both sufficient and necessary for convexity of \(\Psi\), under Assumption 1.) So the potential wells for different losses are shaped roughly similarly, as seen in Figure 4.1.

Lemma 2 tells us that the potential well is easy to optimize under any of the given conditions. Note that these conditions encompass convex surrogate losses commonly used in ERM, including all such “margin-based” losses (convex univariate functions of \(z_j g_j\)). These constitute a large class of losses introduced primarily for their favorable computational properties relative to direct 0-1 loss minimization.

\(^2\)If \(\Gamma\) does not have a unique inverse, our arguments also work, mutatis mutandis, with the pseudoinverse \(\Gamma^{-1}(m) = \inf\{g \in [-1, 1] : \Gamma(g) \geq m\}\).
Figure 4.1. At left are plots of potential wells. At right are optimal prediction functions \( g \), as a function of ensemble score. Both are shown for various losses, as listed in Section 4.2.3.

An easily optimized potential well benefits us, because our learning problem basically consists of optimizing it over the unlabeled data, as we will soon make explicit. The function we will actually need to optimize is in terms of the dual parameters, so we call it the slack function, expanding the definition of Chapter 3 to general losses.

**Definition 3** (Slack Function for General Losses). Let \( \sigma \geq 0^p \) be a weight vector over \( \mathcal{H} \) (not necessarily a distribution). The vector of ensemble scores is \( F^\top \sigma = (x_1^\top \sigma, \ldots, x_n^\top \sigma) \), whose elements’ magnitudes are the margins. The prediction slack function is

\[
\gamma(\sigma, b) := \gamma(\sigma) := -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma) \tag{4.6}
\]

An optimal weight vector \( \sigma^* \) is any minimizer of the slack function: \( \sigma^* \in \arg \min_{\sigma \geq 0^p} [\gamma(\sigma)]. \)

**4.2.1 Solution of the Game**

These are used to describe the minimax equilibrium of the game (4.2), in our main result.
Theorem 4. The minimax value of the game (4.2) is

$$\min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n, \frac{1}{n}Fz \geq b} \ell(z, g) = V = \frac{1}{2} \gamma(\sigma^*) = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma) \right]$$

The minimax optimal predictions are defined as follows: for all \( j \in [n] \),

$$g^*_j := g_j(\sigma^*) = \begin{cases} 
-1 & \text{if } x_j^\top \sigma^* \leq \Gamma(-1) \\
\Gamma^{-1}(x_j^\top \sigma^*) & \text{if } x_j^\top \sigma^* \in (\Gamma(-1), \Gamma(1)) \\
1 & \text{if } x_j^\top \sigma^* \geq \Gamma(1) 
\end{cases} \quad (4.7)$$

\( g^*_j \) is always an increasing sigmoid-shaped function, as shown in Figure 4.1.

We can also redo the proof of Theorem 4 when \( g \in [-1,1]^n \) is not left as a free variable set in the game, but instead is preset to \( g(\sigma) \) as in (4.7) for some (possibly suboptimal) weight vector \( \sigma \).

Observation 5. For any weight vector \( \sigma_0 \geq 0^p \), the worst-case loss after playing \( g(\sigma_0) \) is bounded by

$$\max_{z \in [-1,1]^n, \frac{1}{n}Fz \geq b} \ell(z, g(\sigma_0)) \leq \frac{1}{2} \gamma(\sigma_0)$$

The proof is a simpler version of that of Theorem 4; there is no minimum over \( g \) to deal with, and the minimum over \( \sigma \geq 0^p \) in Equation (4.16) is upper-bounded by using \( \sigma_0 \). This result is an expression of weak duality in our setting, and generalizes Observation 3, from the previous chapter.

4.2.2 The Ensemble Aggregation Algorithm

Theorem 4 defines a prescription for aggregating the given ensemble predictions on the test set. This can be stated in terms of a learning algorithm and a prediction
Learning: Minimize the slack function $\gamma(\sigma)$, finding the minimizer $\sigma^*$ that achieves $V$. This is a convex optimization under broad conditions (Lemma 2), and when the test examples are i.i.d., the $\Psi$ term is a sum of $n$ i.i.d. functions. As such it is readily amenable even to standard first-order optimization methods which require only $O(1)$ test examples at once. In practice, learning employs such methods to approximately minimize $\gamma$, finding some $\sigma_A$ such that $\gamma(\sigma_A) \leq \gamma(\sigma^*) + \varepsilon$ for some small $\varepsilon$. Standard convex optimization methods will do this because the slack function is Lipschitz, as Lemma 2 shows (combined with the observation that $\|b\|_\infty \leq 1$).

Prediction: Predict $g(\sigma^*)$ on any test example, as indicated in (4.7). This decouples the prediction task on each test example, which is as efficient as $p$-dimensional linear prediction, requiring $O(p)$ time and memory. After finding an $\varepsilon$-approximate minimizer $\sigma_A$ in the learning step as above, Observation 5 tells us that the prediction $g(\sigma_A)$ has loss guaranteed to be within $\varepsilon^2$ of $V$.

In particular, note that there is no algorithmic dependence on $n$ in either step in a statistical learning setting, so our transductive formulation is no less tractable than a stochastic optimization setting in which i.i.d. data arrive one at a time.

4.2.3 Examples of Different Losses

To further illuminate Theorem 4, we detail a few special cases in which $\ell_+, \ell_-$ are explicitly defined. These losses may be found throughout the literature on decision theory and learning; for further information, see [RW10]. The key functions $\Psi$ and $g^*$ are listed for these losses in Table 4.1, and in most cases plotted in Figure 4.1. We can see
that the nonlinearities used for $g^*$ are all sigmoids, which arises solely from the minimax structure and the box constraints inherent in the classification game (more details in Section 4.5.1).

- **0-1 Loss**: Here $g_j$ is taken to be a randomized binary prediction; this case was developed in the previous chapter.

- **Log Loss**

- **Square Loss**

- **Cost-Weighted Misclassification (Quantile) Loss**: This is defined with a parameter $c \in [0, 1]$ representing the relative cost of false positives vs. false negatives, making the Bayes-optimal classifier the $c$-quantile of the conditional label distribution ([She05]).

- **Exponential Loss**

- **Logistic Loss**

- **Hellinger Loss**: This is typically given for $p, y \in [0, 1]$ as

  \[
  \frac{1}{2} \left( (\sqrt{p} - \sqrt{y})^2 + (\sqrt{1-p} - \sqrt{1-y})^2 \right)
  \]

  Our formulation is equivalent when the prediction and label are rescaled to $[-1, 1]$.

- **“AdaBoost Loss”**: If the goal of AdaBoost ([SF12]) is interpreted as class probability estimation, the implied loss is proper and given in [BSS05, RW10].

- **Absolute Loss and Hinge Loss**: The absolute loss can be defined by $\ell_\pm^{\text{abs}}(g_j) = 1 \pm g_j$, and the hinge loss also has $\ell_\pm^{\text{abs}}(g_j) = 1 \pm g_j$ since the kink in the hinge loss only lies at $g_j = \mp 1$. These partial losses are the same as for 0-1 loss up to scaling.
and therefore all our results for $\Psi$ and $g^*$ are as well. So these losses are not shown in Table 4.1.

### 4.2.4 Technical Discussion

There are two notable ways in which the result of Theorem 4 is particularly convenient and general. First, the fact that $\ell(z, g)$ can be non-convex in $g$, yet solvable as a convex problem, is a major departure from previous work. Second, the solution has a convenient dependence on $n$ like in Chapter 3 (cf. [BF15a]), simply averaging a function over the unlabeled data; this is not only mathematically convenient but also again makes stochastic optimization practical using only constant memory.

Both these favorable properties stem from the structure of the binary classification problem. We can describe this by examining the proof in Sec. the optimization problem we constructed in Equation (4.17). In it, the constraints which do not explicitly appear with Lagrange parameters are all box, or $L_\infty$ norm, constraints. These decouple over the $n$ test examples, which is a critical reason we are able to obtain results for such general non-convex problems. It allows us to reduce the problem to the one-dimensional optimization at the heart of Equation (4.17), which we can solve with ad hoc methods. When we do so, the $g_i$ are “clipped” sigmoid functions because of the bounding effect of the $[-1, 1]$ constraint and loss function structure.

We introduce Lagrange parameters $\sigma$ only for the $p$ remaining constraints in the problem, which do not so decouple, applying globally over the $n$ test examples. However, these constraints only depend on $n$ as an average over examples (which how they arise in dual form in Equation (4.20) of the proof), and the loss function itself is also an average (Equation (4.15)). Putting these together shows how the favorable dependence on $n$ comes about here, making an efficient solution possible to a potentially flagrantly non-convex problem. The technique of optimizing only “halfway into” the dual allows
Table 4.1. Some binary classification losses, as in Sec. 4.2.3. For convenience, we write \( \text{clip}(x) = \min(1, \max(-1, x)) \).

<table>
<thead>
<tr>
<th>Loss</th>
<th>Partial Losses</th>
<th>( \Gamma(g) )</th>
<th>( \Psi(m) )</th>
<th>( g_i(\sigma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>( \ell_-(g) = \frac{1}{2} (1 + g) )</td>
<td>( g )</td>
<td>( \max(1,</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = \frac{1}{2} (1 - g) )</td>
<td>( \ln \left( \frac{e^{-g}}{1 + e^{-g}} \right) )</td>
<td>( \ln(1 + e^m) + \ln(1 + e^{-m}) )</td>
<td>( \frac{1 - e^{-x_i^+ \sigma}}{1 + e^{-x_i^+ \sigma}} )</td>
</tr>
<tr>
<td>Log</td>
<td>( \ell_-(g) = \ln \left( \frac{1}{1 + e^{-g}} \right) )</td>
<td>( \ln \left( \frac{g}{1 + g} \right) )</td>
<td>( \ln(1 + e^m) + \ln(1 + e^{-m}) )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = \ln \left( \frac{1}{1 + e^{-g}} \right) )</td>
<td>( \ln \left( \frac{g}{1 + g} \right) )</td>
<td>( \ln(1 + e^m) + \ln(1 + e^{-m}) )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td>Square</td>
<td>( \ell_-(g) = \left( \frac{1 - x}{1 + x} \right)^2 )</td>
<td>( g )</td>
<td>( \begin{cases} -m &amp; m \leq -1 \ \frac{1}{4} (m^2 + 1) &amp; m \in (-1, 1) \ m &amp; m \geq 1 \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = \left( \frac{1 - x}{1 + x} \right)^2 )</td>
<td>( g )</td>
<td>( \begin{cases} -m &amp; m \leq -1 \ \frac{1}{4} (m^2 + 1) &amp; m \in (-1, 1) \ m &amp; m \geq 1 \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td>CW (param. c)</td>
<td>( \ell_-(g) = c (1 + g) )</td>
<td>( g + 2c - 1 )</td>
<td>( \begin{cases} -m &amp; m \leq 2c - 2 \ (2c - 1)m + 4c(1 - c) &amp; m \in (2c - 2, 2c) \ m &amp; m \geq 2c \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma + 1 - 2c) )</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = (1 - c) (1 - g) )</td>
<td>( \ln \left( \frac{e^i}{1 + e^i} \right) )</td>
<td>( e^i - e^{-i} )</td>
<td>( \text{clip} \left( \ln \left( \frac{1}{2} x_i^+ \sigma + \sqrt{1 + \frac{1}{4} (x_i^+ \sigma)^2} \right) \right) )</td>
</tr>
<tr>
<td>Exponential</td>
<td>( \ell_-(g) = e^x )</td>
<td>( e^x )</td>
<td>( \begin{cases} -m + 2/e &amp; m \leq -e + \frac{2}{e} \ \sqrt{4 + m^2} &amp; m \in (-e + \frac{1}{e}, e - \frac{1}{e}) \ m + 2/e &amp; m \geq e - \frac{1}{e} \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = e^{-x} )</td>
<td>( e^x )</td>
<td>( \begin{cases} -m + 2/e &amp; m \leq -e + \frac{2}{e} \ \sqrt{4 + m^2} &amp; m \in (-e + \frac{1}{e}, e - \frac{1}{e}) \ m + 2/e &amp; m \geq e - \frac{1}{e} \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td>Logistic</td>
<td>( \ell_-(g) = \ln(1 + e^x) )</td>
<td>( g )</td>
<td>( \begin{cases} -m + 2 \ln(1 + 1/e) &amp; m \leq -1 \ \ln(1 + e^m) + \ln(1 + e^{-m}) &amp; m \in (-1, 1) \ m + 2 \ln(1 + 1/e) &amp; m \geq 1 \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = \ln(1 + e^{-x}) )</td>
<td>( g )</td>
<td>( \begin{cases} -m + 2 \ln(1 + 1/e) &amp; m \leq -1 \ \ln(1 + e^m) + \ln(1 + e^{-m}) &amp; m \in (-1, 1) \ m + 2 \ln(1 + 1/e) &amp; m \geq 1 \end{cases} )</td>
<td>( \text{clip}(x_i^- \sigma) )</td>
</tr>
<tr>
<td>Hellinger</td>
<td>( \ell_-(g) = \sqrt{\frac{1 - x}{1 + x}} - \sqrt{\frac{1 + x}{1 - x}} )</td>
<td>( \sqrt{\frac{m}{m + 1 - m^2}} - \sqrt{\frac{1 - m}{m + 1}} )</td>
<td>( \begin{cases} m \leq -1 \ \frac{1 + m}{\sqrt{m + 1}} \frac{1}{2} \end{cases} )</td>
<td>( \begin{cases} \text{sgn}(x_i \sigma) \frac{\sqrt{2 - (x_i \sigma)^2}}{\sqrt{1 + (x_i \sigma)^2}} &amp;</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = 1 - \sqrt{\frac{1 + x}{1 - x}} )</td>
<td>( \sqrt{\frac{m + 1}{m - 1}} - \sqrt{\frac{1 - m}{m + 1}} )</td>
<td>( \begin{cases} m \leq -1 \ \frac{1 + m}{\sqrt{m + 1}} \frac{1}{2} \end{cases} )</td>
<td>( \begin{cases} \text{sgn}(x_i \sigma) \frac{\sqrt{2 - (x_i \sigma)^2}}{\sqrt{1 + (x_i \sigma)^2}} &amp;</td>
</tr>
<tr>
<td>“AdaBoostr”</td>
<td>( \ell_-(g) = \sqrt{\frac{1 - x}{1 + x}} )</td>
<td>( \frac{2e}{\sqrt{1 + e^x}} )</td>
<td>( \left( \frac{\sqrt{m + 4 - m}}{\sqrt{m + 4 - m}} \right)^{1/2} + \left( \frac{\sqrt{m + 4 - m}}{\sqrt{m + 4 - m}} \right)^{1/2} )</td>
<td>( \frac{x_i \sigma}{\sqrt{</td>
</tr>
<tr>
<td></td>
<td>( \ell_+(g) = \sqrt{\frac{1 + x}{1 - x}} )</td>
<td>( \frac{2e}{\sqrt{1 + e^x}} )</td>
<td>( \left( \frac{\sqrt{m + 4 - m}}{\sqrt{m + 4 - m}} \right)^{1/2} + \left( \frac{\sqrt{m + 4 - m}}{\sqrt{m + 4 - m}} \right)^{1/2} )</td>
<td>( \frac{x_i \sigma}{\sqrt{</td>
</tr>
</tbody>
</table>
us to readily manipulate the minimax problem exactly without using an approximation like weak duality, despite the lack of convexity in $g$.

A final remaining bit of structure in the binary classification problem – the linearity of $\ell(z, g)$ in $z$ – is used in Section 4.4 to make a further sweeping generalization of Theorem 4.

The conditions defining “general losses” in Assumption 1 are notable for their generality. Differentiability is convenient, but by no means necessary. We only choose to use it because first-order conditions are a convenient way to establish convexity of the potential well in Lemma 2. It is never used elsewhere, including in the minimax argument of Theorem 4.

Also, we assert that the monotonicity condition on $\ell_\pm$ is natural for a loss function. If it were otherwise, for instance if there were $g_1 < g_2 < g_3$ s.t. $\ell_-(g_1) = \ell_-(g_3) \neq \ell_-(g_2)$, then this would indicate that the loss function is not simply a function of the disparity between labels; in this case perhaps a different loss function is warranted. In technical terms, the minimax manipulations we use to prove Theorem 4 are structured to be valid even if $\ell_\pm$ are non-monotonic; but in this case, $g_j^*$ could turn out to be multi-valued and hence not a true function of the example’s margin. So we consider there to be significant evidence that our assumption on the losses is necessary, but do not prove it here.

As we observed for the 0-1 loss in Chapter 3, the minimax structure’s decoupling of the data for optimization purposes is remarkably powerful, because the original primal problem is jointly over the entire dataset, avoiding further independence or decoupling assumptions.

### 4.3 Related Work and Extensions

A number of comments are in order.

Chapter 3 (cf. [BF15a]) addresses a problem, 0-1 loss minimization, that is well
known to be strongly NP-hard when solved directly over labeled data ([GR09]). It is crucial here to formulate it in the transductive setting, in which the marginal distribution of the unlabeled data is known (reflected in the large amount of unlabeled data that we consider jointly). It gives the dual problem an independently interesting interpretation, so the learning problem is on the always-convex Lagrange dual function and is therefore tractable.

This work generalizes that idea, as the possibly non-convex partial losses are minimized transductively via a straightforward convex optimization. A similar formal technique, including the crucial use of $L_\infty$-norm constraints and averaging over $n \gg p$ to decompose optimization efficiently over many examples, is used implicitly, for a different purpose, in the “drifting” repeated game analysis of boosting ([SF12], Sec. 13.4.1). Existing boosting work ([SF12]) is loosely related to our approach in being a transductive game invoked to analyze ensemble aggregation, but it does not consider unlabeled data and draws its power instead from being a repeated game.

Our transductive formulation involves no surrogates or relaxations of the loss, which we view as a significant advantage over previous work – it allows us to bypass the consistency and agnostic-learning discussions ([Zha04, BJM06]) common to ERM methods that use convex risk minimization. Convergence analyses of such methods make heavy use of convexity of the losses and are generally done presupposing a linear weighting on $h \in \mathcal{H}$ ([TDS15]), whereas in our work such structure emerges directly from Lagrange duality and involves no convexity to derive the worst-case-optimal predictions. Prior work does express the idea that is our explicit conclusion – the learning problem is completely determined by the choice of loss function.

Our assumptions on the loss, particularly condition (C) of Lemma 2, have arisen independently in the setting of online learning in earlier work of Haussler et al. ([HKW98]). This is suggestive of open connections between our techniques and
their, though their learning setting does appear unrelated. We also note with interest that
the same family of general losses was defined by [She05] in the ERM setting (dubbed
“F-losses” there) – in which condition (C) of Lemma 2 also has significance ([She05],
Prop. 2) – but has seemingly not been revisited thereafter.

Indeed, we emphasize that our result on the minimax equilibrium (Theorem 4)
holds for all such general losses – the slack function may not be convex and easy to
optimize unless the further conditions of Lemma 2 are met, but the interpretation of
the optimum decision rule in terms of margins and sigmoid functions remains. All
this emerges from the intrinsic decision-theoretic structure of the problem. Beyond our
related discussion in Section 4.5.1, another basic observation is that the function $g(x^T \sigma)$
is always increasing in $x^T \sigma$ for general losses, because the score function $\Gamma$ is increasing.

This monotonicity typically needs to be assumed in generalized linear models
([MN89]) and related settings, which use such objects for prediction. The score function
can be thought of as analogous to the link function used by GLMs (log loss corresponds
to the scaled and shifted logit link, as seen from Table 4.1), with its inverse being used
for prediction. Further fleshing out the links between our minimax analysis and GLM
learning would be interesting future work.

### 4.3.1 Weighted Test Sets, Covariate Shift, and Label Noise

Though our results here deal with binary classification of a uniformly-weighted
test set, note that our formulation deals with a weighted test set with only a modification
to the slack function:

**Theorem 6.** For any vector $r \geq 0^n$,

$$
\min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n} \frac{1}{n} \sum_{j=1}^{n} r_j \ell(z_j, g_j) = \frac{1}{2} \min_{\sigma \geq 0^n} \left[ -b^T \sigma + \frac{1}{n} \sum_{j=1}^{n} r_j \Psi \left( \frac{x^T \sigma}{r_j} \right) \right]
$$
With $\sigma^*_r$ defined as the minimizer to the above, the optimal predictions $g^* = g(\sigma^*_r)$, as in Theorem 4.

Such weighted classification can be parlayed into algorithms for general supervised learning problems via learning reductions ([BLZ08]). Allowing weights on the test set for the evaluation is tantamount to accounting for known covariate shift in our setting; it would be interesting, though outside our scope, to investigate scenarios with more uncertainty.

**4.3.2 Uniform Convergence Bounds for $b$**

Given $b$ as a lower bound on ensemble classifier losses, the slack function can be efficiently approximately optimized, translating into a worst-case prediction loss bound, as we have seen in Section 4.2.2. But there is typically error in estimating $b$, often quantified by uniform convergence ($L_\infty$) bounds on $b$. If one would like to incorporate the two-sided constraint on $b$, the solution to our game involves the dual ($L_1$) norm of the dual vector $\sigma$.

**Theorem 7.** We have

\[
\min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n, \|Fz-b\|_\infty \leq \epsilon} \ell(z, g) = \min_{\sigma \in \mathbb{R}^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma) + \epsilon \|\sigma\|_1 \right]
\]

Let $\sigma^*_r$ be the minimizer of the right-hand side above. Then the optimal $g^* = g(\sigma^*_r)$, the same function of the optimal weighting as in (4.7).

(This is proved exactly like Theorem 4, but using Lemma 13 instead of Lemma 11 in that proof.) So when the ensemble losses are uniformly bounded, we are now searching over all vectors $\sigma$ (not just nonnegative ones) in an $L_1$-regularized version of the original optimization problem in Theorem 4. To our knowledge, this particular analysis has been
addressed in prior work only for the special case of the entropy objective function on the probability simplex – [DPS04] discusses that special case further.

4.4 Constraints on General Losses for Binary Classification

In the previous sections of this chapter, we allow the evaluation function of the game to be a general loss, but assume that the constraints (our information about the ensemble) are in terms of zero-one loss. This section shows how to relax that assumption, allowing each classifier $h_i$ to constrain the test labels $z$, not with the zero-one loss of $h_i$’s predictions, but rather with some other general loss.

This is possible because when the true labels are binary, all the losses we consider are linear in $z$. This is seen in (4.3), which can be rewritten as:

$$
\ell(z, g) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{2} \left[ \ell_+(g_j) + \ell_-(g_j) \right] - \frac{1}{2n} z^\top [\Gamma(g)].
$$

Accordingly, recall that $h_i \in [-1, 1]^n$ is the vector of test predictions of hypothesis $h_i$. Suppose we have an upper bound on the generalization loss of $h_i$, i.e. $\ell(z, h_i) \leq \epsilon_i$. If we define

$$
b_i^\ell := \frac{1}{n} \sum_{j=1}^{n} \left[ \ell_+(h_i(x_j)) + \ell_-(h_i(x_j)) \right] - 2\epsilon_i
$$

(4.8)

then we can use the definition of $\ell(z, g)$ to write

$$
\ell(z, h_i) \leq \epsilon_i \iff \frac{1}{n} z^\top [\Gamma(h_i)] \geq b_i^\ell
$$

(4.9)

Now (4.9) is a linear constraint on $z$, just like each of the error constraints earlier considered in (3.2). We can derive an aggregation algorithm with constraints like (4.9), using essentially the same analysis as employed in Section 4.2 to solve the game (4.2).

In summary, any classifier can be incorporated into our framework for aggregation
if we have a generalization loss bound on it, where the loss can be any of the losses we have considered. This allows an enormous variety of constraint sets, as each classifier considered can have constraints corresponding to any number of loss bounds on it, including 0 (it can be omitted from the constraint set, providing no information about \( z \)), or \( > 1 \) (it can conceivably have multiple loss bounds using different losses). For instance, \( h_1 \) can yield a constraint corresponding to a zero-one loss bound, \( h_2 \) can yield one constraint corresponding to a square loss bound and another corresponding to a zero-one loss bound, and so on.

### 4.4.1 Matching Objective and Constraint Losses

Despite this generality, we can glean some intuition about the aggregation method for general losses. To do so in the rest of this section, we only consider the case when each classifier contributes exactly one constraint to the problem, and the losses used for these constraints are all the same as each other and as the loss \( \ell \) used in the objective function. In other words, the minimax prediction problem we consider is

\[
V^\ell := \min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n} \ell(z, g) = \min_{g \in [-1,1]^n} \max_{z \in [-1,1]^n} \ell(z, g) \quad (4.10)
\]

The matrix \( F \) and the slack function from earlier this chapter are therefore redefined for the purposes of this section:

\[
F^\ell_{ij} := \Gamma(h_i(x_j)) = \ell_-(h_i(x_j)) - \ell_+(h_i(x_j)) \quad (4.11)
\]

\[
\gamma^\ell(\sigma, b^\ell) := \gamma^\ell(\sigma) := -[b^\ell]^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi \left( [\Gamma(x_j)]^\top \sigma \right) \quad (4.12)
\]
where $b^\ell = (b_1^\ell, \ldots, b_p^\ell)^\top$ and the vectors $x_j$ are now from the new unlabeled data matrix $F_{ij}^\ell$. The game (4.10) is clearly of the same form as the earlier formulation (4.2). Therefore, its solution has the same structure as in Theorem 4, proved using that theorem’s proof:

**Theorem 8.** The minimax value of the game (4.10) is $V := \frac{1}{2} \gamma^\ell(\sigma^{\ell*}) := \min_{\sigma \geq 0} \frac{1}{2} \gamma^\ell(\sigma)$.

The minimax optimal predictions are defined as follows: for all $j \in [n],$

$$g^*_j := g_j(\sigma^{\ell*}) = \begin{cases} -1 & \text{if } [\Gamma(x_j)]^\top \sigma^{\ell*} \leq \Gamma(-1) \\ \Gamma^{-1}( [\Gamma(x_j)]^\top \sigma^{\ell*} ) & \text{if } [\Gamma(x_j)]^\top \sigma^{\ell*} \in (\Gamma(-1), \Gamma(1)) \\ 1 & \text{if } [\Gamma(x_j)]^\top \sigma^{\ell*} \geq \Gamma(1) \end{cases}$$

(4.13)

This provides a concise characterization of how to solve the semi-supervised binary classification game for general losses. Though on the face of it Theorem 8 is a much stronger result than even Theorem 4, we cannot neglect statistical issues. The loss bounds $\epsilon_i^\ell$ on each classifier are estimated using a uniform convergence bound over the ensemble with loss $\ell$, but the data now considered are not the ensemble predictions, but the predictions passed through the score function $\Gamma$. This can be impractical for losses like log loss, for which $\Gamma$ is unbounded, and therefore uniform convergence to estimate $b_i^\ell$ in (4.9) is much looser than for 0-1 loss.

But such issues are outside our scope here, and our constrained minimax results hold in any case. They may be useful to obtain semi-supervised learnability results for different losses from tighter statistical characterizations, which we consider an interesting open problem.
4.4.2 Beating the Best Classifier and the Best Weighted Majority

We re-emphasize that our guarantee here is the strongest possible worst-case guarantee, because our method is the minimax algorithm. This type of statement lacks precedent in this area, so we conclude our discussions by highlighting a few simple corollaries when the loss is general.

In minimizing the slack function over the dual parameters $\sigma$, we perform at least as well as the weighting $\sigma^i \geq 0^p$ that puts weight 1 on $h_i$ and 0 on the remaining classifiers $h_{i'} \neq i$. In other words, our predictor always has the option of simply choosing the best single classifier $i^*$ and guaranteeing its loss bound $\epsilon^{\ell}_{i^*}$. Consequently, our predictor’s loss is always at most that of any single classifier, proving the following observation.

**Proposition 9.** $V^\ell \leq \epsilon^{\ell}_i$ for any classifier $i \in [p]$ and any loss $\ell$.

We provide a short alternative proof of this in Section 4.5.2 using the definitions of this section to better illuminate how they fit together, even though the proposition is evident from the fact that we are minimizing over $\{\sigma : \sigma \geq 0^p\} \ni \sigma^i$.

Our minimax guarantee means that given the ensemble loss constraints $b^\ell$, our algorithm automatically admits superior worst-case loss bounds to any predictor, including any weighted majority vote as well.

**Proposition 10.** $V^\ell$ has a better expected test loss guarantee than any weighted majority vote.

The precise meaning of Proposition 10 should be stressed. It is saying that any weighted majority vote could have a $z$, consistent with the information in the ensemble constraints, that leads to more errors than the error guarantee for $g^*$. It does not imply that the actual performance of $g^*$ will beat every weighted majority, because the constraints could still be unhelpful and the bounds loose.
The empirical results in the following chapters suggest that this and the statistical issues we have discussed in estimating \( \mathbf{b} \) are major considerations in the empirical performance of such algorithms, but can be dealt with to realize significant practical benefits from unlabeled data.

### 4.5 Supporting Results and Proofs

#### 4.5.1 Proof of Theorem 4

The main hurdle here is the constrained maximization over \( \mathbf{z} \). For this we use the following result, a basic application of Lagrange duality.

**Lemma 11.** For any \( \mathbf{a} \in \mathbb{R}^n \),

\[
\max_{\mathbf{z} \in [-1,1]^n, \frac{1}{n} \mathbf{Fz} \geq \mathbf{b}} \, \frac{1}{n} \mathbf{z}^\top \mathbf{a} = \min_{\sigma \geq 0} \left[ -\mathbf{b}^\top \sigma + \frac{1}{n} \left\| \mathbf{F}^\top \sigma + \mathbf{a} \right\|_1 \right]
\]

With this we prove the main result, followed by some comments on the proof.

**Proof of Theorem 4.** First note that \( \ell(\mathbf{z}, \mathbf{g}) \) is linear in \( \mathbf{z} \),

\[
V = (4.3) = \frac{1}{2} \min_{\mathbf{g} \in [-1,1]^n} \max_{\mathbf{z} \in [-1,1]^n, \frac{1}{n} \mathbf{Fz} \geq \mathbf{b}} \frac{1}{n} \sum_{j=1}^n \left[ \ell_+(g_j) + \ell_-(g_j) + z_j \left( \ell_+(g_j) - \ell_-(g_j) \right) \right]
\]

Here we can rewrite the constrained maximization over \( \mathbf{z} \) using Lemma 11:

\[
\max_{\mathbf{z} \in [-1,1]^n, \frac{1}{n} \mathbf{Fz} \geq \mathbf{b}} \frac{1}{n} \sum_{i=1}^n z_i \left( \ell_+(g_i) - \ell_-(g_i) \right) = \max_{\mathbf{z} \in [-1,1]^n, \frac{1}{n} \mathbf{Fz} \geq \mathbf{b}} \frac{1}{n} \mathbf{z}^\top \left[ \Gamma(\mathbf{g}) \right]
\]

\[
= \min_{\sigma \geq 0} \left[ -\mathbf{b}^\top \sigma + \frac{1}{n} \left\| \mathbf{F}^\top \sigma - \Gamma(\mathbf{g}) \right\|_1 \right] \quad \text{(4.14)}
\]
Substituting (4.14) into (4.3) and simplifying,

\[ V = \frac{1}{2} \min_{g \in [-1, 1]^n} \left[ \frac{1}{n} \sum_{j=1}^{n} [\ell_+(g_j) + \ell_-(g_j)] + \max_{\vec{z} \in [-1, 1]^n, \frac{1}{n} F \vec{z} \geq \vec{b}} \frac{1}{n} \sum_{j=1}^{n} z_j (\ell_+(g_j) - \ell_-(g_j)) \right] \]

(4.15)

\[ = \frac{1}{2} \min_{g \in [-1, 1]^n} \left[ \frac{1}{n} \sum_{j=1}^{n} [\ell_+(g_j) + \ell_-(g_j)] \right] + \min_{\sigma \geq 0^p} \left[ -\vec{b}^\top \sigma + \frac{1}{n} \left\| F^\top \sigma - \Gamma(g) \right\|_1 \right] \]

(4.16)

\[ = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -\vec{b}^\top \sigma + \frac{1}{n} \sum_{j=1}^{n} [\ell_+(g_j) + \ell_-(g_j)] \right] + \min_{\sigma \geq 0^p} \left[ \frac{1}{n} \left\| F^\top \sigma - \Gamma(g) \right\|_1 \right] \]

(4.17)

The absolute value breaks down into two cases, so the inner minimization’s objective can be simplified:

\[ \ell_+(g_j) + \ell_-(g_j) + |x_j^\top \sigma - \Gamma(g_j)| = \begin{cases} 
2\ell_+(g_j) + x_j^\top \sigma & \text{if } x_j^\top \sigma \geq \Gamma(g_j) \\
2\ell_-(g_j) - x_j^\top \sigma & \text{if } x_j^\top \sigma < \Gamma(g_j) 
\end{cases} \]

(4.18)

Suppose \( g_j \) falls in the first case, so that \( x_j^\top \sigma \geq \Gamma(g_j) \). From Assumption 1, \( 2\ell_+(g_j) + x_j^\top \sigma \) is decreasing in \( g_j \), so it is minimized for the greatest \( g_j^* \leq 1 \) s.t. \( \Gamma(g_j^*) \leq x_j^\top \sigma \). Since \( \Gamma(\cdot) \) is increasing, exactly one of two subcases holds:

1. \( g_j^* \) is such that \( \Gamma(g_j^*) = x_j^\top \sigma \), in which case the minimand (4.18) is \( \ell_+(g_j^*) + \ell_-(g_j^*) \)

2. \( g_j^* = 1 \) so that \( \Gamma(g_j^*) = \Gamma(1) < x_j^\top \sigma \), in which case the minimand (4.18) is \( 2\ell_+(1) + x_j^\top \sigma \)

A precisely analogous argument holds if \( g_j \) falls in the second case, where \( x_j^\top \sigma < \Gamma(g_j) \). Putting the cases together, we have shown the form of the summand \( \Psi \).
piecewise over its domain, so (4.17) is equal to $\frac{1}{2} \min_{\sigma \geq 0^p} [\gamma(\sigma)]$.

We have proved the dependence of $g^*_j$ on $x\top_j \sigma^*$, where $\sigma^*$ is the minimizer of the outer minimization of (4.17). This completes the proof. \hfill \Box

**Proof of Lemma 11.** We have

\begin{equation}
\frac{1}{n} \max_{z \in [-1,1]^n, Fz \geq nb} -\frac{1}{n} z\top a = \frac{1}{n} \max_{z \in [-1,1]^n} \min_{\sigma \geq 0^p} \left( z\top a + \sigma\top (Fz - nb) \right) \tag{4.19}
\end{equation}

where (a) is by the minimax theorem ([Sio58]).

\begin{equation}
\frac{1}{n} \min_{\sigma \geq 0^p} \max_{z \in [-1,1]^n} \left( z\top (a + F\top \sigma) - nb\top \sigma \right) \tag{4.20}
\end{equation}

\begin{equation}
= \frac{1}{n} \min_{\sigma \geq 0^p} \left( \|a + F\top \sigma\|_1 - nb\top \sigma \right) = \min_{\sigma \geq 0^p} \left( -nb\top \sigma + \frac{1}{n} \|F\top \sigma + a\|_1 \right) \tag{4.21}
\end{equation}

4.5.2 Other Proofs

**Lemma 12.** The function $\ell_+ (\Gamma^{-1}(m)) + \ell_- (\Gamma^{-1}(m))$ is convex for $m \in (\Gamma(-1), \Gamma(1))$ under any of the conditions of Lemma 2.

**Proof of Lemma 12.** Define $F(m) = \ell_+ (\Gamma^{-1}(m)) + \ell_- (\Gamma^{-1}(m))$. By basic properties of the derivative,

\begin{equation}
\frac{d}{dm} \left[ \Gamma^{-1}(m) \right] = \frac{1}{\Gamma'(\Gamma^{-1}(m))} = \frac{1}{\ell'_- (\Gamma^{-1}(m)) - \ell'_+ (\Gamma^{-1}(m))} \geq 0 \tag{4.22}
\end{equation}

where the last inequality follows by Assumption 1. Therefore, by the chain rule and (4.22),

\begin{equation}
F'(m) = \frac{\ell'_- (\Gamma^{-1}(m)) + \ell'_+ (\Gamma^{-1}(m))}{\ell'_- (\Gamma^{-1}(m)) - \ell'_+ (\Gamma^{-1}(m))} \tag{4.23}
\end{equation}
From this, we calculate $F''(m)$, writing $\ell'_\pm (\Gamma^{-1}(m))$ and $\ell''_\pm (\Gamma^{-1}(m))$ as simply $\ell'_\pm$ and $\ell''_\pm$ for clarity:

$$
F''(m) = \frac{d[\Gamma^{-1}]}{dm} \left( \ell'_-(\Gamma^{-1}(m)) - \ell'_+(\Gamma^{-1}(m)) \right)^2 \left[ (\ell'_- - \ell'_+) (\ell''_+ + \ell''_-) - (\ell'_+ + \ell'_-)(\ell''_- - \ell''_+) \right]
$$

From (4.22), observe that the term $(a) = (\ell'_-(\Gamma^{-1}(m)) - \ell'_+(\Gamma^{-1}(m)))^{-3} \geq 0$. Therefore, it suffices to show that the other term is $\geq 0$. But this is equal to

$$
(\ell'_- - \ell'_+) (\ell''_+ + \ell''_-) - (\ell'_+ + \ell'_-)(\ell''_- - \ell''_+) = 2(\ell'_- \ell''_+ - \ell''_- \ell'_+)
$$

This proves that condition (C) of Lemma 2 is sufficient for convexity of $F$ (and indeed necessary also, under Assumption 1 on the partial losses).

We now address the other conditions of Lemma 2. (A) implies (C), because by Assumption 1, $\ell'_-, \ell''_+, \ell''_-$ are $\geq 0$ and $\ell'_+ \leq 0$, so (4.24) is $\geq 0$ as desired.

Finally we prove that (B) implies (C). If $\ell$ is proper, then it is well known (e.g. Thm. 1 of [RW10], and [BSS05]) that for all $x \in (-1, 1)$,

$$
\frac{\ell'_-(x)}{1+x} = -\frac{\ell'_+(x)}{1-x}
$$

(This is a simple and direct consequence of stationary conditions from the properness definition.)

Define the function $w(x) = \frac{\ell'_-(x)}{1+x} = -\frac{\ell'_+(x)}{1-x}$; we drop the argument and write it
and its derivative as $w$ and $w'$ for clarity. By direct computation,

$$\ell_+'' - \ell_+'' = [(1+x)w(w+(x-1)w')] - [(w+(1+x)w')(x-1)w]$$

$$= [(1+x)w^2 + (x^2-1)ww'] - [(x-1)w^2 + (x^2-1)ww'] = 2w^2 \geq 0$$

so (C) is true as desired.

\[\Box\]

Proof of Lemma 2. Continuity follows by checking $\Psi(m)$ at $m = \pm 1$.

For Lipschitzness, note that for $m \in (\Gamma(-1), \Gamma(1))$, by (4.23),

$$\Psi'(m) = \ell'_-(\Gamma^{-1}(m)) + \ell'_+(\Gamma^{-1}(m))$$

$$= -1 + \frac{2\ell'_-(\Gamma^{-1}(m))}{\ell'_-(\Gamma^{-1}(m)) - \ell'_+(\Gamma^{-1}(m))}$$

$$= 1 - \frac{2(-\ell'_+(\Gamma^{-1}(m)))}{\ell'_-(\Gamma^{-1}(m)) - \ell'_+(\Gamma^{-1}(m))}$$

Using Assumption 1 on the partial losses, equations (4.26) and (4.27) respectively make clear that $\Psi'(m) \geq -1$ and $\Psi'(m) \leq 1$ on this interval. Since $\Psi'(m)$ is $-1$ for $m < \Gamma(-1)$ and $1$ for $m > \Gamma(1)$, it is 1-Lipschitz.

As for convexity, since $\Psi$ is linear outside the interval $(\Gamma(-1), \Gamma(1))$, it suffices to show that $\Psi(m)$ is convex inside this interval, which is shown in Lemma 12. \[\Box\]

Lemma 13. For any $a \in \mathbb{R}^n$,

$$\max_{\|Fz-b\|_\infty \leq \epsilon} \frac{1}{n} z^\top a = \min_{\sigma \in \mathbb{R}^p} \left[ -b^\top \sigma + \frac{1}{n} \left\| F^\top \sigma + a \right\|_1 + \epsilon \|\sigma\|_1 \right]$$
Proof.

\[
\max_{\mathbf{z} \in [-1,1]^n, \|\frac{1}{n} \mathbf{Fz} - \mathbf{b}\|_\infty \leq \varepsilon} \frac{1}{n} \mathbf{z}^\top \mathbf{a} = \max_{\mathbf{z} \in [-1,1]^n, \|\frac{1}{n} \mathbf{Fz} - \mathbf{b}\|_\infty \leq \varepsilon} \frac{1}{n} \mathbf{z}^\top \mathbf{a}
\]

\[
= \frac{1}{n} \max_{\mathbf{z} \in [-1,1]^n} \min_{\lambda, \xi \geq 0} \left[ \mathbf{z}^\top \mathbf{a} + \lambda^\top (-\mathbf{Fz} + n \mathbf{b} + n\varepsilon \mathbf{1}^n) + \xi^\top (\mathbf{Fz} - n \mathbf{b} + n\varepsilon \mathbf{1}^n) \right]
\]

\[
= \frac{1}{n} \min_{\lambda, \xi \geq 0} \max_{\mathbf{z} \in [-1,1]^n} \left[ \mathbf{z}^\top (\mathbf{a} + \mathbf{F}^\top (\xi - \lambda)) + n \mathbf{b}^\top (\xi - \lambda) + n\varepsilon \mathbf{1}^n \top (\xi + \lambda) \right]
\]

\[
= \frac{1}{n} \min_{\lambda, \xi \geq 0} \left[ \|\mathbf{a} + \mathbf{F}^\top (\xi - \lambda)\|_1 - n \mathbf{b}^\top (\xi - \lambda) + n\varepsilon \mathbf{1}^n \top (\xi + \lambda) \right]
\]

where the interchanging of min and max is again justified by the minimax theorem ([Sio58]), since the objective is linear in each variable and one of the constraint sets is closed.

Suppose for some \( j \in [n] \) that \( \xi_j > 0 \) and \( \lambda_j > 0 \). Then subtracting \( \min(\xi_j, \lambda_j) \) from both does not affect the value \([\xi - \lambda]_j\), but always decreases \([\xi + \lambda]_j\), and therefore always decreases the objective function. Therefore, we can w.l.o.g. assume that \( \forall j \in [n] : \min(\xi_j, \lambda_j) = 0 \). Defining \( \sigma_j = \xi_j - \lambda_j \) for all \( j \) (so that \( \xi_j = [\sigma_j]_+ \) and \( \lambda_j = [\sigma_j]_- \)), the last equality above becomes

\[
\frac{1}{n} \min_{\lambda, \xi \geq 0} \left[ \|\mathbf{a} + \mathbf{F}^\top (\xi - \lambda)\|_1 - n \mathbf{b}^\top (\xi - \lambda) + n\varepsilon \mathbf{1}^n \top (\xi + \lambda) \right]
\]

\[
= \frac{1}{n} \min_{\sigma \in \mathbb{R}^p} \left[ \|\mathbf{a} + \mathbf{F}^\top \sigma\|_1 - n \mathbf{b}^\top \sigma + n\varepsilon \|\sigma\|_1 \right]
\]

\( \square \)

Proof of Theorem 6. The proof is quite similar to that of Theorem 4, but generalizes it.
First note that we have

\[
\max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} \frac{1}{n} \sum_{i=1}^{n} r_j z_j (\ell_+(g_j) - \ell_-(g_j)) = \max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} -\frac{1}{n} z^T [r \circ \Gamma(g)] = \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \frac{1}{n} \| F^T \sigma - (r \circ \Gamma(g)) \|_1 \right]
\]

(4.28)

where the last equality uses Lemma 11.

Therefore, using (4.28) on the left-hand side of what we wish to prove,

\[
V = \frac{1}{2} \min_{g \in [-1,1]^p} \left[ \frac{1}{n} \sum_{j=1}^{n} r_j [\ell_+(g_j) + \ell_-(g_j)] + \max_{z \in [-1,1]^n, \frac{1}{n} Fz \geq b} \frac{1}{n} \sum_{i=1}^{n} r_j z_j (\ell_+(g_j) - \ell_-(g_j)) \right] = \frac{1}{2} \min_{g \in [-1,1]^p} \left[ \frac{1}{n} \sum_{j=1}^{n} r_j [\ell_+(g_j) + \ell_-(g_j)] + \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \frac{1}{n} \sum_{j=1}^{n} | x_j^T \sigma - r_j \Gamma(g_j) | \right] \right] = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \frac{1}{n} \sum_{j=1}^{n} \min_{g \in [-1,1]^p} \left( r_j [\ell_+(g_j) + \ell_-(g_j)] + | x_j^T \sigma - r_j \Gamma(g_j) | \right) \right]
\]

(4.29)

This is solved by a case analysis essentially identical to the argument used to prove Theorem 4 (given in full in [BF16b]).

\[\square\]

Proof of Prop. 9. Consider the weighting \( \sigma^i \) as above. Then

\[
V^\ell = \frac{1}{2} \min_{\sigma \geq 0^p} \gamma^\ell(\sigma) \leq \frac{1}{2} \gamma^\ell(\sigma^i) = -b^\ell_i + \frac{1}{n} \sum_{j=1}^{n} \Psi(\Gamma(h_i(x_j)))
\]

(4.30)

Since \( h_i(x_j) \in [-1,1] \ \forall j \), we have \( \Gamma(h_i(x_j)) \in [\Gamma(-1), \Gamma(1)] \). Using the definitions of
Ψ and $b^\ell$ (in Equation (4.8)), (4.30) can therefore be rewritten as

$$2V^\ell \leq -b_i^\ell + \frac{1}{n} \sum_{j=1}^{n} \left[ \ell_+ (\Gamma^{-1}(\Gamma(h_i(x_j)))) + \ell_- (\Gamma^{-1}(\Gamma(h_i(x_j)))) \right]$$

$$= -\frac{1}{n} \sum_{j=1}^{n} \left[ \ell_+ (h_i(x_j)) + \ell_- (h_i(x_j)) \right] + 2\varepsilon_i^\ell + \frac{1}{n} \sum_{j=1}^{n} \left[ \ell_+ (h_i(x_j)) + \ell_- (h_i(x_j)) \right] = 2\varepsilon_i^\ell$$

**Acknowledgements**

This chapter contains material to appear as a paper at the conference on Advances in Neural Information Processing Systems, 2016 (“Optimal Binary Classifier Aggregation for General Losses,” Balsubramani and Freund). The dissertation author was the primary investigator and author of this paper.
Chapter 5

Improving Random Forests with Semi-Supervised Specialists

5.1 Introduction

Ensemble-based learning is a very successful approach to learning classifiers, including well-known methods like boosting ([SF12]), bagging ([Bre96]), and random forests ([Bre01]). The power of these methods has been clearly demonstrated in open large-scale learning competitions such as the Netflix Prize ([Kor09]) and the ImageNet Challenge ([RDS+15]). In general, these methods train a large number of “base” classifiers and then combine them using a (possibly weighted) majority vote. By aggregating over classifiers, ensemble methods reduce the variance of the predictions, and sometimes also reduce the bias ([SFBL98]).

The ensemble methods above rely solely on a labeled training set of data. In this chapter, we propose an ensemble method that uses a large unlabeled data set in addition to the labeled set, building on the core formulation of Chapter 3. Our main contributions here are to extend and apply those results with a new algorithm in the context of random forests ([Bre01]) and to perform experiments in which we show that, when the number of labeled examples is small, our algorithm’s performance is at least that of random forests, and often significantly better.
To achieve this, we extend the “muffled” worst-case-optimal framework of Chapter 3 ([BF15a]) to handle *specialists* – ensemble classifiers which only venture to predict on a subset of the data, and abstain from predicting on the rest. Specialists can be very useful in targeting regions of the data on which to precisely suggest a prediction.

The high-level idea of our algorithm is to artificially generate new specialists from the ensemble. We incorporate these, and the targeted information they carry, into the muffled framework. The resulting aggregated predictor inherits the advantages of the framework, which we recapitulate here:

(A) **Efficient**: Learning reduces to solving a scalable convex optimization, and test-time prediction is as efficient and parallelizable as linear prediction.

(B) **Versatile/robust**: No assumptions about the structure or origin of the predictions or labels.

(C) **No introduced parameters**: The aggregation method is completely data-dependent.

(D) **Safe**: Accuracy guaranteed to be at least that of the best classifier in the ensemble (cf. Prop. 9 in Ch. 4).

We develop these ideas in the rest of this chapter, first specifying how to incorporate specialists and the resulting learning algorithm in Section 5.2, then performing an exploratory evaluation of the algorithm on data in Section 5.3.

Though the muffled framework and our extensions to it can be applied to any ensemble of arbitrary origin, in this chapter we focus on random forests, which have been repeatedly demonstrated to have state-of-the-art, robust classification performance in a wide variety of situations ([CKY08]). We use a random forest as a base ensemble whose predictions we aggregate. But unlike conventional random forests, we do not simply
take a majority vote over tree predictions, instead using a unlabeled-data-dependent aggregation strategy dictated by the worst-case framework we employ.

5.2 Learning with Specialists

Recall the notation of Chapter 3, defining the objects of study of our formulation, including the classifier ensemble $\mathcal{H}$, unlabeled data $\mathcal{F}$, and true labels $\mathcal{z}$.

We examine a generalized scenario in which each classifier in the ensemble can abstain on any subset of the examples instead of predicting $\pm 1$. It is a specialist that predicts only over a subset of the input, and we think of its abstain/participate decision being randomized in the same way as the randomized label on each example. In this section, we extend the “muffled” framework of Chapters 3 to arbitrary specialists, and discuss the semi-supervised learning algorithm that results.

In our formulation, suppose that for a classifier $i \in [p]$ and an example $x$, the classifier decides to abstain with probability $1 - v_i(x)$. But if the decision is to participate, the classifier predicts $h_i(x) \in [-1, 1]$ as previously. Our only assumption on $\{v_i(x_1), \ldots, v_i(x_n)\}$ is the reasonable one that $\sum_{j=1}^n v_i(x_j) > 0$, so classifier $i$ is not a useless specialist that abstains everywhere.

The constraint on classifier $i$ is now not on its correlation with $\mathcal{z}$ on the entire test set, but on the average correlation with $\mathcal{z}$ restricted to occasions on which it participates. So for some $[b_S]_i \in [0, 1],$

$$\sum_{j=1}^n \left( \frac{v_i(x_j)}{\sum_{k=1}^n v_i(x_k)} \right) h_i(x_j) z_j \geq [b_S]_i$$

(5.1)

Define $\rho_i(x_j) := \frac{v_i(x_j)}{\sum_{k=1}^n v_i(x_k)}$ (a distribution over $j \in [n]$) for convenience. Now redefine
our unlabeled data matrix as follows:

\[
S = n \begin{pmatrix}
\rho_1(x_1)h_1(x_1) & \rho_1(x_2)h_1(x_2) & \cdots & \rho_1(x_n)h_1(x_n) \\
\vdots & \vdots & \ddots & \vdots \\
\rho_p(x_1)h_p(x_1) & \rho_p(x_2)h_p(x_2) & \cdots & \rho_p(x_n)h_p(x_n)
\end{pmatrix}
\]  

(5.2)

Then the constraints (5.1) can be written as \( \frac{1}{n}Sz \geq b_S \), analogous to Chapter 3’s prediction game’s constraints of \( \frac{1}{n}Fz \geq b \).

To summarize, our specialist ensemble aggregation game is stated as

\[
V_S := \min_{z \in [-1,1]^n} \max_{g \in [-1,1]^n} \frac{1}{n} z^T g \quad \text{subject to} \quad \frac{1}{n}Sz \geq b_S
\]  

(5.3)

We can immediately solve this game from Thm. 4, with \((S,b_S)\) simply taking the place of \((F,b)\).

**Theorem 5 (Solution of the Specialist Aggregation Game).** The *awake ensemble prediction* w.r.t. weighting \( \sigma \geq 0^p \) on example \( x_i \) is \( [S^T \sigma]_i = n \sum_{j=1}^p \rho_j(x_i)h_j(x_i)\sigma_j \). The slack function is now

\[
\gamma_S(\sigma) := -b_S^T \sigma + \frac{1}{n} \sum_{j=1}^n \max \left( \left| [S^T \sigma]_j \right| , 1 \right)
\]  

(5.4)

The minimax value of this game is \( V_S := \frac{1}{2} \gamma_S(\sigma_5^*) := \frac{1}{2} \min_{\sigma \geq 0^p}[\gamma_S(\sigma)] \). The minimax optimal predictions are defined as follows: for all \( i \in [n] \),

\[
[g_S^*]_i \equiv g_S(\sigma_5^*) = \begin{cases} 
[S^T \sigma_5^*]_i & \left| [S^T \sigma_5^*]_i \right| < 1 \\
\text{sgn}\left( [S^T \sigma_5^*]_i \right) & \text{otherwise}
\end{cases}
\]

In the no-specialists case, the vector \( \rho_i \) is the uniform distribution \( \left( \frac{1}{n}, \ldots, \frac{1}{n} \right) \) for
any \( i \in [p] \), and the problem reduces to the original prediction game of Chapter 3. As in the original prediction game, the minimax equilibrium depends on the data only through the ensemble predictions, but these are now of a different form. Each example is now weighted proportionally to \( \rho_j(x_i) \). So on any given example \( x_i \), only hypotheses which participate on it will be counted; and those that specialize the most narrowly, and participate on the fewest other examples, will have more influence on the eventual prediction \( g_i \), ceteris paribus.

### 5.2.1 Creating Specialists for an Algorithm

We can now present the main ensemble aggregation method of this paper, which creates specialists from the ensemble, adding them as additional constraints (rows of \( S \)). The algorithm, HEDGECLIPPER, instantiates our specialist learning framework with a random forest ([Bre01]). As an initial exploration of the framework here, random forests are an appropriate base ensemble because they are known to exhibit state-of-the-art performance ([CKY08]). Their well-known advantages also include scalability, robustness (to corrupt data and parameter choices), and interpretability; each of these benefits is shared by our aggregation algorithm, which consequently inherits them all.

Furthermore, decision trees are a natural fit as the ensemble classifiers because they are inherently hierarchical. Intuitively (and indeed formally too, as in [LJ06]), they act like nearest-neighbor (NN) predictors w.r.t. a distance that is “adaptive” to the data. So each tree in a random forest represents a somewhat different, nonparametric partition of the data space into regions in which one of the labels \( \pm 1 \) dominates. Each such region corresponds exactly to a leaf of the tree.

The idea of HEDGECLIPPER is simply to **consider each leaf in the forest as a specialist**, which predicts only on the data falling into it. By the NN intuition above, these specialists can be viewed as predicting on data that is near them, where the supervised
training of the tree attempts to define the purest possible partitioning of the space. A pure partitioning results in many specialists with \( [b_S]_i \approx 1 \), each of which contributes to the awake ensemble prediction w.r.t. \( \sigma^* \) over its domain, to influence it towards the correct label (inasmuch as \( [b_S]_i \) is high).

Though the idea is complex in concept for a large forest with many arbitrarily overlapping leaves from different trees, it fits the worst-case specialist framework of the previous sections. So the algorithm is still essentially linear learning with convex optimization, as we have described; it is given as Alg. 1.

\[ 
\textbf{Algorithm 1. HEDGECLIPPER} 
\]

1: **Input:** Labeled set \( L \), unlabeled set \( U \)

2: Using \( L \), grow trees \( \mathcal{T} = \{T_1, \ldots, T_p\} \) (regularized; see Sec. 5.2.2)

3: Using \( L \), estimate \( b_S \) on \( \mathcal{T} \) and its leaves

4: Using \( U \), (approximately) optimize (5.4) to estimate \( \sigma^*_S \)

5: **Output:** The estimated weighting \( \sigma^*_S \), for use at test time

### 5.2.2 Discussion

Trees in random forests have thousands of leaves or more in practice. As we are advocating adding so many extra specialists to the ensemble for the optimization, it is natural to ask whether this erodes some of the advantages we have claimed earlier.

Computationally, it does not. When \( \rho_j(x_i) = 0 \), i.e. classifier \( j \) abstains deterministically on \( x_i \), then the value of \( h_j(x_i) \) is irrelevant. So storing \( S \) in a sparse matrix format is natural in our setup, with the accompanying performance gain in computing \( S^\top \sigma \) while learning \( \sigma^* \) and predicting with it. This turns out to be crucial to efficiency – each tree induces a partitioning of the data, so the set of rows corresponding to any tree contains \( n \) nonzero entries *in total*. This is seen in Fig. 5.1.
Figure 5.1. A schematic of how the forest structure is related to the unlabeled data matrix $S$, with a given example $x$ highlighted. The two colors in the matrix represent $\pm 1$ predictions, and white cells abstentions.

Statistically, the situation is more complex. On one hand, there is no danger of overfitting in the traditional sense, regardless of how many specialists are added. Each additional specialist can only shrink the constraint set that the adversary must follow in the game (5.3). It only adds information about $z$, and therefore the performance $V_S$ must improve, if the game is solved exactly.

However, for learning we are only concerned with *approximately* optimizing $\gamma_S(\sigma)$ and solving the game. This presents several statistical challenges. Standard optimization methods do not converge as well in high ambient dimension, even given the structure of our problem. In addition, random forests practically perform best when each tree is grown to overfit. In our case, on any sizable test set, small leaves would cause some entries of $S$ to have large magnitude, $\gg 1$. This can foil an algorithm like HEDGECLIPPER by causing it to vary wildly during the optimization, particularly since
those leaves’ \( b_s \) values are only roughly estimated.

From an optimization perspective, some of these issues can be addressed by e.g. (pseudo)-second-order methods ([DHS11]), whose effect would be interesting to explore in future work. Our implementation opts for another approach – to grow trees constrained to have a nontrivial minimum weight per leaf. Of course, there are many other ways to handle this, including using the tree structure beyond the leaves; we just aim to conduct an exploratory evaluation here, as several of these areas remain ripe for future research.

### 5.3 Experimental Evaluation

We now turn to evaluating HEDGECLIPPER on publicly available datasets. Our implementation uses minibatch SGD with learning rate schedule \( \alpha \propto \frac{1}{\sqrt{t}} \) to optimize (5.3), runs in Python on top of the popular open-source learning package scikit-learn, and runs out-of-core (\( n \)-independent memory), taking advantage of the scalability of our formulation.\(^1\) The datasets are drawn from UCI/LibSVM as well as data mining sites like Kaggle, and no further preprocessing was done on the data. We refer to “Base RF” as the forest of constrained trees from which our implementation draws its specialists. We restrict the training data available to the algorithm, using mostly supervised datasets because these far outnumber medium/large-scale public semi-supervised datasets. Unused labeled examples are combined with the test examples (and the extra unlabeled set, if any is provided) to form the set of unlabeled data used by the algorithm. Further information and discussion on the protocol is in the appendix.

Class-imbalanced and noisy sets are included to demonstrate the aforementioned practical advantages of HEDGECLIPPER. Therefore, AUC is an appropriate measure of performance, and these results are in Table 5.1. Results are averaged over 10 runs, each

\(^1\)It is possible to make this footprint independent of \( d \) as well by hashing features [WDL+09], not done here.
drawing a different random subsample of labeled data. The best results according to a paired t-test are in bold.

We find that the use of unlabeled data is sufficient to achieve improvements over even traditionally overfitted RFs in many cases. Notably, in most cases there is a significant benefit given by unlabeled data in our formulation, as compared to the base RF used. The boosting-type methods also perform fairly well, as we discuss in the next section.

![Figure 5.2](image)

**Figure 5.2.** Class-conditional “awake ensemble prediction” \((\mathbf{x}^\top \sigma^*)\) distributions, on SUSY. Rows (top to bottom): \(\{1K, 10K, 100K\}\) labels. Columns (left to right): \(\alpha = \{1.0, 0.3, 3.0\}\), and the base RF.

The awake ensemble prediction values \(\mathbf{x}^\top \sigma\) on the unlabeled set are a natural way to visualize and explore the operation of the algorithm on the data, in an analogous way to the margin distribution in boosting [SFBL98]. One representative sample is in Fig. 5.2, on SUSY, a dataset with many (5M) examples, roughly evenly split between \(\pm 1\). These plots demonstrate that our algorithm produces much more peaked class-conditional ensemble prediction distributions than random forests, suggesting margin-based learning applications. Changing \(\alpha\) alters the aggressiveness of the clipping, inducing a more or less peaked distribution. The other datasets without dramatic label imbalance show very similar qualitative behavior in these respects, and these plots help choose \(\alpha\) in practice (see appendix).

Toy datasets with extremely low dimension seem to exhibit little to no significant improvement from our method. We believe this is because the distinct feature splits found by the random forest are few in number, and it is the diversity in ensemble predictions
Table 5.1. Area under ROC curve for HEDGECLIPPER vs. supervised ensemble algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># training</th>
<th>HedgeClipper</th>
<th>Random Forest</th>
<th>Base RF</th>
<th>AdaBoost Trees</th>
<th>MART [Fre01]</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>kagg-prot</td>
<td>10</td>
<td>0.567</td>
<td>0.509</td>
<td>0.500</td>
<td>0.520</td>
<td>0.497</td>
<td>0.510</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.714</td>
<td>0.665</td>
<td>0.656</td>
<td>0.681</td>
<td>0.666</td>
<td>0.688</td>
</tr>
<tr>
<td>ssl-text</td>
<td>10</td>
<td>0.586</td>
<td>0.517</td>
<td>0.512</td>
<td>0.556</td>
<td>0.553</td>
<td>0.501</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.765</td>
<td>0.551</td>
<td>0.542</td>
<td>0.596</td>
<td>0.569</td>
<td>0.552</td>
</tr>
<tr>
<td>kagg-cred</td>
<td>100</td>
<td>0.558</td>
<td>0.518</td>
<td>0.501</td>
<td>0.528</td>
<td>0.542</td>
<td>0.502</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.602</td>
<td>0.534</td>
<td>0.510</td>
<td>0.585</td>
<td>0.565</td>
<td>0.505</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.603</td>
<td>0.563</td>
<td>0.535</td>
<td>0.586</td>
<td>0.566</td>
<td>0.510</td>
</tr>
<tr>
<td>a1a</td>
<td>100</td>
<td>0.779</td>
<td>0.619</td>
<td>0.525</td>
<td>0.680</td>
<td>0.682</td>
<td>0.725</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.808</td>
<td>0.714</td>
<td>0.655</td>
<td>0.734</td>
<td>0.722</td>
<td>0.768</td>
</tr>
<tr>
<td>w1a</td>
<td>100</td>
<td>0.543</td>
<td>0.510</td>
<td>0.505</td>
<td>0.502</td>
<td>0.513</td>
<td>0.509</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.651</td>
<td>0.592</td>
<td>0.520</td>
<td>0.695</td>
<td>0.689</td>
<td>0.671</td>
</tr>
<tr>
<td>covtype</td>
<td>100</td>
<td>0.735</td>
<td>0.703</td>
<td>0.661</td>
<td>0.709</td>
<td>0.732</td>
<td>0.515</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.764</td>
<td>0.761</td>
<td>0.715</td>
<td>0.730</td>
<td>0.761</td>
<td>0.524</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.809</td>
<td>0.822</td>
<td>0.785</td>
<td>0.759</td>
<td>0.788</td>
<td>0.515</td>
</tr>
<tr>
<td>ssl-secstr</td>
<td>10</td>
<td>0.572</td>
<td>0.574</td>
<td>0.535</td>
<td>0.563</td>
<td>0.557</td>
<td>0.555</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.656</td>
<td>0.645</td>
<td>0.610</td>
<td>0.643</td>
<td>0.637</td>
<td>0.629</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.687</td>
<td>0.682</td>
<td>0.646</td>
<td>0.690</td>
<td>0.689</td>
<td>0.683</td>
</tr>
<tr>
<td>SUSY</td>
<td>1K</td>
<td>0.776</td>
<td>0.769</td>
<td>0.764</td>
<td>0.747</td>
<td>0.771</td>
<td>0.720</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.785</td>
<td>0.787</td>
<td>0.784</td>
<td>0.787</td>
<td>0.789</td>
<td>0.759</td>
</tr>
<tr>
<td></td>
<td>100K</td>
<td>0.799</td>
<td>0.797</td>
<td>0.797</td>
<td>0.797</td>
<td>0.796</td>
<td>0.779</td>
</tr>
<tr>
<td>epsilon</td>
<td>1K</td>
<td>0.651</td>
<td>0.659</td>
<td>0.645</td>
<td>0.718</td>
<td>0.726</td>
<td>0.774</td>
</tr>
<tr>
<td>webspam-uni</td>
<td>1K</td>
<td>0.936</td>
<td>0.928</td>
<td>0.920</td>
<td>0.923</td>
<td>0.928</td>
<td>0.840</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.967</td>
<td>0.970</td>
<td>0.957</td>
<td>0.945</td>
<td>0.953</td>
<td>0.901</td>
</tr>
</tbody>
</table>

that enables HEDGECLIPPER to clip (weighted majority vote) dramatically and achieve its performance gains.

On the other hand, given a large quantity of data, our algorithm is able to learn significant structure; the minimax structure appears appreciably close to reality, as evinced by the results on large datasets.

5.4 Related Work

This paper’s framework and algorithms are superficially reminiscent of boosting, another paradigm that uses voting behavior to aggregate an ensemble and has a game-theoretic intuition ([SF12, FS96]). There is some work on semi-supervised versions of boosting ([KMJJL09]), but it departs from this principled structure and has little in common with our approach. Classical boosting algorithms like AdaBoost ([FS97]) make
no attempt to use unlabeled data. It is an interesting problem to incorporate boosting ideas into our formulation, particularly since the two boosting-type methods acquitted themselves well in the results of this chapter, and can pack information parsimoniously into many fewer ensemble classifiers than random forests. We explore this in Chapter 6.

There is a long-recognized connection between transductive and semi-supervised learning, and our method bridges these two settings. Popular variants on supervised learning such as the transductive SVM ([Joa99]) and graph-based or nearest-neighbor algorithms, which dominate the semi-supervised literature ([ZG09]), have shown promise largely in data-poor regimes because they face major scalability challenges. Our focus on ensemble aggregation instead allows us to keep a computationally inexpensive linear formulation and avoid considering the underlying feature space of the data. Largely unsupervised ensemble methods have been explored especially in applications like crowdsourcing, in which the method of ([DS79]) gave rise to a plethora of Bayesian methods under various conditional independence generative assumptions on $\mathbf{F}$ ([PSNK14]). Using tree structure to construct new features has been applied successfully, though without guarantees ([MTJ07]).

Learning with specialists has been studied in an adversarial online setting as in the work of [FSSW97]. Though that paper’s setting and focus is different from ours, the optimal algorithms it derives also depend on each specialist’s average error on the examples on which it is awake.

5.5 Additional Information on Experiments

5.5.1 Datasets

All data from the challenges (e.g. kagg-cred) lacked test labels, so the results reported are averaged over 10 random splits of the training data. Information on the
Table 5.2. Datasets used for HEDGECLIPPER experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data sizes $(m/n)$</th>
<th>Dim. $d$</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>kagg-prot</td>
<td>3751 / 1776</td>
<td>1776</td>
<td>Kaggle challenge [kag12]</td>
</tr>
<tr>
<td>ssl-text</td>
<td>1500 / 11960</td>
<td>11960</td>
<td>[CSZ06]</td>
</tr>
<tr>
<td>kagg-cred</td>
<td>150000 / 10</td>
<td>10</td>
<td>Kaggle challenge [kag11]</td>
</tr>
<tr>
<td>a1a</td>
<td>1605 / 30956</td>
<td>123</td>
<td>LibSVM</td>
</tr>
<tr>
<td>w1a</td>
<td>2477 / 47272</td>
<td>300</td>
<td>LibSVM</td>
</tr>
<tr>
<td>covtype</td>
<td>581012 / 54</td>
<td>54</td>
<td>LibSVM</td>
</tr>
<tr>
<td>ssl-secstr</td>
<td>83679 (unlabeled:1189472)</td>
<td>315</td>
<td>[CSZ06]</td>
</tr>
<tr>
<td>SUSY</td>
<td>5000000 / 18</td>
<td>18</td>
<td>UCI</td>
</tr>
<tr>
<td>HIGGS</td>
<td>11000000 / 28</td>
<td>28</td>
<td>UCI</td>
</tr>
<tr>
<td>epsilon</td>
<td>500000 / 2000</td>
<td>2000</td>
<td>PASCAL Large Scale Learning Challenge 2008</td>
</tr>
<tr>
<td>webspm-uni</td>
<td>350000 / 254</td>
<td>254</td>
<td>LibSVM</td>
</tr>
</tbody>
</table>

datasets used:

5.5.2 Algorithms

In all cases, the random forests were grown with default parameters for the feature and data splits (bootstrap data sample of input data size, and $\sim \sqrt{d}$ features considered per split), and 100 trees was standard. Varying these changes the induced diversity of the trees/partitions and may fundamentally affect the output of our algorithm, but exploration of such aspects is left to future work. All the comparator algorithms were also run with scikit-learn’s default parameters – in many cases like RFs, they are fairly insensitive to parameter choice.

To overcome the statistical issues discussed in Sec. 5.2.2, we found we needed to enforce some regularization on the tree used. We chose to impose a constraint on the minimum number of training examples in any leaf of the tree. This constraint was imposed as a parameter to grow the forest; thereafter, we could use all resulting leaves as
specialists. To avoid any leaf specialist weights being too large but still collect as many leaf specialists as possible, we set the minimum number of examples per leaf to 10 with $\geq 1K$ labeled examples, and to 4 otherwise.

We also tried an alternative way of avoiding small specialists: to simply grow an unregularized forest and then filter out leaves, selecting only large enough leaves as specialists. This generally performed comparably or worse, consistent with the intuition that the diversity in unregularized tree predictions often manifests largely on small leaves.

As pointed out in the paper, estimating $b$ is an important step in an implementation. Accordingly, we used a bootstrap sample to do so; this performed comparably to holding out a validation set from a constant fraction of the labeled data. We observe throughout that it seems to matter far less how well the ensemble is trained, and more how well $b$ is estimated; so we elected to only keep a modicum of labeled data for actually training the ensemble, and most for estimation.

A notable issue we encountered is the setting of the “noise” rescaling factor $\alpha$. We found MARVIN to be relatively insensitive to the precise choice of $\alpha$, so it essentially sufficed for our experiments to try three choices: $\{0.3, 1.0, 3.0\}$. The last is $> 1.0$, and therefore does not have an interpretation in terms of uniform label noise, but it is certainly a valid computational tool.

Which of these three $\alpha$s to choose? Generally, we found that choosing $\alpha = 0.3$ does not hurt performance, because our performance goals are often met best by separating the class-conditional peaks as much as possible. This is dangerous for more class-unbalanced datasets like kagg-cred, however, for which the default $\alpha = 1.0$ works best. A useful heuristic which we used to choose $\alpha$ is simply to look at the class-conditional awake ensemble prediction distribution as plotted in Fig. 5.2; the distribution can be roughly estimated and plotted on the fly, and we can quickly ascertain sensible choices of parameters like $\alpha$. These choices appear to matter less at larger scale.
Acknowledgements

This chapter contains material from the conference on Advances in Neural Information Processing Systems, 2015 (“Scalable Semi-Supervised Classifier Aggregation,” Balsubramani and Freund). The dissertation author was the primary investigator and author of this paper.
Chapter 6

Muffled Incremental Combination of Classifiers

6.1 Introduction

The *boosting* approach to learning binary classifiers is to construct a weighted-majority ensemble of them by incrementally adding base classifiers ([SF12]). This process is guided by a potential function that is an upper bound on the training error. The weights assigned to the training examples communicate the gradient of the potential function to the base learner. Like other supervised learning algorithms, boosting algorithms require a sufficiently large labeled training set in order to produce an accurate classifier. Such algorithms do not make use of unlabeled data, which is typically much more abundant.

On the other hand, semi-supervised learning approaches ([CSZ06]) attempt to use both labeled and unlabeled training examples. The basic idea in many approaches is to augment the labeled set by inferring the label of unlabeled examples from their labeled neighbors in some way. Such inference uses “hallucinated labels” on the unlabeled data that tend to agree with the labeled ones (e.g. [BNS06]).

In this chapter, we explore a semi-supervised learning approach that uses the opposite strategy. Instead of using the unlabeled examples to augment the guidance of
the labeled examples, we use the unlabeled examples to \textit{muffle} the effect of the labeled examples, and hallucinate labels which tend to oppose the labeled ones.

In this chapter, we devise algorithms which build empirically on the semi-supervised “muffled” formulation we have developed in Chapter 3 and subsequent chapters. The intuition behind the basic algorithm for 0-1 loss in Chapter 3 is that the aggregated prediction on unlabeled examples should be between $-1$ and $+1$. The average potential $\Psi(x) = \max(|x|, 1)$ of the unlabeled data regularizes the problem by encouraging us to put weight on classifiers which disagree, so that the margin stays low. (Plotted earlier in Fig. 3.1, we reproduce the potential well $\Psi$ as Fig. 6.1 for easy reference.)

The “muffling” behavior occurs while minimizing the slack function when the aggregate prediction (score) on an unlabeled example is outside $[-1, 1]$, because moving its score towards this interval tends to lower the slack function. As we will explore, muffling can be represented by assigning to the unlabeled example a hallucinated label that is the opposite of the predicted label. Thm. 4 provides a direct proof that this strategy generalizes well by directly addressing test error, even though it contrasts starkly with \textit{max-margin} approaches known to generalize in fully supervised settings ([SFBL98]).

We refer to the minimax framework of this and previous chapters as \textit{muffled learning} to emphasize its learning principle of actively distrusting overly confident
predictions, counteracting the guidance of the labeled examples. As a consequence of Thm. 4, it always performs at least as well as any single classifier when \( b \) is estimated accurately – in other words, unlabeled data do not hurt. We follow our previous work ([BF15a, BF15b]) in re-emphasizing that while the transductive setting is convenient for a clean muffled formulation, in this case it is not a restrictive assumption for high \( n \), since the data are i.i.d. (see Sec. 6.9 for details).

For the remainder of this chapter, we investigate ways to minimize the slack function generally (for any ensemble) and practically. All of these are algorithms to find a weight vector \( \sigma \) that leads to a good predictor, and to learn the associated ensemble. So our algorithms always inherit the aforementioned prediction-based advantages of the muffled learning framework.

We use this muffling principle to devise several simple algorithms, notably a sequential scheme (MARVIN) that incorporates ensemble classifiers one at a time into an aggregated classifier. At every step, it chooses a new classifier that tends to disagree with the current majority opinion on examples in the unlabeled set.

The rest of the chapter is organized as follows. In Section 6.2 we describe the algorithm MARVIN. In Section 6.3 we describe the algorithm HEDGEMOWER, a way of exploiting the partitionings used by ensemble classifiers like decision trees. Section 6.4 contains a comparative experimental evaluation of our algorithm on a number of datasets. In Section 15.3 we draw some conclusions from the experiments, and in Section 15.3.2 we make connections to past and future work.

## 6.2 An Algorithm for Incrementally Aggregating Classifiers

Directly minimizing the slack function with an ensemble generated a priori like a random forest can enjoy some practical success, but it can be too conservative (Chapter
5, see [BF15b]), because the bound \( V \) on error is too loose. However, adding more classifiers to any ensemble can only lower its \( V \), because \( z \) is at least as constrained after the addition. Therefore, a natural strategy to mitigate the bound’s looseness is to call upon a larger ensemble.

So we elect to build our predictor incrementally, from classifiers in a very large ensemble \( \mathcal{H} \). Supervised boosting algorithms have long ([Fre95]) done this efficiently by accessing a learning algorithm that the booster calls as needed on a carefully constructed input dataset, to generate classifiers one at a time. We use a similar learning oracle as a subroutine – it returns a classifier from \( \mathcal{H} \) that approximately minimizes error on its input data among \( h \in \mathcal{H} \). This is efficiently implemented for many hypothesis classes, like decision trees, linear classifiers, and other supervised learning approaches – our method is capable of using any of these.

So in this chapter, we modify the setting of Chapter 5 (from Chapter 3) slightly, so that we have two types of data drawn i.i.d. from the same distribution: a labeled set \( L = \{ (x^L_1, y^L_1), \ldots, (x^L_m, y^L_m) \} \) and an unlabeled set \( U = \{ x^U_1, \ldots, x^U_n \} \), with \( |L| = m \) and \( |U| = n \). We have at our disposal an ensemble \( \mathcal{H} \) of classifiers, which may be very large, using the learning oracle described earlier.

Our algorithm, Marvin, repeatedly requests the classifier in \( \mathcal{H} \) that minimizes error on inputs comprised of \( m + n \) weighted examples: the \( m \) labeled examples in \( L \), and the \( n \) unlabeled examples in \( U \) with purposefully hallucinated labels that change every iteration. \(^1\) This means that the effective dimensionality of our weight vector (and therefore of every example \( x \)) is \( t \) on iteration \( t \), rather than being a fixed value \( p \). So we abuse notation somewhat by writing the score of example \( x \) with weight vector \( \sigma \) as \( \langle x, \sigma \rangle \), where the inner product defined depends on the iteration number.

\(^1\) The actual algorithm run in our experiments is a minibatch out-of-core version adapted slightly for the stochastic setting (Sec. 6.9).
Algorithm 2. MARVIN

Input: Labeled set $L = \{(x^L_1, y^L_1), \ldots, (x^L_m, y^L_m)\}$, unlabeled set $U = \{x^U_1, \ldots, x^U_n\}$

Initialize weights: $\sigma^0 = 0$, so that $\langle x, \sigma^0 \rangle = 0$ for all $x \in U$

for $t = 1$ to $T$

Hallucinate label $\tilde{y}^t_j$ for each $x^U_j \in U$:

$$\tilde{y}^t_j = -\sgn(\langle x^U_j, \sigma^{t-1} \rangle) \cdot 1(\langle x^U_j, \sigma^{t-1} \rangle \geq 1)$$

Find a classifier $h^t \in H$ that approximately minimizes weighted error over combined data:

$$h^t = \arg \max_{h \in H} \left[ \frac{1}{m} \sum_{i=1}^{m} y^L_i h(x^L_i) + \frac{1}{n} \sum_{j=1}^{n} \tilde{y}^t_j h(x^U_j) \right]$$

Add $h^t$ to predictor with weight found by line search, forming $\sigma^t$ from $\sigma^{t-1}$.

For MARVIN-C only: Total correction – minimize the slack function over the ensemble so far: $h^1, \ldots, h^t$

For MARVIN-D only: If $h^t$ is a decision tree, add all internal nodes of $h^t$ too and then perform total correction (see Sec. 6.3).

end for

Output: Predictor $g^T(x) = \text{clip}(\langle x, \sigma^T \rangle) := \max(-1, \min(\langle x, \sigma^T \rangle, 1))$
**MARVIN** is straightforward to specify (Alg. 2), with no parameters to tune. It ignores all currently hedged unlabeled examples because they are already minimizing $\Psi$, and sends every clipped unlabeled example to the error-minimizing oracle with a hallucinated label of the minority prediction, to encourage its margin towards zero. Labeled examples are sent to the oracle unchanged, and the data are weighted so that $L$ and $U$ have equal weights when no unlabeled data are hedged (see Eq. (6.1)).

The **MARVIN** update can be seen as greedy coordinate descent on the slack function in the high-dimensional space spanned by $\mathcal{H}$ (Sec. 6.7). This dimensionality is a thorny theoretical issue, so that even though the slack function is convex, a practical step size schedule is not easily understood using optimization-based analysis of coordinate descent. Even seeming niceties like proportionality constants are of great importance in ensuring that the method converges quickly, and a good choice depends on the interactions between dimensions (ensemble classifiers) in complex ways. All these considerations motivate us to use line search in this chapter to find the appropriate step size, replacing the schedule $\propto \frac{1}{\sqrt{t}}$ used in Chapter 5 ([BF15a]). This is crucial to achieving quicker convergence and enabling our total correction results (details in Sec. 6.3.1).

Another way to improve performance that we experiment with follows the example of the *totally corrective* technique for supervised boosting ([WLR06]). After adding each new ensemble classifier, this approach minimizes the objective function over the entire cumulative ensemble so far. It is especially appropriate in our case because the slack function is convex, so such a method will make progress. We return to this idea in Sec. 6.4, where we implement **MARVIN-C**, the totally corrective version of **MARVIN**.
6.3 Maximizing the Performance of an Ensemble of Trees

We now address the vanilla non-sequential setting, as experimented with in Chapter 5. In that work (paper at [BF15b]), we augmented the ensemble with specialists constructed from the leaves of the trees each of which predicts only on the data falling into it, contributing local information about the true labels. Recall that this algorithm, called HEDGECLIPPER, optimizes the slack function using standard stochastic gradient descent with a step size schedule $\propto t^{-1/2}$.

6.3.1 Line Search

We improve the random forest aggregation method HEDGECLIPPER in several ways. One is to use line search. This is motivated by the reasons described in the previous section, but there is an even more important reason to use line search in the specialist setting – the slack function is scaled very differently along different dimensions, making a fixed step size schedule even less effective because of high variance in the direction being chosen.

In this and later sections of the chapter, we use a modified (memoized) version of golden section search for our line search, which is crucial to the performance of our algorithms. It is described well in the original paper ([Kie53]) and commonly in textbooks; we use the version given in the scipy.optimize Python package.

6.3.2 Wilson’s Score Interval for Estimating $b$

We also make another improvement to HEDGECLIPPER, extending the idea to handle all internal nodes, not just the potentially prohibitively small-support leaves. So we calculate the components of $b$, i.e. the errors of all the specialists representing internal nodes, simultaneously. These errors are close to their estimates from the labeled data with
We now specify how each specialist’s bound is computed. For each node, we are estimating a binomial proportion, say $p$, using an estimate $\hat{p}$ computed with $m$ samples. The natural option for this is the Wald confidence interval with width $\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$. However, this fails to provide adequate coverage in two regimes of interest in decision tree partitionings: a small number of labeled data falling into a leaf, and very skewed leaves ($\hat{p} \approx 0$ or 1).

To maintain coverage of our interval in these situations, we calculate $b$ using the lower bound provided by Wilson’s score interval ([Wil27]; also [LR06a], Example 11.2.7) for each node of the tree. This follows accepted practice for estimating $p$ in the aftermentioned regimes of interest ([BCD01]). Figure 6.2 depicts the effect of using Wilson’s interval, even on small pure leaves – it implements nonuniform shrinkage of all errors towards $1/2$ to ensure good coverage, in keeping with the conservatism of
muffling. The only parameter here is the confidence level (i.e. the allowed probability of failure) – a higher such probability makes the prediction more aggressive, resulting in most of the internal nodes getting ”mowed down” to a Wilson lower-bound of 0, as seen in Figure 6.2. We call the resulting algorithm HEDGMOWER, minimizing the slack function over the random forest trees combined with all their internal nodes. See Sec. 6.8 for a full specification of the algorithm (Alg. 3) and of Wilson’s score interval.

6.3.3 Discussion

We find that line search and Wilson’s interval are crucially important to our empirical performance. Line search results in significant improvements over HEDGECLIPPER’s SGD with a stepsize schedule, even without any additional specialist nodes. To highlight this, in addition to HEDGMOWER we implement a simplified algorithm we call HEDGMOWER-1, which simply minimizes the slack function using the complete random forest trees, without further augmenting the ensemble with any specialists.

We are adding specialists of many various sizes representing variation at many scales, represented by a different scaling factor for each column of F (recall the specialist setup Sec. 5.2), so that dimensions are “unnormalized” by design. An open problem of Chapter 5 ([BF15b]) is to use second-order or other convex optimization methods, to continue to make progress in the optimization despite such multiscale issues. This is still unexplored as written, but we believe our approach of first-order line search works satisfactorily. Fig. 6.3 shows the effect of HEDGMOWER and HEDGMOWER-1 for a couple of datasets; the muffling effect of Ψ (recall Fig. 6.1) is readily apparent, particularly when specialist knowledge is incorporated (HEDGMOWER, right column of Fig. 6.3).
Figure 6.3. Score distributions of four different classifiers in the columns from left to right – respectively logistic regression, random forests, HEDGEMOWER-1, and HEDGEMOWER. Top row: covtype dataset. Bottom row: ssl-secstr. Blue dashed lines are at scores of ±1, the inflection points of $\Psi$.

6.4 Empirical Results

We now turn to implementing MARVIN and HEDGEMOWER on a variety of datasets. We summarize the algorithms we implement; the first four have each been described in previous sections, and MARVIN-D combines the two ideas.

- **HEDGEMOWER-1** – Minimize slack function using just the non-specialist trees of a random forest (RF), and none of their specialists.

- **HEDGEMOWER** – Add all internal nodes of RF trees as specialists, minimize slack function.

- **MARVIN** – Add one (non-specialist) decision tree at a time.

- **MARVIN-C** – Like MARVIN, but with total correction each timestep.

- **MARVIN-D** – Similar to MARVIN-C (with total correction), but adding a (non-specialist) tree and its internal nodes each timestep, like HEDGEMOWER.

Our implementations of our new algorithms are in Python, and use the open-source package scikit-learn. We restrict the labeled data available to the algorithm by
various orders of magnitude when feasible, to explore its effect. Unused labeled examples are combined with the test examples (and the extra unlabeled set, if any is provided) to form the set of unlabeled data used by the algorithm. All algorithms are run with 100 base unregularized decision trees as ensemble classifiers where applicable. Class-imbalanced and noisy datasets are included, so that AUC is an appropriate measure of performance. Results and 95% confidence intervals are given from 20 Monte Carlo trials, each starting with a different random subsample of labeled data. Further information\(^2\) on the sources of the datasets can be found in Table 6.2.

We compare our algorithms’ performance to that of standard supervised ensemble algorithms under the same conditions – AdaBoost and LogitBoost, random forests (100 trees, default parameters as in Chapter 5) as a high-performance supervised ensemble algorithm, and logistic regression.

We find that one or more of our new algorithms is sufficient to achieve significant improvements over the baselines in all cases. The results are in Table 6.1, further discussed in Sec. 15.3.

Many of our datasets are large enough that \( U \) will not fit in memory, making the batch boosting method impractical. However, there is a fairly straightforward minibatch remedy: store only a fixed-size minibatch of unlabeled examples, and periodically replace this batch (or similar, e.g. in a streaming setting, replace a randomly selected example in the batch with each new example that arrives). This is explained in Sec. 6.9.

The only parameter adjustment done for the new algorithms is of the failure probability used in calculating the Wilson bounds (details in Sec. 6.8). This applies to all our new algorithms, because all use line search, which requires \( b \) (just one component at a time for MARVIN, and many at once for the other four algorithms). The situation is particularly complex for MARVIN-D, which needs enough labeled data to estimate \( b \)

\(^2\)Our code is available at https://github.com/aikanor/marvin.
Table 6.1. Area under ROC curve for MARVIN and HEDGE MOWER variants, and supervised ensemble algorithms, all run with/to 100 ensemble classifiers unless otherwise stated. All MARVIN variants are run with unregularized decision tree weak learners. The "# relevant nodes..." column refers to the average number of internal nodes not mowed down by Wilson's interval when running the "HEDGE MOWER" column, as a fraction of the total number of internal nodes. 95% confidence intervals indicated using 20 Monte Carlo trials, with results in bold being for the "best" algorithm for each row (achieving the highest lower bound), along with any others with confidence intervals overlapping it.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># labeled</th>
<th>HEDGE-MOWER-1</th>
<th>HEDGE-MOWER</th>
<th># relevant nodes / # total nodes per use</th>
<th>MARVIN</th>
<th>MARVIN-C</th>
<th>MARVIN-D</th>
<th>RF</th>
<th>Ada-Boost</th>
<th>LogitBoost</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>kagg-prot</td>
<td>100</td>
<td>0.58 ± 0.02</td>
<td>0.66 ± 0.00</td>
<td>2.7 ± 0.00</td>
<td>0.25 ± 0.00</td>
<td>0.96 ± 0.06</td>
<td>0.85 ± 0.12</td>
<td>0.64 ± 0.04</td>
<td>0.65 ± 0.05</td>
<td>0.66 ± 0.06</td>
<td>0.64 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.57 ± 0.05</td>
<td>0.76 ± 0.05</td>
<td>2.7 ± 0.00</td>
<td>0.30 ± 0.02</td>
<td>0.68 ± 0.07</td>
<td>0.84 ± 0.03</td>
<td>0.64 ± 0.04</td>
<td>0.65 ± 0.05</td>
<td>0.66 ± 0.06</td>
<td>0.64 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>2500</td>
<td>0.82 ± 0.03</td>
<td>0.58 ± 0.02</td>
<td>61.2 ± 0.03</td>
<td>0.30 ± 0.04</td>
<td>0.89 ± 0.05</td>
<td>0.58 ± 0.04</td>
<td>0.80 ± 0.00</td>
<td>0.79 ± 0.06</td>
<td>0.75 ± 0.04</td>
<td>0.95 ± 0.00</td>
</tr>
<tr>
<td>covtype</td>
<td>1K</td>
<td>0.84 ± 0.02</td>
<td>0.82 ± 0.02</td>
<td>21.1 ± 0.02</td>
<td>0.84 ± 0.05</td>
<td>0.88 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.72 ± 0.03</td>
<td>0.62 ± 0.04</td>
<td>0.71 ± 0.04</td>
<td>0.71 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.81 ± 0.03</td>
<td>0.87 ± 0.03</td>
<td>25.4 ± 0.03</td>
<td>0.84 ± 0.05</td>
<td>0.88 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.72 ± 0.04</td>
<td>0.62 ± 0.04</td>
<td>0.71 ± 0.04</td>
<td>0.71 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.94 ± 0.04</td>
<td>0.89 ± 0.05</td>
<td>190.1 ± 0.05</td>
<td>0.86 ± 0.05</td>
<td>0.87 ± 0.05</td>
<td>0.86 ± 0.05</td>
<td>0.71 ± 0.04</td>
<td>0.62 ± 0.04</td>
<td>0.71 ± 0.04</td>
<td>0.71 ± 0.03</td>
</tr>
<tr>
<td>ssl-secrets</td>
<td>100</td>
<td>0.58 ± 0.06</td>
<td>0.55 ± 0.11</td>
<td>2.1 ± 0.06</td>
<td>0.30 ± 0.03</td>
<td>0.64 ± 0.06</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>1K</td>
<td>0.70 ± 0.04</td>
<td>0.68 ± 0.09</td>
<td>14.7 ± 0.04</td>
<td>0.64 ± 0.04</td>
<td>0.62 ± 0.04</td>
<td>0.64 ± 0.04</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.74 ± 0.06</td>
<td>0.71 ± 0.06</td>
<td>17.7 ± 0.06</td>
<td>0.68 ± 0.04</td>
<td>0.62 ± 0.04</td>
<td>0.64 ± 0.04</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
<td>0.55 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.78 ± 0.08</td>
<td>0.81 ± 0.03</td>
<td>4.1 ± 0.03</td>
<td>0.74 ± 0.02</td>
<td>0.66 ± 0.04</td>
<td>0.68 ± 0.04</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
</tr>
<tr>
<td>adult</td>
<td>1K</td>
<td>0.58 ± 0.01</td>
<td>0.85 ± 0.02</td>
<td>29.1 ± 0.04</td>
<td>0.74 ± 0.03</td>
<td>0.72 ± 0.04</td>
<td>0.74 ± 0.02</td>
<td>0.72 ± 0.04</td>
<td>0.72 ± 0.04</td>
<td>0.72 ± 0.04</td>
<td>0.72 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.68 ± 0.05</td>
<td>0.65 ± 0.09</td>
<td>5.2 ± 0.04</td>
<td>0.73 ± 0.08</td>
<td>0.69 ± 0.08</td>
<td>0.69 ± 0.08</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
<td>0.66 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.74 ± 0.03</td>
<td>0.75 ± 0.04</td>
<td>17.4 ± 0.04</td>
<td>0.86 ± 0.06</td>
<td>0.87 ± 0.05</td>
<td>0.87 ± 0.04</td>
<td>0.87 ± 0.04</td>
<td>0.87 ± 0.04</td>
<td>0.87 ± 0.04</td>
<td>0.87 ± 0.04</td>
</tr>
<tr>
<td>ssl-text</td>
<td>1K</td>
<td>0.74 ± 0.03</td>
<td>0.75 ± 0.04</td>
<td>31.1 ± 0.04</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
<td>0.92 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.93 ± 0.015</td>
<td>0.96 ± 0.004</td>
<td>205.7 ± 0.05</td>
<td>0.99 ± 0.04</td>
<td>0.97 ± 0.00</td>
<td>0.97 ± 0.00</td>
<td>0.97 ± 0.00</td>
<td>0.97 ± 0.00</td>
<td>0.97 ± 0.00</td>
<td>0.97 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>10K</td>
<td>0.82 ± 0.02</td>
<td>0.83 ± 0.015</td>
<td>25.7 ± 0.04</td>
<td>0.78 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
</tr>
<tr>
<td>cod-rna</td>
<td>1K</td>
<td>0.87 ± 0.005</td>
<td>0.82 ± 0.009</td>
<td>18.7 ± 0.007</td>
<td>0.73 ± 0.004</td>
<td>0.88 ± 0.005</td>
<td>0.83 ± 0.005</td>
<td>0.78 ± 0.003</td>
<td>0.77 ± 0.006</td>
<td>0.77 ± 0.006</td>
<td>0.77 ± 0.006</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.84 ± 0.004</td>
<td>0.83 ± 0.009</td>
<td>15.7 ± 0.004</td>
<td>0.84 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.83 ± 0.04</td>
</tr>
<tr>
<td>SUSY</td>
<td>1K</td>
<td>0.82 ± 0.02</td>
<td>0.83 ± 0.015</td>
<td>25.7 ± 0.04</td>
<td>0.78 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
<td>0.76 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>5K</td>
<td>0.87 ± 0.005</td>
<td>0.82 ± 0.009</td>
<td>18.7 ± 0.007</td>
<td>0.73 ± 0.004</td>
<td>0.88 ± 0.005</td>
<td>0.83 ± 0.005</td>
<td>0.78 ± 0.003</td>
<td>0.77 ± 0.006</td>
<td>0.77 ± 0.006</td>
<td>0.77 ± 0.006</td>
</tr>
</tbody>
</table>
for a growing ensemble including many specialists. It is certainly possible that further tuning of this nature will lead to better performance in the future, but we aim to highlight the approaches’ simplicity and generality in this initial evaluation.

6.5 Discussion

The results of Table 1 show that we achieve significant improvement over the baselines in all cases. The situation is more unclear when choosing between the new algorithms, an exciting source of future open problems. However, we can still deduce some statements and recommendations from Table 1.

For a combination of simplicity, speed, and good performance with low variance, we recommend MARVIN-C. It only adds one new classifier per iteration, and its results dominate MARVIN across experiments; total correction with line search is effective on the convex slack function. As the culmination of our ideas in this chapter, MARVIN-D might be expected to perform best overall. This is possibly true for many datasets, but we cannot conclude this in general, because MARVIN-D often has high variance. We believe this has to do with the complex way in which it uses labeled data, both online and to estimate specialist errors. Further exploration appears warranted, because such optimization was out of our scope here.

We find that the algorithms here converge quickly in a number of ways. Our results typically can be achieved with a small fraction of the unlabeled data available; beyond this point, we believe that there is a statistical bottleneck in estimating the first term of the slack function, involving $b$. In addition, the table makes clear the profound effect of Wilson’s interval on HEDGEMOWER, which uses all internal nodes. Other heuristics, like selecting just the top $k$ nodes by Wilson score for some $k$, perform almost as well (not shown, see full version ([BF16a])).

All this makes the final decision rule of our algorithms essentially a thresholded
linear combination of a few white-box tree learners, which has the advantageous side effect of being easily interpretable - the score is just an additive combination of specialist rules, each of which can be written as a decision rule (involving both the asleep/awake status and predictions when awake), similar to alternating decision trees ([FM99]) or similar tree ensembles.

6.6 Related and Future Work

Semi-supervised learning has been an active area of research over the last decade ([CSZ06]), mostly involving graph-based methods like label propagation that operate pairwise on the data, and also including the transductive SVM [Joa99] and other algorithms [ZG09]. These generally try to locate the decision boundary at low-density regions of the unlabeled data ([CZ05]); when formulated as max-margin methods, they stand in stark contrast to the muffled min-margin idea for generalization. The labels typically agree with the labeled data and some type of unlabeled cluster structure used as a regularizer ([BNS06]), while our regularizer is in the same spirit but encourages the opposite muffling behavior. Other more coarse-grained methods using discriminative statistics [CSZ06, QSCL09] are more in the spirit of our algorithms.

Semi-supervised algorithms for boosting have previously drawn some attention for their applications, notably in [GLB08, KMJJL09], which also hallucinate labels over the unlabeled data; but they do not use the muffling framework. The only practical work which does is our aforementioned method of the previous chapter ([BF15b]). Another fascinating open problem, building further on these ideas, is how to target areas of the space with specialist classifiers as part of the incremental process, rather than just using the specialists provided by decision trees.

This chapter is related to the significant existing supervised boosting literature ([SF12]). Such algorithms concern the incremental classifier aggregation idea, and
generally attempt to minimize some convex upper bound on error on $L$. It would be of interest to incorporate other notions from boosting theory, like weak learnability, into our framework, or investigate if they are even necessary.

### 6.7 Derivation of the MARVIN Update Rule

The idea of hallucinating labels on unlabeled data is not new to semi-supervised learning (see Sec. 15.3.2). But in our case, we can show that MARVIN is approximately a greedy coordinate descent update on the slack function, as long as $b$ can be estimated reasonably accurately using $L$:

$$
\gamma(\sigma_T) = -\langle b, \sigma_T \rangle + \frac{1}{n} \sum_{j=1}^{n} \Psi \left( \langle x^U_j, \sigma_T \rangle \right) 
$$

$$
\approx \frac{1}{m} \sum_{j=1}^{m} \left[ -y_j^L \sum_{i=1}^{T} h_i(x^L_j) \right] + \frac{1}{n} \sum_{j=m+1}^{m+n} \Psi \left( \sum_{t=1}^{T} h_t(x^U_j) \right) 
$$

$$
= -\sum_{i=1}^{T} \frac{1}{m} \sum_{j=1}^{m} \left( y_j^L h_i(x^L_j) \right) + \frac{1}{n} \sum_{j=m+1}^{m+n} \Psi \left( \sum_{t=1}^{T} h_t(x^U_j) \right) 
$$

Taking the partial derivative with respect to any $\{h_i\}_{i=1}^{p}$ and minimizing over $i$ for the steepest descent direction, we get the MARVIN update.

### 6.8 Generalization and Estimating $b$

Here we expand on the discussions of Section 6.3.

#### 6.8.1 Wilson’s Interval

Wilson’s score interval is specified as follows, for a binomial proportion ([Wil27]).

Our problem of bounding the error rate of a specialist from data is like determining the unknown bias $p \in [0, 1/2]$ of a biased coin that comes up heads (i.e. 1) with probability $p$, using $m$ random flips $A_1, \ldots, A_m \in \{0, 1\}$ to estimate a high-probability upper bound
for $p$. In our case, $p$ is the error rate of the specialist, and $m$ the number of labeled data predicted upon by the specialist and used to estimate its error. We would like a high-probability upper bound $p_u$ for $p$, with specified failure probability $\alpha \in [0, 1]$; we wish that $p \leq p_u$ w.p. $1 - \alpha$ over the $m$ coin flips (the labeled data).

The most apparent unbiased estimator of $p$ is $\hat{p} = \frac{1}{m} \sum_{i=1}^{m} A_i$. The commonly used Wald confidence interval uses the fact that $m\hat{p}$ is binomially distributed: $m\hat{p} \sim Bin(m, p)$, and plugs in $\hat{p}$ instead of $p$ for the bias of the binomial. This results in an upper bound on error of:

$$ p_u = \hat{p} + \sigma z_\alpha $$

where $\sigma = \sqrt{\frac{\hat{p}(1 - \hat{p})}{m}}$

where $z_\alpha$ is the $(1 - \alpha)$ quantile of the standard normal distribution.

Wilson’s interval instead shows shrinkage toward $\frac{1}{2}$, along with additive bias and variance corrections:

$$ p_u = \bar{p} + \bar{\sigma} z_\alpha $$

where

$$ \bar{p} = \frac{\hat{p} + \frac{z_\alpha^2}{2m}}{1 + \frac{z_\alpha^2}{m}} $$

$$ \bar{\sigma} = \sqrt{\frac{\frac{\hat{p}(1 - \hat{p})}{m} + \frac{z_\alpha^2}{4m^2}}{1 + \frac{z_\alpha^2}{m}}} $$

The interval is derived by considering the behavior of the exact binomial tail for low $n$, the regime in which approximations to such an exact tail fail. Our usage often requires this, particularly when considering leaves of an unregularized decision tree as specialists. Wilson’s interval is therefore numerically appropriate for our usage; further discussions on numerical stability can be found in the excellent overview of [BCD01]; see also [LR06a].
6.8.2 Other Details

We selected the allowed failure probability for Wilson’s score interval by cross-validating among only a few values (or not at all), depending on the number of labeled data $m$: \{0.01\} for $m = 1K$, \{0.001,0.005\} for $m = 10K$, \{0.001\} for $m = 100K$. These work across datasets, and across our new algorithms, to give significant performance improvements. Small $m$ values are more problematic to deal with – the predictor has naturally higher variance – so for those experiments we choose from \{0.003,0.01,0.03,0.1\}.

All our algorithms put labeled data to two different uses: training the ensemble itself (RF or incrementally), and estimating $b$. For HEDGE\textsc{Mower} and HEDGE\textsc{Mower-1}, we recommend using less training data in general when $m$ is large (as supervised generalization anyway limits the predictive power of each ensemble classifier), in favor of more accurately estimating $b$. For MARVIN, a similar situation occurs: on each iteration $t$, the classifier being added to the ensemble must have its $b_t$ estimated in order to proceed with the line search. We use $1/4$ of the labeled data to train and $3/4$ to measure $b$; this works well across all experiments.

6.9 Implementation Details

Many of our datasets are large enough that $U$ will not fit in memory, making the batch boosting method seem impractical. However, there is a fairly straightforward minibatch remedy: store only a fixed-size minibatch of unlabeled examples, and periodically resample this entire minibatch. Similar methods refine this further – in a streaming setting, one can replace a randomly selected example in the batch with each new example that arrives.

This works because in the muffled formulation, the unlabeled data only enter into the optimization through the average value of the potential well. Since the data are i.i.d.,
Algorithm 3. HEDGE MOWER

**INPUT:** Size-\( m \) labeled set \( L \), size-\( n \) unlabeled set \( U \), number of trees \( p \), Wilson interval tail probability \( \alpha \).

**Partition** \( L \) at random into sets \( L_1 \) (of size \( \frac{m}{4} \)) and \( L_2 \).

**Train** random forest with \( p \) trees using \( L_1 \).

**Calculate** \( b \) using \( L_2 \), for the \( p \) trees and all their internal nodes, using the lower bound of Wilson’s interval with tail prob. \( \alpha \).

**Prune** \( b \) to leave only nodes with Wilson lower bounds \( \geq 0 \).

(Optional: Further prune, by Wilson scores or otherwise. HEDGE MOWER-1 prunes away all except the \( p \) original non-specialist trees.)

**Minimize slack function** (approximately) using \( b \) and \( U \):

\[
\sigma^* \approx \arg\min_{\sigma \geq 0^p} \left[ -\langle b, \sigma \rangle + \frac{1}{n} \sum_{j=1}^{n} \Psi(\langle x_j^U, \sigma \rangle) \right] \tag{6.3}
\]

(Done in this chapter with gradient descent using line search.)

**OUTPUT:** Predictor \( g(x) = \text{clip}(\langle x, \sigma^* \rangle) \)

This can be estimated with just a small sample, so roughly speaking, the transductive setting converges to the i.i.d. setting when \( n \) is high (abundant unlabeled data). This could conceivably be done for the labeled data as well, but to compare to batch supervised baselines we limit \( m \) to fit in our memory in this chapter.

There are potentially situations in which the hierarchical partitioning of the data is known by other means than a decision tree, such as an unsupervised clustering method. In such cases, the labeled data can exclusively be used to estimate \( b \) for each specialist, but the specialist predictions are not defined a priori. Here it is natural to associate each node with just one prediction: the majority label of the data falling within it.

We experimented with this method using the partitioning defined by the decision tree (not shown), and it performs comparably to or slightly worse than the methods in this chapter; we believe this is because decision trees are learning a supervised hierarchical partition, so the granularity of their predictions is useful. Further discussion is outside our scope here, but this variant might be better for distributed applications, where each
Table 6.2. Information about the datasets used in Chapter 6.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># labeled</th>
<th># unlabeled</th>
<th>Dim.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>kagg-prot</td>
<td>3750</td>
<td>1776</td>
<td>125</td>
<td>Kaggle challenge [kag12]</td>
</tr>
<tr>
<td>ssl-text</td>
<td>1500</td>
<td>11960</td>
<td></td>
<td>[CSZ06]</td>
</tr>
<tr>
<td>kagg-cred</td>
<td>150K</td>
<td>10</td>
<td></td>
<td>Kaggle challenge [kag11]; imbalanced (&lt; 10% positives)</td>
</tr>
<tr>
<td>adult</td>
<td>32561</td>
<td>123</td>
<td></td>
<td>LibSVM</td>
</tr>
<tr>
<td>covtype</td>
<td>581012</td>
<td>54</td>
<td></td>
<td>LibSVM</td>
</tr>
<tr>
<td>ssl-secstr</td>
<td>83679</td>
<td>1189472</td>
<td>315</td>
<td>[CSZ06]</td>
</tr>
<tr>
<td>cod-rna</td>
<td>59535</td>
<td></td>
<td>157413</td>
<td>LibSVM</td>
</tr>
<tr>
<td></td>
<td>train,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>271617 test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSY</td>
<td>5M</td>
<td>10</td>
<td></td>
<td>UCI</td>
</tr>
</tbody>
</table>

node can store its own one-bit prediction irrespective of the others.
Chapter 7

Optimal Classifier Aggregation for Decomposable Multiclass Losses

The ideas developed in the previous chapters for binary classification extend also to multiclass predictors. Here, the data are each labeled with one of \( K \geq 2 \) classes. Due to the multiclass setting, we make some changes to the earlier formalism of Chapter 4 in Section 7.1, introducing a multiclass generalization of our ensemble aggregation game. Section 7.2 then defines a decomposability condition satisfied by a large class of losses, including multiclass log loss and all the general losses for the \( K = 2 \) case explored in Chapter 4.

The game is solved for these losses in Section 7.3. A particularly notable corollary of the result is a convex-optimization learning algorithm for a “softmax” artificial neuron, as commonly used in much recent work on deep learning ([LBH15]).

Strictly speaking, our results here subsume those of Chapter 4, but separate proofs are given for both, because the multiclass proof is significantly longer by necessity.

7.1 Preliminaries

In this setting, the labels are naturally encoded in a \( K \)-dimensional space. Exactly one label is correct, so after incorporating randomization of the labels as before, the label
space is isomorphic to a probability simplex in $K$ dimensions. We translate this into our notation in which the randomized presence or absence of each class is indicated by a number in $[-1, 1]$, so that example $i$’s label $z^i$ belongs to $\Delta^K := \{ z \in [-1, 1]^K : 1^\top z = -K + 2 \}$.

The algorithm’s prediction $g^i$ is in $\Delta^K$ as well.

Again, our goal is to minimize the average worst-case loss over the data, this time using some multiclass loss function $\ell : [-1, 1]^K \times [-1, 1]^K \mapsto \mathbb{R}$. The adversary is still constrained by our information about the true labels.

The unlabeled data $F$ is now not only indexed by the classifier $h \in [p]$ and datapoint $i \in [n]$, but also the class $k \in [K]$; it can now be considered a third-order tensor rather than a matrix. We write $F^i_{k,h}$ as the value of the $k^{th}$ coordinate of the prediction of classifier $h$ on example $i$.

Similarly, the incoming data can be represented as vectors, not matrices – example $i \in [n]$ can be written as

$$X^i = \begin{pmatrix} F^i_{1,1} & F^i_{2,1} & \cdots & F^i_{k,1} \\ \vdots & \vdots & \cdots & \vdots \\ F^i_{1,p} & F^i_{2,p} & \cdots & F^i_{k,p} \end{pmatrix} \in [-1, 1]^{k \times n}$$

We write the $k^{th}$ column of $X^i$ as $x^i_k \in [-1, 1]^p$.

Each coordinate $k \in [K]$ takes on values in $[-1, 1]$ for each data example. We therefore treat the situation similarly to the basic binary classification scenario for each example, so that the true labels $z^1, \ldots, z^n$ are constrained for each coordinate $k \in [K]$:

$$\forall h \in [p] : \frac{1}{n} \sum_{i=1}^{n} z^i_k F^i_{k,h} \geq [b_k]_h$$  \hspace{1cm} (7.1)$$

for some known $b_1, \ldots, b_K \in \mathbb{R}^p$ (e.g. measured from labeled data, as in previous chapters), and known $F$ (from unlabeled data).
The loss minimization game that results is therefore as follows:

\[
V := \min_{g^1, \ldots, g^p \in \Delta^K} \max_{z^1, \ldots, z^p \in \Delta^K: \forall k \in [K], \forall h \in [p]: \frac{1}{n} \sum_{i=1}^{n} I(F^i_k, h \geq b_k)} \frac{1}{n} \sum_{i=1}^{n} \ell(z^i, g^i)
\]  

(7.2)

7.2 Decomposable Losses

The multidimensional label space is in general much more complex than the one-dimensional space that we considered for binary classification. However, most multiclass losses of interest have additional structure that we can exploit to vastly simplify the problem, and then apply our ideas from the two-class setting. Namely, the key observation is that many multiclass losses \emph{decompose additively} over the \(K\) dimensions.

\textbf{Definition 14} (Additively Decomposable Loss). \emph{Call a continuous loss} \(\ell(z^i, g^i) : \mathbb{R}^K \times \mathbb{R}^K \mapsto \mathbb{R}\) \emph{additively decomposable} if it can be written in the form

\[
\ell(z^i, g^i) = \sum_{k=1}^{K} \ell_k(z^i_k, g^i_k) = \sum_{k=1}^{K} \left( \frac{1 + z^i_k}{2} \right) \ell_k(g^i_k)
\]  

(7.3)

for some univariate partial losses \(\ell_k\). \emph{We call the loss monotonically additively decomposable (MAD) if each} \(\ell_k(\cdot)\) \emph{is decreasing}.

One leading example of a MAD loss is the multiclass logarithmic loss between a true label vector \(z^i \in \Delta^K\) and a prediction \(g^i \in \Delta^K\), written as

\[
\ell^{\log}(z^i, g^i) := \sum_{k=1}^{K} \left( \frac{1 + z^i_k}{2} \right) \ln \left( \frac{2}{1 + g^i_k} \right)
\]  

(7.4)

Similarly, we can consider a large class of multiclass losses that can be encoded as a pairwise cost matrix between labels \(C \in \mathbb{R}^{K \times K}\), where \(C_{ij}\) is the loss (cost) suffered
when \( i \) is the actual label and \( j \) the predicted label. One example in this family is the multiclass 0-1 loss, where \( C \) has zeroes on the diagonal and ones elsewhere. For such loss functions, the expected loss \( \ell_C \) can be written as

\[
\ell_C^C(z^j, g^i) := K \sum_{k=1}^{K} \sum_{m=1}^{K} \left( \frac{1 + z^i_k}{2} \right) \left( \frac{1 + g^i_m}{2} \right) C_{k,m} \tag{7.5}
\]

In this section, we develop powerful extensions of our results to multiclass losses that are decomposable, as in the above examples. The upshot is that any monotonically decomposable multiclass loss fits into our classifier aggregation framework, leading to optimal linear-time semi-supervised aggregation algorithms similar to those we have previously described.

In fact, all the results for general binary classification losses, stated earlier in this chapter, are subsumed by our results for MAD losses, because any binary classification loss with monotonic partial losses is a MAD loss with \( K = 2 \).

### 7.3 Main Result

The main result of this chapter solves Eq. (7.2).

**Theorem 15.** Suppose \( \ell_k \) is monotonically decomposable for each \( k \in [K] \). The minimax value of the game (7.2) is

\[
V = \frac{1}{2} \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \left[ -\sum_{k=1}^{K} \left( b_k + \frac{1}{n} \sum_{i=1}^{n} x^i_k \right)^\top \sigma_k + \frac{2}{n} \sum_{i=1}^{n} \min_{g^i \in \Delta^K} \max_{k \in [K]} \left[ \ell_k(g^i_k) + \sigma_k^\top x^i_k \right] \right]
\tag{7.6}

This result (proved in Sec. 7.4) is structurally similar to our main result from Chapter 3, except that it replaces the potential well by a minimax problem depending on the partial losses \( \ell_k \) and the scores \( \sigma_k^\top x^i_k \). This has no simple analytical form in general.
Algorithm 4. Water-filling algorithm for Theorem 16

1: Initialize $g_1 = g_2 = \cdots = g_K = 0$.
2: W.l.o.g. consider the coordinates ordered such that

$$\ell_1(g_1) + s_1 \geq \ell_2(g_2) + s_2 \geq \cdots \geq \ell_K(g_K) + s_K$$

3: if $\ell_K(g_K) + s_K$ is infinite then
4: Initialize $g_1 = g_2 = \cdots = g_K = \varepsilon$ for some small $\varepsilon \to 0^+$.
5: end if
6: for $i = 1$ to $K - 1$ do
7: Define $C_i := \ell_i(g_i) + s_i - (\ell_{i+1}(g_i) + s_{i+1})$.
8: Define $\lambda_i := \ell_i^{-1}(\ell_i(g_i) - C_i)$ for all $j = 1, 2, \ldots, i$.
9: if $\sum_i \lambda_i > 1$ then
10: Define $C'_i := \max\{C \geq 0 : \sum_i \ell_i^{-1}(\ell_i(g_i) - C) \leq 1\}$.
11: Set $g_j$ to $\ell_j^{-1}(\ell_j(g_j) - C'_i)$ for all $j = 1, 2, \ldots, i$.
12: Return $g^* = (g_1; g_2; \ldots; g_K)$.
13: else
14: Set $g_i = \lambda_i$ for all $j = 1, 2, \ldots, i$.
15: end if
16: end for
17: // At this point, the values $\{\ell_k(g_k) + s_k\}_{k \in [K]}$ are equal.
18: Define $C_K := \max\{C \geq 0 : \sum_i \ell_i^{-1}(\ell_i(g_i) - C) \leq 1\}$.
19: Set $g_j$ to $\ell_j^{-1}(\ell_j(g_j) - C_K)$ for all $j = 1, 2, \ldots, K$.
20: Return $g^* = (g_1; g_2; \ldots; g_K)$.

(although it does in cases of interest – see Sec. 7.3.1), but we can express the optimal strategy $g^*$ as the result of an efficient algorithm.

**Theorem 16.** For any scores $\{s_k : k = 1, \ldots, K\}$, consider the game

$$\min_{g \in \Delta^K} \max_{k \in [K]} [\ell_k(g_k) + s_k]$$

Then the minimax optimal predictions $g^*$ are specified by a “water-filling” algorithm written as Alg. 4.

This is proved in Sec. 7.5.
7.3.1 Example: Optimality of the Softmax Artificial Neuron

Theorem 16 and Alg. 4 may appear opaque at first. To illustrate the result’s consequences, it can be instantiated with the example of the logarithmic loss from earlier.

It turns out that the optimal decision function for multiclass log loss is a “softmax” artificial neuron, learnable with an efficient convex optimization:

**Proposition 6.** For the log loss \( \ell_{\log} \) as defined in Eq. (7.4), the value of the game (7.2) is

\[
V = \frac{1}{2} \min_{\sigma_1, \ldots, \sigma_K \geq 0^v} \left[ -\sum_{k=1}^K \left( b_k + \frac{1}{n} \sum_{i=1}^n x_k^i \right) \sigma_k + \frac{2}{n} \sum_{i=1}^n \ln \left( \sum_{j=1}^K \exp \left( \sigma_j^\top x_i^j \right) \right) \right]
\]  

(7.8)

If \( \sigma_1^*, \ldots, \sigma_K^* \) are the minimizers above, then for \( k \in [K] \), the optimal prediction on example \( i \) is

\[
g_k^* = -1 + 2 \frac{\exp \left( \sigma_k^\top x_i^k \right)}{\sum_{k=1}^K \exp \left( \sigma_k^\top x_i^k \right)}
\]

Note that the potential well becomes the log-sum-exp function, well known to be convex and 1-Lipschitz.

**Proof of Prop. 6.** We use Theorem 16 to solve the inner min-max problem of Theorem 15 for any example \( i \in [n] \), and then finish the proof by invoking Theorem 15.

First we run the algorithm of Theorem 16 (Alg. 4). Consider any example \( i \), and define the scores \( s_k^i := \sigma_k^\top x_k^i \) for all \( i \). Alg. 4 essentially skips directly past the for loop to Line 19, putting nonzero weight on each class because the partial losses \( \ell_k^i (g_k^i) = \ln \left( \frac{2}{1 + g_k^i} \right) \) diverge approaching zero (we omit the “log” and “i” superscripts hereafter when clear from context in discussing example \( i \)).

Therefore, for some constant \( Z \),

\[
\ell_1 (g_1^*) + s_1 = \cdots = \ell_K (g_K^*) + s_K = Z
\]
so for all \( k \in [K] \),
\[
    g^*_k = \ell_k^{-1}(Z - s_k) = 2\exp(s_k - Z) - 1
\]

The vector \( g^* \) represents a probability distribution with probabilities that we write as
\[
    p^*_k \propto \exp(s_k - Z).
\]
Observe that \( Z \) corresponds to the normalization factor making \( \sum_k p^*_k = 1 \), so that
\[
    p^*_k = \frac{\exp(s_k)}{\sum_k \exp(s_k)}, \quad Z = \ln \left( \sum_k \exp(s_k) \right)
\]
which is exactly the softmax function. This means that for any \( i \in [n] \),
\[
\begin{align*}
    \min_{g^i} \max_z \left[ \ell_k(g^i_k) + \sigma^T_k x^i_k \right] &= \min_{g^i} \max_z \ln \left( \sum_{j=1}^K \exp \left( \sigma^T_j x^i_j \right) \right) \\
    &= \ln \left( \sum_{j=1}^K \exp \left( \sigma^T_j x^i_j \right) \right)
\end{align*}
\]
Substituting this into Theorem 15 gives the result.

\[\square\]

### 7.4 Proof of Theorem 15

**Proof of Theorem 15.** First note that by definition of a MAD loss,
\[
    V = \min_{g^1, \ldots, g^K} \max_{z^1, \ldots, z^K} \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \left( \frac{1 + z^i_k}{2} \right) \ell_k(g^i_k) \quad (7.9)
\]
\[
    = \frac{1}{2} \min_{r^1, \ldots, r^K} \left[ \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \ell_k(g^i_k) + \max_{z^1, \ldots, z^K} \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K z^i_k \ell_k(g^i_k) \right] \quad (7.10)
\]

where the constraints on \( g^i, z^i \) are omitted for convenience.

The adversary’s constrained optimization over \( z \) can be solved by introducing a Lagrangian vector \( \sigma_k \geq \mathbf{0}^p \) for the set of \( p \) constraints applying to each coordinate.
$k \in [K]$. 

\[
\max_{z^1, \ldots, z^n \in \Delta^K; \forall k \in [K], \forall h \in [p]} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} z^i_k \ell_k (g^i_k) \quad \text{subject to} \quad \frac{1}{n} \sum_{i=1}^{n} z^i_k \geq [b_k]_h.
\]

\[
= \max_{z^1, \ldots, z^n \in \Delta^K} \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \left[ \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} z^i_k \ell_k (g^i_k) + \sum_{k=1}^{K} \sum_{h=1}^{p} \sigma_{k,h} \left( \frac{1}{n} \sum_{i=1}^{n} z^i_k F^i_{k,h} - [b_k]_h \right) \right] \quad \text{(a)}
\]

\[
= \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \max_{z^1, \ldots, z^n \in \Delta^K} \left[ - \sum_{k=1}^{K} b_k^\top \sigma_k + \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} z^i_k \ell_k (g^i_k) + \sigma_k^\top x^i_k \right] 
\]

\[
= \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \left[ - \sum_{k=1}^{K} b_k^\top \sigma_k + \frac{1}{n} \sum_{i=1}^{n} \max_{z^1 \in \Delta^K} \sum_{k=1}^{K} z^i_k \ell_k (g^i_k) + \sigma_k^\top x^i_k \right] 
\] (7.11)

where (a) follows by an application of the minimax theorem (Theorem 25), and by rearranging terms.

Therefore, substituting (7.11) into (7.9),

\[
V = \frac{1}{2} \min_{g^1, \ldots, g^n} \left[ \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \ell_k (g^i_k) \right] + \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \left[ - \sum_{k=1}^{K} b_k^\top \sigma_k + \frac{1}{n} \sum_{i=1}^{n} \max_{z^1 \in \Delta^K} \sum_{k=1}^{K} z^i_k \ell_k (g^i_k) + \sigma_k^\top x^i_k \right]
\]

\[
= \frac{1}{2} \min_{\sigma_1, \ldots, \sigma_K \geq 0^p} \left[ - \sum_{k=1}^{K} b_k^\top \sigma_k + \frac{1}{n} \sum_{i=1}^{n} \min_{g^1 \in \Delta^K} \max_{z^1 \in \Delta^K} \sum_{k=1}^{K} \ell_k (g^i_k) + z^i_k \left( \ell_k (g^i_k) + \sigma_k^\top x^i_k \right) \right] 
\] (7.12)

We have decomposed the problem to depend on an inner zero-sum game between the algorithm and the adversary over each data point, with each side playing a probability distribution:

\[
\min_{g^1 \in \Delta^K} \max_{z^1 \in \Delta^K} \sum_{k=1}^{K} \ell_k (g^i_k) + z^i_k \left( \ell_k (g^i_k) + \sigma_k^\top x^i_k \right) 
\] (7.13)
The game (7.13) is dealt with by a separate argument, laid out in Lemma 17, which states that

\[
\text{Eq. (7.13)} = \min_{g^i \in \Delta^K} \max_{k \in [K]} \left[ \ell_k(g^i_k) + \sigma^\top_k x^i_k \right] - \sum_{k=1}^K \sigma^\top_k x^i_k
\]

Using this result to simplify (7.12),

\[
V = \frac{1}{2} \min_{\sigma_1, \ldots, \sigma_K \geq 0} \left[ -\sum_{k=1}^K b_k^\top \sigma_k + \frac{1}{n} \sum_{i=1}^n \left( 2 \min_{g^i \in \Delta^K} \max_{k \in [K]} \left[ \ell_k(g^i_k) + \sigma^\top_k x^i_k \right] - \sum_{k=1}^K \sigma^\top_k x^i_k \right) \right]
\]

(7.14)
as desired.

**Lemma 17.** For any constants \( \{s_k : k = 1, \ldots, K\} \),

\[
\min \max \sum_{k=1}^K [\ell_k(g_k) + z_k (\ell_k(g_k) + s_k)] = 2 \min \max \left[ \ell_k(g_k) + s_k \right] - \sum_{k=1}^K s_k
\]

Proof of Lemma 17. Recall that \( \Delta^K := \{z \in [-1, 1]^K : 1^\top z = -K + 2\} \). We address the inner maximization over \( z \) by introducing a scalar Lagrange parameter \( \eta \in \mathbb{R} \) for the lone equality constraint on \( z \), which decomposes the problem into a tractable one.

\[
\max_{z \in \Delta^K} \sum_{k=1}^K z_k (\ell_k(g_k) + s_k) = \max_{z \in [-1, 1]^K} \min_{\eta} \left[ \sum_{k=1}^K z_k (\ell_k(g_k) + s_k) + \eta \left( -K + 2 - 1^\top z \right) \right]
\]

\[
= \min_{\eta} \max_{z \in [-1, 1]^K} \left[ \sum_{k=1}^K z_k (\ell_k(g_k) + s_k) + \eta \left( -K + 2 - 1^\top z \right) \right]
\]

\[
= \min_{\eta} \left[ \sum_{k=1}^K \max_{z_k \in [-1, 1]} z_k (\ell_k(g_k) + s_k - \eta) - \eta (K - 2) \right]
\]

(7.15)

where \((a)\) follows by the minimax theorem (Theorem 25, valid because the constraint set
for $z$ is closed). Write the objective function of (7.15) as

$$A(\eta) := \sum_{k=1}^{K} |\ell_k(g_k) + s_k - \eta| - \eta (K - 2)$$

Note that this is convex in $\eta$, with subdifferential

$$\partial A(\eta) = \left\{ -K + 2 + |\{k : \eta > \ell_k(g_k) + s_k\}| 
- |\{k : \eta < \ell_k(g_k) + s_k\}| + \alpha \{k : \eta = \ell_k(g_k) + s_k\} : \alpha \in [-1, 1] \right\}$$

Define $\eta_{\text{max}} := \max_{k \in [K]} [\ell_k(g_k) + s_k]$. We now argue that

$$\eta_{\text{max}} \in \arg\min_{\eta} A(\eta)$$

by using first-order derivative conditions on $A(\eta)$.

For any $\eta > \eta_{\text{max}}$, we can see that $|\{k : \eta > \ell_k(g_k) + s_k\}| = K$. So for any $d \in \partial A(\eta)$,

$$d \geq -K + 2 + |\{k : \eta > \ell_k(g_k) + s_k\}| - |\{k : \eta \leq \ell_k(g_k) + s_k\}| = 2$$

and we see that $\eta \notin \arg\min_{\eta} A(\eta)$.

Next, consider any $\eta < \eta_{\text{max}}$. Then there is at least one $k^* \in \arg\max_k [\ell_k(g_k) + s_k]$ such that $\eta < \ell_{k^*}(g_{k^*}) + s_{k^*}$, and consequently $|\{k : \eta < \ell_k(g_k) + s_k\}| \geq 1$. Therefore, from the definition of $\partial A(\eta)$, we can conclude that for any $d \in \partial A(\eta)$,

$$d \leq -K + 2 + |\{k : \eta \geq \ell_k(g_k) + s_k\}| - |\{k : \eta < \ell_k(g_k) + s_k\}|$$

$$\leq -K + 2 + (K - 1) - 1 = 0$$
Since $A$ is convex in $\eta$, this means that when searching for some $\eta^* \in \arg \min_\eta A(\eta)$, we can w.l.o.g. consider $\eta^* \geq \eta_{\max}$. From this and our previous considerations showing $\arg \min_\eta A(\eta) \subseteq [2, \infty)$ for $\eta < \eta_{\max}$, we can conclude that $\eta_{\max} \in \arg \min_\eta A(\eta)$.

Combining this information with (7.15),

$$\max_{z \in \Delta^K} \sum_{k=1}^K z_k (\ell_k(g_k) + s_k) = \sum_{k=1}^K |\ell_k(g_k) + s_k - \eta_{\max}| - \eta_{\max} (K-2) \quad (7.16)$$

Using (7.16), we now can address the outer minimization over $g$:

$$\min_{g \in \Delta^K} \max_{z \in \Delta^K} \sum_{k=1}^K \left[ \ell_k(g_k) + z_k (\ell_k(g_k) + s_k) \right]$$

$$= \min_{g \in \Delta^K} \left[ \sum_{k=1}^K \left[ \ell_k(g_k) + |\ell_k(g_k) + s_k - \eta_{\max}| \right] - \eta_{\max} (K-2) \right]$$

Since $\eta_{\max} = \max_{k=1,...,K} [\ell_k(g_k) + s_k]$, the summand is

$$\ell_k(g_k) + |\ell_k(g_k) + s_k - \eta_{\max}| = \ell_k(g_k) - (\ell_k(g_k) + s_k - \eta_{\max}) = \eta_{\max} - s_k$$

and therefore, the game can be simplified:

$$\min_{g \in \Delta^K} \max_{z \in \Delta^K} \sum_{k=1}^K \left[ \ell_k(g_k) + z_k (\ell_k(g_k) + s_k) \right] = \min_{g \in \Delta^K} \left[ \sum_{k=1}^K [\eta_{\max} - s_k] - \eta_{\max} (K-2) \right]$$

$$= 2 \min_{g \in \Delta^K} \eta_{\max} - \sum_{k=1}^K s_k = 2 \min_{g \in \Delta^K} \max_{k \in [K]} [\ell_k(g_k) + s_k] - \sum_{k=1}^K s_k$$

\qed
7.5 Proof of Theorem 16

Proof of Theorem 16. Define \( \delta_k := \ell_k(g_k) + s_k \) for convenience. Algorithm 4 seeks to minimize an objective \( \tau := \max_{k \in [K]} \delta_k \), by monotonically adding probability mass to the coordinates of \( g^* \) until it runs out of its “budget” of unit probability mass. (The initialization step of the algorithm can be thought of as initializing \( g_1 = \cdots = g_K = 0 \); the preprocessing \( \text{if} \) condition of Line 3 serves only to handle cases like log loss, for which the partial losses diverge at zero.)

By always adding only to the coordinate(s) with maximum \( \delta_k \), the algorithm lowers \( \ell_k(g_k) \) and lowers \( \tau \). Observe that adding to any coordinate that does not currently maximize \( \delta_k \) wastes some of its budget; that probability mass would be better used on the maximum coordinate(s). Similarly, when multiple coordinates are tied for the maximum \( \delta_k \), the algorithm should decrease them all at the same rate. Therefore, over the course of the algorithm as the \( \delta_k \) values change, coordinates enter the set \( T := \arg \max_{k \in [K]} \delta_k \) and subsequently never leave it.

So it is sufficient to prove that the algorithm always adds only to the coordinates in \( T \), in such a way that \( T \) never shrinks. We do this for two phases of the algorithm: the first phase in which the first \( K - 1 \) coordinates join \( T \) one by one (the \( \text{for} \) loop), and the second phase in which the last coordinate joins \( T \) and all coordinates \( g_1, \ldots, g_k \) are increased.

First phase:

Define for \( i \in [K - 1] \) the following thresholds:

\[
\alpha_i = \min \{ \alpha \geq 0 : \ell_i(\alpha) + s_i = \ell_{i+1}(0) + s_{i+1} \}
\]

When coordinate \( i \) is being added to the set \( T \) for the first time, an amount \( \alpha_i \) can
be added to it before another coordinate joins $T$. This is what is done by each iteration $i$ of the for loop.

On iteration $i$ of the loop, if the budget of probability mass does not run out, Line 14 of the algorithm is reached, so $g_i$ is set to $\lambda_i$. Unraveling the definition of $\lambda_i$, this means that $\forall j \in [i]$, 

$$\ell_j(g_j) - \ell_j(\lambda_j) = C_i$$

so that $\delta_1, \ldots, \delta_i$ are equal and none of them leave the set $T$ (this corresponds to the set $T = [i]$). For coordinate $j = i$, we know that $g_i = 0$ and so 

$$\lambda_i = \ell_i^{-1}(\ell_i(0) - C_i) = \ell_i^{-1}(\ell_i(0) - \ell_i(0) - s_i + (\ell_{i+1}(0) + s_{i+1}))$$

$$= \ell_i^{-1}(\ell_{i+1}(0) + s_{i+1} - s_i)$$

Matching definitions, $\lambda_i$ is equivalent to $\alpha_i$, so iteration $i \in [K - 1]$ stops exactly when $\delta_i$ enters the set $T$, as desired.

Similarly, when the budget of probability mass runs out on iteration $i$ of the loop, Line 11 of the algorithm is reached, and $\forall j \in [i]$ we can show that $\ell_j(g_j) - \ell_j(\lambda_j) = C_i'$, so that again no coordinate leaves the set $T$.

**Second phase:**

If the last coordinate $g_K$ is increased before the budget runs out, the algorithm reaches Line 19, and returns a value of $\xi_j := \ell_j^{-1}(\ell_j(g_j) - C_K)$. Similar to the first phase analysis, it is clear from the definitions that for all $j$, $C_K = \ell_j(g_j) - \ell_j(\xi_j)$, so the change in $\delta_j$ is a constant $C_K$ for all $j$, and all coordinates remain in $T$ for the second phase, as desired. □
Chapter 8

Learning to Abstain from Binary Prediction

We address how to learn a binary classifier capable of abstaining from making a label prediction. Such a classifier hopes to abstain where it would be most inaccurate if forced to predict, so it has two goals in tension with each other: minimizing errors, and avoiding abstaining unnecessarily often.

In this work, we exactly characterize the best achievable tradeoff between these two goals in a general semi-supervised setting, given an ensemble of classifiers of varying competence as well as unlabeled data on which we wish to predict or abstain. We give an algorithm for learning a classifier which trades off its errors with abstentions in a minimax optimal manner. This algorithm is as efficient as linear learning and prediction, and comes with strong and robust theoretical guarantees. Our analysis extends to a large class of loss functions and other scenarios, including ensembles comprised of “specialist” classifiers that can themselves abstain.

8.1 Introduction

Consider a general practice physician treating a patient with unusual or ambiguous symptoms. The general practitioner often does not have the capability to confidently
diagnose such an ailment. The doctor is faced with a difficult choice: either *commit* to a potentially erroneous diagnosis and act on it, which can have catastrophic consequences; or *abstain* from any such diagnosis and refer the patient to a specialist or hospital instead, which is safer but will certainly cost extra time and resources.

Such a situation motivates the study of classifiers which are able not only to form a hypothesis about the correct classification, but also abstain entirely from making a prediction if they are not confident enough. Abstaining can be very beneficial, because a sufficiently self-aware abstaining classifier might abstain on examples on which it is most unsure about the label; this lowers the prediction error it suffers when it does commit to a prediction. Like the doctor in the example above, however, there is typically no use in abstaining on all data, so the amount of overall abstaining is somehow restricted. The classifier must essentially allocate a limited amount of abstaining where it will most reduce error.

In decision theory and machine learning, there has been much historical work on learning such abstaining classifiers (e.g. [Cho57, Cho70, HLS97, Tor00]), where the setting is often dubbed “classification with a reject option.” This prior work has had a central focus: characterizing the *tradeoff* between the abstain rate, and the probability of error when committing to predict (the *error rate*). The optimal tradeoff is achieved by the set of classifiers which minimize error rate for a given abstain rate, and vice versa; such classifiers are *Pareto optimal*, and the set of them is called the Pareto frontier between abstain and error rates.

The tradeoff has previously been examined theoretically using algorithms that are overly conservative by design in deciding to abstain, in order to prove error bounds. So the Pareto frontier between abstain rate and error rate has only been determined at a handful of points despite a spate of recent activity on the problem ([BW08, WY11, EYW11, EYW12, ZC14]).
In this work, our main contribution is to describe the Pareto frontier \textit{completely} for the very general semi-supervised learning scenario addressed by our muffled framework of previous chapters, in which labeled and unlabeled data are both available. So we assume we are given an ensemble of binary classifiers with which to build our aggregated abstaining classifier, along with (bounds on) the ensemble classifiers’ error rates, estimated from labeled data.

We generalize the formulation of Chapters 3 and 4 by exactly specifying the Pareto optimal frontier when learning an abstaining classifier in this setting. We also give an efficient algorithm for learning the abstaining classifiers realizing this optimal tradeoff, which achieve the lowest possible error rate for any given abstain rate. The resulting guarantees on error as a function of abstain rate, and their dependence on the structure of the unlabeled ensemble predictions, are without precedent in the literature and are all unimprovable in this setting, by virtue of the minimax arguments we use to derive them.

This chapter builds towards such results in Section 8.2, an introduction to our techniques on a common and natural model of abstaining: as a third possible outcome with a fixed cost independent of the true labels.

We then arrive at the heart of the chapter in Section 8.3, building on the previous sections to establish the Pareto optimal tradeoff between abstaining and erring. Our analysis also shows that this Pareto frontier is exactly the same whether abstaining is penalized explicitly with a fixed cost for each abstained-upon example (as in Section 8.2) or is simply restricted to not occur too often overall (Section 8.3), the two prevailing models of abstaining in the literature.

In both Section 8.2 and Section 8.3, we also derive the minimax optimal algorithm that realizes this Pareto optimal tradeoff. To be specific, the tradeoff is a one-dimensional curve (Fig. 8.1, which we derive in Theorem 11), parametrized by the allowed abstain rate $\alpha$. The algorithm takes $\alpha$ as an input, and outputs a decision rule with the lowest
Figure 8.1. The abstain/error Pareto frontier in our setting: Minimum achievable error rate $V(\alpha)$ for an aggregated classifier with allowed abstain rate $\leq \alpha$. Formalized in Theorem 11.

possible error bound among all classifiers with abstain rate $\alpha$. The method resembles that of Ch. 3, again involving a simple learning algorithm and prediction rule as scalable as a linear learning algorithm like the perceptron, and without parameters to tune.

Having established the tradeoff and how to realize it optimally and efficiently, we discuss related work in Section 8.4. Finally, to show the robustness of our analysis, we re-analyze the fixed-cost model in Section 8.5 with the large class of other loss functions introduced in Chapter 4, which includes all convex surrogate losses used for ERM as well as many non-convex losses.

Many of the proofs of this section are voluminous without adding much new insight, so we defer them to Sec. B. As a last note, we fix our terminology. We build a classifier capable of two types of output on an input data point: abstaining, and predicting a label (that is, not abstaining). This is to avoid confusion, since there appear to be no standard names for these concepts in the abstaining literature.
8.2 Abstaining with a Fixed Cost

For the rest of this chapter, we build on the muffled learning setup of Chapter 3, extending it to be able to learn a classifier that can abstain. Recall the definitions from that setting, including the classifier ensemble $\mathcal{H}$, unlabeled data $\mathcal{F}$, true labels $\mathbf{z}$, and ensemble correlation bounds $\mathbf{b}$.

We first need to specify how to formalize the decision-making process of the abstaining classifier we generate. This is done w.l.o.g. by allowing the classifier to abstain on each test example $j \in [n]$ with some probability $1 - p_j$, and predict $g_j$ with the remaining probability $p_j \in [0, 1]$. Therefore, the classifier now plays two vectors, the probabilities of prediction $\mathbf{p} \in [0, 1]^n$ and the predictions $\mathbf{g} \in [-1, 1]^n$, while the adversary again plays the true labels $\mathbf{z}$ constrained as in Chapter 3.

The objective function must reflect the tradeoff between abstaining and erring. In the literature, this is often modeled by assessing a pre-specified cost $c \in [0, \frac{1}{2}]$ for an abstention, somewhere between the loss of predicting correctly and guessing randomly. This induces the algorithm to be self-aware when it abstains – it will abstain whenever it would expect to incur a higher loss, $> c$, by predicting against the adversary.

Formally, given a value of $c$, the game proceeds as follows:

- Algorithm plays probabilities of prediction $\mathbf{p} \in [0, 1]^n$, and predictions $\mathbf{g} \in [-1, 1]^n$.
- Adversary plays true labels $\mathbf{z} \in [-1, 1]^n$ such that $\frac{1}{n}\mathbf{Fz} \geq \mathbf{b}$.
- Algorithm suffers expected loss $p_j\ell(z_j, g_j) + (1 - p_j)c$ on each test example, where $
\ell(z_j, g_j) = \frac{1}{2}(1 - z_jg_j)$ is the zero-one loss as in Chapter 3.
We wish to minimize the worst-case expected loss on the test data:

\[ V_c := \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \max_{z \in [-1,1]^n} \ell(z, g) = \frac{1}{2} \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \max_{z \in [-1,1]^n} \frac{1}{n} \sum_{j=1}^n \left( p_j (1 - z_j g_j) + 2 (1 - p_j) c \right) \]  

(8.1)

Our goal is now to find the optimal \( g_c^*, p_c^* \) which minimize the above. To do this, we define a new potential function we will minimize over the unlabeled data.

**Definition 7.** Define the **abstaining potential well** given abstaining cost \( c \leq \frac{1}{2} \):

\[
\Psi(m, c) = \begin{cases} 
|m| + 2c(1 - |m|) & |m| \leq 1 \\
|m| & |m| > 1 
\end{cases} 
\]  

(8.2)

In particular, note that this generalizes the potential well of Chapter 3, which we continue to write as a univariate function \( \Psi(m) \); so we have \( \Psi(m) = \Psi(m, \frac{1}{2}) = \max(|m|, 1) \). The potential wells are plotted as a function of \( m \) for different values of \( c \) in Fig. 8.2. As \( c \) decreases with all else held equal, the potential well’s shape changes so that more examples tend to have lower-magnitude margins at the optimum, and so they get abstained upon more. It is easy to see this in the extreme case – the \( c = 0 \) potential well is minimized by abstaining on all examples, so that they are all at the bottom of the well with zero margin.

**Theorem 8.** The minimax value of the game (8.1) for any positive cost \( c \leq \frac{1}{2} \) is

\[
V_c = \frac{1}{2} \min_{\sigma \geq 0} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma, c) \right] 
\]  

(8.3)

If we define \( \sigma_c^* \) to be the minimizing weight vector in this optimization, the minimax
optimal predictions $g^*_c$ and participation probabilities $p^*_c$ can be defined for each example $j \in [n]$ in the test set as:

$$p^*_{c,j} = \min \left( 1, \left| x_j^\top \sigma^*_c \right| \right), \quad g^*_{c,j} = \text{sgn} \left( x_j^\top \sigma^*_c \right)$$

Just as in Chapter 3, we have reduced the learning problem to a convex optimization of a well-behaved function, and the optimal prediction on each test example depends only on that example’s learned margin. This gives an efficient algorithm to optimally solve the cost-sensitive abstain scenario in the semi-supervised setting.

Observe that $\Psi(m, c)$ is increasing in $c$ for any $m$, so for any $c \leq \frac{1}{2}$,

$$V_c = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma, c) \right] \leq \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi(x_j^\top \sigma) \right] = V_{\text{PRED}}$$

which shows that allowing abstaining does help in the worst case; its benefit can be quantified in a way that depends intimately on the (unlabeled) data distribution.

As in the prediction setting, the key quantity which encodes the dependencies of the ensemble on unlabeled data is $\sigma^*_c$, which is in general different from $\sigma^*_{1/2} = \sigma^*$ in the prediction-only setting. The optimal strategies for the abstaining classifier are...
very simple, and are independent of \( c \) given the \( \sigma_i^\ast \)-weighted margin – in particular, after abstaining it is no longer optimal to “hedge,” or randomize, the predictions at all.

Instead, the prediction is a simple majority vote, while the optimal classifier abstains with nonzero probability on exactly the examples it would hedge in the non-abstain case (those with margin in \([-1, 1]\)). Observe that this reduces to the non-abstaining algorithm of Chapter 3 for \( c = \frac{1}{2} \) despite the apparent difference in \( g^\ast \), since abstaining in this case is equivalent to predicting one of the two labels uniformly at random.

To summarize, we have established a clean minimax solution to the problem of abstaining with a specified cost, extending the classic fully supervised work of [Cho70]. To our knowledge, this and previous work in this model considered a supervised setting with no access to unlabeled data, so we believe we have contributed the first principled semi-supervised abstaining classifier in the fixed-cost model.

However, we believe this model is still an unsatisfactory way to study the central abstain-error tradeoff, because there may be no clear way to choose \( c \). We remedy this in the next section.

### 8.3 Predicting with Constrained Abstain Rate

In this section, we directly study the central tradeoff between abstain rate and error rate without using \( c \), and derive Fig. 8.1.

As shown in Fig. 8.1, classifiers on the Pareto frontier minimize their error rate under a given constraint on abstain rate. So the algorithm is now given a constant \( \alpha \in [0, 1] \); we simply restrict our abstain rate to be \( \leq \alpha \), and derive the minimax optimal error-minimizing strategy among all predictors with abstain rate thus restricted.

The game protocol with the adversary is now a slightly altered version of the previous game:
• Algorithm plays probabilities of prediction $\mathbf{p} \in [0, 1]^n$ such that $\frac{1}{n} \mathbf{1}^\top \mathbf{p} \geq 1 - \alpha$, and predictions $\mathbf{g} \in [-1, 1]^n$.

• Adversary plays true labels $\mathbf{z} \in [-1, 1]^n$ such that $\frac{1}{n} \mathbf{F} \mathbf{z} \geq \mathbf{b}$.

• Algorithm suffers expected error $p_j \ell(z_j, g_j)$ on each test example.

This deals directly with the abstain-error tradeoff that we have described earlier. Raising $\alpha$ always lowers the classifier’s error rate in this scenario, because the examples on which it abstains by default contribute zero to the expected error, so abstaining can only lower the error contribution relative to predicting.

As in the previous section, our goal is to minimize the worst-case expected error when predicting on the test data (w.r.t. the randomized labeling $\mathbf{z}$), which we write $\ell_p(\mathbf{z}, \mathbf{g}) := \frac{1}{n} \sum_{j=1}^n p_j \ell(z_j, g_j) := \frac{1}{2n} \sum_{j=1}^n p_j (1 - z_j g_j)$. We can again write our worst-case prediction scenario as a zero-sum game:

$$V(\alpha) := \min_{\mathbf{g} \in [-1,1]^n} \min_{\mathbf{p} \in [0,1]^n} \max_{\frac{1}{n} \mathbf{1}^\top \mathbf{p} \geq 1 - \alpha} \max_{\frac{1}{n} \mathbf{F} \mathbf{z} \geq \mathbf{b}} \ell_p(\mathbf{z}, \mathbf{g})$$

$$= \min_{\mathbf{g} \in [-1,1]^n} \min_{\mathbf{p} \in [0,1]^n} \max_{\frac{1}{2n} \sum_{j=1}^n p_j (1 - z_j g_j)}$$

Next, we exactly compute the optimal $\mathbf{g}^*$ (and $V(\alpha)$) given $\alpha$, and derive an efficient algorithm for learning them.

### 8.3.1 Solving the Game

The solution to the game (8.4) depends on the optimum of an appropriate abstaining potential as before (all proofs hereafter deferred to appendix).
Theorem 9. Given some $\alpha \in (0, 1)$, the minimax value of the game (8.4) is

$$V(\alpha) = \frac{1}{2} \max_{\lambda \geq 0} \left[ \min_{\sigma \geq 0^p} \left( -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi \left( x_j^\top \sigma, \frac{\lambda}{2} \right) \right) - \lambda \alpha \right]$$ (8.6)

If we define $\sigma^*(\alpha)$ to be the minimizing weight vector in this optimization, the minimax optimal predictions $g^*$ and participation probabilities $p^*$ can be defined for each example $j \in [n]$ in the test set.

$$p_j^* = \min \left( 1, |x_j^\top \sigma^*(\alpha)| \right), \quad g_j^* = \text{sgn} \left( x_j^\top \sigma^*(\alpha) \right)$$

For the rest of this subsection, we elaborate on the structure of the solution in Theorem 9. Define the maximizing value of $\lambda$, written as a function of the abstain rate, to be $\lambda^*(\alpha)$. Also define the minimand over $\sigma$ as $\gamma(\sigma, \lambda) := -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \Psi \left( x_j^\top \sigma, \frac{\lambda}{2} \right)$, and its minimizing $\sigma$ as a function of $\lambda$, i.e. $\sigma^*(\lambda) = \arg \min_{\sigma \geq 0^p} \gamma(\sigma, \lambda)$. Now

$$\partial \left[ \gamma(\sigma, \lambda) \right] \partial \lambda = \frac{1}{n} \sum_{j=1}^n \partial \left[ \Psi \left( x_j^\top \sigma, \frac{\lambda}{2} \right) \right] = \frac{1}{n} \sum_{j=1}^n \left[ 1 - |x_j^\top \sigma| \right]_+$$

so $\gamma(\sigma, \lambda)$ is linear in $\lambda$. Therefore, the maximand over $\lambda$ is concave, because it is a minimum of linear functions, so that we can define the concave function

$$w(\lambda) := \min_{\sigma \geq 0^p} \gamma(\sigma, \lambda)$$

This concavity means $w'(\lambda)$ is decreasing\(^1\), so we can deal with its pseudoinverse $(w')^{-1}(\alpha) := \inf \{ \lambda \geq 0 : w'(\lambda) \leq \alpha \}$.

\(^1\)Strictly speaking, we mean any element of the subdifferential set; our proofs remain essentially unaltered with this change.
appear unclear how to efficiently solve this algorithmically. However, $\lambda$ is the Lagrange parameter for the upper-bound constraint $\alpha$, so that any $\alpha$ maps to a $\lambda^*(\alpha)$, and vice versa.

**Proposition 10.** For a given $\alpha$, the $\lambda$ value corresponding to $\alpha$ is $\lambda^*(\alpha) = (w')^{-1}(\alpha - 1)$, such that $\frac{1}{n}\sum_{j=1}^{n} p_j^*(\sigma^*[\lambda^*(\alpha)], \lambda^*(\alpha)) = 1 - \alpha$.

Prop. 10 implies that the maximizing value $\lambda^*(\alpha)$ is decreasing in $\alpha$, again identifying the bijection between $\lambda \geq 0$ and $\alpha \in [0, 1]$. The bijection is such that the abstain rate is exactly $\alpha$, so that as we might expect, the adversary should never leave slack in the abstain rate constraint.

Returning to the saddle point problem in Theorem 9, it is not clear how to efficiently calculate $\lambda^*(\alpha)$ for a given $\alpha$ and achieve a specific point on the optimal tradeoff curve. However, we can try the algorithm with $\lambda$ as an input parameter, governing the optimal weight vector $\sigma^*(\lambda)$. Running it in parallel for a range of values of $\lambda \geq 0$, we can learn a set of classifiers which approximates the optimal Pareto frontier, and then choose one with an abstain rate to our liking.

### 8.3.2 The Pareto Frontier and Discussion

Recall that the central tradeoff here is between $V(\alpha)$, the minimax attainable error rate, and $\alpha$. The main result of this chapter describes this Pareto frontier in our general setting, and fully explains Fig. 8.1.

**Theorem 11.** For $\alpha \geq 0$, $V(\alpha)$ is the convex, decreasing, nonnegative function

$$V(\alpha) = \frac{1}{2}(w(\lambda^*(\alpha)) - \alpha \lambda^*(\alpha))$$

(8.7)

In particular, $V(0) = V_{\text{PRED}}$ and $V(1) = 0$, and $V'(\alpha) = -\frac{1}{2}\lambda^*(\alpha)$.

Since $V(\alpha)$ is convex and decreasing on $[0, 1]$, $V'(\alpha) \leq \frac{V(1)-V(\alpha)}{1-\alpha} = -\frac{V(\alpha)}{1-\alpha}$. So
the curve $V(\alpha)$ is always decreasing at least as fast as its secant to the point $(\alpha = 1, V(1) = 0)$. At any $\alpha$, this means there is a marginal benefit to abstaining, relative to the error incurred by predicting with $\alpha$ constant.

### 8.4 Discussion and Related Work

The solution to the game of Section 8.3 closely resembles that of Section 8.2, with $\lambda$ replacing $2c$ in the same solution path. This close correspondence is perhaps not surprising when viewed as a consequence of Lagrange duality; the cost $c$ is dual to the abstain rate $\alpha$. However, it appears to be a new contribution to the abstaining literature.

There is a large, theoretically principled literature on expressing unsureness in binary classification in various settings, ranging from selective sampling in an online learning setting ([CBL06]) to more benign statistical learning settings like ours. We focus on the latter here as being more relevant; although our formulation is game-theoretic, the “first moment” constraints imposed by the ensemble $b$ in this chapter successfully handle i.i.d. or similar data, and cannot be imposed in online learning scenarios. Though there are many practical approaches to abstaining (e.g. [HLS97]), they are most often Bayesian (similar to the early work which introduced the cost model ([Cho70])), and are not analyzed under model misspecification.

Characterizing the abstain/error tradeoff has been a central goal of previous work since the early Bayesian and cost-based work on the topic ([Cho57, Cho70]). In statistical learning, analyses of the tradeoff have only been conducted in restricted situations. Generally, they either hold for general ensembles in the realizable case, in which there is an error-free classifier in the ensemble ([EYW10, EYW12]); or hold for very specific hypothesis classes, as in regularized risk minimization approaches for linear learning ([BW08, WY11]).

The possibility of zero-error learning with some nonzero abstention rate is one of
the highlights of the former line of work, shown by [EYW10]. The algorithms in such work basically track disagreements among a version space—a set of “good” classifiers. Such algorithms hope to eventually converge on the best classifier in the space as ERM would, and as a consequence asymptotically cannot perform better than it as the muffled formulation can.

The work of [FMS04] stands apart from other prior work in analyzing an ensemble abstaining scheme that predicts only on high-margin examples. However, their analysis is fundamentally different from ours, being basically an ERM-type analysis that converges to the best single classifier’s predictions, albeit with possibly faster convergence than ERM. As discussed at length in earlier chapters (papers [BF15a, BF15b]), our formulation incorporates voting behavior to cancel errors in the ensemble, and can therefore yield much better results than the best classifier $h^*$ while still guaranteeing that it will perform at least as well as $h^*$.

In using the semi-supervised muffled formulation, we also inherit the approach’s flexibility. Ch. 5 has already showed that just a small modification to $F$ suffices to handle ensembles of specialists, each of which predicts only on some arbitrary subset of the data and abstains on the rest. Armed with this ability, we can use our formulation to form multilayer stacks of specialists, since the abstaining classifiers learned by our methods in this chapter are also specialists. The analysis generalizes in other ways as well, including to weighted datasets (Chapter 4) and of course to other loss functions, as we show in Section 8.5.

To our knowledge, there exist no theoretically motivated semi-supervised ap-

---

2 Following [EYW10], we can look for the tantalizing possibility of learning to predict with zero error for some $\alpha < 1$, when the aggregated classifier is able to abstain exactly where it is unsure and predict perfectly otherwise. This is at the very least conceivable in our framework because $g_j^*$ is a weighted majority vote, and does not internally randomize its predictions at all. By Theorem 11, since $V(1) = 0$ and $V(\alpha)$ is convex and decreasing, zero-error prediction can happen exactly when $V'(\alpha) = -\frac{1}{2}\lambda^*(\alpha) = 0$ for some $\alpha < 1$. Working through the definition of $\lambda^*$, this happens when $w'(0) < 0$, a general criterion based on the unlabeled data distribution ([Bal16]).
proaches to abstaining, let alone methods that work with general ensembles and perform better than ERM. There is also a dearth of work in the literature which attempts to address multicriterion optimization problems like this bicriterion one; though many learning problems feature multiple objectives, they are typically scalarized ([CBL06, JS08]).

Our algorithms are interpretable in terms of familiar notions like margin, which cannot be overlooked as a significant potential practical advantage. The abstaining probability is monotonic in the margin’s magnitude, which is highly suggestive of margin-based approaches to “sureness” and their applications like active sampling ([TK02]). So we believe this work throws open directions of empirical inquiry, though an experimental evaluation of our proposed methods is outside our scope in this chapter.

8.5 Abstaining with General Losses

In this final section, we show that the abstaining formulation set up in this chapter also extends to the very general class of loss functions introduced in Chapter 4. This section contains no new algorithmic ideas relative to the rest of this chapter, only demonstrating that our analysis is robust – it does not rely crucially on properties of the 0-1 loss.

Here, we will re-analyze the fixed cost model of Section 8.2, this time allowing the aggregated classifier’s prediction performance to be measured by other losses than 0-1. This allows us to minimax optimally handle a variety of prediction tasks, such as predicting Bayesian-style probabilities (log loss) or coping with asymmetric misclassification costs. We omit the constrained-abstain-rate setting here.

Recall, just for this section, the setup and notation of Chapter 4, particularly the definition of the potential well $\Psi$ and its properties, from Def. 1 and Lem. 2 respectively.

Here we assume that one of the conditions of Lem. 2 hold and therefore that $\Psi$ is convex. We also assume $\Psi$ to be differentiable, purely for convenience in stating
our results; the example of the 0-1 loss shows that a nondifferentiable \( \Psi \) is no obstacle to analysis. Using these assumptions, we now turn to analyze the fixed-cost model of abstaining.

We follow the protocol of Section 8.2, seeking to minimize the worst-case expected loss on the test data:

\[
V_c := \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \max_{z \in [-1,1]^n} \ell(z, g)
\]

\[
= \min_{g,p} \max_{z \in [-1,1]^n} \frac{1}{n} \sum_{j=1}^{n} \left[ p_j \left( \frac{1 + z_j}{2} \right) \ell(g_j) + \left( \frac{1 - z_j}{2} \right) \ell(-g_j) \right] + (1 - p_j) c
\]

\[
= \frac{1}{2} \min_{g,p} \max_{z \in [-1,1]^n} \frac{1}{n} \sum_{j=1}^{n} \left( p_j \left[ \ell(g_j) + \ell(-g_j) + z_j \left( \ell(g_j) - \ell(-g_j) \right) \right] + 2(1 - p_j) c \right)
\]

where the later equalities omit the constraints on \( g, p \) for brevity.

Extending the previous model, our goal is now to find the optimal \( g_c^*, p_c^* \) which minimize the above. To do this, the following “potential”-like function and its derivatives play a major role.

**Lemma 12.** Define the function

\[ G(p_j, x_j, \sigma) = p_j \Psi \left( \frac{x_j^\top \sigma}{p_j} \right) \]

Then \( G(0, x_j, \sigma) = \lim_{p_j \to 0^+} p_j \Psi \left( \frac{x_j^\top \sigma}{p_j} \right) = |x_j^\top \sigma| \). Also, \( G \) is convex in \( p_j \), so that

\[ K_{j, \sigma}(p_j) := \frac{\partial [G(p_j, x_j, \sigma)]}{\partial p_j} = \Psi \left( \frac{x_j^\top \sigma}{p_j} \right) - \frac{x_j^\top \sigma}{p_j} \Psi' \left( \frac{x_j^\top \sigma}{p_j} \right) \]

is increasing in \( p_j \) for \( p_j \geq 0 \).
Since this function is increasing in $p_j$, we can deal with its inverse $K_{j,\sigma}^{-1}(\lambda)$, also an increasing function.

A few other definitions will be necessary to set up our main result. Define the function

$$
\phi(x) = \Psi(x) - x\Psi'(x)
$$

Now $\phi(x)$ is nonnegative, because $\phi\left(\frac{x^\top \sigma}{p_j}\right) = K_{j,\sigma}(p_j)$, which a straightforward calculation verifies is always nonnegative for all $j$. Also, $\phi'(x) = -x\Psi''(x)$, so $\phi(x)$ is decreasing for $x \geq 0$ and increasing for $x < 0$.

Therefore, we can define $\phi_+^{-1}$ to be the inverse of the mapping $\{(x, \phi(x)) : x \geq 0\}$, and $\phi_-^{-1}$ to be the inverse of the mapping $\{(x, \phi(x)) : x < 0\}$, and $\phi_+^{-1}$ is decreasing and $\phi_-^{-1}$ is increasing. So if $\phi(x) = \lambda$, then $x = \begin{cases} 
\phi_+^{-1}(\lambda) & , x \geq 0 \\
\phi_-^{-1}(\lambda) & , x < 0 
\end{cases}$.

\textit{Definition 13.} Define the \textbf{abstaining potential} function:

$$
Q_j(\sigma, \lambda) = \begin{cases} 
(x^\top \sigma)\Psi\left(\phi_\text{sgn}(x^\top \sigma)^{-1}(\lambda)\right) & \lambda \leq K_{j,\sigma}(1) \\
\Psi\left(x^\top \sigma\right) - \lambda & \lambda > K_{j,\sigma}(1)
\end{cases} \tag{8.10}
$$

$Q_j$ is clearly convex in $x^\top \sigma$, and therefore convex in $\sigma$. Finally, we can compute that for any positive $\lambda_0$,

$$
\frac{\partial}{\partial \lambda} \left[ Q_j(\sigma, \lambda) \right] \bigg|_{\lambda = \lambda_0} = -p_j^\ast(\sigma, \lambda_0) = \begin{cases} 
-K_{j,\sigma}^{-1}(\lambda_0) & \lambda_0 \leq K_{j,\sigma}(1) \\
-1 & \lambda_0 > K_{j,\sigma}(1)
\end{cases} \tag{8.11}
$$

so since $K_{j,\sigma}^{-1}$ is increasing, $Q_j$ is concave in $\lambda$. 
Theorem 14. The minimax value of the game (8.8) is

$$V_c = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^{n} Q_j(\sigma, 2c) \right] + c$$

If we define $\sigma_c^*$ to be the minimizing weight vector in this optimization, the minimax optimal predictions $g_{c,j}^*$ and participation probabilities $p_{c,j}^*$ are defined as follows for each example $j \in [n]$ in the test set.

$$p_{c,j}^* = \min \left( 1, K_{j,c}^{-1}(2c) \right) \quad (8.12)$$

$$g_{c,j}^* = \begin{cases} 
-1 & \text{if } x_j^\top \sigma_c^* \leq p_{c,j}^* \Gamma(-1) \\
\Gamma^{-1} \left( \frac{x_j^\top \sigma_c^*}{p_{c,j}^*} \right) & \text{if } x_j^\top \sigma_c^* \in p_{c,j}^*(\Gamma(-1), \Gamma(1)) \\
1 & \text{if } x_j^\top \sigma_c^* \geq p_{c,j}^* \Gamma(1) 
\end{cases} \quad (8.13)$$
Chapter 9

MARVIN as a Function Approximator

In this section, we analyze the non-asymptotic convergence rate of a simplified version of MARVIN, the algorithm of Chapter 6, in a setting where the time horizon $T$ is known in advance. This algorithm builds an ensemble sequentially, one classifier at a time, using labeled and unlabeled data in the muffled formulation. It builds a gradually improving approximation to the true labels, by combining classifiers in $\mathcal{H}$.

The analysis in this chapter outlines a very general type of function approximation guarantee, allowing us to characterize the performance of MARVIN’s iteratively constructed approximation. We take advantage of similar explorations in a line of theoretical work on pseudorandomness and complexity theory; in particular, our proof technique is essentially from Impagliazzo ([Imp95]).

As mentioned in Chapter 6, the MARVIN update is an instance of greedy coordinate descent on the slack function. When the dimension is moderate and finite, this procedure’s rate of convergence has been explored in the optimization literature ([Nes12] and references therein, with asymptotic convergence results dating at least to [Zou60]). But such analysis depends on the ambient dimension ([NSL+15]) and is therefore not appropriate for MARVIN, which is motivated in Ch. 6 by the idea of efficiently aggregating a very large ensemble.

This is the main technical challenge surmounted in this chapter: analyzing the
convergence in a more meaningful manner that does not depend on the ambient dimension, but rather on the competence of the learner used by the algorithm.

For simplicity, we assume that each hypothesis in the ensemble predicts deterministically, so each $h_i$ is a vector of $n$ bits.

9.1 Algorithm and Setup

As described in Alg. 2, MARVIN works on a labeled set $L$ and an unlabeled set $U$, drawn i.i.d. from the same distribution. Our simplified version is different in one significant way: it takes steps of constant size $\alpha = \frac{1}{\sqrt{T}}$ instead of using line search. This new algorithm is given here in Alg. 5, written to emphasize the learning oracle (“learner”) that is called each iteration to generate a new classifier for the ensemble.

Algorithm 5. Fixed Step Size Version of MARVIN

**Input:** Labeled set $L = \{(x_1^L, y_1^L), \ldots, (x_m^L, y_m^L)\}$, unlabeled set $U = \{x_1^U, \ldots, x_n^U\}$, time horizon $T$, learner for ensemble $\mathcal{H}$

**Initialize weights:** $\sigma^0 = 0$, so that $\langle x, \sigma^0 \rangle = 0$ for all $x \in U$

for $t = 1$ to $T$
do

Hallucinate labels for $U$: for $j = 1, \ldots, n$, set

$$\tilde{y}_{t,j} = \text{sgn}(\langle x_j^U, \sigma^{t-1} \rangle) 1(|\langle x_j^U, \sigma^{t-1} \rangle| \geq 1)$$

Call learner to find $h_t \in \mathcal{H}$ that approximately maximizes

$$c(h) = \frac{1}{m} \sum_{i=1}^{m} y_i^L h(x_i^L) + \frac{1}{n} \sum_{j=1}^{n} (-\tilde{y}_{t,j}) h(x_j^U)$$

Add $h_t$ to predictor with weight $\frac{1}{\sqrt{T}}$, so that $\sigma^t$ has all $t$ components $= \frac{1}{\sqrt{T}}$.

end for

**Output:** Predictor $g_T(x) = \text{clip}(\langle x, \sigma^T \rangle)$

The actual performance of this learning algorithm is limited by two factors:

1. Estimation: The labeled and unlabeled data are only finite random samples, so
the slack function being optimized is a necessarily inexact estimate.

2. **Approximation:** The incremental addition of hypotheses should bestow gradually better approximation error over time, but the ultimate error is limited by the expressivity of $\mathcal{H}$. The true labels are not perfectly generated by any aggregate of classifiers in $\mathcal{H}$.

This is a commonly occurring tradeoff, particularly in incremental learning ([BB08]). We typically have relatively scarce labeled data $m \ll n$, so that the estimation error is governed by uniform convergence of the ensemble classifier errors over labeled data.

The approximation error means that with a general $\mathcal{H}$, we cannot hope to guarantee convergence to a classifier with zero error, but rather only guarantee convergence to the best achievable aggregate of classifiers in $\mathcal{H}$. If $\mathcal{H}$ is complex enough, it will achieve very low approximation error, but require a larger quantity of data to learn (estimation) accurately with.

**In this analysis, we skirt estimation issues by taking $m, n$ sufficiently large, focusing only on approximation.**

### 9.1.1 Definitions

The convexity of the slack function lends this optimization problem an interesting structure, different from traditional boosting.

Consider the data distribution $D$ over the data and label spaces $\mathcal{X} \times \mathcal{Y}$, from which the labeled and unlabeled examples are sampled i.i.d. Denote the marginal distribution of $D$ over $\mathcal{X}$ by $\nu(x)$.

In this chapter, the number of data is taken to be very large, so we use slightly different notation to represent functions of data – a classifier $h$ is a binary-valued function $h(x)$ over $\mathcal{X}$ instead of an $n$-dimensional vector. Similarly, we can define some other functions of an example $x \in \mathcal{X}$, which were represented by vectors in previous chapters:
• Define the true label function $z(x) \in \{-1, 1\}$, which we seek to predict. It can be any binary function on $\mathcal{X}$, possibly with internal randomization. \(^1\)

• Define the step size $\alpha := \frac{1}{\sqrt{T}}$.

• Define the predicted labels of Alg. 5 as

$$g_t(x) := \text{clip}\left(\sum_{i=1}^t h_i(x) \sigma^t(h_i)\right) = \text{clip}\left(\sum_{i=1}^t \alpha h_i(x)\right)$$

• Rewrite the hallucinated labels $\tilde{y}_{i,1}, \ldots, \tilde{y}_{i,n}$ of Alg. 5 as

$$\zeta_t(x) := \begin{cases} 0 & |\sum_{i=1}^t \alpha h_i(x)| < 1 \\ \text{sgn}(\sum_{i=1}^t \alpha h_i(x)) & |\sum_{i=1}^t \alpha h_i(x)| \geq 1 \end{cases}$$

Also define the inner product of bounded functions $f, g \in L^1(\nu)$ on the data space $\mathcal{X}$ as:

$$\langle f(x), g(x) \rangle := \mathbb{E}_{x \sim \nu} [f(x)g(x)] = \mathbb{E}_x [f(x)g(x)]$$

(In previous chapters when we considered $n$ unlabeled data, the equivalent operation would be taking the average $\frac{1}{n} \sum_{i=1}^n f(x_i)g(x_i)$.)

In a similar spirit, since there are many classifiers in $\mathcal{H}$, in this chapter we write the label correlation bound for classifier $h$ as $b(h)$. The weight put by vector $\sigma^t$ on classifier $h$ is $\sigma^t(h)$; for Alg. 5, it is nonzero only in multiples of $\alpha$, on classifiers returned by the learner.

\(^1\)In Bayesian classification terms, this represents the conditional label distribution.
9.1.2 Greedy Coordinate Descent as Residual Fitting

In the chosen limit \( m, n \to \infty \), the goal is to minimize the slack function, which stands in for the minimax test error:

\[
\gamma(\sigma^t) = -\sum_{i=1}^{t} \alpha b(h_i) + \mathbb{E}_x \left[ \Psi \left( \sum_{i=1}^{t} \alpha h_i(x) \right) \right]
\]

\[
\approx -\sum_{i=1}^{t} \alpha \langle z(x), h_i(x) \rangle + \mathbb{E}_x \left[ \Psi \left( \sum_{i=1}^{t} \alpha h_i(x) \right) \right]
\]

When the ensemble \( \mathcal{H} \) is finite and moderate, existing guarantees tell us that greedy coordinate descent converges to the minimum of the slack function, because it is convex. Each step corresponds to adding a classifier to the ensemble, in a similar way to boosting.

Now we broaden our focus to general, possibly infinite ensembles. Taking the partial derivative of \( \gamma(\sigma^{t-1}) \) w.r.t. any classifier \( h \in \mathcal{H} \),

\[
-\frac{\partial}{\partial h} \left[ \gamma(\sigma^{t-1}) \right] = \langle z(x), h(x) \rangle - \langle \zeta_{t-1}(x), h(x) \rangle = \langle h(x), z(x) - \zeta_{t-1}(x) \rangle
\]

Maximizing over \( h \), a greedy coordinate descent update on the slack function is in the direction of the classifier

\[
h_t = \arg \max_{h \in \mathcal{H}} \langle h(x), z(x) - \zeta_{t-1}(x) \rangle
\]

which is the Marvin update direction.

\(^2\gamma(\cdot)\) is not smooth, so strictly speaking we must optimize over elements of the subdifferential set, which means considering margin-1 examples \( x_i \) with randomized labels with any bias \( \in [0, 1] \) \( \text{sgn}(\mathbf{x}_i^\top \sigma^*) \). We instead only consider them to have deterministic labels \( \text{sgn}(\mathbf{x}_i^\top \sigma^*) \), meaning we are not exactly performing greedy coordinate descent on the slack function. However, the distinction is not material to us, because the results are stable to perturbations. So “general position” can be enforced, for instance by adding an unbiased Gaussian perturbation with small variance \( \varepsilon \) to the score of each example; this will w.h.p. result in almost all of the examples \( \mathbf{x} \) having margin \( |\mathbf{x}^\top \sigma^*| \neq 1 \), and eliminate the problem at an arbitrarily small additive cost of \( \leq \varepsilon \).
This has a facile interpretation in terms of residual fitting. Recall that the task here is to iteratively build an approximation to \( z(x) \). Consider \( \zeta_{t-1}(x) \) as this approximation after \( t - 1 \) steps. Then \( z(x) - \zeta_{t-1}(x) \) can be thought of as the residual label signal after \( t - 1 \) iterations, and the algorithm’s update at time \( t \) (in Eq. (9.1)) is in the direction which best fits the residual.

In fact, the label approximation \( \zeta_{t-1} \) is essentially the adversary’s optimal label strategy given \( \sigma_{t-1} \), as mentioned in Ch. 3 ([BF15a]); in the muffled formulation, the algorithm plays against this labeling. So the greedy coordinate update rule for selecting each new classifier \( h_t \) corresponds exactly to an iterative procedure of fitting the residual with respect to the minimax optimal adversarial labeling.

### 9.2 Convergence of the Algorithm

By definition of the update in (9.1),

\[
\langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \leq \varepsilon \iff \forall h \in \mathcal{H} : \langle h(x), z(x) - \zeta_{t-1}(x) \rangle \leq \varepsilon
\]

For small \( \varepsilon \), think of this as a nearly hopeless situation for learning: no classifier in \( \mathcal{H} \) fits the residual \( z(x) - \zeta_{t-1}(x) \) well, as none have correlation more than \( \varepsilon \) with it. In this situation, when classifiers are added to the ensemble, progress is quite slow, with the slack function decreasing at a rate \( \leq \varepsilon \).

As time increases, the best available classifier \( h_t \) can be expected to have progressively smaller correlation with the residual \( \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \), because the objective function is convex and continuous. So convergence to the optimum can be studied through this quantity \( \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \).
9.2.1 Convergence Rate

Specifically, for any target error \( \varepsilon \), we can define the last time that the average residual correlation is at least \( \varepsilon \):

\[
T_\varepsilon := \sup \left\{ T : \frac{1}{T} \sum_{t=1}^{T} \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \geq \varepsilon \right\}
\]

The main result of this chapter shows that this time is fairly small – for any \( \varepsilon > 0 \), after \( O(\varepsilon^{-2}) \) iterations, the average residual correlation never exceeds \( \varepsilon \).

**Theorem 18** (Convergence Rate of Alg. 5). Algorithm 5 guarantees that for any \( \varepsilon > 0 \),

\[
T_\varepsilon \leq \frac{81}{\varepsilon^2}.
\]

Equivalently,

\[
\forall T > \frac{81}{\varepsilon^2} : \quad \forall h \in \mathcal{H} : \quad \frac{1}{T} \sum_{t=1}^{T} \langle h(x), z(x) - \zeta_{t-1}(x) \rangle < \varepsilon
\]

Theorem 18 follows from upper and lower bounds on a “potential function” which plays a central role: the time-averaged correlation of the new classifier with the residual labels.

\[
\phi(T) := \frac{1}{T} \sum_{t=1}^{T} \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \tag{9.2}
\]

\( \phi \) takes two averages: over times \( t \in [T] \), and over data \( x \). It can correspondingly be bounded in two separate ways. First, the contribution over all \( x \) can be lower-bounded when averaged over times \( t \leq T_\varepsilon \); and second, the contribution for each \( x \) over all \( t \) can be upper-bounded using the structure of the successive residual fitting procedure (Theorem 19). Putting together these bounds on \( \phi(T_\varepsilon) \) gives the main result.

**Proof of Theorem 18.** By definition of \( T_\varepsilon \), observe that

\[
\frac{1}{T_\varepsilon} \sum_{t=1}^{T_\varepsilon} \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \geq \varepsilon
\]

Meanwhile, Theorem 19 upper bounds the potential \( \phi(T) \) at any \( T \), in particular \( T_\varepsilon \).
Combining these two bounds on the potential $\phi(T_\varepsilon)$,

$$\varepsilon T_\varepsilon \leq \sum_{t=1}^{T_\varepsilon} \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \leq 9 \sqrt{T_\varepsilon}$$

Solving for $T_\varepsilon$ shows that $T_\varepsilon \leq \frac{81}{\varepsilon^2}$. The remaining proposition follows because

$$\langle h(x), z(x) - \zeta_{t-1}(x) \rangle \leq \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle$$

by definition of $h_t$ for any $t$.

It therefore remains to upper-bound the potential $\phi(T_\varepsilon)$, done in the following theorem.

**Theorem 19.** For any example $x$ and time horizon $T$,

$$\sum_{t=1}^{T} \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle \leq 9 \sqrt{T}$$

The proof of Theorem 19 is the heart of our argument, and is in Sec. 9.4. It uses the core potential decomposition and bounding technique of Impagliazzo ([Imp95]), adapted to our variant setting and potential function.

**9.2.2 Discussion**

Theorem 18 can be seen in a slightly different light by averaging over times $[T]$ in a different order. To see this, define the average hallucinated labels

$$\bar{\zeta}_T(x) := \frac{1}{T} \sum_{t=1}^{T} \zeta_{t-1}(x) = \frac{1}{T} \sum_{t=1}^{T} 1 \left( \left| \sum_{i=1}^{t-1} \alpha h_i(x) \right| \geq 1 \right) \text{ sgn} \left( \sum_{i=1}^{t-1} \alpha h_i(x) \right)$$
**Corollary 15 (Convergence of \( \hat{\zeta}_T \)).** Algorithm 5 guarantees that for any \( \varepsilon > 0 \),

\[
\forall T > \frac{81}{\varepsilon^2} : \quad \forall h \in \mathcal{H} : \quad \langle h(x), z(x) - \hat{\zeta}_T(x) \rangle < \varepsilon
\]

Cor. 15 casts \( \hat{\zeta}_T(x) \) as the algorithm’s approximation to \( z(x) \), constructed over \( T \) timesteps. It follows directly from Theorem 18 by averaging over times inside the upper bound \( \varepsilon \).

The paper [Imp95] shows that every example’s total contribution to the potential over time is relatively low. From a slightly different perspective, this is key to a very general approximation result about Boolean functions, as discussed in detail in the powerful work of [TTV09]. There, a very similar result (to our Lemma 21) is used in a constructive proof of existence of a function approximator in an exceptionally general setting.

Compared to that work, we take more interest in the details of the algorithm used in that construction, which remarkably corresponds almost exactly to MARVIN as given in Alg. 5. The resulting analysis in this chapter inherits the generality of the theoretical techniques of [Imp95, TTV09], and their approximation convergence rate of \( O(T^{-1/2}) \). This may be statistically interesting if there is a large population of labeled and unlabeled data from which \( L \) and \( U \) are sampled using \( O(m + n) \) memory each iteration, a solution recommended in Ch. 6 when the data are abundant but memory is not. In this case, the statistical lower bound on convergence from anti-concentration may match the approximation rate; we leave the formalization of this argument as an interesting open problem.
9.3 Hard Core Sets

If run long enough, boosting algorithms ([SF12]) can be considered to separate out a “hard core” of the data – a subset of the data whose labels have low correlation with any classifier in the ensemble ([Imp95]). Indeed, the constructive technique of [Imp95] is used in that paper to prove a fundamental result about hard core sets.

So a natural question is whether a similar situation arises with the iterative approximation procedure of MARVIN. Here we prove that this is the case. We specify “hard core” distributions over the data which arise from the dynamics of Algorithm 5, and over which the true label is almost perfectly decorrelated with every ensemble classifier. These distributions tend to put weight on examples which remain in the hedged region for many iterations.

Define the set of all label approximations of the form of $\zeta_t$ that are constructible from some $t$ functions $h_1, \ldots, h_t \in \mathcal{H}$:

$$S_t = \left\{ H(x) : \exists h_1, \ldots, h_t \in \mathcal{H} \text{ s.t. } H(x) = \begin{cases} 0 & |\sum_{i=1}^{t} \alpha h_i(x)| < 1 \\ \text{sgn} (\sum_{i=1}^{t} \alpha h_i(x)) & |\sum_{i=1}^{t} \alpha h_i(x)| \geq 1 \end{cases} \right\}$$

Note that $S_{t-1} \subseteq S_t$ under broad conditions on the ensemble $\mathcal{H}$.

Then if the label function is weakly hard to predict for any aggregate of classifiers in $\mathcal{H}$, we show that a distribution constructed according to the dynamics of Algorithm 5 decorrelates any $h \in \mathcal{H}$.

**Theorem 20** (Weak Hardness Implies Hard Core Distributions). Fix any $\epsilon, \delta > 0$ and

---

3 E.g., it suffices for $\mathcal{H}$ to contain the all-zeroes classifier (attaining $\sim 50\%$ error regardless of the data).
time horizon $T$. Suppose that for any $f \in S_T$,

$$\Pr(f(x) \neq z(x)) \geq \delta$$

i.e. $z(x)$ is hard on average to predict with classifiers in $S_T$.

(I) There exists a “hard core” distribution $\mu(x) \propto \frac{1}{T} \sum_{t=1}^{T} |z(x) - \zeta_{t-1}(x)|$ over $X$ such that

$$\forall h \in \mathcal{H}: \quad \Pr_{\mu}(h(x) \neq z(x)) \geq \frac{1}{2} - \frac{9}{4\delta \sqrt{T}}$$

(II) Define the stopping time $\tau_\varepsilon := \inf\{t \leq T : \langle h_t(x), z(x) - \zeta_{t-1}(x) \rangle < \varepsilon\} \leq T_\varepsilon$.

Then a hard core distribution $\mu_t(x) \propto |z(x) - \zeta_{t-1}(x)|$ can be defined at time $t$ over $X$, so that

$$\forall h \in \mathcal{H}: \quad \Pr_{\mu_{\tau_\varepsilon}}(h(x) \neq z(x)) \geq \frac{1}{2} - \frac{9}{4\delta \sqrt{\tau_\varepsilon}}$$

The parts of Theorem 20 illustrate the versatility of the proof technique. Part (I) shows that there is a hard core distribution at all times, although it is not clear whether $\mu(x)$ is sparse because of the averaging over $t \in [T]$. Part (II)’s hard core distribution $\mu_t(x)$ remedies this since its support is clearly only on hedged examples and on clipped incorrect predictions.

9.3.1 Discussion

A similar object to our hard core distribution has been shown to play a crucial role in the convergence dynamics of boosting ([KS99, Tel12]), where it can be understood to emphasize examples that are unavoidably difficult or noisy: the asymptotic dregs of
the iterative fitting procedure when it is run for many iterations. Our results contrast somewhat with this and are more like [Imp95] – they are non-asymptotic, and quantify the degree of decorrelation in the hard core as a function of the expressivity of \( \mathcal{H} \) with respect to the label function \( z(x) \), and the size of the hard core.

Practical supervised boosting algorithms will tend to put more weight on the hard core, but the \( \text{min} \)-margin muffled formulation prescribes the opposite, and puts no weight on hedged unlabeled examples in determining \( h_t \). Instead, the dataset given to the learner is made more difficult by hallucinating the minority label on clipped examples, which similarly emphasizes the hedged examples when combined with the effect of labeled data.

### 9.3.2 Proof of Theorem 20

**Proof of Theorem 20.** Let \( h \in \mathcal{H} \) be any ensemble classifier.

**Proving (I).** The key to this result is the decomposition

\[
|z(x) - \zeta_{t-1}(x)| \mathbf{1}(h(x) = z(x)) = \frac{1}{2} [h(x)(z(x) - \zeta_{t-1}(x)) + |z(x) - \zeta_{t-1}(x)|] \quad (9.3)
\]

which can be verified to hold for any \( h, z \) predicting binary labels (e.g. by enumerating the possible values of the variables \( h(x) \in \{-1, 1\}, z(x) \in \{-1, 1\}, \zeta_{t-1}(x) \in \{-1, 0, 1\} \)).

Define a probability distribution over the data \( \mu_t(x) \propto |z(x) - \zeta_{t-1}(x)| \). Taking expectations of both sides over \( x \), Eq. (9.3) implies the following convenient identity:

\[
\Pr_{\mu_t} (h(x) = z(x)) = \frac{\mathbb{E}_{x}[|z(x) - \zeta_{t-1}(x)| \mathbf{1}(h(x) = z(x))]}{\mathbb{E}_{x}[|z(x) - \zeta_{t-1}(x)|]} = \frac{1}{2} \left( 1 + \frac{\mathbb{E}_{x}[h(x)(z(x) - \zeta_{t-1}(x))]}{\mathbb{E}_{x}[|z(x) - \zeta_{t-1}(x)|]} \right) \quad (9.4)
\]
The denominator of (9.4) can be rewritten, because $z(x)$ is binary:

$$\mathbb{E}_x [|z(x) - \zeta_{t-1}(x)|] = 2\Pr (z(x) \neq \zeta_{t-1}(x)) \geq 2\delta$$

where the inequality follows by the average-case hardness property of $z(x)$ since $\zeta_{t-1} \in S_t$.

Substituting this into (9.4) and averaging both sides over $t \in [T]$,

$$\frac{1}{T} \sum_{t=1}^{T} \Pr_{\mu_t} (h(x) = z(x)) \leq \frac{1}{2} + \frac{1}{4\delta} \left( \frac{1}{T} \sum_{t=1}^{T} \langle h(x), z(x) - \zeta_{t-1}(x) \rangle \right) \leq \frac{1}{2} + \frac{9}{9\sqrt{T}}$$

where the second inequality uses the definition of $h_t$ in the update rule.

**Proving (II).** The proof is similar to that of part (I) of this result, bounding (9.4) at time $t = \tau_{\varepsilon}$ using the definitions of $\tau_{\varepsilon}$ and $h_{\tau_{\varepsilon}}$:

$$\Pr_{\mu_{\tau_{\varepsilon}}} (h(x) = z(x)) \leq \frac{1}{2} \left( 1 + \frac{\varepsilon}{\mathbb{E}_x [|z(x) - \zeta_{\tau_{\varepsilon}-1}(x)|]} \right) \leq \frac{1}{2} \left( 1 + \frac{\varepsilon}{2\delta} \right)$$

It is clear that $\tau_{\varepsilon} < T_{\varepsilon}$, so using Theorem 18, $\varepsilon \leq \frac{9}{\sqrt{T_{\varepsilon}}}$, giving the result. \qed

### 9.4 Proof of Theorem 19

Theorem 19 is proved with two lemmas. These decompose the potential around the continuous, Lipschitz decision rule of $g_{t-1}(x)$.

**Lemma 21.** For any $x$ and $T$,

$$\sum_{t=1}^{T} h_t(x) (z(x) - g_{t-1}(x)) \leq \frac{4}{\alpha} + \frac{\alpha T}{2}$$
Lemma 22. For any $x$ and $T$,

$$
\sum_{t=1}^{T} h_t(x) (g_{t-1}(x) - \zeta_{t-1}(x)) \leq \frac{4}{\alpha} + \frac{\alpha T}{2}
$$

Proof of Theorem 19. Adding Lemma 21 and Lemma 22 gives that for any $x$,

$$
\sum_{t=1}^{T} h_t(x) (z(x) - \zeta_{t-1}(x)) \leq \frac{8}{\alpha} + \alpha T
$$

The result follows from substituting $\alpha$ and taking expectations of both sides w.r.t. $x$. □

We adapt the construction used to prove Theorem 3.1 in [TTV09] (see discussion for more details), in order to prove Lemma 21, and then Lemma 22.

Proof of Lemma 21. This proof follows that of Claim 3.4 of [TTV09] almost exactly. Fix any $x$ and define $\Delta_t(x) = z(x) - g_t(x)$. Observe that since $h_t(x)$ are binary $\in \{-1, 1\}$, we have $\Delta_t(x) - \Delta_{t+1}(x) \in \{\alpha, -\alpha, 0\}$.

Now, we partition the timesteps according to the value of $\Delta_t(x)$. For integer $r \in \left[-\frac{1}{\alpha}, \frac{1}{\alpha}\right]$, define ascending and descending level sets as follows:

$$
U_r = \{t : \Delta_{t-1} = z(x) + (r-1)\alpha, \quad \Delta_t = z(x) + r\alpha\}
$$

$$
L_r = \{t : \Delta_{t-1} = z(x) + r\alpha, \quad \Delta_t = z(x) + (r-1)\alpha\}
$$

We write $L = \cup_r L_r, U = \cup_r U_r$. We break up our potential accordingly:

$$
\sum_{t=1}^{T} h_t(x) \Delta_{t-1}(x) = \sum_{t \in L \cup U} h_t(x) \Delta_{t-1}(x) + \sum_{t \in L \cup U} h_t(x) \Delta_{t-1}(x)
$$

(A) (B)

The rest of the proof consists of bounding these two terms.
Bounding (A). First consider term (A) of (9.5). We know for all $r$ that $||U_r| - |L_r|| \leq 1$, because $\Delta_t$ only changes in steps of $\alpha$, and must be less than, equal to, or greater than $z(x) + r\alpha$ for any $r$. The key observation is that we can pair up times $t_1 \in U_r$ with $t_2 \in L_r$, because for any such $t_1, t_2$, $h_{t_1}(x) = -h_{t_2}(x)$. Therefore,

$$h_{t_1}(x)\Delta_{t_1-1}(x) + h_{t_2}(x)\Delta_{t_2-1}(x) = -h_{t_1}(x)\alpha \leq \alpha$$

by definition. So for every $r$, pairing up times in this fashion, we find that

$$\sum_{t \in U_r \cup L_r} h_t(x)\Delta_{t-1}(x) = \sum_{t \in U_r} h_t(x)\Delta_{t-1}(x) + \sum_{t \in L_r} h_t(x)\Delta_{t-1}(x) \leq \frac{\alpha}{2} (|U_r| + |L_r|) + 2$$

where the bound is tight, and the additive constant 2 accounts for the unpaired value of $h_t(x)\Delta_{t-1}(x)$ – there can be at most one. Consequently,

$$\sum_{t \in L \cup U} h_t(x)\Delta_{t-1}(x) = \sum_{r} \sum_{t \in U_r \cup L_r} h_t(x)\Delta_{t-1}(x) \leq \frac{\alpha}{2} \sum_{r} (|U_r| + |L_r|) + 2 \left( \frac{2}{\alpha} \right)$$

(9.6)

$$= \frac{\alpha}{2} (|U| + |L|) + \frac{4}{\alpha} \leq \frac{\alpha T}{2} + \frac{4}{\alpha}$$

(9.7)

Bounding (B). Next, consider term (B) of (9.5), i.e. times $t \notin L \cup U$; by definition, here $g_{t-1}(x) = g_t(x)$. We partition such times into sets $T_P = \{ t : g_{t-1}(x) = g_t(x) = 1 \}$ and $T_N = \{ t : g_{t-1}(x) = g_t(x) = -1 \}$ corresponding to our predicted label on $x$.

Consider first $T_P$. All times in $T_P$ fall into intervals of the form $(t_1, t_2] \subseteq T_P$ for some $t_1, t_2$. We will show that the contribution of each such interval to (9.5) is nonnegative. W.l.o.g., we assume that $t_1 \notin T_P$ – that is, we widen the interval as much as possible.

Now the contribution of such an interval $(t_1, t_2]$, by definition of $T_P$, is

$$\sum_{t \in [t_1, t_2]} h_t(x)\Delta_{t-1}(x) = \sum_{t \in (t_1, t_2]} h_t(x) (z(x) - g_t(x)) = (z(x) - 1) \sum_{t \in (t_1, t_2]} h_t(x)$$
But for at least half the times in any such interval \((t_1, t_2]\), \(h_t\) must be positive; because by definition of \(T_P\) and the interval, \(g_{t_1-1}(x) \neq g_{t_1}(x) = \cdots = g_{t_2}(x) = 1\).

This implies that \(\sum_{t \in (t_1, t_2]} h_t(x) \geq 0\). Since \(z(x) - 1 \leq 0\), we see that for all such intervals \((t_1, t_2]\),

\[
\sum_{t \in (t_1, t_2]} h_t(x) \Delta_{t-1}(x) \leq 0
\]

Summing over all such intervals in \(T_P\), observe that the contribution of \(T_P\) is nonpositive:

\[
\sum_{t \in T_P} h_t(x) \Delta_{t-1}(x) \leq 0.
\]

The argument for \(t \in T_N\) is exactly analogous, so

\[
\sum_{t \in T_P \cup T_N} h_t(x) \Delta_{t-1}(x) = (B) \leq 0.
\]

Combining with (9.9) and substituting into (9.5) finishes the proof.

The argument of the proof of Lemma 21 can now be adapted to prove Lemma 22.

**Proof of Lemma 22.** Fix any \(x\). First observe that either \(\zeta_{t-1}(x) = 0\) or \(\zeta_{t-1}(x) = g_{t-1}(x)\), so that

\[
\sum_{t=1}^{T} h_t(x) (g_{t-1}(x) - \zeta_{t-1}(x)) = \sum_{r: \zeta_{t-1}(x) = 0} h_t(x) g_{t-1}(x)
\]

To prove the result, we will bound the right-hand expression.

Define, for integer \(r \in \{\left\lceil -\frac{1}{\alpha} \right\rceil, \ldots, \left\lceil \frac{1}{\alpha} \right\rceil\},\)

\[
U_r = \{t: g_{t-1}(x) = (r - 1)\alpha, \quad g_t(x) = r\alpha\}
\]

\[
L_r = \{t: g_{t-1}(x) = r\alpha, \quad g_t(x) = (r - 1)\alpha\}
\]

and \(L = \bigcup_r L_r, U = \bigcup_r U_r\).

We know for all \(r\) that \(||U_r| - |L_r|| \leq 1\), because \(g_{t-1}\) only changes in steps of \(\alpha\). The key observation is that we can pair up times \(t_1 \in U_r\) with \(t_2 \in L_r\), because...
\( h_{t_1}(x) \neq h_{t_2}(x) \) for any such \( t_1, t_2 \). Therefore,

\[
h_{t_1}(x)g_{t_1-1}(x) + h_{t_2}(x)g_{t_2-1}(x) = \alpha
\]

by definition. So for any \( r \), pairing up times in this fashion, we find that

\[
\sum_{t \in U_r \cup L_r} h_t(x)g_{t-1}(x) = \sum_{r} \sum_{t \in U_r \cup L_r} h_t(x)g_{t-1}(x) \leq \frac{\alpha}{2} (|U_r| + |L_r|) + 2 \tag{9.8}
\]

where the bound accounts for the unpaired value of \( h_t(x)g_{t-1}(x) \) – there can be at most one.

Using this and (9.8), we conclude that

\[
\sum_{t \in L \cup U} h_t(x)g_{t-1}(x) = \sum_{r} \sum_{t \in U_r \cup L_r} h_t(x)g_{t-1}(x) \leq \frac{\alpha}{2} \sum_{r} (|U_r| + |L_r|) + 2 \left( \frac{\alpha}{\alpha} \right) \leq \frac{\alpha T}{2} + \frac{4}{\alpha} \tag{9.9}
\]

Finally, note that the set \( L \cup U \) can be partitioned into three sets: \( U_{\lceil -\frac{1}{\alpha} \rceil}, L_{\lceil \frac{1}{\alpha} \rceil} \), and \( \{ t : \zeta_{t-1}(x) = 0 \} \). For \( t \in U_{\lceil -\frac{1}{\alpha} \rceil} \), we have \( -h_t(x) = -1 = g_{t-1}(x) = \zeta_{t-1}(x) \), so

\[
\sum_{t \in U_{\lceil -\frac{1}{\alpha} \rceil}} h_t(x) (g_{t-1}(x) - \zeta_{t-1}(x)) = 0
\]

Exactly analogously, \( \sum_{t \in L_{\lceil \frac{1}{\alpha} \rceil}} h_t(x) (g_{t-1}(x) - \zeta_{t-1}(x)) = 0 \).
Putting these together with (9.9),

\[
\frac{\alpha T}{2} + \frac{4}{\alpha} \geq \sum_{t \in L \cup U} h_t(x)g_{t-1}(x)
\]

\[
= \sum_{t : \xi_{t-1}(x) = 0} h_t(x)g_{t-1}(x) + \sum_{t \in U_{\lceil -\frac{1}{\alpha} \rceil}} h_t(x)g_{t-1}(x) + \sum_{t \in L_{\lceil \frac{1}{\alpha} \rceil}} h_t(x)g_{t-1}(x)
\]

\[
= \sum_{t=1}^{T} h_t(x) \left( g_{t-1}(x) - \xi_{t-1}(x) \right)
\]

which finishes the proof. \qed
Chapter 10

Perspectives, Extensions, and Open Problems

In this chapter, we collect a number of striking features of the “muffled” formulation and algorithms we have introduced, and state some open problems that we believe are of particular interest.

Notably, the muffled formulation applies well beyond the setting of binary (or multiclass) classifier aggregation. In the previous chapters, an unlabeled data example is considered to be a vector of the ensemble’s predictions, i.e. a vector of binary features. But nowhere in any of the proofs is a restriction on $F$ used, so the muffled algorithms we have developed handle even data with real-valued features, not just bit-vector data. This is implicit in our specialist formulation, in which each “feature” (classifier) is scaled differently. The muffled formulation is robust to this because each feature represents a different constraint. So as long as the components of $b$ are scaled correctly to be feasible and useful constraints, our algorithms will automatically adapt to features with different scales.

Another important perspective on our muffled formulation traces back to its origination in a linear program (our original muffled analysis of the material of Ch. 3, in [BF15a]), which we showed to be efficiently solvable in $O(p)$ dimensions despite
considering all $2^n$ possible labelings! It is the joint consideration of these, through efficient duality-based methods generalizing linear programming, that gives the muffled method power and flexibility, and its strongest possible loss guarantee. We view it as remarkable that binary classification, in as general a setting as Chapter 4, has the intrinsic additive structure required to admit efficient minimax solutions with highly nontrivial behavior (this structure is discussed when explicating our duality-based proof techniques, in Chapter 4).

There are a number of extensions we are currently working on, notably to unsupervised autoencoding (building on Ch. 4), to regression, and to active learning. In particular, we believe the muffled formulation has significant potential for the latter, in quantifying the benefits of unlabeled data in a fine-grained manner.

The fine structure of unlabeled data plays a notable role in the uniform concentration of estimated classifier errors in a function class $\mathcal{H}$, through the Rademacher complexity of $\mathcal{H}$ and similar objects ([BM02]). This directly impacts how we estimate $b$, but we have avoided the problem in this dissertation. Addressing this rigorously with our formulation is one of the more interesting learning-theoretic open problems we suggest.

Finally, we plan to explore practical applications at larger scale to investigate the space of ensembles that can be aggregated. For instance, decision trees can be inappropriate in high dimension, and efficient linear classifiers could be used instead, as when the data are sparse. The specialists we generate in algorithms like MARVIN are not restricted to being generated by tree partitionings. Other ways of dividing the data like clustering methods can provide valuable unsupervised information that our muffled algorithms can use efficiently and directly.

The optimal decision rules in muffled learning can also be viewed as artificial neurons, as we have pointed out. These originated from neuroscience in very early work ([MP43]), but have more recently enjoyed tremendous success as building blocks of deep
networks for prediction with the use of labeled and unlabeled data ([LBH15]). Our work suggests possible connections to deep learning – concretely, it would be of interest to implement the learning algorithm for the “softmax” artificial neuron (Chapter 7), and also try to aggregate deeply learned representations of data with the muffling framework.

A tremendous influence on modern deep learning has been the Parallel Distributed Processing (PDP) connectionist approach to neuroscience, developed in the 1980s ([RM87]). To the reader who has indulged our speculations so far, we close with a passage from this work that we find suggestive:

“We see the kinds of phenomena we have been studying as products of a kind of constraint satisfaction procedure in which a very large number of constraints act simultaneously to produce the behavior. Thus, we see most behavior not as the product of a single, separate component of the cognitive system, but as the product of large [sic] set of interacting components, each mutually constraining the others [sic] and contributing in its own way to the globally observable behavior of the system.”

– Rumelhart & McClelland ([RM87], p. 76)
Part II

Sequential Processes and Stopping Times
Chapter 11

Prologue: A Game with Exponential Martingales

In this part of the dissertation, we contribute to the theory of martingales, stochastic processes with nearly arbitrary dependence on the past. The full mathematical definition of these requires measure theory (e.g. [Kal06], Ch. 6). We defer this to the later chapters as needed for formal proofs, in favor of a somewhat more intuitive and less rigorous description in this chapter, neglecting some technical niceties.

The intuitions developed in this chapter are closely linked to our technical contributions which follow in succeeding chapters. This chapter also serves as a gentle primer to martingales for the unfamiliar reader, though some familiarity with probability theory is assumed.

11.1 Martingales Describe Fair Games

A martingale $M_t$ can be thought of as describing the winnings of a player betting repeatedly in a fair game against a casino ([SV01]). The word origin is instructive: it arises from a betting strategy popular in 18th century France, in which a gambler bets on repeated flips of a fair coin, winning their stake if they call correctly and losing it (going bankrupt) otherwise. A gambler following this betting strategy bets a dollar on
the first round and repeatedly doubles the amount staked until they win a round, at which point they stop. If this occurs at time 1, the gambler wins $1; and indeed at any time $N$, the total amount won is $-1 - 2 - 4 - \cdots - 2^{N-1} + 2^N = 1$. And since a head will always eventually come up, this strategy appears at first to turn a fair coin toss into a guaranteed payoff of $1$.

Of course, this is intuitively impossible – “you can’t beat the system!” But the gaps in our definitions above do suggest how to define such betting strategies, to analyze with probability theory.

Specifically, the martingale betting strategy requires an arbitrarily high bankroll (line of credit), which we might reasonably forbid in gambling scenarios. So in fair games amenable to analysis, it is reasonable to suppose that the gambler can never have an infinite stake $M_t$ in expectation. Formally, $\mathbb{E} [|M_t|] < \infty$ for all times (that is, $\{M_t\}_{t=0,1,2,\ldots}$ are uniformly integrable). Also, because the game (coin toss) is fair, the gambler should break even, on average, given the past; we require

$$\mathbb{E} [M_t \mid \text{information up to time } t-1] = M_{t-1}$$

Any set of random variables $\{M_t\}_{t=0,1,\ldots}$ satisfying these two conditions is what we call a martingale.

We use this as a springboard to a few further definitions. Define $\mathcal{F}_t$ as the set of all measurable (colloquially, observable) events up to time $t$ (later this will be formally defined as the $\sigma$-algebra, with $\{\mathcal{F}_t\}_{t\geq 0}$ the natural filtration); this can be thought of as our information at time $t$, so that $\mathbb{E} [M_t \mid \mathcal{F}_{t-1}]$ is the expected value of $M_t$ “given the past.” Then as we stated, $M_t$ is a martingale (resp. supermartingale, submartingale) if it has $\mathbb{E} [M_t \mid \mathcal{F}_{t-1}] = M_{t-1}$ (resp. $\leq M_{t-1}$, $\geq M_{t-1}$) for all $t$. So from our interpretation of $M_t$ as a gambler’s winnings, we see that supermartingales are unfavorable games to the
player, and submartingales favorable.

There is another important characteristic of such games which is also motivated by the betting system discussed above – the choice of a *stopping time* for the betting, which depends on the winnings from previous rounds and is therefore random. In the example above, the stopping time of the betting strategy is the first time $M_t = 1$. We consider a general such stopping time $\tau$, which is allowed to depend in any way on the results of previous rounds (the past), but not the future.

Then the relevant intuition, from the classic text by Doob [Doo84], is:

*If a game is unfavorable to a player, it looks unfavorable to him at random times chosen without foreknowledge.*

Expressing this intuition with the expected winnings, we see that for any super-martingale $M_t$ and any stopping time $\tau$,

$$\mathbb{E}[M_\tau] \leq \mathbb{E}[M_0]$$  \hspace{1cm} (11.1)

The inequality becomes an equality for martingales $M_t$, representing fair games. In probability theory, this is known as the Optional Stopping Theorem.

The statements above about martingales are motivated loosely, but the idea of hedging bets over the $\lambda$-casino’s games to handle unknown stopping times corresponds very closely to the actual mathematics. We use these ideas later in this chapter to prove better concentration inequalities for a simple martingale – a random walk.

### 11.2 A Family of Exponential Martingales

We examine a very basic stochastic process, the *Gaussian random walk* $M_t := \sum_{i=1}^{t} \sigma_i$, where each $\sigma_i$ is an independent standard normal random variable (and therefore
\[ \mathbb{E} \left[ e^{\lambda \sigma_i} \right] = \exp \left( \frac{1}{2} \lambda^2 \right). \]

We start with a basic fact about \( M_t \): for all \( \lambda \in \mathbb{R} \),

\[ \mathbb{E} \left[ e^{\lambda M_t - \frac{1}{2} \lambda^2 t} \mid \mathcal{F}_{t-1} \right] = \mathbb{E} \left[ e^{\lambda \sigma_i - \frac{1}{2} \lambda^2} \right] e^{\lambda M_{t-1} - \frac{1}{2} \lambda^2 (t-1)} = e^{\lambda M_{t-1} - \frac{1}{2} \lambda^2 (t-1)} \] (11.2)

This means that the process \( X^\lambda_t = e^{\lambda M_t - \frac{1}{2} \lambda^2 t} \) is a martingale – an exponential martingale – for any \( \lambda \in \mathbb{R} \).

Eq. (11.2) is true as time becomes continuous and the random walk becomes a Brownian motion, in which case it characterizes the geometric Brownian motion (GBM). One consequence is that at any fixed time \( t \),

\[ \mathbb{E} \left[ e^{\lambda M_t - \frac{1}{2} \lambda^2 t} \right] = 1 \] (11.3)

which characterizes all moments of \( M_t \) at once, since it refers to the moment-generating function (m.g.f.) of \( M_t \). So when proving concentration bounds on \( M_t \), it is enough to consider the exponential martingales indexed by \( \lambda \). The proofs of later chapters involve betting on this (uncountable) slate of fair games, all of which are functions of the underlying martingale \( M_t \). Call this slate of games “the \( \lambda \)-casino.”

### 11.3 Hoeffding’s Large Deviation Bound

First, we explore how to prove the standard bounds on the concentration of such a random walk. (As we will see in the formal proofs in succeeding chapters, what we will say is roughly true for many other random processes besides the Gaussian random walk, such as the sum of fair coin flips and any other process with bounded increments.)

The basic concentration bound for large deviations here expresses the same phenomenon as the central limit theorem (CLT) – that for any fixed time \( t \), \( M_t \) concentrates
around its mean of 0 with tails of probability mass like those of a Gaussian distribution.

Lemma 16 ([Hoe63]). Suppose $M_t$ is a Gaussian random walk. Then for any fixed time $T$, w.p. $\geq 1 - \delta$,

$$|M_T| \leq \sqrt{2T \ln(1/\delta)}$$

To get concentration from this m.g.f., we need to choose a parameter $\lambda$.

Proof of Lemma 16. Set $\lambda = \frac{M_T}{T^2}$ to optimize the exponent of (11.2). This gives

$$1 = \mathbb{E} \left[ \exp \left( \frac{M_T^2}{2T} \right) \right]$$

(11.4)

Defining the event $A = \left\{ \exp \left( \frac{M_T^2}{2T} \right) \geq \frac{1}{\delta} \right\}$,

$$\mathbb{E} \left[ \exp \left( \frac{M_T^2}{2T} \right) \right] \geq \mathbb{E} \left[ \exp \left( \frac{M_T^2}{2T} \right) \mid A \right] \Pr(A) \geq \frac{1}{\delta} \Pr(A)$$

where the last inequality is by definition of the “bad event” $A$. Combining this with (11.4) gives the result.

This result is tight – there are well-known matching anti-concentration bounds ([Slu77]). These constitute a convenient finite-time version of the CLT’s tail distributions.

This proof can be recast in our betting language. We initially have $1 to bet at the $\lambda$-casino from earlier. The casino is offering a $\lambda$-indexed set of fair games, each of which offers a payout of $X_t^\lambda = e^{\lambda M_t - \frac{1}{2} \lambda^2 t}$ per dollar invested, which evolves over time. In this case we know the time $T$ when we are to collect, so we can invest all our money in the game that pays most at $T$, which has parameter $\lambda^* = \frac{M_T}{T^2} = \arg\max_{\lambda} X_T^\lambda$. Betting to maximize our winnings $X_T^\lambda$ leads to the concentration result.
11.4 From Stopping Times to Uniform Concentration

In the next chapter, we prove powerful pathwise results that extend this proof to general stopping times – random times \( \tau \) whose choice only depends on the past (i.e. for any time \( t \), \( \{ \tau \leq t \} \) can be determined from the history up to \( t \)).

This requires two modifications to the proof of Lemma 16, which illustrate the relationships we explore in the rest of Part II.

1. The moment bound is stronger, holding for any stopping time \( \tau \), rather than any fixed time \( T \). The integrand is not \( X_t^{A^*} = \exp \left( \frac{M^2}{2t} \right) \), but another similar process \( Y_t \), so that for any stopping time \( \tau \),

\[
1 = \mathbb{E} [Y_\tau]
\]

2. The “bad event” is defined differently, as \( A = \{ \exists t : Y_t \geq \frac{1}{\delta} \} \). Correspondingly, the stopping time we use is the first time this occurs (or \( \infty \) otherwise): \( \tau_D := \min \{ t : Y_t \geq \frac{1}{\delta} \} \), so that \( \tau_D < \infty \) occurs exactly when \( A \) occurs. So writing out the rest of the proof,

\[
\mathbb{E} [Y_{\tau_D}] \geq \mathbb{E} [Y_{\tau_D} | \tau_D < \infty] \Pr(\tau_D < \infty) \geq \frac{1}{\delta} \Pr(\tau_D < \infty)
\]

because when \( \tau_D < \infty \), \( Y_{\tau_D} \geq \frac{1}{\delta} \) by definition of \( \tau \).

Putting both modifications together,

\[
\delta \geq \Pr(\tau_D < \infty) = \Pr \left( \exists t : Y_t \geq \frac{1}{\delta} \right)
\]

so the process \( Y_t \) w.h.p. is uniformly concentrated (below \( \frac{1}{\delta} \)) for all time.
This derivation shows that stopping times and uniform concentration are related. It also demonstrates how we might go about generalizing the Hoeffding concentration bound – find a process $Y_t$ playing the role of $\exp\left(\frac{M_t^2}{2\tau}\right)$ in that proof, such that for any stopping time $\tau$, $E[Y_\tau] = 1$. In betting terms, we would like the winnings process $Y_t$ to break even at any stopping time. The arguments of this section show that such a $Y_t$, if defined appropriately in terms of $M_t$, can be instrumental in uniformly concentrating $M_t$ for all times simultaneously.

11.5 Uniform Concentration by Mixing Bets at a Stopping Time

To design such a process $Y_t$, recall our discussion at the end of Sec. 11.3, giving the betting interpretation of the Hoeffding bound proof. We had bet in the $\lambda$-casino, which offers a slate of fair games characterizing the process $M_t$ whose concentration we proved. Betting so as to maximize our winnings – i.e., on a single well-chosen game indexed by $\lambda^*$ – gave the Hoeffding concentration bound. The only difference here is that we are required to provide winnings $Y_t$ for a game that is fair at any stopping time.

Since the $\lambda$-casino’s exponential martingale games are all fair, we can distribute our money in any way among them and break even at any stopping time – the winnings will look fair to us “at random times chosen without foreknowledge.” In other words, for any distribution $P$ and stopping time $\tau$,

$$E\left[ E_{\lambda \sim P} \left[ X_{\tau}^\lambda \right] \right] = E_{\lambda \sim P} \left[ E \left[ X_{\tau}^\lambda \right] \right] = 1$$

where the unmarked expectation is over the randomness in $M_t$, and the expectations can be interchanged because $X_{\tau}^\lambda$ is nonnegative.
This suggests that our required \( Y_t \), as discussed in Section 11.4, can be defined as

\[
Y_t := \mathbb{E}_{\lambda \sim P} \left[ X_t^\lambda \right]
\]

our winnings after betting our dollar over the set of games in the \( \lambda \)-casino according to distribution \( P \). Just as with the Hoeffding bound in Sec. 11.3, we aim to maximize our winnings with an appropriately chosen \( P \), to get the tightest possible bound.

Therefore, we are interested in playing a distribution \( P \) that maximizes \( \mathbb{E} [ Y_\tau ] \) against a worst-case \( \tau \), which we can roughly rewrite as the following.

\[
\sup_P \inf_\tau \mathbb{E} \left[ \mathbb{E}_{\lambda \sim P} \left[ X_\tau^\lambda \right] \right] \quad (11.5)
\]

Recall that to prove the Hoeffding bound, we essentially deal with a version of (11.5) with the order of the optimizations interchanged to \( \inf_\tau \sup_P \); this reflects our knowledge of the stopping time, written there as \( T \).

Our winnings here can only be less than that previous case, because we try to perform reasonably well against any unknown \( \tau \), as written in (11.5). If we choose any particular deterministic \( \lambda_0 \) as we did for the Hoeffding bound, the adversary chooses \( \tau \) with knowledge of \( \lambda_0 \). It only needs to consider stopping the process \( X_\tau^{\lambda_0} = \exp \left( \lambda_0 M_t - \frac{1}{2} \lambda_0^2 t \right) \), which can be driven to any small constant by waiting long enough.\(^1\) So we must hedge our bets over different \( \lambda \) values, according to the distribution \( P \).

We prove in Chapter 12 that an essentially optimal choice of \( P \) is to have probability density function

\[
\frac{1}{|\lambda| \left( \ln \frac{1}{|\lambda|} \right)^2} \quad \text{over} \quad \lambda \in [-e^{-2}, e^{-2}] \setminus \{0\}
\]

\(^1\)Due to the strong law of large numbers (\( \frac{M_t}{T} \to 0 \) almost surely).
This choice results in
\[ Y_t = \mathbb{E}_{\lambda \sim P} \left[ X^\lambda_t \right] \leq \exp \left( \frac{M^2 t}{2T} \right) \frac{\ln \left( \ln \left( \frac{\ln(t)}{\delta} \right) \right)}{\ln^2(t)} \]

which is slightly lower than for the Hoeffding bound because of the denominator being \( \ln^2(t) \) instead of 1.

Putting this together with the argument of Section 11.4, we have that w.p. \( \geq 1 - \delta \), for all times simultaneously,\(^2\) \[ Y_t \leq \frac{1}{\delta} \iff |M_t| \lesssim \sqrt{t \ln \frac{\ln(t)}{\delta}}. \]

This iterated logarithm bound extends the Hoeffding bound, which gives a slightly superior rate of \( \sqrt{t \ln \frac{1}{\delta}} \) at the cost of requiring a fixed time horizon. As \( t \to \infty \), it matches the rate of one of the fundamental results of probability theory, the law of the iterated logarithm (LIL; [Khi24]), as is further described in the succeeding chapters.

### 11.6 Related Work

In the intuition provided in this chapter, our bets to maximize expected winnings in the \( \lambda \)-casino can be viewed as yielding concentration bounds. We have viewed the betting strategy as being a distribution \( P \) over the range of \( \lambda \), chosen oblivious to \( \tau \) and all else in the problem, which is why such a strategy can show optimal concentration bounds over time. This technique has broad precedents, which we discuss here.

The general strategy of averaging exponential supermartingales (in our view, betting in the \( \lambda \)-casino of [slightly] unfair games) holds key insights into the uniform concentration of measure over finite times, and is an approach that was introduced by Robbins and colleagues ([DR67a, DR68b, RS70]). In the following chapters, we point out how it can be used to prove, improve, and subsume the standard uniform concentration bounds in the literature. As such, we give different but somewhat redundant proofs even

\(^2\)Here \( \lesssim \) represents \( \leq \) in an approximate sense (up to absolute constants).
of related results in the following chapters, for a better overall view of how the formal
techniques can be used together. This is done in Chapter 12, where we fill in the
technical gaps in the intuitive derivation of this chapter to prove sharp finite-time uniform
concentration bounds for essentially any martingale addressed by the CLT.

The link between repeated fair games and martingales has been explored many
times in depth. Central in modern probability are measure-theoretic treatments, which
recognize the fundamental nature of martingales (e.g. [Kal06, Dur10]); some ([RW00,
Wil91]) place them especially prominently. A distinct approach taken by Shafer and Vovk
([SV01]) is to consider repeated prediction games as the basic concepts of probability
instead of the standard measure-theoretic axioms and probability spaces. This is related
to the work [Vov93], and describes how to prove the LIL by investing winnings against
an adversary.

On a related note, there is a long line of work concerning martingales in online
learning, in which the idea of a game-theoretic supermartingale recurs. We highlight
work on online linear/(convex) optimization, when the martingale is directly observed
as here ([FS97, MA13]). There is also a line of work on so-called drifting games
([Fre95, Sch01, LS14]), which generalize a backward-induction, dynamic programming-
style argument to solve repeated games where the winnings are only calculated at the
end, and the supermartingale represents the potential winnings under optimal future play.

A separate but related tack is taken by work in which the player controls the
stopping time of the process, in which case the problem is one of “optimal stopping.”
Such problems have a well-developed theory also heavily involving backward induction
and martingales ([IPS06]), and there is even a formal duality relation between optimal
stopping and uniform concentration over time ([DK94, Rog02]).
Chapter 12
Sharp Uniform Martingale Concentration Bounds

12.1 Introduction

Martingales are indispensable in studying the temporal dynamics of stochastic processes arising in a multitude of quantitative fields ([Hey72, Per09]). When such processes have complex long-range dependences such as arise in learning, it is often of interest to concentrate martingales uniformly over time ([BDF13]).

On the theoretical side, a fundamental limit to such concentration is expressed by the law of the iterated logarithm (LIL). However, this only concerns asymptotic behavior. In many applications, it is more natural to instead consider concentration that holds uniformly over all finite times.

This chapter presents such bounds for the large classes of martingales which are addressed by Hoeffding ([Hoe63]) and Bernstein ([Fre75]) inequalities. These new results are optimal within small constants, and can be viewed as finite-time generalizations of the upper half of the LIL.

To be concrete, the simplest nontrivial martingale for such purposes is the discrete-time random walk \( \{M_t\}_{t=0,1,2,...} \) induced by flipping a fair coin repeatedly. It can be written as \( M_t = \sum_{i=1}^t \sigma_i \), where \( \sigma_i \) are i.i.d. Rademacher-distributed random variables.
(Pr(σ_i = -1) = Pr(σ_i = +1) = 1/2), so we refer to it as the “Rademacher random walk”; take M_0 = 0 w.l.o.g.

The LIL was first proved for the Rademacher random walk, by Khinchin:

**Theorem 17** (Law of the iterated logarithm ([Khi24])). Suppose M_t is a Rademacher random walk. Then with probability 1,

\[
\limsup_{t \to \infty} \frac{|M_t|}{\sqrt{t \ln \ln t}} = \sqrt{2}
\]

Our main concentration result for the Rademacher random walk generalizes this to hold over finite times.

**Theorem 18.** Suppose M_t is a Rademacher random walk. Then for any \( \delta < 1 \), with probability \( \geq 1 - \delta \), for all \( t \geq 173 \ln \left( \frac{4}{\delta} \right) \) simultaneously, the following are true: \( |M_t| \leq \frac{t}{e^2 (1 + \sqrt{1/3})} \) and

\[
|M_t| \leq \sqrt{3t \left( 2 \ln \ln \left( \frac{5t}{2|M_t|} \right) + \ln \left( \frac{2}{\delta} \right) \right)}
\]

(The latter implies \( |M_t| \leq \max \left( 1, \sqrt{3t \left( 2 \ln \ln \left( \frac{5t}{2} \right) + \ln \left( \frac{2}{\delta} \right) \right)} \right) \).)

Theorem 18 takes the form of the LIL upper bound as \( t \to \infty \) for any fixed \( \delta > 0 \). Interestingly, it also captures a finite-time tradeoff between \( t \) and \( \delta \). The \( \ln \ln t \) term is dominated by the \( \ln \left( \frac{1}{\delta} \right) \) term for \( t \lesssim e^{1/\delta} \). In this regime, the bound is \( O\left( \sqrt{t \ln \left( \frac{1}{\delta} \right)} \right) \), a time-uniform central limit theorem (CLT)-type bound below the LIL rate for small enough \( t \) and \( \delta \).

### 12.1.1 Optimality of Theorem 18

We now show that Theorem 18 is optimal in a very strong sense.

Suppose we are concerned with concentration of the random walk uniformly over time up to some fixed finite time \( T \). If the failure probability \( \delta \lesssim \frac{1}{\ln T} \), then the \( \ln \frac{1}{\delta} \)
term dominates the $\ln \ln t$ term for all $t < T$. In this case, the bound of Theorem 18 is $\mathcal{O} \left( \sqrt{t \ln \frac{1}{\delta}} \right)$ uniformly over $t < T$. This is optimal even for a fixed $t$ by binomial tail lower bounds.

The more interesting case for our purposes is when $\delta \gtrsim \frac{1}{\ln T}$, in which case Theorem 18 gives a concentration rate of $\mathcal{O} \left( \sqrt{t \ln t + t \ln \frac{1}{\delta}} \right)$, uniformly over $t < T$. As $T$ and $t$ increase without bound for any fixed $\delta > 0$, this rate becomes $\mathcal{O} \left( \sqrt{\ln t} \right)$, which is unimprovable by the LIL.

But the tradeoff between $t$ and $\delta$ given in Theorem 18 is also essentially optimal, as shown by the following result.

**Theorem 19.** There are absolute constants $C_1, C_2$ such that the following is true. Fix a finite time $T > C_2 \ln \left( \frac{2}{\delta} \right)$, and fix any $\delta \in \left[ \frac{4}{\ln((T-1)/3)}, \frac{1}{C_1} \right]$. Then with probability at most $1 - \delta$, for all $t \in \left[ C_2 \ln \left( \frac{2}{\delta} \right), T \right)$ simultaneously, $|M_t| \leq \frac{2}{3e} t$ and

$$|M_t| \leq 2 \sqrt{t \ln \left( \frac{2t}{|M_t| + \sqrt{t/3}} \right) + \ln \left( \frac{1}{C_1 \delta} \right)}$$

(It suffices if $C_1 = (\frac{420}{117})^2$ and $C_2 = 164$.)

There are no previous results in this vein, to our knowledge. So a principal contribution of this chapter is to characterize the tradeoff between $t$ and $\delta$ in uniform concentration of measure over time, which is not addressed by the classical LIL of Theorem 17. Theorems 18 and 19 constitute a sharp finite-time version of the LIL, analogous to how the Hoeffding bound for large deviations is a sharp finite-time version of the CLT’s Gaussian tail for a fixed time.

### 12.1.2 Chapter Outline

The proofs of both Theorems 18 and 19 are possibly of independent interest, and the rest of this chapter details and builds on them.
The proof of Theorem 18 extends the exponential moment method, the standard way of proving classical Chernoff-style bounds which hold for a fixed time. We use a technique, manipulating stopping times of a particular averaged supermartingale, which generalizes easily to many discrete- and continuous-time martingales, allowing us to prove iterated-logarithm concentration bounds for them as well. These martingale generalizations are given in Section 12.2, with discussion of even more general settings like continuous time.

The proof of Theorem 18 is of interest in demonstrating our techniques, so we include it here in Section 12.3. It has a loose proportionality constant and is not particularly direct, to better illustrate our technique. But the tightest and sharpest proof we know for the concentration bounds can be found as the main proof of Ch. 15.

Other proof details are deferred to the appendices. The proof of the anti-concentration bound of Theorem 19 basically inverts the argument used to prove the concentration bounds; the details involve some lengthy analytic estimates and so are deferred to Section C.1. Finally, Section C.2 contains some ancillary results, and formalizes extensions discussed in the previous sections.

12.2 Uniform Concentration Bounds for Martingales

Here we present extensions of the random walk concentration result of Theorem 18 to broader classes of martingales. Some notation must be established first, recapitulating our setup using martingales from earlier.

We study the behavior of a real-valued stochastic process \( M_t \) in a filtered probability space \( (\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P) \), where \( M_0 = 0 \) w.l.o.g. For simplicity, only the discrete-time case \( t \in \mathbb{N} \) is considered hereafter; the results and proofs in this chapter extend to continuous time as well (Remark 1). Define the difference sequence \( \xi_t = M_t - M_{t-1} \) for all \( t \) (\( \xi_t \) being \( \mathcal{F}_t \)-measurable), and the cumulative conditional variance and quadratic variation:
\[ V_t = \sum_{i=1}^{t} \mathbb{E} \left[ \xi_i^2 \mid \mathcal{F}_{i-1} \right] \quad \text{and} \quad Q_t = \sum_{i=1}^{t} \xi_i^2 \] respectively.

Also recall the following standard definitions. A martingale \( M_t \) (resp. supermartingale, submartingale) has \( \mathbb{E} [\xi_t \mid \mathcal{F}_{i-1}] = 0 \) (resp. \( \leq 0 \), \( \geq 0 \)) for all \( t \). A stopping time \( \tau \) is a function on \( \Omega \) such that \( \{ \tau \leq t \} \in \mathcal{F}_t \ \forall t \); notably, \( \tau \) can be infinite with positive probability ([Dur10]; [Doo84], Sec. 2.III.7).

### 12.2.1 Uniform Second-Moment Martingale Concentration

A few pertinent generalizations of Theorem 18 are now presented. The first is a direct iterated-logarithm analogue of Hoeffding’s inequality ([Hoe63]) for martingales.

**Theorem 20 (Uniform Hoeffding Bound).** Let \( M_t \) be a martingale, and suppose there are constants \( \{c_i\}_{i \geq 1} \) such that for all \( t \geq 1 \), \( |M_t - M_{t-1}| \leq c_t \) w.p. 1. Fix any \( \delta < 1 \) and define \( \tau_0 = \min \{ s : \sum_{i=1}^{s} c_i^2 \geq 173 \ln \left( \frac{4}{3} \right) \} \). Then with probability \( \geq 1 - \delta \), for all \( t \geq \tau_0 \) simultaneously, \( |M_t| \leq \frac{\sum_{i=1}^{t} c_i^2}{e^2 \left( 1 + \sqrt{1/3} \right)} \) and

\[
|M_t| \leq \sqrt{3 \left( \sum_{i=1}^{t} c_i^2 \right) \left( 2 \ln \ln \left( \frac{3 (\sum_{i=1}^{t} c_i^2)}{2 |M_t|} \right) + \ln \left( \frac{2}{\delta} \right) \right)}.
\]

A uniform counterpart to Bernstein’s inequality can be derived similarly.

**Theorem 21 (Uniform Bernstein Bound).** Let \( M_t \) be a martingale with uniformly bounded differences: \( |M_t - M_{t-1}| \leq e^2 \) w.p. 1 for all \( t \geq 1 \). Fix any \( \delta < 1 \) and define \( \tau_0 = \min \{ s : 2(e - 2)V_s \geq 173 \ln \left( \frac{4}{3} \right) \} \). Then with probability \( \geq 1 - \delta \), for all \( t \geq \tau_0 \) simultaneously, \( |M_t| \leq \frac{2(e-2)}{e^2 \left( 1 + \sqrt{1/3} \right)} V_t \) and

\[
|M_t| \leq \sqrt{6(e - 2)V_t \left( 2 \ln \ln \left( \frac{3(e-2)V_t}{|M_t|} \right) + \ln \left( \frac{2}{\delta} \right) \right)}.
\]

As with other Bernstein-type inequalities, the boundedness assumption on \( \xi_t \) can
be replaced by higher moment conditions (e.g. Lemma 61 and the preceding discussion).

The proofs of Theorems 20 and 21 are nearly identical to that of Theorem 18. Further details on these topics are given in Section C.2.2.

12.2.2 Discussion of Results

Some remarks on our results are in order.

Remark 1 (Extension to Continuous Time). In many cases, our uniform results can be generalized to continuous-time martingales with an almost unchanged proof (e.g., this is true for the Wiener process \( W_t \)). Further explanation of this depends on the proof details, and therefore is deferred to Section 12.2.3.

Remark 2 (Extension to Super(sub)martingale Bounds). One-sided variants of Theorem 18 hold in many cases for super- (resp. sub-) martingales, giving a uniform upper (resp. lower) bound identical to that in Theorem 18. When the Doob-Meyer decomposition ([Dur10]) applies, such bounds are immediate.

Remark 3 (Removal of Initial Time Conditions). The upper concentration bounds in this chapter include an initial time condition \( t \geq \tau_0 \). For Theorem 2, it is straightforward to remove this condition without degrading the result: if \( t < \tau_0 \), an explicit union bound over fixed-time Hoeffding bounds immediately gives that \( |M_t| \leq O \left( \ln \left( \frac{1}{\delta} \right) \right) \) w.h.p. for all \( t < \tau_0 \). Combining this with Theorem 2 gives a uniform bound, over all times \( t \geq 0 \), of

\[
|M_t| \leq O \left( \sqrt{t \left( \ln \ln t + \ln \frac{1}{\delta} \right)} + \ln \left( \frac{1}{\delta} \right) \right).
\]

The \( t \)-independent additive \( \ln \left( \frac{1}{\delta} \right) \) term matches standard Bernstein/Bennett concentration inequalities ([BLM13a]) for a fixed time.

To extend this argument to the upper bounds of Theorems 20 and 21, such an explicit union bound is no longer usable. But our proof techniques using stopping times extend naturally to these cases – see Section C.2.3.

When considering uniform martingale concentration over all times without an explicit union bound, the basic tools are Doob’s maximal inequality for nonnegative
supermartingales ([Dur10], Exercise 5.7.1), Hoeffding’s maximal inequality ([Hoe63]), and Freedman’s Bernstein-type inequality ([Fre75]). These can all be easily proved with the techniques of this chapter (similar to the proof of Theorem 26). However, the latter two results are fundamentally weaker than ours, in that they only hold uniformly over a finite time interval, and degrade to triviality as the interval grows infinite.

Our uniform Hoeffding/Bernstein-type bounds in Section 12.2.1 achieve optimal rates in the variance and $\delta$ parameters as well. These generalize martingale LILs like the classic result of Stout ([Sto70]), which for large classes of martingales makes a statement similar to Theorem 17, except concerning the ratio $|M_t|/\sqrt{V_t \ln \ln V_t}$. The finite-time upper Hoeffding-bound LIL can be proved with an epoch-based approach ([JMNB14]) standard in proofs of the (asymptotic) LIL ([Dur10]). Our technique can be viewed as generalizing that idea using stopping time manipulations.

Theorem 18’s tradeoff between $t$ and $\delta$ describes some of the interplay between the CLT and the LIL when uniform bounds are taken of partial sums of suitable i.i.d. variables. A similar question has been explored with a different statistical emphasis by the classic paper [DE56] and a line of subsequent work, though only as $t \to \infty$ to our knowledge.

### 12.2.3 Discussion of Proof

Most of the tools used in this proof, particularly optional stopping as in Theorem 33, extend seamlessly to the continuous-time case. The main potential obstacle to this is in the first step – establishing an exponential supermartingale construction of the form of Lemma 24. This is easily done in many situations of interest, as demonstrated by the archetypal result that the standard geometric Brownian motion $\exp \left( \lambda W_t - \frac{\lambda^2 t}{2} \right)$ is precisely a martingale for any $\lambda \in \mathbb{R}$, where $W_t$ is the standard Wiener process.

A direct antecedent to this chapter is a pioneering line of work by Robbins and
colleagues ([DR67a, RS70, Rob70a]) that investigates the powerful method of averaging martingales. For the most part, it only considers the asymptotic regime, though Darling and Robbins ([DR67a]) do briefly treat finite times (with far weaker $\delta$ dependence). More recently, de la Peña et al. ([dlPnKL07] and references therein) revisit their techniques with a different emphasis and normalization for $M_t$.

The idea of using stopping times in the context of uniform martingale concentration goes back at least to work of Robbins ([Rob70a]) with Siegmund ([RS70]), and was then notably used by Freedman ([Fre75]).

Our proof techniques are conceptually related to ideas from Shafer and Vovk ([SV01], Ch. 5), who describe how to view the LIL as emerging from a game. Departing from traditional approaches, they motivate the exponential supermartingale construction directly using Taylor expansions and prove the (asymptotic) LIL by averaging such supermartingales.

Two final connections bear mentioning. First, our proof technique incorporates relative variation ($|M_t|/U_t$) at multiple scales at once. There is an analogy to well-developed general chaining techniques ([Tal05]) that have been used to great effect to uniformly bound processes indexed on metric spaces; this is possible for us because the index set (time) is totally ordered. Second, it has been previously noted ([Rog02, HK04]) that there is a duality relationship between martingale stopping times and uniform concentration, and our technique provides another connection between them.

### 12.3 Proof of Theorem 18 and Concentration Bounds

Define the (deterministic) process $U_t = t$ (a notational convenience, to ease extension of this proof to the martingale case discussed in Section 12.2). Also define $k := \frac{1}{3}$ and $\lambda_0 := \frac{1}{e^2(1+\sqrt{k})}$.

In this section, we prove the following bound, which is a slightly more precise
version of Theorem 18:

**Theorem 22.** Let $M_t$ be a Rademacher random walk. Fix any $\delta < 1$ and define the time $\tau_0 = \min\left\{ s : U_s \geq \frac{2}{\lambda_0} \ln\left(\frac{4}{\delta}\right) \right\}$. Then with probability $\geq 1 - \delta$, for all $t \geq \tau_0$ simultaneously, $|M_t| \leq \lambda_0 U_t$ and

$$|M_t| \leq \frac{2U_t}{1 - k} \ln \left(\frac{2\ln^2 \left(\frac{U_t}{(1 - \sqrt{k}) |M_t|}\right)}{\delta}\right)$$

The proof invokes the Optional Stopping Theorem.

**Theorem 23 (Optional Stopping for Nonnegative Supermartingales ([Dur10], Theorem 5.7.6)).** Let $M_t$ be a nonnegative supermartingale. Then if $\tau$ is a (possibly infinite) stopping time, $E[M_{\tau}] \leq E[M_0]$.

The version we use here explicitly exploits the favorable convergence properties of nonnegative supermartingales when the stopping time is infinite.

Our argument begins by appealing to a standard exponential supermartingale construction.

**Lemma 24.** The process $X^\lambda_t := \exp\left(\lambda M_t - \frac{\lambda^2}{2} U_t \right)$ is a supermartingale for any $\lambda \in \mathbb{R}$.

**Proof.** Using Hoeffding’s Lemma, for any $\lambda \in \mathbb{R}$ and $t \geq 1$, $E[\exp(\lambda \xi_t) | \mathcal{F}_{t-1}] \leq \exp\left(\frac{\lambda^2}{8} (2^2)\right) = \exp\left(\frac{\lambda^2}{2}\right)$.

Therefore, $E[\exp(\lambda \xi_t - \frac{\lambda^2}{2}) | \mathcal{F}_{t-1}] \leq 1$, so $E[X^\lambda_t | \mathcal{F}_{t-1}] \leq X^\lambda_{t-1}$.

The result is derived through various manipulations of this supermartingale $X^\lambda_t$.

For the rest of the proof, for all $t$, assume that $M_t \neq 0$. This is with full generality, because when $M_t = 0$, the bound of Theorem 18 trivially holds.
12.3.1 A Time-Uniform Law of Large Numbers

The desired result, Theorem 22, uniformly controls \( |M_t| / \sqrt{U_t \ln \ln U_t} \), but we first control \( |M_t| / U_t \). This generalizes the (strong) law of large numbers ([S]LLN), for any failure probability \( \delta > 0 \), uniformly over finite times. While a weaker result than Theorem 22, this concisely demonstrates our principal proof techniques, and we independently also use it to prove the main bound.

The first step is to establish a moment bound which holds at any stopping time, by averaging supermartingales from the family \( \{ \exp \left( \lambda M_t - \frac{\lambda^2}{2} U_t \right) \} \) \( \lambda \in \mathbb{R} \) using a particular weighting over \( \lambda \).

**Lemma 25.** For any stopping time \( \tau \), \( \mathbb{E} \left[ \exp \left( \lambda_0 |M_\tau| - \frac{\lambda_0^2}{2} U_\tau \right) \right] \leq 2. \)

**Proof.** Recall the definition of \( X_\tau^\lambda \) from Lemma 24. Here we set the free parameter \( \lambda \) in the process \( X_\tau^\lambda \) to get a process \( Y_t \). \( \lambda \) is set stochastically: \( \lambda \in \{-\lambda_0, \lambda_0\} \) with probability \( \frac{1}{2} \) each. After marginalizing over \( \lambda \), the resulting process is

\[
Y_t = \frac{1}{2} \exp \left( \lambda_0 M_t - \frac{\lambda_0^2}{2} U_t \right) + \frac{1}{2} \exp \left( -\lambda_0 M_t - \frac{\lambda_0^2}{2} U_t \right) \\
\geq \frac{1}{2} \exp \left( \lambda_0 |M_t| - \frac{\lambda_0^2}{2} U_t \right)
\]

(12.1)

Now take \( \tau \) to be any stopping time as in the lemma statement.

Then \( \mathbb{E} \left[ \exp \left( \lambda_0 M_\tau - \frac{\lambda_0^2}{2} U_\tau \right) \right] = \mathbb{E} \left[ X_\tau^{\lambda = \lambda_0} \right] \leq 1 \), where the inequality is by the Optional Stopping Theorem (Theorem 33). Similarly, \( \mathbb{E} \left[ X_\tau^{\lambda = -\lambda_0} \right] \leq 1. \)

So \( \mathbb{E} [Y_\tau] = \frac{1}{2} \left( \mathbb{E} \left[ X_\tau^{\lambda = -\lambda_0} \right] + \mathbb{E} \left[ X_\tau^{\lambda = \lambda_0} \right] \right) \leq 1. \) Combining this with (13.3) gives the result. \( \square \)

A particular setting of \( \tau \) extracts the desired uniform LLN bound from Lemma 25.
Theorem 26. Fix $\delta > 0$. With probability $\geq 1 - \delta$, for all $t \geq \min \left\{ t : U_t \geq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right) \right\}$ simultaneously,

$$\frac{|M_t|}{U_t} \leq \lambda_0$$

Proof. For convenience define $\tau_1 = \min \left\{ t : U_t \geq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right) \right\}$. Define the stopping time $\tau = \min \left\{ t \geq \tau_1 : \frac{|M_t|}{U_t} > \lambda_0 \right\}$. Then it suffices to prove that $P(\tau < \infty) \leq \delta$.

On the event $\{ \tau < \infty \}$, we have $\frac{|M_t|}{U_t} > \lambda_0$ by definition of $\tau$. Therefore, using Lemma 25,

$$2 \geq \mathbb{E} \left[ \exp \left( \lambda_0 |M_\tau| - \frac{\lambda_0^2}{2} U_\tau \right) \right] \geq \mathbb{E} \left[ \exp \left( \lambda_0 |M_\tau| - \frac{\lambda_0^2}{2} U_\tau \right) | \tau < \infty \right] P(\tau < \infty)\tag{a}$$

$$> \mathbb{E} \left[ \exp \left( \lambda_0^2 U_\tau - \frac{\lambda_0^2}{2} U_\tau \right) \right] P(\tau < \infty) \geq \frac{2}{\delta} P(\tau < \infty)\tag{b}$$

where $(a)$ uses that $\frac{|M_t|}{U_t} > \lambda_0$ when $\tau < \infty$, and $(b)$ uses $U_\tau \geq U_{\tau_1} \geq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right)$. Therefore, $P(\tau < \infty) \leq \delta$, as desired. $\square$

The process $U_t$ is increasing in any case of interest, implying that $|M_t|/U_t \leq \lambda_0$ uniformly in $t$ after some finite initial time. The setting of $\lambda_0$ happens to fit with the rest of our main proof, but this choice of $\lambda_0$ in Theorem 26 is arbitrary. The same proof method in fact defines a family of bounds parametrized by $\lambda_0$; collectively, these express the SLLN for finite times.

12.3.2 Proof of Theorem 22

We proceed to prove Theorem 22, using the SLLN bound of Theorem 26 and its proof techniques.

Preliminaries

A little further notation is required for the rest of the proof.
For any event $E \subseteq \Omega$ of nonzero measure, let $\mathbb{E}_E [\cdot]$ denote the expectation restricted to $E$, i.e. $\mathbb{E}_E [f] = \frac{1}{P(E)} \int_E f(\omega)P(d\omega)$ for a measurable function $f$ on $\Omega$. Similarly, dub the associated measure $P_E$, where for any event $\Xi \subseteq \Omega$ we have $P_E(\Xi) = \frac{P(E \cap \Xi)}{P(E)}$.

Consider the “good” event of Theorem 26, in which its uniform deviation bound holds w.p. $\geq 1 - \delta$ for some $\delta$; call this event $A_\delta$. Formally,

$$A_\delta = \left\{ \omega \in \Omega : \frac{|M_t|}{U_t} \leq \lambda_0 \quad \forall t \geq \min \left\{ s : U_s \geq \frac{2}{\lambda_0^2 \ln \left( \frac{2}{\delta} \right)} \right\} \right\}$$

(12.2)

Theorem 26 states that $P(A_\delta) \geq 1 - \delta$.

It will be necessary to shift sample spaces from $A_\delta$ to $\Omega$. The shift should be small in measure because $P(A_\delta) \geq 1 - \delta$; this is captured by the following simple observation.

**Lemma 27.** Define $A_\delta$ as in (12.2). For any nonnegative random variable $X$ on $\Omega$,

$$\mathbb{E}_{A_\delta} [X] \leq \frac{1}{1 - \delta} \mathbb{E} [X]$$

**Proof.** Since $X \geq 0$, using Theorem 26,

$$\mathbb{E} [X] = \mathbb{E}_{A_\delta} [X] P(A_\delta) + \mathbb{E}_{A^c_\delta} [X] P(A^c_\delta) \geq \mathbb{E}_{A_\delta} [X] (1 - \delta). \quad \square$$

Define $X_t^\lambda$ as in Lemma 24. The idea of the proof is to choose $\lambda$ stochastically from a probability space $(\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)$. The parameter $\lambda$ is chosen independently of $\xi_1, \xi_2, \ldots$, so that $X_t^\lambda$ is defined on the product space. Write $\mathbb{E}_\lambda [\cdot]$ to denote the expectation with respect to $(\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)$.

To be consistent with previous notation, we continue to write $\mathbb{E} [\cdot]$ to denote the expectation w.r.t. the original probability space $(\Omega, \mathcal{F}, P)$ which encodes the stochasticity of $M_t$. As mentioned earlier, we use subscripts for expectations conditioned on events in this space, e.g. $\mathbb{E}_{A_\delta} [X]$. As an example, $\mathbb{E}_{\Omega} [\cdot] = \mathbb{E} [\cdot]$. 

Proof of Theorem 22

The main result can now be proved. The first step is to choose $\lambda$ stochastically in the supermartingale $X_t^\lambda$ and bound the effect of averaging over $\lambda$ (analogous to Lemma 25 in the proof of the bootstrap bound).

Lemma 28. Define $\tau_0$ as in Theorem 22, and $A_\delta$ as in (12.2) for any $\delta$. Then for any stopping time $\tau \geq \tau_0$,

$$\mathbb{E}_{A_\delta} \left[ \mathbb{E} \left[ X_t^\lambda \right] \right] \geq \mathbb{E}_{A_\delta} \left[ \frac{2 \exp \left( \frac{M_t^2}{2U_t} (1 - k) \right)}{\ln^2 \left( \frac{U_t}{(1 - \sqrt{k})|M_t|} \right)} \right]$$

The proof relies on estimating an integral for any outcome in the “good” event $A_\delta$, and is deferred to Section C.2.

Lemma 28 can be converted into the desired uniform bound using a particular choice of stopping time, analogously to how the bootstrap bound Theorem 26 is derived from Lemma 25. However, this time a shift in sample spaces is also needed to yield Theorem 22, since Lemma 28 uses $A_\delta$ instead of $\Omega$.

Proof of Theorem 22. Define the stopping time

$$\tau = \min \left\{ t \geq \tau_0 : |M_t| > \lambda_0 U_t \lor \left( |M_t| \leq \lambda_0 U_t \land |M_t| > \frac{2U_t}{1 - k} \ln \left( \frac{2 \ln^2 \left( \frac{U_t}{(1 - \sqrt{k})|M_t|} \right)}{\delta} \right) \right) \right\}$$

(12.3)
It suffices to prove that \( P(\tau = \infty) \geq 1 - \delta \). On the event \( \{\tau < \infty \cap A_{\delta/2}\} \), we have

\[
|M_\tau| > \sqrt{\frac{2U_\tau}{1-k}} \ln \left( \frac{2 \ln^2 \left( \frac{U_\tau}{(1-\sqrt{k})|M_\tau|} \right)}{\delta} \right) \iff \frac{2 \exp \left( \frac{M_\tau^2}{2U_\tau} (1-k) \right)}{\ln^2 \left( \frac{U_\tau}{(1-\sqrt{k})|M_\tau|} \right)} > \frac{4}{\delta}
\]

Therefore, using Lemma 28 and the nonnegativity of \( \frac{2 \exp \left( \frac{M_\tau^2}{2U_\tau} (1-k) \right)}{\ln^2 \left( \frac{U_\tau}{(1-\sqrt{k})|M_\tau|} \right)} \) on \( A_{\delta/2} \),

\[
2 \geq \frac{1}{1 - \frac{\delta}{2}} = \frac{\mathbb{E}^\lambda \left[ \mathbb{E} \left[ X_0^\lambda \right] \right]}{1 - \frac{\delta}{2}} \geq \frac{\mathbb{E}^\lambda \left[ \mathbb{E} \left[ X_\tau^\lambda \right] \right]}{1 - \frac{\delta}{2}} \geq \mathbb{E}^\lambda \left[ \mathbb{E}_{A_{\delta/2}} \left[ X_\tau^\lambda \right] \right] \geq \mathbb{E}^\lambda \left[ \mathbb{E}_{A_{\delta/2}} \left[ X_\tau^\lambda \right] \right] \\
\geq \mathbb{E}_{A_{\delta/2}} \left[ \mathbb{E}^\lambda \left[ X_\tau^\lambda \right] \right] \geq \mathbb{E}_{A_{\delta/2}} \left[ \frac{2 \exp \left( \frac{M_\tau^2}{2U_\tau} (1-k) \right)}{\ln^2 \left( \frac{U_\tau}{(1-\sqrt{k})|M_\tau|} \right)} \right] \]
\[
\geq \mathbb{E}_{A_{\delta/2}} \left[ \frac{2 \exp \left( \frac{M_\tau^2}{2U_\tau} (1-k) \right)}{\ln^2 \left( \frac{U_\tau}{(1-\sqrt{k})|M_\tau|} \right)} \right] \mid \tau < \infty \quad P_{A_{\delta/2}}(\tau < \infty) \geq \frac{4}{\delta} P_{A_{\delta/2}}(\tau < \infty)
\]

where (a) is by Optional Stopping (Theorem 33; note that \( \tau \) can be unbounded), (b) is by Lemma 27, (c) is by Tonelli’s Theorem, (d) is by Lemma 28, and (e) is by the definitions of \( A_{\delta/2} \) in (12.2) and of \( \tau \) in (12.3).

After simplification, this gives

\[
P_{A_{\delta/2}}(\tau < \infty) \leq \frac{\delta}{2} \implies P_{A_{\delta/2}}(\tau = \infty) \geq 1 - \frac{\delta}{2} \quad (12.4)
\]
Therefore,

\[ P(\tau = \infty) \geq P(\{\tau = \infty\} \cap A_{\delta/2}) \overset{(a)}{=} P_{A_{\delta/2}}(\tau = \infty)P(A_{\delta/2}) \]

\[ \overset{(b)}{\geq} \left( 1 - \frac{\delta}{2} \right) \left( 1 - \frac{\delta}{2} \right) \geq 1 - \delta \]

where (a) uses the definition of $P_{A_{\delta/2}}(\cdot)$ and (b) uses (12.4) and Theorem 26. The result follows. \qed
Chapter 13

Uniform PAC-Bayes Concentration of Martingale Mixtures

13.1 Introduction

We have seen that despite their mighty generality, martingales exhibit essentially the same well-understood concentration behavior as simple random walks.

Even more powerful concentration results can be obtained by considering aggregates of many martingales, averaged according to some distribution. Though these too have long been studied asymptotically ([Tom75]), their non-asymptotic study was only initiated by a recent paper of Seldin et al. ([SLCB+12]), which proves the sharpest known results on concentration of martingale mixtures, uniformly over the mixing distribution in a “PAC-Bayes” sense which is essentially optimal for such bounds ([Ban06]). This is motivated by and originally intended for applications in learning theory, as further discussed in that paper.

In this manuscript, we simplify, strengthen, and subsume results of [SLCB+12]. While that paper follows classical central-limit-theorem-type concentration results in focusing on an arbitrary fixed time, we instead leverage a recent method in Chapter 12 to achieve concentration that is uniform over finite times, extending the law of the iterated logarithm (LIL), with a rate at least as good as [SLCB+12] and often far superior.
In short, our bounds on mixtures of martingales are uniform both over the mixing distribution in a PAC-Bayes sense, and over finite times, all simultaneously (and optimally). This has no precedent in the literature.

### 13.1.1 Preliminaries

To formalize this, consider a set of discrete-time stochastic processes \( \{M_t(h)\}_{h \in \mathcal{H}} \), in a filtered probability space \((\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)\), where \( \mathcal{H} \) is possibly uncountable.\(^1\) Define the corresponding difference sequences \( \xi_t(h) = M_t(h) - M_{t-1}(h) \) (which are \( \mathcal{F}_t \)-measurable for any \( h, t \)) and conditional variance processes \( V_t(h) = \sum_{i=1}^t \mathbb{E} \left[ \xi_i^2(h) \mid \mathcal{F}_{i-1} \right] \). The mean of the processes at time \( t \) w.r.t. any distribution \( \rho \) over \( \mathcal{H} \) (as we write) is denoted by \( \mathbb{E}_\rho[M_t] := \mathbb{E}_{h \sim \rho}[M_t(h)] \), with \( \mathbb{E}_\rho[V_t] \) being defined similarly. For brevity, we drop the index \( h \) when it is clear from context.

Also recall the following standard definitions from martingale theory. For any \( h \in \mathcal{H} \), a martingale \( M_t \) (resp. supermartingale, submartingale) has \( \mathbb{E}[\xi_t \mid \mathcal{F}_{t-1}] = 0 \) (resp. \( \leq 0 \), \( \geq 0 \)) for all \( t \). A stopping time \( \tau \) is a function on \( \Omega \) such that \( \{\tau \leq t\} \in \mathcal{F}_t \) for all \( t \); notably, \( \tau \) can be infinite with positive probability ([Dur10]).

It is now possible to state our main result.

**Theorem 29 (PAC-Bayes Martingale Bernstein LIL Concentration).** Let \( \{M_t(h)\}_{h \in \mathcal{H}} \) be a set of martingales and fix a distribution \( \pi \in \Delta(\mathcal{H}) \) and a \( \delta < 1 \). Suppose the difference sequence is uniformly bounded: \( |M_t(h) - M_{t-1}(h)| \leq e^2 \) for all \( t \geq 1 \) and \( h \in \mathcal{H} \) w.p. 1. Then with probability \( \geq 1 - \delta \), the following is true for all \( \rho \in \Delta(\mathcal{H}) \).

Suppose

\[
\tau_0(\rho) = \min \left\{ s : 2(e-2)\mathbb{E}_\rho[V_s] \geq 2e^4 \left(1 + \sqrt{1/3}\right)^2 \left(\ln \left(\frac{4}{\delta}\right) + KL(\rho \mid\mid \pi)\right) \right\}
\]

\(^1\)As in Chapter 12, we just consider discrete time for convenience; the core results and proofs in this paper extend to continuous time as well, as well as other generalizations discussed in that paper.
Then for all $t \geq \tau_0(\rho)$ simultaneously, $|\mathbb{E}_\rho [M_t]| \leq \frac{2(e-2)}{e^2(1+\sqrt{1/3})} \mathbb{E}_\rho [V_t]$ and

$$
|\mathbb{E}_\rho [M_t]| \leq \sqrt{6(e-2)|\mathbb{E}_\rho [V_t]| \left(2 \ln \left( \frac{3(e-2)|\mathbb{E}_\rho [V_t]|}{|\mathbb{E}_\rho [M_t]|} \right) + \ln \left( \frac{2}{\delta} \right) + KL(\rho \| \pi) \right)}
$$

(This bound is implicit in $|\mathbb{E}_\rho [M_t]|$, but notice that either $|\mathbb{E}_\rho [M_t]| \leq 1$ or the iterated-logarithm term can be simply treated as $2 \ln \ln \left( \frac{3(e-2)|\mathbb{E}_\rho [V_t]|}{|\mathbb{E}_\rho [M_t]|} \right)$, making the bound explicit.)

As we mentioned, Theorem 29 is uniform over $\rho$ and $t$, allowing us to track mixtures of martingales tightly as they evolve. The martingales indexed by $\mathcal{H}$ can be highly dependent on each other, as they share a probability space. For instance, $M_{t_0}(h_0)$ can depend in arbitrary fashion on $\{M_{t \leq h_0}(h)\}_{h \neq h_0}$; it is only required to satisfy the martingale property for $h_0$, as further discussed in [SLCB+12]. So these inequalities have found use in analyzing sequentially dependent processes such as those in reinforcement learning and bandit algorithms, where the choice of prior $\pi$ can be tailored to the problem ([SASt+11]) and the posterior can be updated based on information learned up to that time.

The method of proof is essentially that used in Chapter 12. Our main observation in this manuscript is that this proof technique is, in a technical sense, quite complementary to the fundamental method used in all PAC-Bayes analysis ([Ban06]). This allows us to prove our results in a sharper and more streamlined way than previous work ([SLCB+12]).

13.1.2 Discussion

Let us elaborate on these claims. Theorem 29 can be compared directly to the following PAC-Bayes Bernstein bound from [SLCB+12] that holds for a fixed time:
Theorem 30 (Restatement\(^2\) of Thm. 8 from [SLCB\(^+\)12]). Fix any \( t \). For \( \rho \) such that \( KL(\rho \mid\mid \pi) \) is sufficiently low compared to \( \mathbb{E}_\rho[V_t] \),

\[
|\mathbb{E}_\rho[M_t]| \leq \sqrt{(1+e)^2(e-2)\mathbb{E}_\rho[V_t] \left( \ln \ln \left( \frac{(e-2)t}{\ln(2/\delta)} \right) + \ln \left( \frac{2}{\delta} \right) + KL(\rho \mid\mid \pi) \right)}
\]

This bound is inferior to Theorem 29 in two significant ways: \(^3\) it holds for a fixed time rather than uniformly over finite times, and has an iterated-logarithm term of \( \ln \ln t \) rather than \( \ln \ln V_t \). The latter is a very significant difference when \( V_t \ll t \), which is precisely when Bernstein inequalities would be preferred to more basic inequalities like Chernoff bounds.

Put differently, our non-asymptotic result, like those of Chapter 12, adapts correctly to the scale of the problem. Indeed, Theorem 29 is optimal by the (asymptotic) martingale LIL, e.g. the seminal result of Stout ([Sto70]); this is true non-asymptotically too, by the main anti-concentration bound of Ch. 12. All these optimality results are for a single martingale, but suffice for the PAC-Bayes case as well; and the additive \( KL(\rho \mid\mid \pi) \) cost of uniformity over \( \rho \) is unimprovable in general also, by standard PAC-Bayes arguments.

**Proof Overview**

Our method follows that of Ch. 12, departing from the more traditionally learning-theoretic techniques used in [SLCB\(^+\)12]. We embark on the proof by introducing a standard exponential supermartingale construction that holds for any of the martingales \( \{M_t(h)\}_{h \in \mathcal{H}} \) (this is identical to Lemma 59).

\(^2\)The actual statement of the theorem in [SLCB\(^+\)12], though not significantly different, is more complicated because of a few inconvenient artifacts of that paper’s more complicated analysis, none of which arise in our analysis.

\(^3\)As discussed in previous chapters (cf. [Bal15b, Bal15a]), the result can be extended to less general settings as for Hoeffding bounds, though our comparison with the results of [SLCB\(^+\)12] remains the same.
Lemma 31. Suppose $|\xi_t| \leq \epsilon^2$ a.s. for all $t$. Then for any $h \in \mathcal{H}$, the process $X_t^\lambda(h) := \exp(\lambda M_t(h) - \lambda^2(e - 2)V_t(h))$ is a supermartingale in $t$ for any $\lambda \in \left[-\frac{1}{\epsilon^2}, \frac{1}{\epsilon^2}\right]$.

The classical martingale Bernstein inequality for a given $h$ and fixed time $t$ can be proved by using Markov’s inequality with $E[X_t^{\lambda^*}]$, where $\lambda^* \propto \frac{|M_t|}{V_t}$ is tuned for the tightest bound, and can be thought of as setting the relative scale of variation being measured.

The proof technique of this paper and its advantages over previous work are best explained by examining how to set the scale parameter $\lambda$.

Choosing the Scale Parameter

On a high level, the main idea of Chapter 12 is to average over a random choice of the scale parameter $\lambda$ in the supermartingale $X_t^\lambda$. This allows manipulation of a stopped version of $X_t^\lambda$, i.e. $X_t^\lambda$ for a particular stopping time $\tau$. So the averaging technique in Chapter 12 can be thought of as using “many values of $\lambda$ at once,” which is necessary when dealing with the stopped process because $\tau$ is random, and so is $\frac{|M_\tau|}{V_\tau}$.

All existing PAC-Bayes analyses achieve uniformity in $\rho$ through the Donsker-Varadhan variational characterization of the KL divergence:

Lemma 32 (Donsker-Varadhan Lemma ([DV76])). Suppose $\rho$ and $\pi$ are probability distributions over $\mathcal{H}$, and let $f(\cdot) : \mathcal{H} \mapsto \mathbb{R}$ be a measurable function. Then

$$E_\rho[f(h)] \leq KL(\rho \mid\mid \pi) + \ln(E_\pi[e^{f(h)}])$$

This introduces a $KL(\rho \mid\mid \pi)$ term into the bounding of $X_t^\lambda$. However, the optimum $\lambda^*$ is then dependent on the unknown $\rho$. The solution adopted by existing PAC-Bayes martingale bounds ([SLCB+12] and variants) is again to use “many values of $\lambda$ at once.” In prior work, this is done explicitly by taking a union bound over a grid of carefully
chosen $\lambda$s.

Our main technical contribution is to recognize the similarity between these two problems, and to use the (tight) stochastic choice of $\lambda$ in Chapter 12 as a solution to both problems at once, achieving the optimal bound of Theorem 29.

### 13.2 Overview of Proof

We now outline the complete proof of Theorem 29, following the presentation of Chapter 12 closely. Define $U_t := 2(e - \lambda)V_t$. As in Chapter 12, our proof invokes the Optional Stopping Theorem, in particular a version for nonnegative supermartingales that neatly exploits their favorable convergence properties:

Theorem 33 (Optional Stopping for Nonnegative Supermartingales ([Dur10], Theorem 5.7.6)). If $M_t$ is a nonnegative supermartingale and $\tau$ a (possibly infinite) stopping time, $E[M_\tau] \leq E[M_0]$.

We also use the exponential supermartingale construction of Lemma 31, which we assume to hold for $M_t(h) \forall h \in \mathcal{H}$ since they are all martingales whose concentration we require.

Our goal is to control $E_\rho[M_t]$ in terms of $E_\rho[U_t]$, so it is tempting to try to show that $e^{\lambda E_\rho[M_t] - \lambda^2 E_\rho[U_t]}$ is an exponential supermartingale. However, this is not generally true; and even if it were, would only control $E[e^{\lambda E_\rho[M_t] - \lambda^2 E_\rho[U_t]}]$ for a fixed $\rho$, not in a PAC-Bayes sense.

We instead achieve uniform control over $\rho$ by using the Donsker-Varadhan lemma to mediate the Optional Stopping Theorem every time it is used in the proof of the single-martingale finite LIL in Chapter 12. This is fully captured in the following key result, a powerful extension of a standard moment-generating function bound that is uniform in $\rho$ and has enough freedom (an arbitrary stopping time $\tau$) to be converted into a
time-uniform bound.

**Lemma 34.** Choose any probability distribution $\pi$ over $\mathcal{H}$. Then for any stopping time $\tau$ and $\lambda \in \left[ -\frac{1}{e^2}, \frac{1}{e^2} \right]$, simultaneously for all distributions $\rho$ over $\mathcal{H}$,

$$
\mathbb{E} \left[ e^{\lambda \mathbb{E}_\rho[M_\tau] - \frac{\lambda^2}{2} \mathbb{E}_\rho[U_\tau]} \right] \leq e^{KL(\rho||\pi)}
$$

**Proof.** Using Lemma 32 with the function $f(h) = \lambda M_\tau(h) - \frac{\lambda^2}{2} U_\tau(h)$, and exponentiating both sides, we have for all posterior distributions $\rho \in \Delta(\mathcal{H})$ that

$$
e^{\lambda \mathbb{E}_\rho[M_\tau] - \frac{\lambda^2}{2} \mathbb{E}_\rho[U_\tau]} \leq e^{KL(\rho||\pi)} \mathbb{E}_\pi \left[ e^{\lambda M_\tau - \frac{\lambda^2}{2} U_\tau} \right] \quad (13.1)
$$

Therefore,

$$
\mathbb{E} \left[ e^{\lambda \mathbb{E}_\rho[M_\tau] - \frac{\lambda^2}{2} \mathbb{E}_\rho[U_\tau]} \right] \overset{(a)}{\leq} e^{KL(\rho||\pi)} \mathbb{E}_\pi \left[ e^{\lambda M_\tau - \frac{\lambda^2}{2} U_\tau} \right] \overset{(b)}{\leq} e^{KL(\rho||\pi)} \text{ where (a) is from (13.1) and Tonelli’s Theorem, and (b) is by Lemma 31 and Optional Stopping (Thm. 33).}
$$

Just as a bound on the moment-generating function of a random variable is the key to proving tight Hoeffding and Bernstein concentration of that variable, Lemma 34 is, exactly analogously, the key tool used to prove Theorem 29.

The remainder of the proof follows Chapter 12 extremely closely but uses the more general (13.1) instead of the supermartingale construction used there.

### 13.3 Proof of Theorem 29

For the rest of this section, define $U_t := 2(e - 2)V_t$, $k := \frac{1}{3}$, and $\lambda_0 := \frac{1}{e^{2/3 + \sqrt{k}}}$. 
13.3.1 A PAC-Bayes Uniform Law of Large Numbers

First, we define the stopping time
\[ \tau_0(\rho) := \min \left\{ t : \mathbb{E}_\rho [U_t] \geq \frac{2}{\lambda_0^2} \left( \ln \left( \frac{2}{\delta} \right) + KL(\rho \mid \mid \pi) \right) \right\} \]

and the following “good” event:
\[ B_\delta = \left\{ \omega \in \Omega : \forall \rho \in \Delta(\mathcal{H}), \quad \frac{\left| \mathbb{E}_\rho [M_t] \right|}{\mathbb{E}_\rho [U_t]} \leq \lambda_0 \quad \forall t \geq \tau_0(\rho) \right\} \quad (13.2) \]

Our first result introduces the reader to our main proof technique; it is a generalization of the law of large numbers (LLN) to our PAC-Bayes martingale setting.

**Theorem 35.** Fix any \( \delta > 0 \). With probability \( \geq 1 - \delta \), the following is true for all \( \rho \) over \( \mathcal{H} \): for all \( t \geq \tau_0(\rho) \),
\[ \frac{\left| \mathbb{E}_\rho [M_t] \right|}{\mathbb{E}_\rho [U_t]} \leq \lambda_0 \]

To prove this, we first manipulate Lemma 34 so that it is in terms of \( \left| \mathbb{E}_\rho [M_t] \right| \).

**Lemma 36.** Choose any prior \( \pi \in \Delta(\mathcal{H}) \). For any stopping time \( \tau \) and all distributions \( \rho \) over \( \mathcal{H} \),
\[ \mathbb{E} \left[ \exp \left( \lambda_0 \left| \mathbb{E}_\rho [M_t] \right| - \frac{\lambda_0^2}{2} \mathbb{E}_\rho [U_t] \right) \right] \leq 2e^{KL(\rho \mid \mid \pi)} \]

**Proof.** Lemma 34 describes the behavior of the process \( \chi^\lambda_t = e^{\lambda \mathbb{E}_\rho [M_t] - \frac{\lambda^2}{2} \mathbb{E}_\rho [U_t]} \). Define \( Y_t \) to be the mean of \( \chi^\lambda_t \) with \( \lambda \) being set stochastically: \( \lambda \in \{-\lambda_0, \lambda_0\} \) with probability \( \frac{1}{2} \) each. After marginalizing over \( \lambda \), the resulting process is
\[ Y_t = \frac{1}{2} \exp \left( \lambda_0 \mathbb{E}_\rho [M_t] - \frac{\lambda_0^2}{2} \mathbb{E}_\rho [U_t] \right) + \frac{1}{2} \exp \left( -\lambda_0 \mathbb{E}_\rho [M_t] - \frac{\lambda_0^2}{2} \mathbb{E}_\rho [U_t] \right) \]
\[ \geq \frac{1}{2} \exp \left( \lambda_0 \left| \mathbb{E}_\rho [M_t] \right| - \frac{\lambda_0^2}{2} \mathbb{E}_\rho [U_t] \right) \]
Now take $\tau$ to be any stopping time as in the lemma statement. Then
\[
\mathbb{E} \left[ \exp \left( \lambda_0 \mathbb{E}_\rho [M_{\tau}] - \frac{\lambda_0^2}{2} \mathbb{E}_\rho [U_{\tau}] \right) \right] \leq e^{KL(\rho \| \pi)}
\]
by Lemma 34. Similarly, $\mathbb{E} \left[ X_{\tau}^{\lambda = -\lambda_0} \right] \leq e^{KL(\rho \| \pi)}$. So $\mathbb{E} [Y_{\tau}] = \frac{1}{2} \left( \mathbb{E} \left[ X_{\tau}^{\lambda = -\lambda_0} \right] + \mathbb{E} \left[ X_{\tau}^{\lambda = \lambda_0} \right] \right) \leq e^{KL(\rho \| \pi)}$. Combining this with (13.3) gives the result.

A particular setting of $\tau$ extracts the desired uniform LLN bound from Lemma 36.

**Proof of Theorem 35.** Define the stopping time
\[
\tau = \min \left\{ t : \exists \rho \in \Delta(H) \text{ s.t. } t \geq \tau_0(\rho) \text{ and } \frac{\mathbb{E}_\rho [M_t]}{\mathbb{E}_\rho [U_t]} > \lambda_0 \right\}
\]
Then it suffices to prove that $P(\tau < \infty) \leq \delta$.

On the event $\{ \tau < \infty \}$, we have for some $\rho$ that $\frac{\mathbb{E}_\rho [M_{\tau}]}{\mathbb{E}_\rho [U_{\tau}]} > \lambda_0$ by definition of $\tau$. Therefore, using Lemma 36, we can conclude that for this $\rho$,
\[
2e^{KL(\rho \| \pi)} \geq \mathbb{E} \left[ \exp \left( \lambda_0 \frac{\mathbb{E}_\rho [M_{\tau}]}{\mathbb{E}_\rho [U_{\tau}]} - \frac{\lambda_0^2}{2} \frac{\mathbb{E}_\rho [U_{\tau}]}{\mathbb{E}_\rho [U_{\tau}]} \right) \mid \tau < \infty \right] P(\tau < \infty)
\]
\[
\geq \left( \frac{2}{\delta} e^{KL(\rho \| \pi)} \right) P(\tau < \infty)
\]
where $(b)$ uses $\mathbb{E}_\rho [U_{\tau}] \geq \mathbb{E}_\rho [U_{\tau_0}] \geq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} e^{KL(\rho \| \pi)} \right)$. Therefore, $P(\tau < \infty) \leq \delta$, as desired.

**13.3.2 Proof of Theorem 29**

For any event $E \subseteq \Omega$ of nonzero measure, let $\mathbb{E}_E [\cdot]$ denote the expectation restricted to $E$, i.e. $\mathbb{E}_E [f] = \frac{1}{P(E)} \int_E f(\omega)P(d\omega)$ for a measurable function $f$ on $\Omega$. 
Similarly, dub the associated measure $P_E$, where for any event $\Xi \subseteq \Omega$ we have $P_E(\Xi) = \frac{P(E \cap \Xi)}{P(E)}$.

Theorem 35 shows that $P(B_\delta) \geq 1 - \delta$. It is consequently easy to observe that the corresponding restricted outcome space can still be used to control expectations.

**Proposition 37.** For any nonnegative r.v. $X$, we have $E_{B_\delta}[X] \leq \frac{1}{1 - \delta} E[X]$.

**Proof.** Using Thm. 35, $E[X] = E_{B_\delta}[X] P(B_\delta) + E_{B_\delta}^c[X] P(B_\delta) \geq E_{B_\delta}[X] (1 - \delta)$. $\square$

Just as in Chapter 12, the idea of the main proof is to choose $\lambda$ stochastically from a probability space $(\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)$ such that $P_\lambda(d\lambda) = \frac{d\lambda}{|\lambda| \ln \frac{1}{|\lambda|}}$ on $\lambda \in [-e^{-2}, e^{-2}] \setminus \{0\}$. The parameter $\lambda$ is chosen independently of $\xi_1, \xi_2, \ldots$, so that $X^\lambda$ is defined on the product space. Write $E^\lambda[\cdot]$ to denote the expectation with respect to $(\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)$.

To be consistent with previous notation, we continue to write $E[\cdot]$ to denote the expectation w.r.t. the original probability space $(\Omega, \mathcal{F}, P)$. As mentioned earlier, we use subscripts for expectations conditioned on events in this space, e.g. $E_{A_\delta}[X]$. (As an example, $E_{\Omega}[\cdot] = E[\cdot]$.)

Before going on to prove the main theorem, we need one more result that controls the effect of averaging over $\lambda$ as above.

**Lemma 38.** For all $\rho \in \Delta(\mathcal{H})$ and any $\delta$, the following is true: for any stopping time $\tau \geq \tau_0(\rho)$,

$$E_{B_\delta} \left[ E^\lambda \left[ e^{\lambda E_{\rho}[M_\tau] - \frac{\lambda^2}{2} E_{\rho}[U_\tau]} \right] \right] \geq E_{B_\delta} \left[ 2 \exp \left( \frac{E_{\rho}[M_\tau]^2}{2 E_{\rho}[U_\tau]} (1 - k) \right) \right] \ln^2 \left( \frac{E_{\rho}[U_\tau]}{(1 - \sqrt{k}) |E_{\rho}[M_\tau]|} \right)$$

Lemma 38 is precisely analogous to Lemma 28 and proved using exactly the same calculations, so its proof is omitted here. Now we can prove Theorem 29.
Proof of Theorem 29. The proof follows precisely the same method as that of Chapter 12, but with a more nuanced setting of the stopping time $\tau$. To define it, we first for convenience define the deterministic function

$$
\zeta_t(\rho) = \frac{2\mathbb{E}_\rho[U_t]}{1-k} \ln \left( \frac{2 \ln^2 \left( \frac{\mathbb{E}_\rho[U_t]}{(1-\sqrt{k})\mathbb{E}_\rho[M_t]} \right)}{\delta} e^{KL(\rho||\pi)} \right)
$$

Now we define the stopping time

$$
\tau = \min \left\{ t : \exists \rho \in \Delta(H) \text{ s.t. } t \geq \tau_0(\rho) \text{ and } \left| \mathbb{E}_\rho[M_t] \right| > \zeta_t(\rho) \right\}
$$

The rest of the proof shows that $P(\tau = \infty) \geq 1 - \delta$, and involves nearly identical calculations to the main proof of Chapter 12 (with $\mathbb{E}_\rho[M_t], \mathbb{E}_\rho[U_t], B_\delta$ replacing what that paper writes as $M_t, U_t, A_\delta$).

It suffices to prove that $P(\tau = \infty) \geq 1 - \delta$. On the event $\left\{ \left\{ \tau < \infty \right\} \cap B_{\delta/2} \right\}$, by definition there exists a $\rho$ s.t.

$$
\left| \mathbb{E}_\rho[M_\tau] \right| > \zeta(\rho) = \frac{2\mathbb{E}_\rho[U_\tau]}{1-k} \ln \left( \frac{2 \ln^2 \left( \frac{\mathbb{E}_\rho[U_\tau]}{(1-\sqrt{k})\mathbb{E}_\rho[M_\tau]} \right)}{\delta} e^{KL(\rho||\pi)} \right)
$$

I.e. a $\rho$ such that

$$
2 \exp \left( \frac{\mathbb{E}_\rho[M_\tau]^2}{2\mathbb{E}_\rho[U_\tau]} (1-k) \right) > \frac{4}{\delta} e^{KL(\rho||\pi)} \quad (13.5)
$$

Consider this $\rho$. Using the nonnegativity of

$$
\frac{2 \exp \left( \frac{\mathbb{E}_\rho[M_\tau]^2}{2\mathbb{E}_\rho[U_\tau]} (1-k) \right)}{\ln^2 \left( \frac{\mathbb{E}_\rho[U_\tau]}{(1-\sqrt{k})\mathbb{E}_\rho[M_t]} \right)} \text{ on } B_{\delta/2} \text{ and letting}
$$
\[ Z_\tau^\lambda := e^{\lambda E_\rho[M_t] - \frac{\lambda^2}{2} E_\rho[U_t]} , \]

\[
2 e^{KL(\rho||\pi)} \geq \frac{e^{KL(\rho||\pi)}}{1 - \frac{\delta}{2}} \geq \frac{e^{\lambda \mathbb{E}[Z_\tau^\lambda]}}{1 - \frac{\delta}{2}} \geq e^{\lambda \mathbb{E}[B_{\delta/2}] Z_\tau^\lambda} \geq e^{\lambda \mathbb{E}[Z_\tau^\lambda]} \\
\geq 2 \exp \left( \frac{E_\rho[M_t]^2}{2 E_\rho[U_t]} (1 - k) \right) \ln \frac{E_\rho[M_t]}{E_\rho[U_t]} \left( \frac{1}{1 - \sqrt{k}} \right) P_{B_{\delta/2}}(\tau < \infty) \\
\geq \frac{4}{\delta} e^{KL(\rho||\pi)} P_{B_{\delta/2}}(\tau < \infty)
\]

where (a) is by Lemma 34, (b) is by Prop. 37, (c) is by Tonelli’s Theorem, (d) is by Lemma 38, and (e) is by (13.5).

After simplification, this gives

\[
P_{B_{\delta/2}}(\tau < \infty) \leq \delta / 2 \implies P_{B_{\delta/2}}(\tau = \infty) \geq 1 - \frac{\delta}{2} \tag{13.6}
\]

and using Theorem 35 – that \( P(B_{\delta/2}) \geq 1 - \frac{\delta}{2} \) – concludes the proof. \( \square \)
Chapter 14

Sequential Nonparametric Testing with the Law of the Iterated Logarithm

This chapter concerns an application of the finite-time LIL results to nonparametric statistical hypothesis testing.

14.1 Introduction

Nonparametric statistical decision theory poses the problem of making a decision between a null \((H_0)\) and alternate \((H_1)\) hypothesis over a dataset with the aim of controlling both false positives and false negatives (in statistics terms, maximizing power while controlling type I error), all without making assumptions about the distribution of the data being analyzed. Such hypothesis testing is based on a “stochastic proof by contradiction” – the null hypothesis is thought of by default to be true, and is rejected only if the observed data are statistically very unlikely under the null.

There is increasing interest in solving such problems in a “big data” regime, in which the sample size \(N\) can be huge. We present a sequential testing framework for these problems that is particularly suitable for two related scenarios prevalent in many applications:

1. The dataset is extremely large and high-dimensional, so even a single pass through
it is prohibitive.

2. The data is arriving as a stream, and decisions must be made with minimal storage.

Sequential tests have long been considered strong in such settings. Such a test accesses the data in an online/streaming fashion, assessing after every new datapoint whether it then has enough evidence to reject the null hypothesis. However, prior work tends to be univariate or parametric or asymptotic, while we are the first to provide non-asymptotic guarantees on multivariate nonparametric problems.

To elaborate on our motivations, suppose we have a gigantic amount of data from each of two unknown distributions, enough to detect even a minute difference in their means $\mu_1 - \mu_2$ if it exists. Further suppose that, unknown to us, deciding whether the means are equal is actually statistically easy ($|\mu_1 - \mu_2|$ is large), meaning that one can conclude $\mu_1 \neq \mu_2$ with high confidence by just considering a tiny fraction of the dataset. Can we take advantage of this, despite our ignorance of it?

A naive solution would be to discard most of the data and run a batch (offline) test on a small subset. However, we do not know how hard the problem is, and hence do not know how large a subset will suffice — sampling too little data might lead to incorrectly not rejecting the null, and sampling too much would unnecessarily waste computational resources. If we somehow knew $\mu_1 - \mu_2$, we would want to choose the fewest number of samples (say $n^*$) to reject the null while controlling type I error at some target level.

14.1.1 Overview of Our Approach

Our sequential test solves the problem by automatically stopping after seeing about $n^*$ samples, while still controlling type I and II errors almost as well as the equivalent linear-time batch test. Without knowing the true problem difficulty, we are able to detect it with virtually no computational or statistical penalty. We devise and
formally analyze a sequential algorithm for a variety of problems, starting with a basic
test of the bias of a coin, and developing this to nonparametric two-sample mean testing,
with further extensions to general nonparametric two-sample and independence testing.

Our proposed procedure only keeps track of a single scalar test statistic, which
we construct to be a zero-mean random walk under the null hypothesis. It is used to
test the null hypothesis each time a new data point is processed. A major statistical
issue is dealing with the apparent multiple hypothesis testing problem – if our algorithm
observes its first rejection of the null at time $t$, it might raise suspicions of being a false
rejection, because $t - 1$ hypothesis tests were already conducted and the $t$-th may have
been rejected purely by chance. Applying some kind of multiple testing correction, like
the Bonferroni or Benjamini-Hochberg procedure ([BH95]), is exceedingly conservative
and produces very suboptimal results over a large number of tests. However, since the
random walk moves only a relatively small amount every iteration, the tests are far from
independent.

Formalizing this intuition requires adapting the law of the iterated logarithm
(LIL), with which we control for type I error (when $H_0$ is true). The LIL can be described
as follows. Imagine tossing a fair coin, assigning $+1$ to heads and $-1$ to tails, and
keeping track of the sum $S_t$ of $t$ coin flips. The LIL asserts that asymptotically, $S_t$
always remains bounded between $\pm \sqrt{2t \ln \ln t}$ (and this “envelope” is tight).

When $H_1$ is true, we prove that the sequential algorithm does not need the whole
dataset as a batch algorithm would, but automatically stops after processing “just enough”
data points to detect $H_1$, depending on the unknown difficulty of the problem being
solved. The near-optimal nature of this adaptive type II error control (when $H_1$ is true) is
again due to the LIL.

As alluded to earlier, all of our test statistics can be thought of as random walks,
which behave like $S_t$ under $H_0$. The LIL then characterizes how such a random walk
behaves under $H_0$ – our algorithm will keep observing new data since the random walk values will simply bounce around within the LIL envelope. Under $H_1$, the random walk is designed to have nonzero mean, and hence will eventually stray outside the LIL envelope, at which point the process stops and rejects the null hypothesis.

For practically applying this argument to finite samples and reasoning about type II error and stopping times, we cannot use the classical asymptotic form of the LIL typically stated in textbooks such as [Fel50], instead adapting the finite-time LIL of Chapter 12 ([Bal15b]). As we will see, the technical contribution is necessary to investigate the stopping time, and control type I and II errors non-asymptotically and uniformly over all $t$.

In summary, our sequential testing framework has the following properties:

(A) Under $H_0$, it controls type I error, using a finite-time LIL computable in terms of empirical variance.

(B) Under $H_1$, and with type II error controlled at a target level, it automatically stops after seeing the same number of points as the corresponding computationally-constrained oracle batch algorithm.

(C) Each update takes $O(d)$ time and constant memory.

In later sections, we develop formal versions of these statements. The statistical observations, particularly the stopping time, follow from the finite-time LIL through simple concentration of measure arguments that extend to very general sequential testing settings, but have seemingly remained unobserved in the literature for decades because of the finite-time LIL necessary to make them.
14.1.2 Outline of the Chapter

We begin by describing a sequential test for the bias of a coin in Section 14.2. We then provide a sequential test for nonparametric two-sample mean testing in Section 14.3. We run extensive simulations in Section 14.4 to bear out predictions of our theory, followed by a comparison to the extensive existing literature on the subject in Section 14.5. We also include extensions to general nonparametric two-sample and independence testing problems, in Section 14.6. Some less instructive proofs, and more detailed notes about the algorithm, are deferred to the appendices (Sec. D).

1: Fix $N$ and compute $p_N$
2: if $S_N > p_N$ then
3: Reject $H_0$
4: else
5: Fail to reject $H_0$
6: end if

1: Fix $N$
2: for $n = 1$ to $N$ do
3: Compute $q_n$
4: if $S_n > q_n$ then
5: Reject $H_0$ and return
6: end if
7: end for
8: Fail to reject $H_0$

Figure 14.1. Batch (left) and sequential (right) tests.

14.2 Detecting the Bias of a Coin

This section will illustrate how a simple sequential test can perform statistically as well as the best batch test in hindsight, while automatically stopping essentially as soon as possible. We will show that such early stopping can be viewed as quite a general consequence of concentration of measure. Just for this section, let $K$ represent a constant that may take different values on each appearance, but is always absolute.

Consider observing i.i.d. binary flips $A_1, A_2, \ldots \in \{-1, +1\}$ of a coin, which may be fair or biased towards $+1$, with $P(A_i = +1) = \rho$. We want to test for fairness, detecting unfairness as soon as possible. Formulated as a hypothesis test, we wish to test,
for $\delta \in (0, \frac{1}{2}]$:

$$H_0 : \rho = \frac{1}{2} \quad \text{vs.} \quad H_1(\delta) : \rho = \frac{1}{2} + \delta$$

For any sample size $n$, the natural test statistic for this problem is $S_n = \sum_{i=1}^{n} A_i$. $S_n$ is a (scaled) simple mean-zero random walk under $H_0$. A standard hypothesis testing approach to our problem is a basic batch test involving $S_N$, which tests for deviations from the null for a fixed sample size $N$ (Fig. 14.1, left). A basic Hoeffding bound shows that

$$S_N \leq \sqrt{\frac{N}{2} \ln \frac{1}{\alpha}} =: p_N$$

with probability $\geq 1 - \alpha$ under the null, so type I error is controlled at level $\alpha$:

$$P_{H_0}(\text{reject } H_0) = P_{H_0}(S_N > p_N) \leq e^{-2p_N^2/N} = \alpha.$$

### 14.2.1 A Sequential Test

The main test we propose will be a sequential test as in Fig. 14.1. It sees examples as they arrive one at a time, up to a large time $N$, the maximum sample size we can afford. The sequential test is defined with a sequence of positive thresholds $\{q_n\}_{n \in [N]}$. We show how to set $q_n$ to justify statements (A) and (B) in Section 14.1.1.

**Type I Error.** Just as the batch threshold $p_N$ is determined by controlling the type I error with a concentration inequality, the sequential test also chooses $q_1, \ldots, q_N$ to control the type I error at $\alpha$:

$$P_{H_0}(\text{reject } H_0) = P_{H_0}(\exists n \leq N : S_n > q_n) \leq \alpha$$

(14.1)

This inequality concerns the uniform concentration over infinite tails of $S_n$, but what $\{q_n\}_{n \in [N]}$ satisfies it? Asymptotically, the answer is governed by the LIL, reproduced
Theorem 39 (Law of the iterated logarithm ([Khi24])). With probability 1,

\[ \limsup_{n \to \infty} \frac{S_n}{\sqrt{n \ln \ln n}} = \sqrt{2} \]

The LIL says that \( q_n \) should have a \( \sqrt{n \ln \ln n} \) asymptotic dependence on \( n \), but does not specify its \( \alpha \) dependence.

Our sequential testing insights rely on the stronger non-asymptotic LIL of Ch. 12: with probability at least \( 1 - \alpha \), we have \( |S_n| \leq \sqrt{Kn \ln \left( \frac{\ln n}{\alpha} \right)} =: q_n \) simultaneously for all \( n \geq K \ln \left( \frac{\ln n}{\alpha} \right) := n_0 \). This choice of \( q_n \) satisfies (14.1) for \( n_0 \leq n \leq N \), and specifies the sequential test as in Fig. 14.1. (Choosing \( q_n \) this way is unimprovable in all parameters up to absolute constants, by the anti-concentration bound of Chapter 12.)

Type II Error. For practical purposes, \( \sqrt{\ln \ln n} \leq \sqrt{\ln \ln N} \) can be treated as a small constant (even when \( N = 10^{20}, \sqrt{\ln \ln N} < 2 \)). Hence, \( q_N \approx p_N \) (more discussion in the appendices), and the power is:

\[ P_{H_1(\delta)} \left( \exists n \leq N : S_n > q_n \right) \geq P_{H_1(\delta)} \left( S_N > q_N \right) \approx P_{H_1(\delta)} \left( S_N > p_N \right) \]

(14.2)

(14.3)

So the sequential test is essentially as powerful as a batch test with \( N \) samples (and similarly the \( n^{th} \) round of the sequential test is like an \( n \)-sample batch test). The iterated-logarithm correction is effectively in the argument of a Gaussian CDF, so our approximation loses a negligible amount.

Early Stopping. The standard motivation for using sequential tests is that they often require few samples to reject statistically distant alternatives. To investigate this with
our working example, suppose $N$ is large and the coin is actually biased, with a fixed unknown $\delta > 0$. Then, if we somehow had full knowledge of $\delta$ when using the batch test and wanted to ensure a desired type II error $\beta < 1$, we would use just enough samples $n^*_\beta(\delta)$ (written as $n^*$ in context):

$$n^*_\beta(\delta) = \min \{ n : P_{H_1(\delta)}(S_n \leq p_n) \leq \beta \}$$  \hspace{1cm} (14.4)

so that for all $n \geq n^*_\beta(\delta)$, since $p_n = o(n)$,

$$\beta \geq P_{H_1(\delta)}(S_n \leq p_n) = P_{H_1(\delta)}(S_n - n\delta \leq p_n - n\delta)$$

$$\geq P_{H_1(\delta)}(S_n - n\delta \leq -Kn\delta)$$  \hspace{1cm} (14.5)

Examining (14.5), note that $S_n - n\delta$ is a mean-zero random walk. Therefore, standard lower bounds for the binomial tail tell us that $n^*_\beta(\delta) \geq \frac{K\ln(1/\beta)}{\delta^2}$ suffices, and no test can statistically use much less than $n^*_\beta(\delta)$ samples under $H_1(\delta)$ to control type II error at $\beta$.

How many samples does the sequential test use? The quantity of interest is the test’s stopping time $\tau$, which is $< N$ when it rejects $H_0$ and $N$ otherwise. In fact, the expected stopping time is close to $n^*$ under any alternate hypothesis:

**Theorem 40.** For any $\delta$ and any $\beta > 0$, there exist absolute constants $K_1, K_2$ such that

$$\mathbb{E}_{H_1}[\tau] \leq \left(1 + \frac{K_1\beta^{K_2}}{\ln\frac{1}{\beta}}\right) n^*_\beta(\delta)$$

Theorem 40 shows that the sequential test stops roughly as soon as we could hope for, under any alternative $\delta$, despite our ignorance of $\delta$! We will revisit these ideas in our two-sample sequential test of Section 14.3.1.
14.2.2 Discussion

Note the generality of the ideas in this section. Theorem 40 is proved for biased
coin flips, but it uses only basic concentration of measure ideas: upper and lower bounds
on the tails of a statistic that is a cumulative sum incremented each timestep. Many
natural test statistics follow this scheme, particularly those that can be efficiently updated
on the fly. Our main sequential two-sample test in the next section does also.

Theorem 40 is notable for its uniformity over $\delta$ and $\beta$. Note that $q_n$ (and therefore
the sequential test) are independent of both of these – we need only to set a target type I
error bound $\alpha$. Under any alternative $\delta > 0$, the theorem holds for all $\beta$ simultaneously.
As $\beta$ decreases, $n^*_\beta(\delta)$ (written as $n^*$ in context) of course increases, but the leading
multiplicative factor $\left(1 + \frac{K_1 K_2}{\ln \beta}\right)$ decreases. In fact, with an increasingly stringent
$\beta \to 0$, we see that $\frac{\mathbb{E}_{H_1}[\tau]}{n^*} \to 1$; so the sequential test in fact stops closer to $n^*$, and
hence $\tau$ is almost deterministically best possible. Indeed, the proof of Theorem 40 also
shows that $P_{H_1}(\tau \geq n) \leq e^{-Kn\delta^2}$ for $n > n^*$, so the probability of lasting $n$ steps falls
off exponentially in $n$, and is therefore quite sharply concentrated near the optimum
$n^* = n^*_\beta(\delta)$.

We formalize this precise line of reasoning completely non-asymptotically in an
even stronger high-dimensional setting, in the analysis of our main two-sample test in
the next section.

14.3 Two-Sample Mean Testing

In this section, we present our main sequential two-sample test. Assume that
we have samples $X_1, \ldots, X_n, \cdots \sim P$ and $Y_1, \ldots, Y_n, \cdots \sim Q$, with $P, Q$ being unknown
arbitrary continuous distributions on $\mathbb{R}^d$ with means $\mu_1 = \mathbb{E}_{X \sim P}[X], \mu_2 = \mathbb{E}_{Y \sim Q}[Y]$, and
we need to test

\[ H_0 : \mu_1 = \mu_2 \quad \text{vs.} \quad H_1 : \mu_1 \neq \mu_2 \]  

(14.6)

Denote covariances of \( P, Q \) by \( \Sigma_1, \Sigma_2 \) and \( \Sigma := \frac{1}{2}(\Sigma_1 + \Sigma_2) \). Define \( \delta := \mu_1 - \mu_2 \) so that \( \delta = 0 \) under \( H_0 \). Let \( \Phi(\cdot) \) denote the standard Gaussian CDF, and \([\ln \ln(x)]^+ := \ln \ln[\max(x, e^x)]\).

### 14.3.1 A Linear-Time Sequential Test

Our sequential test follows the scheme in Fig. 14.1, so we only need to specify a sequence of rejection thresholds \( q_n \). To do this, we denote

\[
h_i = (X_{2i-1} - Y_{2i-1})^\top (X_{2i} - Y_{2i}).
\]

and define our sequential test statistic as the following *stochastic process* evolving with \( n \):

\[
T_n = \sum_{i=1}^{n} h_i.
\]

Under \( H_0 \), \( \mathbb{E}[h_i] = 0 \), and \( T_n \) is a zero-mean random walk.

**Proposition 23.** \( \mathbb{E}[T_n] = \mathbb{E}[h] = n\|\delta\|^2 \), and

\[
\text{var}(T_n) = n\text{var}(h) = n(4\text{tr}(\Sigma^2) + 4\delta^\top \Sigma \delta) =: nV_0.
\]

We assume – for now – that our data are bounded, i.e.

\[
\|X\|, \|Y\| \leq 1/2,
\]
so that by the Cauchy-Schwarz inequality, w.p. 1,

\[ |T_n - T_{n-1}| = |(X_{2n-1} - Y_{2n-1})^\top (X_{2n} - Y_{2n})| \leq 1 \]

Since \( T_n \) has bounded differences, it exhibits Gaussian-like concentration under the null.

We examine the cumulative variance process of \( T_n \) under \( H_0 \),

\[
\sum_{i=1}^{n} \mathbb{E} \left[ (T_i - T_{i-1})^2 \mid h_{1:(i-1)} \right] = \sum_{i=1}^{n} \text{var}(h_i) = nV_0
\]

Using this, we can control the behavior of \( T_n \) under \( H_0 \) as follows (proof in Sec. D.3).

**Theorem 41.** Take any \( \xi > 0 \). Then with probability \( \geq 1 - \xi \), for all \( n \) simultaneously,

\[
|T_n| < C_0(\xi) + \sqrt{2C_1nV_0[\ln \ln(nV_0) + (nV_0)nV_0 \ln \left( \frac{4}{\xi} \right)}}
\]

where \( C_0(\xi) = 3(e - 2)e^2 + 2 \left( 1 + \sqrt{\frac{1}{3}} \right) \ln \left( \frac{8}{\xi} \right) \), and \( C_1 = 6(e - 2) \).

Unfortunately, we cannot use the uniform Bernstein LIL of Thm. 21 (from Ch. 12) directly to get computable deviation bounds for type I error control, because the covariance matrix \( \Sigma \) is unknown a priori. \( nV_0 = n(4\text{tr}(\Sigma^2) + 4\delta^\top \Sigma \delta) \) must instead be estimated on the fly as part of the sequential test, and its estimate must be concentrated tightly and uniformly over time, so as not to present a statistical bottleneck if the test runs for a long time. We prove such a result, necessary for sequential testing, relating \( nV_0 \) to the empirical variance process \( \hat{\Sigma}_n = \sum_i h_i^2 \).

**Lemma 42.** With probability \( \geq 1 - \xi \), for all \( n \) simultaneously, there is an absolute constant \( C_3 \) such that

\[
nV_0 \leq C_3(\hat{\Sigma}_n + C_0(\xi))
\]

Its proof uses a self-bounding argument and is Sec. D. Now, we can combine
these to prove a novel uniform empirical Bernstein inequality to (practically) establish concentration of $T_n$ under $H_0$.

**Theorem 43** (Uniform Empirical Bernstein Inequality for Random Walks). Take any $\xi > 0$. Then with probability $\geq 1 - \xi$, for all $n$ simultaneously,

$$|T_n| < C_0(\xi) + \sqrt{2\hat{V}_n^* \left( \lceil \ln \ln \right) + \ln \left( \frac{4}{\xi} \right)}$$

where $\hat{V}_n^* := C_3(\hat{V}_n + C_0(\xi))$, $C_0(\xi) = 3(e - 2)e^2 + 2 \left( 1 + \sqrt{\frac{1}{3}} \right) \ln \left( \frac{6}{\xi} \right)$ and $C_3$ is an absolute constant.

Its proof follows from a union bound on the uniform Bernstein LIL (Thm. 21) and Lem. 42. Thm. 43 depends on $\hat{V}_n$, which is easily calculated by the algorithm on the fly in constant time per iteration. Ignoring constants for clarity, Thm. 43 effectively implies that our sequential test from Figure 14.1 controls type I error at $\alpha$ by setting

$$q_n \propto \ln \left( \frac{1}{\alpha} \right) + \sqrt{2\hat{V}_n \ln \left( \frac{\ln \hat{V}_n}{\alpha} \right)} \quad (14.7)$$

Practically, we suggest using the above threshold with a constant of 1.1 to guarantee type-I error approximately $\alpha$ (this is all one often wants anyway, since any particular choice of $\alpha = 0.05$ is anyway arbitrary). This is what we do in our experiments, with excellent success in simulations. For exact or conservative control, consider using a small constant multiple of the above threshold, such as 2.

The above sequential threshold is remarkable, because within the practically useful and simple expression lies a deep mathematical result – the uniform Bernstein LIL effectively involves a union bound for the error probability over an infinite sequence of times. Other naive attempts to union bound the error probabilities for a possibly infinite
sequential testing procedure are too loose and hence too conservative. Furthermore, the classical LIL is known to be asymptotically tight including constants, and our non-asymptotic LIL is also tight up to small constant factors.

This type-I error control with an implicit infinite union bound surprisingly does not lead to a loss in power. Indeed, our statistic possesses essentially the same power as the corresponding linear-time batch two sample test, and also stops early for easy problems. We make this precise in the following two subsections.

14.3.2 A Linear-Time Batch Test

Here we study a simple linear-time batch two-sample mean test, following the template in Fig. 14.1. Consider the linear-time statistic $T_N = \sum_{i=1}^N h_i$, where, as before, $h_i = (x_{2i-1} - y_{2i-1})^\top (x_{2i} - y_{2i})$. Note that the $h_i$s are also i.i.d., and $T_N$ relies on $2N$ data points from each distribution.

Let $V_{N0}, V_{N1}$ be $\text{var}(T_N) = N \text{var}(h)$ under $H_0, H_1$ respectively. Recalling Proposition 23:

\[ V_{N0} := NV_0 := 4N \text{tr}(\Sigma^2), \]
\[ V_{N1} := NV_1 := N(4 \text{tr}(\Sigma^2) + 4\delta^\top \Sigma \delta). \]

Then since $T_N$ is a sum of i.i.d. variables, the central limit theorem (CLT) implies that (where $\overset{d}{\to}$ is convergence in distribution)

\[ \frac{T_N}{\sqrt{V_{N0}}} \overset{d}{\to}_{H_0} \mathcal{N}(0, 1) \] (14.8a)
\[ \frac{T_N - N\|\delta\|^2}{\sqrt{V_{N1}}} \overset{d}{\to}_{H_1} \mathcal{N}(0, 1) \] (14.8b)
Based on this information, our test rejects the null hypothesis whenever

\[ T_N > \sqrt{V_{N_0}} z_\alpha, \quad \text{(14.9)} \]

where \( z_\alpha \) is the \( 1 - \alpha \) quantile of the standard normal distribution. So Eq. (14.8a) ensures that

\[ P_{H_0} \left( \frac{T_N}{\sqrt{V_{N_0}}} > z_\alpha \right) \leq \alpha, \]

giving us type I error control under \( H_0 \).

In practice, we may not know \( V_{N_0} \), so we standardize the statistic using the empirical variance – since we assume \( N \) is large, these scalar variance estimates do not change the effective power analysis. For non-asymptotic type I error control, we can use an empirical Bernstein inequality [MP09, Thm. 11], based on an unbiased estimator of \( V_N \). Specifically, the empirical variance of \( h_i \)'s (\( \hat{V}_N \)) can be used to reject the null whenever

\[ T_N > \sqrt{2\hat{V}_N \ln(2/\alpha)} + \frac{7N \ln(2/\alpha)}{3(N - 1)}. \quad \text{(14.10)} \]

Ignoring constants for clarity, the empirical Bernstein inequality effectively suggests that the batch test from Figure 14.1 will have type I error control of \( \alpha \) on setting threshold

\[ p_N \propto \ln \left( \frac{1}{\alpha} \right) + \sqrt{2\hat{V}_N \ln \left( \frac{1}{\alpha} \right)} \quad \text{(14.11)} \]

For immediate comparison, we copy below the expression for \( q_n \) from Eq. (14.7):

\[ q_n \propto \ln \left( \frac{1}{\alpha} \right) + \sqrt{2\hat{V}_n \left( \ln \frac{\hat{V}_n}{\alpha} \right)}. \]

This similarity explains the optimal power and stopping time properties, detailed in the
next subsection.

One might argue that if $N$ is large, then $\hat{V}_N \approx V_N$, and in this case we can simply derive the (asymptotic) power of the batch test given in Eq.(14.9) as

$$P_{H_1} \left( \frac{T_N}{\sqrt{V_{N0}}} > z_{\alpha} \right)$$

To derive this expression, we start with

$$= P_{H_1} \left( \frac{T_N - N\|\delta\|^2}{\sqrt{V_{N1}}} > z_{\alpha} \sqrt{\frac{V_{N0}}{V_{N1}} - \frac{N\|\delta\|^2}{\sqrt{V_{N1}}}} \right)$$

This can be simplified further to

$$= \Phi \left( \frac{\sqrt{N}\|\delta\|^2}{\sqrt{8\text{tr}(\Sigma^2) + 8\delta^\top\Sigma\delta}} - z_{\alpha} \sqrt{\frac{\text{tr}(\Sigma^2)}{\text{tr}(\Sigma^2) + \delta^\top\Sigma\delta}} \right)$$

Note that the second term is a constant less than $z_{\alpha}$. As a concrete example, when $\Sigma = \sigma^2 I$, and we denote the signal-to-noise ratio as $\Psi := \|\delta\|/\sigma$, then the power of the linear-time batch test is at least $\Phi \left( \frac{\sqrt{Ny^2}}{\sqrt{8d + 8\Psi^2}} - z_{\alpha} \right)$.

### 14.3.3 Power and Stopping Time of Sequential Test

The striking similarity of Eq. (14.11) and Eq. (14.7), mentioned in the previous subsection, is not coincidental. Indeed, both of these arise out of non-asymptotic versions of CLT-like control and LIL-like control, and we know that in the asymptotic regime for Bernoulli coin-flips, CLT thresholds and LIL threshold differ by just $\propto \sqrt{\ln \ln n}$ factors. Hence, it is not surprising to see the empirical Bernstein LIL match empirical Bernstein thresholds up to $\propto \sqrt{\ln \ln \hat{V}_n}$ factors. Since the power of the sequential test is at least the probability of rejection at the very last step, and since $\sqrt{\ln \ln n} < 2$ even for $n = 10^{20}$, the power of the linear-time sequential and batch tests is essentially the same. However, a sequential test that rejects at the last step is of little practical interest, bringing us to the issue of early stopping.

**Early Stopping.** The argument is again identical to that Section 14.2, proving that $\mathbb{E}_{H_1} [\tau]$ is nearly optimal, and arbitrarily close to optimal as $\beta$ tends to zero. Once more
note that the “optimal” above refers to the performance of the oracle linear-time batch algorithm that was informed about the right number of points to subsample and use for the one-time batch test. Formally, let \( n^*_\beta(\delta) \) denote this minimum sample size for the two-sample mean testing batch problem to achieve a power \( \beta \), the * indicating that this is an oracle value, unknown to the user of the batch test. From Eq. (14.12), it is clear that for \( N \geq \frac{8Tr(\Sigma^2) + 8\delta^T \Sigma \delta}{\|\delta\|^4}(z_\beta + z_{\alpha})^2 \), the power becomes at least \( \beta \). In other words,

\[
n^*_\beta(\delta) \leq \frac{Tr(\Sigma^2) + \delta^T \Sigma \delta}{\|\delta\|^4}8(z_\beta + z_{\alpha})^2
\]

(14.13)

**Theorem 44.** Under \( H_1 \), the sequential algorithm of Fig. 14.1 using \( q_n \) from Eq. (14.7) has expected stopping time \( \propto n^*_\beta(\delta) \).

For clarity, we simplify (14.7) and (14.11) by dropping the initial \( \ln \left( \frac{1}{\alpha} \right) \) additive term since it is soon dominated by the second term and does not qualitatively affect the conclusion.

### 14.3.4 Discussion

This section’s arguments have given an illustration of the flexibility and great generality of the ideas we used to test the bias of the coin. In the two-sample setting, we simply design the statistic \( T_N = \sum_{i=1}^{n} h_i \) to be a mean-zero random walk under the null. As in the coin’s case, our non-asymptotic LIL controls type I error optimally, and the remaining arguments are identical because of the common concentration properties of all random walks.

Our test statistic \( T_N \) is chosen with several considerations in mind. First, the batch test is linear-time in the sample complexity, so we are comparing algorithms with the *same computational budget*, on a fair footing. There exist batch tests using U-statistics that have higher power than ours ([RRP+15]) for a given \( N \), but they use
more computational resources \((O(N^2))\) rather than \(O(N)\)).

Also, the batch statistic is a sum of random increments, a common way to write many hypothesis tests, and one that can be computed on the fly in the sequential setting. Note that \(T_N\) is a scalar, so our arguments do not change with \(d\), and we inherit the favorable high-dimensional statistical performance of the statistic; [RRP+15] has more relevant discussion. The statistic also has been shown to have powerful generalizations in the recent statistics literature, which we discuss in the appendices.

Though we assume data scaled to have norm \(\frac{1}{2}\) for convenience, this can be loosened. Any data with bounded norm \(B > \frac{1}{2}\) can be rescaled by a factor \(\frac{1}{B}\) just for the analysis, and then our results can be used. This results in an empirical Bernstein bound like Thm. 43, but of order \(O\left(C_0(\xi) + \sqrt{\hat{V}_n \ln \left(\frac{\ln(\hat{B}\hat{V}_n)}{\xi}\right)}\right)\). The dependence on \(B\) is very weak, and is negligible even when \(B = \text{poly}(d)\).

In fact, we only require control of the higher moments (e.g. by Bernstein conditions, which generalize boundedness and sub-Gaussianity conditions) to prove the non-asymptotic Bernstein LIL in Ch. 12, exactly as is the case for the usual Bernstein concentration inequalities for averages ([BLM13b]). Therefore, our basic arguments hold for unbounded increments \(h_i\) as well. These results apply to general martingales, so our ideas could conceivably be extended to this setting to devise more data-dependent tests, which would be interesting future work.

14.4 Empirical Evaluation

In this section, we evaluate our proposed sequential test on synthetic data, to validate the predictions made by our theory concerning its type I/II errors and the stopping time.

We simulate data from two multivariate Gaussians \((d = 10)\), motivated by our discussion at the end of Section 14.3.2: each Gaussian has covariance matrix \(\Sigma = \sigma^2 I_d\),
one has mean $\mu_1 = \theta^d$ and the other has $\mu_2 = (\delta, 0, 0, \ldots, 0) \in \mathbb{R}^d$ for some $\delta \geq 0$. We keep $\sigma = 1$ here to keep the scale of the data roughly consistent with the biased-coin example, though we find the scaling of the data makes no practical difference, as we discussed.

\subsection{14.4.1 Running the Test and Type I Error}

Like typical hypothesis tests, ours is designed to control type I error. When implementing our algorithmic ideas, it suffices to set $q_n$ as in (14.7), where the only unknown parameters are proportionality constants $C, C_0$: $q_n \propto C_0 + \sqrt{C\hat{V}_n \left( \ln \ln \hat{V}_n \right)}$. The theory suggests that $C, C_0$ are absolute constants, and prescribes upper bounds for them, which can conceivably be loose because of the analytic techniques we have used. On the other hand, in the asymptotic limit the bounds become tight; the empirical $\hat{V}_n$ converges quickly to its mean $V_n$, and we know from second-moment versions of the LIL that $C = \sqrt{2}$ and $C_0 = 0$ are correct. However, as we consider smaller finite times, that bound must be relaxed (at the extremely low $t = 1$ or 2 when flipping a fair coin, for instance).

Nevertheless, we find that in practice, for even moderate sample sizes like the ones we test here, the same reasonable constants suffice in all our experiments: $C = \sqrt{2}$ and $C_0 = \ln(\frac{1}{\alpha})$, with $C_0$ following Thm. 43 and similar fixed-sample Bennett bounds ([BLM13b]; also see the appendices). The situation is exactly analogous to how the Gaussian approximation is valid for even moderate sample sizes in batch testing, making possible a huge variety of common tests that are asymptotically and empirically correct with reasonable constants to boot.

To be more specific, consider the null hypothesis for the example of the coin bias testing given earlier; these fair coin flips are the most anti-concentrated possible bounded steps, and render our empirical Bernstein machinery ineffective, so they make
a good test case. We choose $C$ and $C_0$ as above, and plot the cumulative probability of type I violations $\Pr_{H_0}(\tau \leq n)$ up to time $n$ for different $\alpha$ (where $\tau$ is the stopping time of the test), with the results in Fig. 14.2. To control type I error, the curves need to be asymptotically upper-bounded by the desired $\alpha$ levels (dotted lines). This does not appear true for our recommended settings of $C, C_0$, but the figure still indicates that type I error is controlled even for very high $n$ with our settings. A slight further raise in $C$ beyond $\sqrt{2}$ suffices to guarantee much stronger control.

Fig. 14.2 also seems to contain linear plots, which we cannot fully explain. For more on provable correctness with low $C$, see the appendices.

![Figure 14.2. Pr$_{H_0}(\tau \leq n)$ for different $\alpha$, on biased coin. Dotted lines of corresponding colors are the target levels $\alpha$.](image)

### 14.4.2 Type II Error and Stopping Time

Now we verify the results at the heart of the paper – uniformity over alternatives $\delta$ of the type II error and stopping time properties.

Fig. 14.3 plots the power of the sequential test $P_{H_1(\delta)}(\tau < N)$ against the maximum runtime $N$ using the Gaussian data, at a range of different alternatives $\delta$; the solid and dashed lines represent the power of the batch test (14.11) with $N$ samples, and the sequential test with maximum runtime $N$. As we might expect, the batch test
Figure 14.3. Power vs. ln(N) for different δ, on Gaussians. Dashed lines represent power of batch test with N samples.

has somewhat higher power for a given sample size, but the sequential test consistently performs well compared to it. The role of N here is basically to set a desired tolerance for error; increasing N does not change the intermediate updates of the algorithm, but does increase the power by potentially running the test for longer. So each curve in Fig. 14.3 illustrates the statistical tradeoff inherent in hypothesis testing against a fixed simple alternative, but the great advantage of our sequential test is in achieving all of them simultaneously with the same algorithm.

To highlight this point, we examine the stopping time compared to the batch test for the Gaussian data, in Fig. 14.4. We see that the distributions of ln(τ) are all relatively concentrated around their medians (marked), which fit well to a slope-2 line. This shows the predicted $\frac{1}{\delta^2}$ dependence on δ. Some more experiments are in the appendices.

14.5 Related Work

**Parametric or asymptotic methods.** Our statements about the control of type I/II errors and stopping times are very general, following up on early sequential analysis work. Most sequential tests operate in Wald’s framework, expounded in [Wal45]. In a seminal line of work, Robbins and colleagues delved into sequential hypothesis testing
Figure 14.4. Distribution of $\log_{1.25}(\tau)$ for $\delta \in \{0.5(1.25)^c : c \in \{7, 6, \ldots, 0\}\}$, so that the abscissa values $\{\log_{1.25}(\frac{1}{c})\}$ are a unit length apart. Dashed line has slope 2.

in an asymptotic sense ([Rob85]). Apart from being asymptotic, their tests were most often for simple hypotheses (point nulls and alternatives), were univariate, or parametric (assuming Gaussianity or known density). That said, two of their most relevant papers are [Rob70b] and [DR67b], which discuss statistical methods related to the LIL. They give an asymptotic version of the argument of Section 14.2, using it to design sequential Kolmogorov-Smirnov tests with power one. Other classic works that mention using the LIL for testing various simple or univariate or parametric problems include [DR68a, DR68b, Lai77, Ler86]. These all operate in the asymptotic limit in which the classic LIL can be used to set $q_N$.

For testing a simple null against a simple alternative, the sequential probability ratio test (SPRT) was proved to be optimal by the seminal work of [WW48], but this applies when both the null and alternative have a known parametric form. The same authors also suggested a univariate nonparametric two-sample test in [WW40], but did not combine these two lines of work.

**Bernstein-based methods.** Finite-time uniform LIL-type concentration tools from Ch. 12 are the linchpins of our analysis, and we adapt them in new ways; but novelty in this respect is not our primary focus here, because less recent concentration bounds
can also be used to yield similar results. It is always possible to use a weighted union bound (allocating failure probability $\xi$ over time as $\xi_n \propto \frac{\xi}{n}$) over fixed-$n$ Bernstein bounds, resulting in a deviation bound of $O\left(\sqrt{V_n \ln \frac{n}{\xi}}\right)$. A more advanced “peeling” argument, dividing time $n$ into exponentially growing epochs, improves the bound to $O\left(\sqrt{V_n \ln \ln n \frac{n}{\xi}}\right)$. This suffices in many simple situations, but in general is still arbitrarily inferior to our bound of $O\left(\sqrt{V_n \ln \ln \frac{V_n}{\xi}}\right)$, precisely in the case $V_n \ll n$ in which we expect the second-moment Bernstein bounds to be most useful over Hoeffding bounds. A yet more intricate peeling argument, demarcating the epochs by exponential intervals in $V_n$ rather than $n$, can be used to achieve our iterated-logarithm rate, in conjunction with the well-known second-order uniform martingale bound due to [Fre75]. This serves as a sanity check on the non-asymptotic LIL bounds of [Bal15b], where it is also shown that these bounds have the best possible dependence on all parameters. However, it can be verified that even a suboptimal uniform concentration rate like $O\left(\sqrt{V_n \ln \frac{V_n}{\xi}}\right)$ would suffice for the optimal stopping time properties of the sequential test to hold, with only a slight weakening of the power.

Bernstein inequalities that only depend on empirical variance have been used for stopping algorithms in Hoeffding races ([LN13]) and other even more general contexts ([MSA08]). This line of work uses the empirical bounds very similarly to us, albeit in the nominally different context of direct estimation of a mean. As such, they too require uniform concentration over time, but achieve it with a crude union bound (failure probability $\xi_n \propto \frac{\xi}{n^2}$), resulting in a deviation bound of $O\left(\sqrt{V_n \ln \frac{n}{\xi}}\right)$. Applying the more advanced techniques above, it may be possible to get our optimal concentration rate, but to our knowledge ours is the first work to derive and use uniform LIL-type empirical Bernstein bounds.

**Practical Usage.** To our knowledge, implementing sequential testing in practice has
previously invariably relied upon CLT-type results patched together with heuristic adjustments of the CLT threshold (e.g. the widely-used scheme for clinical trials of [PPA+77] has an arbitrary conservative choice of $q_n = 0.001$ through the sequential process and $q_N = 0.05 = \alpha$ at the last datapoint). These perform as loose functional versions of our uniform finite-sample LIL upper bound, though without theoretical guarantees. In general, it is unsound to use an asymptotically normal distribution under the null at stopping time $\tau$ – the central limit theorem (CLT) applies to any fixed time $t$, but it may not apply to a random stopping time $\tau$ (see the random-sum CLT of [Ans52], and [Gut12] and references). This has caused myriad practical complications in implementing such tests (see [LS+08], Section 4). One of our contributions is to rigorously derive a directly usable finite-sample sequential test, in a way we believe can be extended to a large variety of testing problems.

We emphasize that there are several advantages to our proposed framework and analysis which, taken together, are unique in the literature. We tackle the multivariate nonparametric (possibly even high-dimensional) setting, with composite hypotheses. Moreover, we not only prove that the power is asymptotically one, but also derive finite-sample rates that illuminate dependence of other parameters on $\beta$, by considering non-asymptotic uniform concentration over finite times. The fact that it is not provable via purely asymptotic arguments is why our optimal stopping property has gone unobserved for a wide range of tests, even as basic as the biased coin. In our more refined analysis, it can be verified (Thm. 40) that the stopping time diverges to $\infty$ when the required type II error $\rightarrow 0$, i.e. power $\rightarrow 1$.

We have presented a sequential scheme for multivariate nonparametric hypothesis testing against composite alternatives, which comes with a full finite-sample analysis in terms of on-the-fly estimable quantities. Its desirable properties include type I error control by considering finite-time LIL concentration; near-optimal type II error compared
to linear-time batch tests, due to the iterated-logarithm term in the LIL; and most importantly, essentially optimal early stopping, uniformly over a large class of alternatives. We presented some simple applications in learning and statistics, but our design and analysis techniques are general, and their extensions to other settings are of continuing future interest.

14.6 Further Extensions to Other Tests

14.6.1 A General Two-Sample Test

Given two independent multivariate streams of i.i.d. data, instead of testing for differences in mean, we could also test for differences in any moment, i.e. differences in distribution, a subtler problem which may require much more data to ascertain differences in higher moments. In other words, we would be testing

\[ H_0 : P = Q \text{ versus } H_1 : P \neq Q \]

One simple way to do this is by using a kernel two-sample test, like the Maximum Mean Discrepancy (MMD) test proposed by [GBR+12]. The population MMD is defined as

\[ \text{MMD}(P, Q) = \sup_{f \in H_k} (\mathbb{E}_{X \sim P} f(X) - \mathbb{E}_{Y \sim Q} f(Y)) \]

where \( H_k \) is the unit ball of functions in the Reproducing Kernel Hilbert Space corresponding to some positive semidefinite Mercer kernel \( k \). One common choice is the Gaussian kernel \( k(a, b) = \exp(-\|a - b\|^2 / \gamma^2) \). With this choice, the population MMD has an interesting interpretation, given by Bochner’s theorem [Rud87] as

\[ \text{MMD} = \int_{\mathbb{R}^d} |\phi_X(t) - \phi_Y(t)|^2 e^{-\gamma^2 \|t\|^2} \, dt \]
where $\varphi_X(t), \varphi_Y(t)$ are the characteristic functions of $P, Q$. This means that the population MMD is nonzero iff the distributions differ (i.e. the alternative holds).

The authors of [GBR+12] propose the following (linear-time) batch test statistic after seeing $2N$ samples: $\text{MMD}_N = \frac{1}{N} \sum_{i=1}^{N} h_i$, where

$$h_i = k(x_{2i}, x_{2i+1}) + k(y_{2i}, y_{2i+1}) - k(x_{2i}, y_{2i+1}) - k(x_{2i+1}, y_{2i})$$

The associated test is consistent against all fixed (and some local) alternatives where $P \neq Q$; see [GBR+12] for a proof, and [RRP+15] for a high-dimensional analysis of this test (in the limited setting of mean-testing that we consider earlier in this paper). Both properties are inherited by the following sequential test.

The sequential statistic we construct after seeing $n$ batches ($2n$ samples) is the random walk $T_n = \sum_{i=1}^{n} h_i$, which has mean zero under the null because $\mathbb{E}[\text{MMD}_N] = \mathbb{E}[h_i] = 0$. The similarity with our mean-testing statistic is not coincidental; when $k(a, b) = a^\top b$, they coincide, further motivating our choice of test statistic $U_n$ earlier in the paper. As before, we use the non-asymptotic LIL to get type I error control, nearly the same power as the linear-time batch test, and also early stopping – much before seeing $N$ points – if the problem at hand is easy.

### 14.6.2 A General Independence Test

Given a single multivariate stream of i.i.d data, where each datapoint is a pair $(X_i, Y_i) \in \mathbb{R}^{p+q}$, the independence testing problem involves testing whether $X$ is independent of $Y$ or not. More formally, we want to test

$$H_0 : X \perp Y \text{ versus } H_1 : X \not\perp Y .$$  \hspace{1cm} (14.14)
A test of linear correlation/covariance only detects linear dependence. As an alternative to this, [SRB07] proposed a population quantity called distance covariance, given by

\[
dCov(X, Y) = \mathbb{E}\|X - X'\| \|Y - Y'\| + \mathbb{E}\|X - X'\| \mathbb{E}\|Y - Y'\| - 2\mathbb{E}\|X - X'\| \|Y - Y''\|
\]

where \((X, Y), (X', Y'), \ldots\) are i.i.d. pairs from the joint distribution on \((X, Y)\).

Remarkably, an alternative representation is

\[
dCov(X, Y) = \int_{\mathbb{R}^{p+q}} |\phi_{X,Y}(t, s) - \phi_X(t)\phi_Y(s)|^2 w(t, s) \, dt \, ds
\]

where \(\phi_X, \phi_Y, \phi_{X,Y}\) are the characteristic functions of the marginals and joint distribution of \(X, Y\) and \(w(t, s) \propto \|t\|^{1+p} \|s\|^{1+q}\). Using this, the paper [SRB07] concludes that \(dCov(X, Y) = 0\) if and only if \(X \perp Y\).

One way to form a linear-time statistic to estimate \(dCov\) is to process the data in batches of size four, i.e. \(B_i = \bigcup_{j=0}^{3} (X_{4i+j}, Y_{4i+j})\), and calculate the scalar

\[
h_i = \frac{1}{6} \sum_{(2)} \|X - X'\| \|Y'' - Y'''\| + \frac{1}{6} \sum_{(2)} \|X - X'\| \|Y - Y'\| - \frac{1}{24} \sum_{4 \times 3} \|X - X'\| \|Y - Y''\|
\]

where the summations are over all possible ways of assigning \((X, Y) \neq (X', Y') \neq (X'', Y'') \neq (X''', Y''')\), each pair being one from \(B_i\). The expectation of this quantity is exactly \(dCov\), and the batch test statistic, given \(2N\) datapoints, is simply \(dCov_N = \frac{1}{N} \sum_{i=1}^{N} h_i\).

As before, the associated test is consistent for any fixed alternatives where \(X \not\perp Y\). Noting that \(\mathbb{E}[dCov_N] = \mathbb{E}[h_i] = 0\) under the null, our random walk after seeing \(n\) batches (i.e. \(4n\) points) will just be \(T_n = \sum_{i=1}^{n} h_i\). As in previous sections, our finite LIL results can be used to get type I error control, and early stopping much before seeing \(N\) points, if the problem at hand is statistically “easy.”
Acknowledgements

This chapter contains material from the Conference on Uncertainty in Artificial Intelligence, 2016 ("Sequential Nonparametric Testing with the Law of the Iterated Logarithm," Balsubramani and Ramdas). The dissertation author was a primary investigator and author of this paper.
Chapter 15

Preventing $P$-Value Manipulation by Stopping

Imagine a statistician trying to collect random samples $X_1, \ldots, X_n$ for a hypothesis test, and willing to collect at most some large number $n$. The test uses some statistic of the data $f(X_1, \ldots, X_n)$ (e.g. the sample mean) to calculate a $p$-value. Under the null hypothesis, $f$ has expected value $\mathbb{E}[f(X_{1:n})]$, which we consider 0 w.l.o.g. To perform the test after having collected some number of samples $t \leq n$, the statistician uses the data so far ($X_1^t$) and approximates $f$ by its conditional expectation $\mathbb{E}[f(X_{1:n}) | X_{1:t}]$, testing whether this has zero mean. For instance, they could be testing whether Gaussian unit-variance data $X_1, \ldots, X_n$ has zero mean, using the sample mean $f$.

An unscrupulous statistician may have goals at odds with the assumptions of standard fixed-sample statistics. For instance, they may be incentivized to report a statistically significant result, which could cause them to choose to stop the test early with $\leq n$ samples if the result appears significant. A natural such approach is to repeatedly gather more samples and keep performing the test on all data gathered so far, “peeking” at the reported $p$-values until one test is low enough to be significant, and reporting that $p$-value.

This form of $p$-value hacking by data peeking is common even among well-
intentioned statisticians and A/B-testers, because collecting more data after a test result that appears significant can be costly, and of seemingly questionable value. However, peeking can severely bias the reported \( p \)-value if the test is repeated sufficiently, since the statistician will only report a low enough \( p \)-value – they are “sampling to reach a foregone conclusion” ([Ans54]). This procedure biases the reported \( p \)-value downward, making it much more likely to report a significant result under the null hypothesis.

However, peeking is widespread for very intuitive reasons in empirical science. For instance, it has long been argued that the statistician’s opinion about the conclusion, as reflected by their sampling rule, should not influence the evidence the data provides against the null hypothesis – “the rules governing when data collection stops are irrelevant to data interpretation” ([ELS63]). Collecting more data (and therefore more evidence) should always help, not invalidate previous results. However, conventional \( p \)-value analyses “depend on the intentions of the investigator” ([Nic00]).

We present a solution to this problem – a way to correct the reported \( p \)-values when peeking is performed in the above scenario, along with many others. Our correction method typically requires no additional computational overhead, and we prove that the corrected \( p \)-value is basically the lowest possible, making our method optimal in a strong sense.

Our approach is to report a strengthened form of the \( p \)-value that is valid for all times, not just the time it is computed – seeing at any point a \( p \)-value of \( \delta \) allows us to reject the null at \( \delta \). This holds regardless of the sampling procedure, and regardless of the outcomes of other tests seen while peeking.

In Section 15.1, we formalize the problem and lay out the argument underlying our claims. Section 15.2 details our correction methods in several successively more general settings, notably a method in Section 15.2.2 that solves our introductory problem, of the unscrupulous statistician stopping a successively conditioned expectation of the
test statistic \( f(X_1, \ldots, X_n) \). The reported robust \( p \)-value is provably close to the lowest possible. This is also extended to a more general scenario in which the statistics observed constitute a nearly arbitrary stochastic process, in Section 15.2.3. Section 15.3 follows with an extended discussion of our results in the context of previous work. The main theoretical contribution of this chapter, a tight improvement of the loose concentration bounds of Ch. 12, is proved in Sec. 15.4.

15.1 Approach: Robustness Against Any Sampling Plan

We are guided by the conventional statistical testing setting, where standard tests are statistically valid without peeking. Here, some sample size \( N \) is prescribed in advance, so that the reported \( p \)-value \( P_N \) is a random variable with

\[
\Pr (P_N \leq \delta) \leq \delta
\]  

This is the defining characteristic of the \( p \)-value that is used in interpreting the results of statistical tests ([LR06b]).

In principle (as when peeking), such a \( p \)-value can be computed after any number of samples \( t \leq N \); write it as \( P_t \). We have seen that for any fixed time \( t \), \( P_t \) satisfies (15.1), i.e.

\[
\forall t : \Pr (P_t \leq \delta) \leq \delta
\]  

Instead of reporting these \( P_t \) values as a conventional statistical test would, we alter them to a stochastic process \( \{A_t\}_{t \geq 1} \) that is robust against all forms of peeking. The statistician gathering the data should be able to decide when to report the result, in any
manner desired based on the past, without invalidating the reported robust \( p \)-value.

We can picture the situation during a given experiment – for instance, a \( z \)-test statistic that is a zero-mean Gaussian under \( H_0 \) – with the reported \( P_t \) values oscillating stochastically with time. Plotting many Monte Carlo trials of such experiments under the null, we get the left side of Fig. 15.1. There are many seemingly significant trials reported by peeking (blue) compared to the type I error violators at \( t = N \) accounted for by \( P_N \) (red), even though many of the blue trials appear significant (low) only very early in the process and not at \( t \approx N \). This shows the magnitude of the problem, even when sample sizes are relatively small, and the situation clearly worsens with a longer horizon as there are more chances for random walks to dip below \( \alpha \).

We aim to inflate these reported \( p \)-values \( P_t \) to the robust process \( A_t \), which can be thought of as collecting the data’s evidence against the null. It starts at the non-significant \( A_0 = 1 \), evolving while peeking occurs at some subset of times, until some time at which the peeking is stopped. As shown on the right side of Fig. 15.1, it is inflated such that the fraction of trials which ever dip below \( \alpha = 0.05 \) is \( \leq \alpha \), down from the large fraction of blue trials shown on the left using \( P \) instead of \( A \).

This sequential view naturally suggests that the best strategy for the unscrupulous statistician, having peeked at a set of \( p \)-values so far, is to report the minimum one among them – the ostensibly most “significant” result seen. So intuitively, our task is to ensure that this minimum reported value is significant.

To reason about this intuition, we use the idea of a stopping time – a random time \( \tau \) which must only depend on the past but can do so in any way, (formally, \{ \tau \leq t \} is measurable w.r.t. the \( \sigma \)-algebra generated by \( X_1,t \) for all times \( t \)). We would like the robust \( p \)-values \( A_t \) to be such that that for any \( \delta \), and any stopping time \( \tau \),

\[
\Pr (A_\tau \leq \delta) \leq \delta
\]
Figure 15.1. Plot of 1,000 trials of the reported values evolving over time up to $N = 100$, when running a $z$-test on the mean of Gaussian data from $\mathcal{N}(\mu, 1)$, under $H_0 : \mu = 0$. The green line is $\alpha = 0.05$. **Left**: The uncorrected $p$-values $P_t$. In red are the $\alpha$ fraction of significant trials with $P_N \leq \alpha$. In blue are the other trials reported as significant by peeking (where $\exists t < N : P_t \leq \alpha$), and the remaining trials are in yellow. **Right**: The robust $p$-values $A_t$, using the method in this paper. In red are the $\alpha$ fraction of trials reported significant by peeking. Note that $A_t \geq P_t$, so the vertical scale of the graphs is different.

In particular, our statistician engaging in data peeking chicanery might set the stopping time to $\tau_p = \min \{t : A_t \leq \delta\}$, where we consider the minimum of an empty set of times to be $\infty$. This means that $\tau_p < \infty \iff A_{\tau_p} \leq \delta$. Consequently, $\{\tau_p < \infty\} = \{\exists t : A_t \leq \delta\}$, and

$$\Pr(\exists t : A_t \leq \delta) \leq \delta \quad (15.3)$$

So we have argued that the desired robust $p$-value process $A_t$ must at least satisfy (15.3), which matches our earlier intuition: $A_t$ must rarely *ever* dip below any $\delta$ under $H_0$, because the peeking statistician is sure to notice whenever it does and play $\tau_p$.

Observe that (15.3) represents a much stronger kind of concentration bound than (15.2). The stochastic process $P_t$ itself is not suitable to use as $A_t$, as it may not satisfy (15.3); though the chance of rejecting the null is low at each time, the chance of ever
rejecting it over time may be quite high, as shown in Fig. 15.1.

To summarize, we find \( A_t \) satisfying Eq. (15.3), which we call a “valid” robust \( p \)-value.

## 15.2 Algorithms

These ideas are formalized for the (very general) scenario with the unscrupulous statistician in the introduction. We start by expressing the uncorrected \( p \)-values \( P_t \) explicitly. These depend on the test statistic \( M_t := \mathbb{E}[f(X_{1:n}) \mid X_{1:t}] \). The increments of \( M_t \) should be conditionally controllable, for which we introduce the following standard definition.

**Definition 24** ([BLM13a]). A zero-mean random variable \( X \) is \((\nu, b)\)-sub-exponential if there exist \( \nu, b \geq 0 \) such that

\[
\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\nu^2 \lambda^2 / 2\right) \quad \forall |\lambda| < \frac{1}{b}
\]

In this chapter, we take \( M_t \) to have conditionally sub-exponential increments (this occurs for example if \( f \) is bounded).

Our analysis relies on the fact that \( M_t \) is a martingale w.r.t. the data \( X_1, \ldots, X_n \) – for all \( t \),

\[
\mathbb{E}[M_t \mid X_{1:(t-1)}] = \mathbb{E}\left[\mathbb{E}[f(X_{1:n}) \mid X_{1:t}] \mid X_{1:(t-1)}\right] = \mathbb{E}\left[f(X_{1:n}) \mid X_{1:(t-1)}\right] = M_{t-1}
\]

This is a well-known construction, called the Doob (or Lévy) martingale for estimating \( f \). In many common statistical tests like the \( z \)-test of Fig. 15.1 with standard normal increments (“the standard \( z \)-test”), the statistic is simply the sum of zero-mean i.i.d. random variables with finite higher moments.
A final note: throughout the rest of this chapter, we write natural logarithms $\ln(x)$ to mean $\ln(\max(e,x))$, so that $\ln\ln x = \ln(\ln(\max(e^e,x)))$ and so on (i.e., there are no issues with the domain of the natural log that necessitate using complex numbers).

15.2.1 Standard $z$-Test

We use the standard $z$-test as a first concrete example, where $M_t$ is the test statistic (i.e. where $Z_i$ are i.i.d. standard normal variables, $M_t = \sum_{i=1}^t Z_i \sim \mathcal{N}(0,t)$).

The event $P_t \leq \delta$ referred to in (15.2) is when the statistic $M_t$ is far from its conditional expectation of zero under $H_0$. This happens with low probability under the null, $\leq \delta$, due to concentration of measure.

As a large-deviation bound, this can be expressed as a Hoeffding concentration inequality describing non-asymptotic CLT-type behavior ([BLM13a]), involving an absolute positive constant $C$:

$$\Pr\left(|M_t| \geq \sqrt{2t \ln \frac{1}{\delta} + C \ln \frac{1}{\delta}}\right) \leq \delta$$  \hspace{1cm} (15.4)

When $V_t$ is sufficiently large compared to $\ln \frac{1}{\delta}$, the term with $C$ becomes negligible. The inequality then becomes equivalent to

$$\forall t : \Pr\left(\exp\left(-\frac{M_t}{2t}\right) \leq \delta\right) \leq \delta$$

Comparing terms with (15.2), we see that we can think of $P_t = \exp\left(-\frac{M_t}{2t}\right)$.

Our correction then simply multiplies $P_t$ by $\ln(t)$, as shown in Algorithm 6.

This corresponds to the relative “correction” in the rates between the Bernstein
Algorithm 6. Robust $p$-Values for a Standard $z$-Test

Set $M_0 := 0$

for $t = 1, 2, \ldots$ do

Observe $X_t$

Compute $M_t := \sum_{i=1}^t X_i$.

Compute $A$-value

$$A_t := \frac{\ln(t)}{\exp\left(\frac{M_t^2}{2t}\right)}$$

If peeking, report any $A_t$ value seen so far (i.e., the minimum one).

end for

bound and the new finite LIL, as viewed from the perspective of $\delta$. A correction of $O(t)$ instead of our $\ln(t)$ would correspond to taking a union bound over the peeking events that happen every timestep (known as the Bonferroni correction, in the parlance of multiple hypothesis testing). Our correction also increases with $t$ but is far milder, taking advantage of the powerful dependences between peeked $p$-values on any given run to transform the concentration of Eq. (15.2) to that of Eq. (15.3).

Though the correction is relatively small, it is powerful enough to be robust to any sampling plan. This addresses the concern about peeking schemes “sampling to a foregone conclusion” – the $A_t$ value at any point can be interpreted as a conventional $p$-value, regardless of what happened in the past or will in the future.

This can make an important difference practically, as the peeking bias on the reported $p$-value can be extremely bothersome after even a moderate number of tests. Figure 15.2 uses Monte Carlo simulations to demonstrate the effect of peeking under the null when running a $z$-test, a common example in the literature.

The actual type I error is clearly significantly above the uncorrected reported value for all $\delta$, so the distribution of the reported values $P_t$ under $H_0$ is nowhere close to the ideal uniform distribution of the $p$-value under the null.
15.2.2 The Introductory Example: Successive Conditional Estimation

We can extend these ideas to the general example of the introduction, in which the test statistic $M_t := \mathbb{E}[f(X_{1:n}) \mid X_{1:t}]$. As a large-deviation bound, this can be expressed in terms of the martingale variance\(^1\)

$$V_t = \sum_{i=1}^{t} \mathbb{E}[ (M_i - M_{i-1})^2 \mid X_{1:(i-1)} ]$$

in a Bernstein concentration inequality that holds at any time $t$, describing non-asymptotic CLT-type behavior ([BLM13a]) just as for (15.4), but with $2(e - 2)V_t$ in place of $t$. Therefore, we see that we can think of $P_t = \exp \left( -\frac{M_t}{4(e - 2)V_t} \right)$.

Our correction then multiplies $P_t$ by $\ln(2(e - 2)V_t)$, shown in Algorithm 7.

---

\(^1\) Just as in Ch. 14, it is typically possible to bound pathwise concentration of an empirically calculated variance process $\hat{V}_t$ to this process $V_t$, and use that if necessary.
Algorithm 7. Robust $p$-Values from Conditionally Estimating a Test Statistic

Set $M_0 := \mathbb{E}[f(X_{1:n})]
\quad$ for $t = 1, 2, \ldots$ do
  Observe $X_t$
  Compute $M_t := \mathbb{E}[f(X_{1:n}) \mid X_{1:t}]$ and $V_t := \sum_{i=1}^{t} \mathbb{E}[(M_i - M_{i-1})^2 \mid X_{1:(i-1)}].$
  Compute A-value
  \[ A_t := \frac{\ln (2(e-2)V_t)}{\exp \left( \frac{M_t^2}{4(e-2)V_t} \right)} \]
  If peeking, report any $A_t$ value seen so far (i.e., the minimum one).
end for

15.2.3 A Robust $p$-Value from General Test Statistics

Suppose the unscrupulous statistician is not restricted to compute nested conditional expectations of a fixed function $f$, but rather is allowed to compute arbitrary\textsuperscript{2} test statistics $T_1, T_2, \ldots, T_n$, and we are able to evaluate their conditional mean and variance given the past under the null distribution. We describe how to extend our approach to be robust to peeking in these extremely general conditions. This is simply a matter of subtracting off conditional bias, and using our earlier method for the residual martingale variation.

We define some standard notation used to discuss the discrete-time real-valued stochastic processes we now deal with. Such a process $X_t$ is defined in a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$. Define the difference sequence $\xi_t = X_t - X_{t-1}$ for all $t$ (with $\xi_t$ being $\mathcal{F}_t$-measurable), and the cumulative conditional variance process $V_t = \sum_{i=1}^{t} \mathbb{E} [\xi_i^2 \mid \mathcal{F}_{i-1}]$. The process is a martingale (resp. supermartingale, submartingale) has $\mathbb{E} [\xi_t \mid \mathcal{F}_{t-1}] = 0$ (resp. $\leq 0$, $\geq 0$) for all $t$.

We use the Doob decomposition of the stochastic process $T$, where $T_0 := 0$ for

\textsuperscript{2}Though we do require uniform integrability $\mathbb{E} [||T_t||] < \infty$, as is standard in martingale theory.
convenience. This decomposition is the observation that $T$ can be written as

$$T_t = D_t + M_t$$

where the process $D$ (the compensator process, or “drift”) and the Doob martingale $M$ (“diffusion”) are defined as

$$D_t := \sum_{i=1}^{t} (\mathbb{E}[T_i | \mathcal{F}_{i-1}] - T_{i-1}) \quad , \quad M_t := \sum_{i=1}^{t} (T_i - \mathbb{E}[T_i | \mathcal{F}_{i-1}])$$

We assume only that the differences $T_i - \mathbb{E}[T_i | \mathcal{F}_{i-1}]$ are bounded, or have $\mathcal{F}_{i-1}$-measurable conditions bounding their higher moments (e.g. they are conditionally sub-exponential $\forall t$). Nothing more is assumed about the distribution of any $T_i$.

The process $D$ is predictable ($D_t$ is $\mathcal{F}_{t-1}$-measurable, so $\mathbb{E}[D_t | \mathcal{F}_{t-1}] = D_t$), and can be computed from the past. The process $M$ is a martingale accounting for all the unpredictable variation in $t$, so $M_t = T_t - D_t$ will be concentrated around zero under the null, and the reported $A_t$ value is calculated as before. The resulting algorithm is in Alg. 8.

**Algorithm 8. Robust $p$-Values from General Test Statistics**

Initialize $T_0 = 0, D_0 = 0, V_0 = 0$

for $t = 1, 2, \ldots$ do

- Compute $\mathbb{E}[T_t | \mathcal{F}_{t-1}]$ and $\mathbb{E}[T_t^2 | \mathcal{F}_{t-1}]$, and thereby
  $$D_t = D_{t-1} + (\mathbb{E}[T_t | \mathcal{F}_{t-1}] - T_{t-1})$$
  $$V_t = V_{t-1} + \mathbb{E} \left[ (T_t - \mathbb{E}[T_t | \mathcal{F}_{t-1}])^2 | \mathcal{F}_{t-1} \right]$$

- Observe $T_t$

- Compute $M_t := T_t - D_t$ and $A_t := \frac{\ln (2(e - 2)V_t)}{\exp \left( \frac{M_t^2}{4(e - 2)V_t} \right)}$

  If peeking, report any $A_t$ value seen so far (i.e., the minimum one).

end for
Note that this algorithm handles as a special case all martingale test statistics, for which $D_t = 0$. An empirical evaluation of this method on nontrivial test statistic martingales is out of the scope of this thesis, but we believe it is of significant interest as future work.

15.2.4 Validity and Optimality

Validity

Our method’s validity – that type I error is as advertised – follows from the non-asymptotic LIL discussed in previous chapters. We prove in the appendices a much tighter, cleaner version of the “concentration” part (“upper half”) of this result:

**Theorem 45.** Choose any integer $v \geq 1$. Suppose $M_t$ is a martingale whose differences $M_t - M_{t-1}$ are $(v, \exp(e^2))$-sub-exponential given $\mathcal{F}_{t-1}$. Then w.p. $\geq 1 - \delta$, for all $t \geq 16 \exp(e^2) \ln \frac{1}{\delta}$, $|M_t| \leq \frac{U_t}{2\exp(e^2)}$ and

$$|M_t| \leq \sqrt{2U_t \left( \ln \frac{1}{\delta} + \ln \ln \frac{1}{\delta} + \ln \ln (U_t) + 4 \ln \ln (U_t) \right)} \quad (15.5)$$

The proof of the result is a refinement of the one in Chapter 12, in order to derive a correction tight enough to be practically relevant. The rate is extremely tight; as $t \to \infty$, the $\ln \ln \cdot$ term dominates, and it has a leading constant of $\sqrt{2}$ for any $v \geq 2$, unimprovable by the asymptotic LIL. Section 15.4 has more details.

Many of the terms are negligible at high times and fixed $\delta$; the $\ln \ln \cdot$ term is dwarfed by $\ln \frac{1}{\delta}$, and also $4 \ln \ln (U_t) \leq 8$ for $U_t < 10^{700}$, so this triply iterated logarithm term can be taken as a constant (negligible) without numerical issues.

All this suggests a uniform upper bound for all sufficiently high $t$ simultaneously
of:

$$|M_t| \leq \sqrt{2U_t \left( \ln \frac{1}{\delta} + \ln \ln (U_t) \right)} \iff \frac{\ln (U_t)}{\exp \left( \frac{M_t^2}{2U_t} \right)} \geq \delta$$

Comparing this result to (15.3), we get our form for $A_t$, which we use throughout this chapter.

$$A_t := \frac{\ln (U_t)}{\exp \left( \frac{M_t^2}{2U_t} \right)}$$

Strictly speaking, $A_t$ is only asymptotically valid. Our recommendation of it for practical use is analogous to using the CLT for fixed-time $P_t$ in practice, instead of a large-deviation Hoeffding or Bernstein bound.

Putting together our discussions and intuition so far, we also observe that our method is valid (in the sense that the reported value has CDF below the uniform distribution, as in Fig. 15.2) even if our setting is relaxed in some ways. Halting the test early, due to unwillingness to collect more than some number $N$ of total samples, can only increase the corrected $p$-value of $\min_{t \leq N} A_t$. This corresponds to collecting evidence against the null, and we will be even more conservative in rejecting it, although Fig. 15.2 shows that this effect diminishes drastically with increasing $N$. Peeking less (such as only every few samples) also makes the concentration of Eq. (15.3) overly conservative, so our method is still valid.

Such concerns may be allayed by working directly from Lemma 45 without dropping terms, making $A_t$ the solution to:

$$A_t = \frac{\ln (U_t) (\ln \ln (U_t))^4}{\exp \left( \frac{M_t^2}{2U_t} \right)}$$

This is a valid choice of $A_t$, satisfying (15.3), and still quite tight (relatively close to our recommended $A_t$, and can be tightened further by taking more of the multiply iterated logarithm terms in Theorem 48).
We use large deviation bounds to get the $P_t = \exp\left(-\frac{M_t}{2V_t}\right)$ in our analysis, and our experiments as well. This is itself more conservative than the common method of using the exact Gaussian CDF, so it is closer to being valid under peeking, yet Fig. 15.2 shows it is still clearly statistically unsafe. This justifies its use as a baseline instead of the Gaussian CDF.

**Optimality**

Increasing $A_t$ sufficiently will always trivially lead to concentration like Eq. (15.3), so there is a potential issue that our correction method is too conservative. But for the statistics we address, no correction method can be much better, because the concentration bound of Eq. (45) enjoys a nearly matching anti-concentration bound. This was proved in Theorem 19, which we recapitulate below for ease of reference.

**Theorem 46.** There are absolute constants $C_1, C_2$ such that the following is true. Fix a finite time $T > C_2 \ln\left(\frac{2}{\delta}\right)$, and fix any $\delta \in \left[\frac{4}{\ln((T-1)/3)}, \frac{1}{C_1}\right]$. Then with probability at most $1 - \delta$, for all $t \in \left[C_2 \ln\left(\frac{2}{\delta}\right), T\right]$ simultaneously, $|M_t| \leq \frac{2}{3e^2}t$ and

$$|M_t| \leq \sqrt{\frac{2}{3}t \left(\ln \ln \left(\frac{2}{3}t\right) + \ln \left(\frac{1}{C_1 \delta}\right)\right)}$$

We believe our method is provably optimal within an absolute constant on the value it reports, i.e. $\Pr(\exists t : A_t \leq \delta) \geq C\delta$ for some absolute constant $C$. There are possible indications of this in Fig. 15.2. But Theorem 46 results in a much lower $A_t$ than recommended by the upper bound, and closing this gap remains an open problem.
15.3 Discussion

15.3.1 Examples of Martingale Test Statistics

Many of the most common types of tests use martingale statistics, and can therefore be viewed as using martingale concentration bounds for a fixed time, specialized to particular situations. As mentioned earlier, such constructions can roughly be thought of as describing statistics which are asymptotically Gaussian under $H_0$, or have otherwise exponential tails (e.g. $\chi^2$ distributions), because of the ubiquitous use of the martingale CLT in deriving such statistics’ null distributions.

For example, here are a few overlapping classes of such statistics:

- **Doob Martingale Tests**: When successive statistics are nested conditional expectations of a random variable $f$, the situation is like in the introductory example; the statistics form a Doob martingale. Any appropriately rescaled sum of i.i.d. random variables is also a Doob martingale.

- **Asymptotically Normal Test Statistics**: This is a massive class of statistics which prototypically includes $z$-statistics, which we consider in the running example used in the figures of previous sections. It also includes a variety of others like $t$- and other “studentized” tests ([dlPLS08]). Maximum-likelihood estimators are often asymptotically normal as a result of similar martingale constructions (e.g. [LR06b], Ch. 12).

- **Other Asymptotic Test Statistic Distributions**: Many other test statistic distributions, such as the $\chi^2$ arising in goodness-of-fit tests, can be derived as combinations of (asymptotically) normally-distributed variables. Such variables are typically martingales following a CLT, so our methods can be used to derive their statistics. Our techniques can also be used on non-Gaussian variables directly: the broad
class of sub-exponential distributions includes Gaussians and any other distribution with exponential tails, and their defining condition specifies a family of exponential supermartingales that we use in our proofs ([BLM13a]).

- **Likelihood-Ratio/Score/Wald Tests and Bayes Factors**: The likelihood ratio \( L_n = \prod_{i=1}^{n} \frac{f(X_i)}{g(X_i)} \) between probability densities \( f \) and \( g \) on i.i.d. data \( X_1, \ldots, X_n \) is a deeply-studied martingale under the null that the data are generated from \( g \). This has motivated very general extensions of the classic Neyman-Pearson theory of testing \( L_n \), to scenarios in which \( n \) is variable ([WW48, HH80]), including tests against composite alternatives ([LR06b]); we discuss this further in Sec. 15.3.2. Related tests often used in Bayesian parameter estimation, the score and Wald tests, are also asymptotically normal and “studentized” respectively, and are classically analyzed with martingales as well using second-order Taylor expansions of the log likelihood. Finally, integrating over a hyperparameter prior to form a Bayes factor is a related method that has been noted to quite naturally yield useful martingales robust to optional stopping ([Vov93, KR95, SSVV11]).

- **U-Statistics and Scan Statistics**: U-statistics are underpinned by a deeply explored martingale structure (see e.g. [KB13]), leading to many more useful applications of the CLT for statistics that we can convert into robust \( p \)-values. Like U-statistics, scan statistics by nature involve dependent random variables, and they have also been shown to be quite generally expressible in martingale form ([PGKS05]).

- **Sampling Without Replacement**: The classic work [Ser74] and more recently [BM15] derive Hoeffding- and Bernstein-type (first- and second-moment) concentration inequalities for sums when sampling without replacement. This obviates independence assumptions on the data being sampled, which is ideally suited for
finite populations and robust learning applications.

15.3.2 Related Work

Treating the sample size as random is central to the area of sequential testing. A prototypical example is the Sequential Probability Ratio Test (SPRT) of Wald and Wolfowitz ([WW48]), which considers the likelihood-ratio martingale and is for a very specific problem in which the null and alternative probability densities are known, so that it stops optimally soon given particular type I and type II error constraints. The likelihood-ratio martingale has been explored for stopping in other contexts as well ([DR68b, RS70, BBW97]), including for composite hypotheses. General martingales can be used in sequential tests, a possibility which has been explored since Wald (see [Rob70a, RS70]).

More recently, the work of Chapter 14 ([BR16]) presents sequential nonparametric two-sample tests in a framework related to ours. However, such work requires changing the algorithm itself to be a sequential test in an appropriate setting, with a specified level of $\alpha$. In contrast, we crucially observe that the setting when conditional means and variances can be conveniently evaluated or bounded, the $p$-value can be reported uniformly over time in very general settings.

Sequential testing involves specifying a type I error a priori (and sometimes also type II, e.g. for the SPRT), while what we are reporting is a minimum significance level at which the data show a deviation from the null. This is exactly analogous to the relationship between Fisher-style significance testing and Neyman-Pearson hypothesis testing – the method of this paper can be considered a robust Fisher-style significance test under martingale nulls, just as sequential testing builds on the Neyman-Pearson framework. Similarly, we do not analyze any alternative hypothesis, which would affect the power of the test (though for some common methods, we have shown in Chapter 14
that power calculations stay roughly the same relative to the batch statistic, so the choice of test statistic governs the power).

We follow a line of work by Vovk and coauthors ([Vov93, SSVV11]), which develops the idea of correcting the $p$-values uniformly over time using a “test martingale,” and contains further historical references on this idea, viewed within the context of Bayes factors and likelihood ratios. This has drawn recent attention for its robustness to optimal stopping as well ([Grü16]). By introducing the finite-time LIL and also showing its anti-concentration, we are able to make the idea broadly applicable even to frequentist tests that use (martingale) CLTs, with a provably optimal method.

This issue’s importance has long been recognized in the practice of statistical testing ([Rob52, AMR69, Nic00, Wag07, SNS11]). The statistician typically does not know their sampling plan, which is necessary for standard hypothesis tests. This is subject to several sources of variation: it could be unethical to continue sampling when a significant effect is detected in a clinical trial ([Ioa08]), or the experimenter could run out of resources to gather more data.

Solutions to this problem are often semi-heuristic and generally involve “spending a budget of $\alpha$,” the willingness to wrongly reject the null, over time. Such methods are widely used ([PPA+77, Poc77]) but are not uniformly robust to sampling strategies, and their execution suffers from many complications which depend on the application ([Poc05]).

### 15.3.3 Proof Discussion

The techniques we use were pioneered in [RS70] (cf. Example 4) in a strictly asymptotic setting for Brownian motion. Our arguments here generalize some of those made in that paper ([RS70], Section 4) to finite times.

Finally, we point out that the essential arguments linking weights in the “scale” $\lambda$-
domain to the iterated-logarithm order of growth of the moment were noticed as early as Feller ([Fel43]). This links back to the remarkable “Erdős-Feller-Kolmogorov-Petrowsky LIL” ([Erd42]) characterization of the optimal rate of growth of the LIL envelope, which appears to bear a resemblance to the proof of Theorem 45. An interesting open problem would be recovering that result as the limit of a non-asymptotic bound proved with our techniques. Also open is whether we can tighten the anti-concentration bound, as discussed earlier.

15.4 Proof of Tight Finite-Time LIL Concentration (Theorem 45)

We prove a generalization of Theorem 45 in this section, which is stated below as Theorem 48. The proof is similar to the main proof of Ch. 12 (in Sec. 12.3), so we adopt the setting of that chapter here.

Therefore, for $M_t$ to be a martingale and $U_t$ to be an increasing “variance” process such that the process

$$X^\lambda_t := \exp \left( \lambda M_t - \frac{\lambda^2}{2} U_t \right)$$

is a supermartingale for any $\lambda \in \left( -\frac{1}{\exp_v(2)}, \frac{1}{\exp_v(2)} \right) \setminus \{0\}$. Section C.2.2 contains more examples of appropriate $U_t$ for various classes of martingales, including any with sub-exponential steps. In particular, the $z$-test correction is derived with $U_t = t$, and the cumulative variance-based corrections of Sections 15.2.2 and 15.2.3 instead use $U_t = 2(e - 2)V_t$.

15.4.1 Proof Overview

Recall the optional stopping theorem for positive supermartingales.

*Theorem 47* (Optional Stopping for Nonnegative Supermartingales ([Dur10], Theorem
Let $M_t$ be a nonnegative supermartingale. Then if $\tau$ is a (possibly infinite) stopping time, $\mathbb{E}[M_\tau] \leq \mathbb{E}[M_0]$.

The leading proportionality constant on the iterated-logarithm term in the results of Ch. 12 is $\sqrt{6}$, a multiplicative $\sqrt{3}$ factor above the LIL’s asymptotic $\sqrt{2}$. This deflates the reported $p$-value by an amount that is exponential in this factor, not multiplicative (recall $A_t \propto \exp \left( -\frac{M_t^2}{C^2 V_t} \right)$ for $C = \sqrt{2}$ or $\sqrt{6}$), which would profoundly worsen the empirical performance of our methods if used directly. The suboptimality is chiefly because the mixing distribution over $\lambda$ is not refined enough to get optimal constants.

Here, we give a countably infinite family of averaging distributions for $\lambda$ that generalize the one used in Ch. 12, and we use them in a more direct proof method with optimal constants as $t$ grows. Theorem 48 is a special case of this result when $v = 2$.

To describe this set, define $\ln_v(x) = \ln \ln \ldots \ln(x)$ and $\exp_v(x) = \exp \exp \ldots \exp(x)$ for $v = 1, 2, \ldots$. The following family of probability distributions is indexed by $v$:

$$P^v_\lambda(d\lambda) = \frac{d\lambda}{|\lambda| \ln_v \left( \frac{1}{|\lambda|} \right) \prod_{i=1}^v \ln_i \left( \frac{1}{|\lambda|} \right)} \quad \text{for } \lambda \in \left[ -\frac{1}{\exp_v(2)}, \frac{1}{\exp_v(2)} \right] \setminus \{0\}$$

Note that $P^1_\lambda$ is used to mix over $\lambda$ in the main proof of Chapter 12.

We use $P^v_\lambda$ (whose PDF we write as $P^v(\lambda)$) to average over the exponential supermartingales in this section, proving the following theorem.

**Theorem 48.** Choose any integer $v \geq 1$. Then w.p. $\geq 1 - \delta$, for all $t \geq 16(\exp_v(2))^2 \ln \frac{1}{\delta}$,

$$|M_t| \leq \frac{U_t}{2\exp_v(2)}$$

and

$$|M_t| \leq \sqrt{2U_t \left( \ln \frac{1}{\delta} + \ln_2 \frac{1}{\delta} + \ln_2 (U_t) + 2 \sum_{i=3}^{v+1} \ln_i (U_t) + 2 \ln_{v+1} (U_t) \right)} \quad (15.6)$$

Define some stopping times associated with each $P^v_\lambda$. The first is the initial time
after which our bounds hold uniformly (see Ch. 12 for details):

$$\tau_v^0 := \min \left\{ t : U_t \geq 16(%exp(v)(2))^2 \ln \frac{1}{\delta} \right\}$$  \hspace{1cm} (15.7)

and the second is the stopping time used to prove uniform concentration bounds.

$$\tau_v = \min \left\{ t \geq \tau_v^0 : |M_t| > \frac{U_t}{2\exp(v)(2)} \lor \left[ |M_t| \leq \frac{U_t}{2\exp(v)(2)} \land |M_t| > \sqrt{2 \left( \ln \frac{1}{\delta} + \ln_2 \frac{1}{\delta} + \ln_2 (U_t) + 2 \sum_{i=3}^{v+1} \ln_i (U_i) + 2\ln_{v+1} (U_i) \right)} \right] \right\}$$  \hspace{1cm} (15.8)

$$\tau_v$$ is basically the stopping time used by a peeking adversary. It is designed so that whenever it is finite, the expected moment-generating function of the stopped martingale mixture is above the large constant $$\frac{1}{\delta}$$.

**Lemma 49.** Suppose $$\tau_v < \infty$$. Then if the averaging distribution over $$\lambda$$ is $$P_v^\lambda$$,

$$\mathbb{E}^\lambda \left[ X_{\tau_v}^\lambda \right] \geq \frac{1}{\delta}$$

The proof of Theorem 48 is then a textbook stopped martingale mixing argument, using these lemmas.

**Proof of Theorem 48.** Let the averaging distribution over $$\lambda$$ be $$P_v^\lambda$$. It suffices to prove that $$P(\tau_v < \infty) \leq \delta$$, which is done as follows:

$$1 = \mathbb{E}^\lambda \left[ \mathbb{E} \left[ X_0^\lambda \right] \right] \overset{(a)}{=} \mathbb{E}^\lambda \left[ \mathbb{E} \left[ X_{\tau_v}^\lambda \right] \right] \overset{(b)}{=} \mathbb{E} \left[ \mathbb{E}^\lambda \left[ X_{\tau_v}^\lambda \right] \right]$$

$$\geq \mathbb{E} \left[ \mathbb{E}^\lambda \left[ X_{\tau_v}^\lambda \right] \mid \tau_v < \infty \right] P(\tau_v < \infty) \overset{(c)}{=} \frac{1}{\delta} \mathbb{P}(\tau_v < \infty)$$
where (a) is by Optional Stopping (Theorem 47), (b) is by Tonelli’s Theorem, and (c) is by Lemma 49.

15.4.2 Proof of Theorem 48 – Details

Proof of Lemma 49. Recall that the probability density function over \( \lambda \) is \( P^\nu(\lambda) \). Write \( \tau \) as \( \tau \) and define \( K_\nu := \frac{1}{\exp(2)} \) for convenience in this proof. Take \( \tau < \infty \) throughout.

\[
\mathbb{E}^\lambda \left[ X^\lambda_\tau \right] = \int_{-K_\nu}^0 X^\lambda_\tau P^\nu(\lambda) + \int_0^{K_\nu} X^\lambda_\tau P^\nu(\lambda)
= \int_{-K_\nu}^0 \exp \left( \lambda M_\tau - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda) + \int_0^{K_\nu} \exp \left( \lambda M_\tau - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda)
\geq \int_0^{K_\nu} \exp \left( \lambda |M_\tau| - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda)
\tag{15.9}
\]

The proof breaks into two cases, depending on how \( \tau_\nu \) is triggered in Eq. (15.8).

Either \( |M_\tau| > \frac{U_\tau}{2} K_\nu \); or \( |M_\tau| \leq \frac{U_\tau}{2} K_\nu \) but \( |M_\tau| \) violates the iterated-logarithm bound.

**First Case:** \( |M_\tau| > \frac{U_\tau}{2} K_\nu \)

If \( |M_\tau| > \frac{U_\tau}{2} K_\nu \), (15.9) becomes

\[
\mathbb{E}^\lambda \left[ X^\lambda_\tau \right] \geq \int_0^{K_\nu} \exp \left( \lambda K_\nu U_\tau - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda)
= \exp \left( \frac{U_\tau K_\nu^2}{8} \right) \int_0^{K_\nu} \exp \left( - \frac{U_\tau}{2} \left( \lambda - \frac{K_\nu}{2} \right)^2 \right) P^\nu(\lambda)
= \exp \left( \frac{U_\tau K_\nu^2}{8} \right) \sqrt{\frac{2\pi}{U_\tau}} \left( 1 - 2\Phi \left( - \frac{1}{2} K_\nu \sqrt{U_\tau} \right) \right)
\]

where the last equality evaluates the Gaussian integral by changing variables to \( x := \sqrt{U_\tau} \left( \lambda - \frac{K_\nu}{2} \right) \), and \( \Phi \) is the standard normal CDF. Now \( U_\tau \) is high enough such that
2\Phi \left( -\frac{1}{2} K \sqrt{U_\tau} \right) \leq \frac{1}{e \sqrt{\pi}}, \text{ giving}

\mathbb{E}^\lambda \left[ X_\tau^\lambda \right] \geq \exp \left( \frac{U_\tau K^2}{8 \sqrt{U_\tau}} \right) \left( \sqrt{2\pi} - \frac{\sqrt{2}}{e} \right) \geq \left( \sqrt{2\pi} - \frac{\sqrt{2}}{e} \right) \exp \left( \frac{U_\tau K^2}{16} \right)

Using \ U_\tau \geq U_{\tau_0} \ gives \ the \ result, \ when \ \mid M_\tau \mid > \frac{U_\tau}{2} K. \ So \ for \ the \ rest \ of \ the \ proof \ we \ can \ assume \ \mid M_\tau \mid \leq \frac{U_\tau}{2} K.

\textbf{Second Case:} \mid M_\tau \mid \leq \frac{U_\tau}{2} K -

Writing \ Z := \frac{\mid M_\tau \mid}{\sqrt{U_\tau}} \ as \ the \ standardized \ process,

\mathbb{E}^\lambda \left[ X_\tau^\lambda \right] \geq \int_0^{K_\tau} \exp \left( \lambda \mid M_\tau \mid - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda) 
\geq \int_0^{2Z/\sqrt{U_\tau}} \exp \left( \lambda Z \sqrt{U_\tau} - \frac{1}{2} \lambda^2 U_\tau \right) P^\nu(\lambda) \quad (15.10)

Completing the square in Eq. (15.10) gives

\mathbb{E}^\lambda \left[ X_\tau^\lambda \right] \geq \exp \left( \frac{1}{2} Z^2 \right) \int_0^{2Z/\sqrt{U_\tau}} \exp \left( -\frac{1}{2} \left( \lambda \sqrt{U_\tau} - Z \right)^2 \right) P^\nu(\lambda) \quad (15.11)
\geq \exp \left( \frac{1}{2} Z^2 \right) P^\nu \left( \frac{2Z}{\sqrt{U_\tau}} \right) \int_0^{2Z/\sqrt{U_\tau}} \exp \left( -\frac{1}{2} \left( \lambda \sqrt{U_\tau} - Z \right)^2 \right) d\lambda \quad (15.12)

where Eq. (15.12) follows by the monotonicity of \ P^\nu(\lambda).

Here we can evaluate the inner Gaussian integral by transforming the integrand to be in terms of \ x := \lambda \sqrt{U_\tau} - Z:

\int_0^{2Z/\sqrt{U_\tau}} \exp \left( -\frac{1}{2} \left( \lambda \sqrt{U_\tau} - Z \right)^2 \right) d\lambda = \sqrt{\frac{2\pi}{U_\tau}} \int_{-Z}^{Z} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} x^2 \right) dx
\quad = \sqrt{\frac{2\pi}{U_\tau}} (\Phi(Z) - \Phi(-Z)) = \sqrt{\frac{2\pi}{U_\tau}} (1 - 2\Phi(-Z))
Substituting this into Eq. (15.12) and using the bound \( \Phi(x) \leq \exp(-x^2/2) \frac{1}{|x|\sqrt{2\pi}} \),

\[
\mathbb{E}^\lambda[X^2_\tau] \geq \exp\left(\frac{1}{2}Z^2\right) P^\nu\left(\frac{2Z}{\sqrt{U_\tau}}\right) \sqrt{\frac{2\pi}{U_\tau}} (1 - 2\Phi(-Z))
\]

\[
\geq \sqrt{\frac{2}{U_\tau}} \exp\left(\frac{1}{2}Z^2\right) P^\nu\left(\frac{2Z}{\sqrt{U_\tau}}\right) \left(\sqrt{\pi} \frac{-\sqrt{2}e^{-Z^2/2}}{|Z|}\right)
\]

\[
\overset{(a)}{\geq} \sqrt{\frac{2}{U_\tau}} \exp\left(\frac{1}{2}Z^2\right) P^\nu\left(\frac{2Z}{\sqrt{U_\tau}}\right) \left(\sqrt{\pi} \frac{-1}{e}\right)
\]

\[
= \frac{\exp\left(\frac{1}{2}Z^2\right)}{\sqrt{2Z} \ln\left(\frac{\sqrt{U_\tau}}{Z}\right) \left[\prod_{i=1}^{v}\ln\left(\frac{\sqrt{U_\tau}}{Z}\right)\right]} \left(\sqrt{\pi} \frac{-1}{e}\right)
\]

\[
\geq \frac{\exp\left(\frac{1}{2}Z^2\right)}{\sqrt{2Z} \ln\left(\frac{\sqrt{U_\tau}}{v}\right) \ln\left(U_\tau\right) \left[\prod_{i=1}^{v}\ln\left(U_\tau\right)\right]} \left(\sqrt{\pi} \frac{-1}{e}\right) \geq \frac{\exp\left(\frac{1}{2}Z^2\right)}{Z \ln\left(U_\tau\right) \left[\prod_{i=1}^{v}\ln\left(U_\tau\right)\right]} \geq \frac{\exp\left(\frac{1}{2}Z^2\right)}{\frac{1}{\delta}}
\]

where \( (a) \) uses that \( Z \geq \sqrt{2} \), so \( \frac{\sqrt{2}e^{-Z^2/2}}{|Z|} \leq \frac{1}{e} \). Using the definition of \( \tau \), Lemma 50, and the assumption \( |M_\tau| \leq U_\tau^2 K_v \), we see that \( \mathbb{E}^\lambda[X^2_\tau] \geq \frac{1}{\delta} \), as desired.

\[\square\]

**Lemma 50.** For any (stopping) time \( \tau \), if

\[
Z \geq \sqrt{2 \left( \ln\frac{1}{\delta} + \ln_2 \frac{1}{\delta} + \ln_2(U_\tau) + 2 \sum_{i=3}^{r+1} \ln_i(U_\tau) + 2 \ln_{i+1}(U_\tau) \right)}
\]

then

\[
\frac{\exp\left(\frac{1}{2}Z^2\right)}{Z \ln\left(U_\tau\right) \left[\prod_{i=1}^{v}\ln\left(U_\tau\right)\right]} \geq \frac{1}{\delta}.
\]

**Proof of Lemma 50.** Let \( Y \) be the lower bound on \( Z \) that we assume in the implication.

We will prove

\[
\frac{\exp\left(\frac{1}{2}Y^2\right)}{Y \ln\left(U_\tau\right) \left[\prod_{i=1}^{v}\ln\left(U_\tau\right)\right]} \geq \frac{1}{\delta}
\]

from which the result follows because the left-hand side is increasing in \( Y \), and \( Z \geq Y \).
By the subadditivity of the $\ln(\cdot)$ function,

$$2\ln Y = \ln 2 + \ln \left( \frac{1}{\delta} + \sum_{i=2}^{v+1} \ln (U_\tau) + 2 \ln_{v+1} (U_\tau) \right)$$

$$\leq \ln 2 + \ln_2 \frac{1}{\delta} + \ln_3 \frac{1}{\delta} + 2 \sum_{i=2}^{v+2} \ln (U_\tau) + 2 \ln_{v+2} (U_\tau)$$

$$\leq 2 \left( \ln_2 \frac{1}{\delta} + \sum_{i=3}^{v+2} \ln (U_\tau) + \ln_{v+2} (U_\tau) \right)$$

(15.13)

where the last inequality uses the bound $(\ln \frac{1}{\delta})^2 \leq \frac{1}{\delta}$. Therefore, by bounding one of the terms in the definition of $Y$ with the inequality $\ln_{v+1} (U_\tau) \geq 2 \ln_{v+2} (U_\tau)$,

$$Y \geq \sqrt{2 \left( \ln \frac{1}{\delta} + \ln_2 \frac{1}{\delta} + \ln_3 \frac{1}{\delta} + 2 \sum_{i=2}^{v+1} \ln (U_\tau) + \ln_{v+2} (U_\tau) \right)}$$

$$= \sqrt{2 \left( \ln \frac{1}{\delta} + \sum_{i=2}^{v+1} \ln (U_\tau) + \ln_{v+1} (U_\tau) \right) + 2 \left( \ln_2 \frac{1}{\delta} + \sum_{i=3}^{v+2} \ln (U_\tau) + \ln_{v+2} (U_\tau) \right)}$$

$$\geq \sqrt{2 \left( \ln \frac{1}{\delta} + \ln Y + \sum_{i=2}^{v+1} \ln (U_\tau) + \ln_{v+1} (U_\tau) \right)}$$

where $(a)$ uses Eq. (15.13). Solving for $\frac{1}{\delta}$, $\frac{\exp(\frac{1}{2}Y^2)}{Y \ln_2 (U_\tau) (\prod_{i=1}^{v+1} \ln (U_\tau))} \geq \frac{1}{\delta}$.

Meeting this definition are all distributions with exponential tails, including the exponential and $\chi^2$ distributions; further characterization is in [BLM13a], where standard Chernoff-type large deviation bounds are proved from the definition above. Notice the difference to sub-Gaussian random variables, for which the inequality is required to hold $\forall \lambda$. 

\[\square\]
Appendix A

Minimax Theorem

The minimax theorem is used in several key places in this dissertation. For all these invocations, it suffices to use Sion’s powerful version:

**Theorem 25** (Application of [Sio58], Cor. 3.3). *Suppose* $X, Y$ *are convex subsets of a linear topological space* (e.g., $\mathbb{R}^d$ *for any finite* $d$) *and at least one is compact. If* $f$ *is a real-valued function on* $X \times Y$ *satisfying*

- $\forall x \in X : f(x, y)$ *is upper semicontinuous and quasiconcave on* $Y$

- $\forall y \in Y : f(x, y)$ *is lower semicontinuous and quasiconvex on* $X$

*then*

$$\inf_{x \in X} \sup_{y \in Y} f(x, y) = \sup_{y \in Y} \inf_{x \in X} f(x, y)$$
Appendix B

Proofs from Chapter 8

Proof of Theorem 8. The proof follows the analysis strategy introduced in Chapter 4 closely. Examining (8.1), note that \( \ell(z,g) \) is linear in \( z \), so the constrained maximization over \( z \) is basically

\[
\max_{z \in [-1,1]^n, \frac{1}{n}Fz \geq b} \left( -\frac{1}{n} \sum_{i=1}^{n} p_j g_j \right) = \max_{z \in [-1,1]^n, \frac{1}{n}Fz \geq b} \left( -\frac{1}{n} z^\top [p \circ g] \right) - \frac{1}{n} \sum_{j=1}^{n} p_j z_j g_j
\]

(B.1)

\[
= \min_{\sigma \geq 0} \left[ -b^\top \sigma + \frac{1}{n} \| F^\top \sigma - (p \circ g) \|_1 \right]
\]

(B.2)

where the last equality uses Lemma 11 (a basic application of Lagrange duality proved in Chapter 4).

Substituting (B.1) into (8.1) and simplifying,

\[
V_c = \frac{1}{2} \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \left[ -\frac{1}{n} \sum_{j=1}^{n} p_j (1 - 2c) + \max_{\sigma \geq 0} \left( -\frac{1}{n} \sum_{j=1}^{n} p_j z_j g_j \right) \right]
\]

\[
= c + \frac{1}{2} \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \left[ -\frac{1}{n} \sum_{j=1}^{n} p_j (1 - 2c) + \min_{\sigma \geq 0} \left( -b^\top \sigma + \frac{1}{n} \| F^\top \sigma - (p \circ g) \|_1 \right) \right]
\]

\[
= \frac{1}{2} \min_{\sigma \geq 0} \left[ -b^\top \sigma + \min_{p \in [0,1]^n} \frac{1}{n} \sum_{j=1}^{n} \left[ 2c + p_j (1 - 2c) + \min_{g_j \in [-1,1]} |x_j^\top \sigma - p_j g_j| \right] \right]
\]

(B.3)
Now we consider only the innermost minimization over $g_j$ in (B.3) for any $j$ (following the procedure of the main proofs of [BF15a, BF16b]).

If $p_j = 0$, then the inner minimization’s objective is simply $\left| x_j^\top \sigma \right|$. Otherwise, it can be seen that

$$
\min_{g_j \in [-1,1]} \left| x_j^\top \sigma - p_j g_j \right| = p_j \min_{g_j \in [-1,1]} \left| \frac{x_j^\top \sigma}{p_j} - g_j \right| = p_j \left( \Psi \left( \frac{x_j^\top \sigma}{p_j} \right) - 1 \right) = \left[ \left| x_j^\top \sigma \right| - p_j \right]_+ \tag{B.5}
$$

with the minimizer $g_j^* = \text{clip} \left( \frac{x_j^\top \sigma}{p_j} \right)$. So we can rewrite Equation (B.3) as

$$
V_c = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^{n} \min_{p_j \in [0,1]} \left[ 2c + p_j(1-2c) + \left[ \left| x_j^\top \sigma \right| - p_j \right]_+ \right] \right] \tag{B.6}
$$

Now the minimand over $p_j$ is convex in $p_j$; since $c \in (0, \frac{1}{2}]$, it is decreasing in $p_j$ for $p_j < \left| x_j^\top \sigma \right|$ and increasing for $p_j > \left| x_j^\top \sigma \right|$. Therefore, it is minimized when $p_j^* = \min \left( \left| x_j^\top \sigma \right|, 1 \right)$. Substituting this $p_j^*$ into the minimization over $p_j$ in Equation (B.6) gives the form of $\Psi(\cdot, c)$, and therefore the result.

Proof of Theorem 9. We begin exactly as in the proof of Theorem 8, by rewriting the
constrained maximization over $z$ in the definition of $V_\alpha$:

$$
V_\alpha = \frac{1}{2} \min_{g \in [-1,1]^n} \min_{\frac{1}{n} \textbf{1}^T p \geq 1 - \alpha} \left[ \frac{1}{n} \sum_{j=1}^{n} p_j + \max_{z \in [-1,1]^n, \frac{1}{n} F z \geq b} - \frac{1}{n} \sum_{j=1}^{n} p_j z_j g_j \right]
$$

$$
= \frac{1}{2} \min_{g \in [-1,1]^n} \min_{\frac{1}{n} \textbf{1}^T p \geq 1 - \alpha} \left[ \frac{1}{n} \sum_{j=1}^{n} p_j + \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \frac{1}{n} \left\| F^T \sigma - (p \circ g) \right\|_1 \right] \right]
$$

$$
= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \min_{\sigma \geq 0^p} \left[ \frac{1}{n} \sum_{j=1}^{n} p_j + \min_{g_j \in [-1,1]} \left[ x^T_j \sigma - p_j g_j \right] \right] \right]
$$

$$
= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \min_{\sigma \geq 0^p} \left[ \frac{1}{n} \sum_{j=1}^{n} p_j + \left[ x^T_j \sigma - p_j \right]_+ \right] \right] (B.7)
$$

where at each step we follow the sequence of steps in the proof of Theorem 8. Again, the minimizer $g_j^* = \text{clip} \left( \frac{x^T_j \sigma}{p_j} \right)$.

Now we simplify Equation (B.7), introducing a Lagrange parameter for the constraint on $p$ involving $\alpha$.

$$
V_\alpha = \frac{1}{2} \min_{p \in [0,1]^n, \sigma \geq 0^p, \frac{1}{n} \textbf{1}^T p \geq 1 - \alpha} \left[ -b^T \sigma + \frac{1}{n} \sum_{j=1}^{n} p_j + \left[ x^T_j \sigma - p_j \right]_+ \right]
$$

$$
= \frac{1}{2} \min_{p \in [0,1]^n} \max_{\lambda \geq 0} \min_{\sigma \geq 0^p} \left[ -b^T \sigma + \lambda (1 - \alpha) + \frac{1}{n} \sum_{j=1}^{n} p_j (1 - \lambda) + \left[ x^T_j \sigma - p_j \right]_+ \right] (B.8)
$$

For any $j \in [n]$, the summand is convex in $p_j$. So we can apply the minimax theorem ([SF12], p. 144) to (B.8), moving $\min_{p \in [0,1]^n}$ inside the maximization while
maintaining equality. This yields

\[ V_\alpha = \frac{1}{2} \max_{\lambda \geq 0} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \lambda (1 - \alpha) + \frac{1}{n} \sum_{j=1}^{n} \min_{p_j \in [0,1]} \left[ p_j (1 - \lambda) + \left[ x_j^\top \sigma - p_j \right]_+ \right] \right] \]

using the fact that the \(L^\infty\) constraints over \(p\) decouple over the coordinates.

For any \(\lambda \geq 0\), the inner minimand over \(p_j\) is decreasing for \(p_j \leq |x_j^\top \sigma|\) and increasing for \(p_j > |x_j^\top \sigma|\). So the minimizing \(p_j^* = \min \left( |x_j^\top \sigma|, 1 \right)\). Substituting this in the minimization gives

\[ V_\alpha = \frac{1}{2} \max_{\lambda \geq 0} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \lambda (1 - \alpha) + \frac{1}{n} \sum_{j=1}^{n} \left( \Psi \left( x_j^\top \sigma, \frac{\lambda}{2} \right) - \lambda \right) \right] \]

giving the result. \(\square\)

**Proof of Prop. 10.** The result follows from checking the first-order optimality condition

\[ \frac{\partial \left[ w(\lambda) - \lambda \alpha \right]}{\partial \lambda} \bigg|_{\lambda = \lambda^*(\alpha)} = 0 \]

\(\square\)

**Proof of Theorem 11.** The definition of \(V(\alpha)\) follows by substituting \(\lambda^*(\alpha)\) in the definition of \(V\) in terms of \(\lambda\). Nonnegativity is immediate by the primal definition of \(V\). As for convexity,

\[ V'(\alpha) = \frac{1}{2} \left( \frac{\partial \left[ \lambda^*(\alpha) \right]}{\partial \alpha} \right) w'(\lambda^*(\alpha)) - \alpha \frac{\partial \left[ \lambda^*(\alpha) \right]}{\partial \alpha} - \lambda^*(\alpha) \]

\[ = \frac{1}{2} \left( \frac{\partial \left[ \lambda^*(\alpha) \right]}{\partial \alpha} \right) [w'(\lambda^*(\alpha)) - \alpha] - \lambda^*(\alpha) \]

\[ = -\frac{1}{2} \lambda^*(\alpha) \leq 0 \]

where the last equality is by subgradient conditions defining \(\lambda^*(\alpha)\) as the maximizer.
over $\lambda$.

We know $\lambda^*(\alpha)$ is decreasing with $\alpha$, so $V''(\alpha) \geq 0$ and $V$ is convex. The boundary conditions at $\alpha = 0, 1$ follow by straightforward computation.

\begin{proof}[Proof of Lemma 12] From the definition of $\Psi$, we have $\lim_{p \to 0^+} \frac{\Psi \left( \frac{x_j^\top \sigma}{p} \right)}{x_j^\top \sigma} = \text{sgn} \left( x_j^\top \sigma \right)$, so $\lim_{p \to 0^+} K \left( p, x_j, \sigma \right) x_j^\top \sigma = \text{sgn} \left( x_j^\top \sigma \right)$, and so $K(0, x_j, \sigma) = 0$. Moreover, $\frac{\partial^2 K(p_j, x_j, \sigma)}{\partial p_j} = \frac{(x_j^\top \sigma)^2}{p_j^3} \Psi'' \left( \frac{x_j^\top \sigma}{p_j} \right) \geq 0$ because $\Psi$ is convex, which makes $G$ convex in $p_j$.
\end{proof}

\begin{proof}[Proof of Theorem 14] Note that $\ell(z, g)$ is linear in $z$, so the constrained maximization over $z$ is basically

$$\max_{z \in [-1, 1]^n, \frac{1}{n} Fz \geq b} \frac{1}{n} \sum_{i=1}^n (1 - a_i) z_i \left( \ell_+ (g_i) - \ell_- (g_j) \right)$$

(B.9)

$$= \max_{z \in [-1, 1]^n, \frac{1}{n} Fz \geq b} - \frac{1}{n} z^\top [p \circ \Gamma \{g\}] = \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \left\| F^\top \sigma - (p \circ \Gamma \{g\}) \right\|_1 \right]$$

(B.10)

where the last equality uses Lemma 11 (a basic application of Lagrange duality proved in Chapter 4).

For convenience define $\xi_j := p_j [\ell_+ (g_j) + \ell_- (g_j)] + 2(1 - p_j)c$. Substituting
\[ V = \frac{1}{2} \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \left[ \frac{1}{n} \sum_{j=1}^{n} \xi_j + \max_{x \in [-1,1]^n} \frac{1}{n} \sum_{j=1}^{n} p_j z_j \left( \ell_+(g_j) - \ell_-(g_j) \right) \right] \]

\[
\frac{1}{2} \min_{g \in [-1,1]^n} \min_{p \in [0,1]^n} \left[ \frac{1}{n} \sum_{j=1}^{n} \xi_j + \min_{\sigma \geq 0^p} \left\| -b^\top \sigma + \frac{1}{n} \left\| F^\top \sigma - (p \circ \Gamma(g)) \right\|_1 \right\| \right] \\
= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \min_{p \in [0,1]^n} \min_{g \in [-1,1]^n} \left[ \frac{1}{n} \sum_{j=1}^{n} \xi_j + \frac{1}{n} \left\| F^\top \sigma - (p \circ \Gamma(g)) \right\|_1 \right] \right] \\
= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \min_{p \in [0,1]^n} \frac{1}{n} \sum_{j=1}^{n} \left[ 2(1 - p_j)c + \min_{g_j \in [-1,1]} U(\sigma, p_j, g_j) \right] \right] \tag{B.11}
\]

where we define \( U(\sigma, p_j, g_j) \) as:

\[ U(\sigma, p_j, g_j) := p_j (\ell_+(g_j) + \ell_-(g_j)) + \left\| x_j^\top \sigma - p_j \Gamma(g_j) \right\|_1. \]

Now we follow the procedure of the main proof of [BF16b], for now considering just the innermost minimization problem \( \min_{g_j \in [-1,1]} U(\sigma, p_j, g_j) \) for any \( j \).

If \( p_j = 0 \) in (B.11), then the inner minimization’s objective is simply \( \left\| x_j^\top \sigma \right\|_1 \).

Otherwise, we note that the absolute value breaks down into two cases, so the inner minimization’s objective can be simplified:

\[
U(\sigma, p_j, g_j) = \begin{cases} 
 p_j \left( 2\ell_+(g_j) + \frac{x_j^\top \sigma}{p_j} \right) & \text{if } \frac{x_j^\top \sigma}{p_j} \geq \Gamma(g_j) \\
 p_j \left( 2\ell_-(g_j) - \frac{x_j^\top \sigma}{p_j} \right) & \text{if } \frac{x_j^\top \sigma}{p_j} < \Gamma(g_j)
\end{cases} \tag{B.12}
\]

This is nearly the same situation faced in the proof of Theorem 4 in Chapter 4 ([BF16b]), except with \( x_j^\top \sigma \) replaced by \( \frac{x_j^\top \sigma}{p_j} \). So we proceed with that argument, finding that:

\[
\min_{g_j \in [-1,1]} U(\sigma, p_j, g_j) = p_j \Psi \left( \frac{x_j^\top \sigma}{p_j} \right)
\]
with the minimizing $g_j^*$ being of the same form as Theorem 4, but dependent on the quantity $\frac{x_j^\top \sigma}{p_j}$ instead of $x_j^\top \sigma$.

So we can rewrite Equation (B.11) as

$$V = Eq. (B.11)$$

$$= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \min_{p_j \in [0,1]} \left[ 2(1 - p_j)c + \min_{g_j \in [-1,1]} U(\sigma, p_j, g_j) \right] \right]$$

$$= \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n \min_{p_j \in [0,1]} \left[ 2(1 - p_j)c + p_j \Psi \left( \frac{x_j^\top \sigma}{p_j} \right) \right] \right]$$

Now the minimand over $p_j$ is convex in $p_j$, by Lemma 12. Using first-order optimality conditions, observe that $p_j^*$ is such that $2c = K_{j,\sigma}(p_j^*)$, if the implied $p_j^*$ is $\leq 1$, and 1 otherwise. Therefore,

$$p_j^* = \min \left( K_{j,\sigma}^{-1}(2c), 1 \right)$$

and so we have

$$V = Eq. (B.11) = \frac{1}{2} \min_{\sigma \geq 0^p} \left[ -b^\top \sigma + \frac{1}{n} \sum_{j=1}^n Q_j(\sigma, 2c) \right] + c \tag{B.13}$$

as desired. \qed
Appendix C
Proofs from Chapter 12

In these proofs, recall the setting of Sec. 12.3 and its definition of $U_t$, and $k := \frac{1}{3}$ and $\lambda_0 := \frac{1}{e^3(1+\sqrt{k})}$.

C.1 Proof of Theorem 19 (Anti-Concentration Bound)

In this section, let $M_t$ be the Rademacher random walk, and $C_1$ be as defined in Theorem 19. Our proof will use submartingales rather than supermartingales; therefore, we will employ a standard optional stopping theorem for submartingales (paralleling Theorem 33):

*Theorem 51 (Optional Stopping for Submartingales ([Dur10])).* Let $M_t$ be a submartingale. Then if $\tau$ is an a.s. bounded stopping time, $E[M_\tau] \geq E[M_0]$.

We will also construct a family of exponential submartingales, analogous to the supermartingale construction of Lemma 24 in the concentration bound proof.

*Lemma 52.* The process $Z_t^\lambda := \exp(\lambda M_t - k\lambda^2 U_t)$ is a submartingale for $\lambda \in \left[-\frac{1}{e^2}, \frac{1}{e^2}\right]$.

*Proof.* We rely on the inequality $\cosh(x) \geq e^{kx^2}$ over $x \in \left[-\frac{1}{e^2}, \frac{1}{e^2}\right]$, so that

\[
E[Z_t^\lambda \mid \mathcal{F}_{t-1}] = E[\exp(\lambda \xi_t) \mid \mathcal{F}_{t-1}] e^{-k\lambda^2 Z_{t-1}^\lambda} = \cosh(\lambda) e^{-k\lambda^2 Z_{t-1}^\lambda} \geq Z_{t-1}^\lambda.
\]

244
C.1.1 Preliminaries and Proof Overview

Many aspects of this proof parallel that of the concentration bound of Theorem 22. Again, the idea is to choose \( \lambda \) stochastically from a probability space \((\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)\) such that
\[
P_\lambda(d\lambda) = \frac{d\lambda}{|\lambda|(\ln|\lambda|)^2};
\]
and the parameter \( \lambda \) is chosen independently of the \( \xi_1, \xi_2, \ldots \), so that \( Z_t^\lambda \) is defined on the product space.

As in the concentration bound proof, write \( \mathbb{E}^\lambda [\cdot] \) to denote the expectation with respect to \((\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)\). For consistency with previous notation, we continue to write \( \mathbb{E} [\cdot] \) to denote the expectation w.r.t. the original probability space \((\Omega, \mathcal{F}, P)\) which encodes the stochasticity of \( M_t \).

Just as for the concentration bound, our proof of this anti-concentration bound reasons about the value of a particular stopping time. However, the stopping time used here is slightly different, because the conditions for convergence of our submartingales are more restrictive than those for supermartingales.

To be concrete, define \( \sigma_\delta := \frac{e^4}{k} \ln \left( \frac{2}{\delta} \right) \). For a given finite time horizon \( T > \sigma_\delta \), we use a stopping time defined as

\[
\tau(T) := \min \left[ \min \left\{ t \in [\sigma_\delta, T) : |M_t| > \frac{2kU_t}{e^2} \right\} \right. 
\begin{align*}
&\left. \left[ |M_t| \leq \frac{2kU_t}{e^2} \wedge |M_t| > \sqrt{2kU_t \ln \left( \frac{\ln \left( \frac{2kU_t}{|M_t| + 2\sqrt{kU_t}|M_t|} \right)}{C_1 \delta} \right)} \right] , T \right) \right] \quad \text{(C.1)}
\]

We also require a moment bound for the \( \lambda \)-mixed submartingales, whose proof is in Section C.1.3.

Lemma 53. For any \( t \), \( \mathbb{E}^\lambda [Z_t^\lambda] \leq G_t := \begin{cases} 
15 \exp \left( \frac{M_t^2}{2kU_t} \right) & \text{if } |M_t| \leq \frac{2k}{e^2} U_t \\
\ln \left( \frac{2M_t}{|M_t| + 2\sqrt{kU_t}} \right) & \text{if } |M_t| > \frac{2k}{e^2} U_t
\end{cases} \)
Finally, we use a “one-sided Lipschitz” characterization of $M_t$—that $G_t$ will not grow too fast when $t \approx \tau(T)$:

**Lemma 54.** For any $T > \frac{e^4}{2} \ln 2$, $G_{\tau(T)} \leq \frac{14}{11} G_{\tau(T) - 1}$.

With these tools, the proof can be outlined.

**Proof Sketch of Theorem 19.** Here fix $T$ and write $\tau = \tau(T)$ as defined in (C.1). It suffices to prove that $P(\tau < T) \geq \delta$ under the given assumption on $\delta$. We have

$$1 \leq \frac{(a)}{\mathbb{E}[Z^\lambda_t]} = \frac{(b)}{\mathbb{E}[Z^\lambda_t]} = \frac{(c)}{\mathbb{E}[G_{\tau}]} \leq \frac{(d)}{\mathbb{E}[G_{\tau} | \tau < T]} P(\tau < T) + \mathbb{E}[G_{\tau} | \tau = T]$$

(C.2)

where (a) is by Optional Stopping (Theorem 51), (b) is by Tonelli’s Theorem, (c) is by Lemma 53, and (d) is by Lemma 54. The result is then proved by upper-bounding $\mathbb{E}[G_{\tau - 1} | \tau < T]$ and $\mathbb{E}[G_{\tau - 1} | \tau = T]$, using the upper bounds on $|M_{\tau - 1}|$ given by the definition of $\tau$.

C.1.2 Full Proof of Theorem 19

**Proof of Theorem 19.** Throughout this proof, fix $T$ and write $\tau = \tau(T)$ as defined in (C.1). Also let $C_1, C_2$ be the absolute constants defined in the theorem statement. It suffices to prove that $P(\tau < T) \geq \delta$ under the given assumption on $\delta$.

We do this by working with (C.2) from the proof sketch; this states that

$$\frac{11}{14} \leq \mathbb{E}[G_{\tau - 1} | \tau < T] P(\tau < T) + \mathbb{E}[G_{\tau - 1} | \tau = T]$$

(C.3)
By definition of $\tau$, we have

$$|M_{\tau-1}| \leq \frac{2k}{e^2} U_{\tau-1} \quad \text{and} \quad |M_{\tau-1}| \leq \sqrt{2kU_{\tau-1} \ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}$$

(C.4)

Therefore, substituting the definition in (C.4) into the definition of $G_{\tau-1}$,

$$G_{\tau-1} = \frac{15 \exp \left( \frac{M_{\tau-1}^2}{4kU_{\tau-1}} \right)}{\ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)} \leq \frac{15}{\sqrt{C_1 \delta \ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}}$$

(C.5)

So from (C.5),

$$\mathbb{E} \left[ G_{\tau-1} \mid \tau = T \right] \leq \mathbb{E} \left[ \frac{15}{\sqrt{C_1 \delta \ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}} \mid \tau = T \right] \leq \frac{15}{\sqrt{C_1 \delta}} = \frac{11}{28}$$

(C.6)

where the last inequality is by the assumption $\delta \geq \frac{4}{\ln(kU_{\tau-1})}$ and Lemma 55.
Also, from (C.5),

\[
\mathbb{E}[G_{\tau-1} \mid \tau < T] \leq \mathbb{E} \left[ \frac{15}{\sqrt{C_1 \delta \ln \left( \frac{\tau - 1}{M_{\tau-1}} \right)}} \mid \tau < T \right]
\]

\[
\leq \mathbb{E} \left[ \frac{15}{\sqrt{C_1 \delta \ln \left( \frac{2kU_{\tau-1}}{\left| M_{\tau-1} \right| + 2\sqrt{kU_{\tau-1}}} \right)}} \mid \tau < T \right]
\]

\[
= \mathbb{E} \left[ \frac{15}{\sqrt{C_1 \delta \ln \left( \left( \frac{1}{e^2} + \frac{1}{\sqrt{kU_{\tau-1}}} \right)^{-1} \right)}} \mid \tau < T \right] \leq \frac{15}{\sqrt{C_1 \delta}}
\]

where (a) uses \( |M_{\tau-1}| \leq \frac{2kU_{\tau-1}}{e^2} \) (by (C.4)) and (b) uses \( U_{\tau-1} = \tau - 1 \geq e^4 \ln 2 - 1 \).

Substituting this and (C.6) into (C.3) gives \( 1 \leq \left( \frac{1}{2\delta} \right) P(\tau < T) + \frac{1}{2} \). Therefore, \( P(\tau < T) \geq \delta \), finishing the proof.

The proof of Theorem 19 requires a supporting lemma, which is proved in Section C.1.4.

**Lemma 55.** Within the event \( \{ \tau(T) = T \} \), if \( \delta \geq \frac{4}{\ln(kU_{T-1})} \), then

\[
\frac{1}{\sqrt{\delta \ln \left( \frac{2kU_{T-1}}{\left| M_{T-1} \right| + 2\sqrt{kU_{T-1}}} \right)}} \leq 1
\]
C.1.3 Supporting Proofs

Proof of Lemma 53. First note that

\[
\mathbb{E}^\lambda \left[ Z^\lambda_t \right] = \int_{-1/e^2}^{0} e^{\lambda M_t - k\lambda^2 U_t} \frac{d\lambda}{-\lambda \left( \ln \frac{1}{\lambda} \right)^2} + \int_{0}^{1/e^2} e^{\lambda M_t - k\lambda^2 U_t} \frac{d\lambda}{-\lambda \left( \ln \frac{1}{\lambda} \right)^2} \\
= \exp \left( \frac{M_t^2}{4kU_t} \right) \left[ \int_{-1/e^2}^{0} e^{-kU_t \left( \frac{\lambda - M_t}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{-\lambda \left( \ln \frac{1}{\lambda} \right)^2} \\
+ \int_{0}^{1/e^2} e^{-kU_t \left( \frac{\lambda - M_t}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \right] \\
\leq 2 \exp \left( \frac{M_t^2}{4kU_t} \right) \int_{0}^{1/e^2} e^{-kU_t \left( \frac{\lambda - M_t}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \quad (C.7) 
\]

Suppose that \( \frac{|M_t|}{2kU_t} \leq \frac{1}{e^2} \). Define an integer \( N \) such that

\[
\frac{|M_t|}{2kU_t} + \sqrt{\frac{N-1}{kU_t}} \leq \frac{1}{e^2} \leq \frac{|M_t|}{2kU_t} + \sqrt{\frac{N}{kU_t}} \quad (C.8) 
\]

(Note that we are guaranteed \( N \geq 1 \) by the assumption \( \frac{|M_t|}{2kU_t} \leq \frac{1}{e^2} \).) Combining this with (C.7),

\[
\mathbb{E}^\lambda \left[ Z^\lambda_t \right] \leq 2 \exp \left( \frac{M_t^2}{4kU_t} \right) \left[ \int_{0}^{\frac{|M_t|}{2k\sqrt{U_t}}} e^{-kU_t \left( \frac{\lambda - |M_t|}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \\
+ \sum_{i=0}^{N-1} \int_{\frac{|M_t|}{2k\sqrt{U_t}} + \sqrt{\frac{i+1}{2kU_t}}}^{\frac{|M_t|}{2k\sqrt{U_t}} + \sqrt{\frac{i}{2kU_t}}} e^{-kU_t \left( \frac{\lambda - |M_t|}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \right] \quad (C.9) 
\]

Now we have

\[
\int_{0}^{\frac{|M_t|}{2k\sqrt{U_t}}} e^{-kU_t \left( \frac{\lambda - |M_t|}{2k\sqrt{U_t}} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \leq \int_{0}^{\frac{|M_t|}{2k\sqrt{U_t}}} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} = \left[ \frac{1}{\ln \frac{1}{\lambda}} \right]_{0}^{\frac{|M_t|}{2k\sqrt{U_t}}} = \frac{1}{\ln \frac{2kU_t}{|M_t|}} 
\]
Substituting this and Lemma 56 into (C.9), we get

\[ \mathbb{E}^\lambda [Z_\lambda^t] = 2 \exp \left( \frac{M_i^2}{4kU_t} \right) \left[ \frac{1}{\ln \left( \frac{2kU_t}{|M_i|} + 2\sqrt{kU_t} \right)} + \frac{1}{\ln \left( \frac{2kU_t}{|M_i| + 2\sqrt{kU_t}} \right)} \left( \frac{e}{e - 1} + \frac{3}{2\ln \left( \frac{e^2}{1 + e^2/\sqrt{k}} \right)} \right) \right] \]

\[ \leq \frac{\exp \left( \frac{M_i^2}{4kU_t} \right)}{\ln \left( \frac{2kU_t}{|M_i| + 2\sqrt{kU_t}} \right)} \left( 6 + \frac{3}{\ln \left( \frac{e^2}{1 + e^2/\sqrt{k}} \right)} \right) \leq \frac{15 \exp \left( \frac{M_i^2}{4kU_t} \right)}{\ln \left( \frac{2kU_t}{|M_i| + 2\sqrt{kU_t}} \right)} = G_t \]

which proves the result when \( |M_i| \leq \frac{2kU_t}{e^2} \).

Alternatively, if \( |M_i| > \frac{2kU_t}{e^2} \), from (C.7) we have

\[ \mathbb{E}^\lambda [Z_\lambda^t] \leq 2 \exp \left( \frac{M_i^2}{4kU_t} \right) \int_0^{1/e^2} e^{-kU_t \left( \lambda - \frac{|M_i|}{2kU_t} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \]

\[ \overset{(a)}{\leq} 2e^2 \exp \left( \frac{M_i^2}{4kU_t} \right) \left( \int_0^{1/e^2} e^{-kU_t \left( \lambda - \frac{|M_i|}{2kU_t} \right)^2} d\lambda \right) \left( \int_0^{1/e^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} \right) \]

\[ = e^2 \exp \left( \frac{M_i^2}{4kU_t} \right) \left( \int_0^{1/e^2} e^{-kU_t \left( \lambda - \frac{|M_i|}{2kU_t} \right)^2} d\lambda \right) \]

\[ \leq e^2 \exp \left( \frac{M_i^2}{4kU_t} \right) \left( \int_{-\infty}^{\frac{|M_i|}{2kU_t}} e^{-kU_t \left( \lambda - \frac{|M_i|}{2kU_t} \right)^2} d\lambda \right) \]

\[ = e^2 \exp \left( \frac{M_i^2}{4kU_t} \right) \left( \frac{\sqrt{\pi}}{2\sqrt{kU_t}} \right) \overset{(b)}{\leq} e^2 \exp \left( \frac{M_i^2}{4kU_t} \right) \sqrt{\frac{\pi}{2\ln(\sqrt{kU_t})}} \leq \frac{7 \exp \left( \frac{M_i^2}{4kU_t} \right)}{\ln \left( \frac{2kU_t}{\sqrt{kU_t}} \right)} \]

where \( (a) \) is by Chebyshev’s integral inequality (Lemma 58), and \( (b) \) is because

\[ \frac{\ln(\sqrt{kU_t})}{\sqrt{kU_t}} \leq 1. \]
Lemma 56. Define $N$ as in (C.8). Then

$$
\sum_{i=0}^{N-1} \int_{\frac{|M_i|}{2kU_t}}^{\frac{|M_i|}{2kU_t} + \sqrt{\frac{i+1}{kU_t}}} e^{-kU_t\left(\lambda - \frac{|M_i|}{2kU_t}\right)^2} \frac{d\lambda}{\lambda (\ln \frac{1}{\lambda})^2}
\leq \frac{1}{\ln \left(\frac{2kU_t}{|M_i| + 2\sqrt{kU_t}}\right)} \left( \frac{e}{e-1} + \frac{3}{2\ln \left(\frac{e^2}{1+e^2/\sqrt{k}}\right)} \right)
$$

Proof. For convenience, define $\mu := \frac{|M_i|}{2kU_t}$ and $\sigma := \frac{1}{\sqrt{kU_t}}$. In particular, this means that $\mu + \sigma \sqrt{N-1} \leq \frac{1}{e^2} \leq \mu + \sigma \sqrt{N}$. 
where (C.10) follows from Lemma 57.

\[ \sum_{i=0}^{N-1} \int_{\frac{\log(N)}{\sigma^2}}^{\frac{\log(N)}{\sigma^2}} e^{-kU_i(\lambda, \frac{\log(N)}{\sigma^2})} \frac{d\lambda}{\ln(\frac{1}{\lambda^2})} = \sum_{i=0}^{N-1} \int_{\mu + \sigma \sqrt{i}}^{\mu + \sigma \sqrt{i+1}} e^{-\frac{(\lambda - \mu)^2}{\sigma^2}} \frac{d\lambda}{\ln(\frac{1}{\lambda^2})} \]

\[ \leq \sum_{i=0}^{N-1} e^{-i} \left( \frac{1}{\ln(\frac{1}{\mu + \sigma \sqrt{i} + 1})} - \frac{1}{\ln(\frac{1}{\mu + \sigma \sqrt{i}})} \right) \]

\[ \leq \sum_{i=0}^{N-1} e^{-i} \left( \frac{1}{\ln(\frac{1}{\mu + \sigma \sqrt{i}})} + \frac{\ln(\frac{\mu + \sigma \sqrt{i + 1}}{\mu + \sigma})}{\ln(\frac{1}{\mu + \sigma \sqrt{i} + 1})} \right) \]

\[ \leq \frac{1}{\ln(\frac{1}{\mu + \sigma})} \sum_{i=0}^{N-1} e^{-i} \left( 1 + \frac{\ln(1 + \frac{\sigma \sqrt{i}}{\mu + \sigma})}{\ln(\frac{1}{\mu + \sigma \sqrt{i}})} \right) \]

\[ \leq \frac{1}{\ln(\frac{1}{\mu + \sigma})} \left( \sum_{i=0}^{\infty} e^{-i} + \sum_{i=0}^{\infty} e^{-i} \left( \frac{\ln(1 + \sqrt{i})}{\ln(1 + e^2)} \right) \right) \]

\[ \leq \frac{1}{\ln(\frac{1}{\mu + \sigma})} \left( \frac{e}{e - 1} + \frac{1}{\ln\left(\frac{e^2}{1 + e^2} / \sqrt{k}\right)} \sum_{i=0}^{\infty} e^{-i} \ln(1 + \sqrt{i}) \right) \]

\[ \leq \frac{1}{\ln(\frac{1}{\mu + \sigma})} \left( \frac{e}{e - 1} + \frac{3}{2 \ln\left(\frac{e^2}{1 + e^2} / \sqrt{k}\right)} \right) \]  

(C.10)

where (C.10) follows from Lemma 57. \[ \square \]

**Lemma 57.** \( \sum_{i=0}^{\infty} e^{-i} \ln(1 + \sqrt{i}) \leq \frac{3}{2} \)

**Proof.** Take \( f(x) = e^{-x/2} \ln(1 + \sqrt{x}) \) for \( x \geq 0 \). Note that the derivative

\[ f'(x) = \frac{1}{2} e^{-x/2} \left( \frac{1}{\sqrt{x}(1 + \sqrt{x})} - \ln(1 + \sqrt{x}) \right) \]
Now, \( \frac{1}{\sqrt{x(1+\sqrt{x})}} \) is monotone decreasing and \( \ln(1 + \sqrt{x}) \) is monotone increasing, so \( f'(x) \) has exactly one root, corresponding to the maximum of \( f(x) \). This can be numerically confirmed to occur at \( x^* \approx 0.745 \), and \( f(x^*) \leq 0.5 \).

So

\[
\sum_{i=0}^{\infty} e^{-i} \ln(1 + \sqrt{i}) \leq \sum_{i=0}^{\infty} e^{-i/2} \left( e^{-i/2} \ln(1 + \sqrt{i}) \right) \leq \frac{1}{2} \sum_{i=0}^{\infty} e^{-i/2} = \frac{\sqrt{e}}{2(\sqrt{e} - 1)} \leq \frac{3}{2}
\]

\( \Box \)

### C.1.4 Ancillary Results and Proofs

**Proof of Lemma 54.** Note that by definition of \( \tau \),

\[
|M_{\tau-1}| \leq \frac{2k}{\varepsilon^2} U_{\tau-1} \quad \text{and} \quad |M_{\tau-1}| \leq \sqrt{2k U_{\tau-1} \ln \left( \frac{\ln \left( \frac{2k U_{\tau-1}}{|M_{\tau-1}|+2\sqrt{k U_{\tau-1}}} \right)}{C_1 \delta} \right)} \tag{C.11}
\]

Firstly, we have

\[
\exp \left( \frac{M_{\tau-1}^2}{4k U_{\tau-1}} \right) \leq \exp \left( \frac{M_{\tau-1}^2}{4k U_{\tau-1}} \right) = \exp \left( \frac{2M_{\tau-1} \xi_{\tau} + \xi_{\tau}^2}{4k U_{\tau-1}} \right) \overset{(a)}{=} \exp \left( \frac{1}{e^2} + \frac{1}{144} \right) \leq \frac{7}{6}
\]

\( \tag{C.12} \)

where \((a)\) is because \( M_{\tau-1} \xi_{\tau} \leq |M_{\tau-1}| \leq \frac{2k}{e^2} U_{\tau-1} \) by (C.11), and because \( \frac{\xi_{\tau}^2}{4k U_{\tau-1}} = \frac{1}{4k(108)} = \frac{1}{144} \).

We write \( \tau(T) \) more concisely as \( \tau \) here, and note that the minimum value of \( T \)
implies that $\tau(T) \geq 108$, a fact we will use throughout the proof. Also, we have

$$\frac{\ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}{\ln \left( \frac{2kU_{\tau}}{|M_{\tau}| + 2\sqrt{kU_{\tau}}} \right)} \leq \frac{\ln \left( \frac{2kU_{\tau}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau}}} \right)}{\ln \left( \frac{2kU_{\tau}}{1 + |M_{\tau-1}| + 2\sqrt{kU_{\tau}}} \right)} \leq \frac{\ln \left( \frac{2kU_{\tau}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau}}} \right)}{\ln \left( \frac{2kU_{\tau}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau}}} \right) - \ln(13/12)} \quad (C.13)$$

where (a) is because $f(x) = \ln \left( \frac{x}{C + \sqrt{2x}} \right)$ is monotone increasing for any $C \geq 0$; and (b) is because $|M_{\tau-1}| + 2\sqrt{kU_{\tau}} \geq 2\sqrt{kU_{\tau}} \geq 12$, so $1 + |M_{\tau-1}| + 2\sqrt{kU_{\tau}} \leq \frac{13}{12} (|M_{\tau-1}| + 2\sqrt{kU_{\tau}})$.

Now we have

$$\frac{2kU_{\tau}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau}}} \geq \frac{2kU_{\tau}}{e^2 U_{\tau-1} + 2\sqrt{kU_{\tau}}} \geq \frac{kU_{\tau}}{\max \left( \frac{2k}{e^2} U_{\tau-1}, 2\sqrt{kU_{\tau}} \right)} \geq \min \left( \frac{e^2}{2k}, \frac{1}{2} \sqrt{kU_{\tau}} \right) \geq 3 \quad (c)$$

where (b) uses (C.11) and (c) uses $U_{\tau} = \tau \geq 108$. This means that from (C.13),

$$\frac{\ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}{\ln \left( \frac{2kU_{\tau}}{|M_{\tau}| + 2\sqrt{kU_{\tau}}} \right)} \leq \frac{\ln(3)}{\ln(3) - \ln(13/12)} \leq 12 \quad (C.14)$$

Note that by (C.11), $|M_{\tau-1}| \leq \frac{2k}{e^2} U_{\tau-1}$, so $G_{\tau-1} = \frac{15 \exp \left( \frac{M_{\tau-1}^2}{2kU_{\tau-1}} \right)}{\ln \left( \frac{2kU_{\tau-1}}{|M_{\tau-1}| + 2\sqrt{kU_{\tau-1}}} \right)}$. 
Suppose $|M_τ| ≤ \frac{2k}{e}U_τ$. Then using Lemma 53, (C.12), and (C.14),

\[
\frac{G_τ}{G_{τ-1}} = \frac{\exp\left(\frac{M_τ^2}{4kU_τ}\right)}{\exp\left(\frac{M_{τ-1}^2}{4kU_{τ-1}}\right)} \frac{\ln\left(\frac{2kU_{τ-1}}{|M_{τ-1}|+2\sqrt{kU_{τ-1}}}\right)}{\ln\left(\frac{2kU_τ}{|M_τ|+2\sqrt{kU_τ}}\right)} \leq \left(\frac{7}{6}\right) \frac{12}{11} = \frac{14}{11}
\]

Alternatively, if $|M_τ| > \frac{2k}{e}U_τ$, we can use Lemma 53 and (C.12) to conclude that

\[
\frac{G_τ}{G_{τ-1}} = \frac{7\exp\left(\frac{M_τ^2}{4kU_τ}\right)}{15\exp\left(\frac{M_{τ-1}^2}{4kU_{τ-1}}\right)} \frac{\ln\left(\frac{2kU_{τ-1}}{|M_{τ-1}|+2\sqrt{kU_{τ-1}}}\right)}{\ln\left(\sqrt{kU_τ}\right)} \leq \frac{7}{6}(1) \leq \frac{14}{11}
\]

\[\square\]

**Proof of Lemma 55.** The definition of $τ(= T)$ and the fact that $|M_{τ-1}| ≥ 0$ imply that

\[
|M_{T-1}| = |M_{τ-1}| \leq \sqrt{2kU_{τ-1} \ln\left(\frac{\ln\left(\frac{2kU_{τ-1}}{|M_{τ-1}|+2\sqrt{kU_{τ-1}}}\right)}{C_1 δ}\right)}
\]

\[
\leq \sqrt{2kU_{T-1} \ln\left(\frac{\ln(2kU_{T-1})}{2C_1 δ}\right)}
\]
Consequently,

\[
\ln \left( \frac{2kU_{T-1}}{|M_{T-1}| + 2\sqrt{kU_{T-1}}} \right) \geq \ln \left( \frac{2kU_{T-1}}{\sqrt{2kU_{T-1} \ln \left( \frac{\ln (kU_{T-1})}{2C_1\delta} \right) + 2\sqrt{kU_{T-1}}}} \right)
\]

\[
= \ln \left( \frac{\sqrt{2kU_{T-1}}}{\sqrt{\ln \left( \frac{\ln (kU_{T-1})}{2C_1\delta} \right) + \sqrt{2}}} \right) \geq \ln \left( \frac{\sqrt{2kU_{T-1}}}{\sqrt{\ln^2 (kU_{T-1}) + \sqrt{2}}} \right)
\]

\[
\geq \ln \left( \frac{\sqrt{kU_{T-1}}}{(kU_{T-1})^{1/16} + 1} \right) \geq \frac{1}{4} \ln (kU_{T-1})
\]

(C.15)

where (a) uses that \( \delta \geq \frac{4}{\ln (kU_{T-1})} \) and \( 8C_1 \geq 1 \), (b) uses that \( \ln (\ln (kU_{T-1})) \leq (kU_{T-1})^{1/8} \), and (c) uses that \( (kU_{T-1})^{1/16} + 1 \leq (kU_{T-1})^{1/4} \) for \( kU_{T-1} = k(\tau - 1) \geq e^4 \ln (2/\delta) - k \geq 26 \). Therefore,

\[
\frac{1}{\sqrt{\delta \ln \left( \frac{2kU_{T-1}}{|M_{T-1}| + 2\sqrt{kU_{T-1}}} \right)} \leq \frac{1}{\sqrt{\frac{\delta}{4} \ln (kU_{T-1})}} \leq 1
\]

after substituting (C.15) and again using \( \delta \geq \frac{4}{\ln (kU_{T-1})} \).

Chebyshev’s Integral Inequality is a standard result, but we give a short proof for completeness.

**Lemma 58 (Chebyshev’s Integral Inequality).** If \( f(x) \) and \( g(x) \) are respectively monotonically increasing and decreasing functions over an interval \((a, b]\), and \( \int_a^b f(x)dx \) and \( \int_a^b g(x)dx \) are both defined and finite, then

\[
\int_a^b f(x)g(x)dx \leq \frac{1}{b-a} \left( \int_a^b f(x)dx \right) \left( \int_a^b g(x)dx \right)
\]

**Proof.** By the monotonicity properties of the functions, we know for any \( x, y \in (a, b] \) that
(f(x) - f(y))(g(x) - g(y)) ≤ 0. Therefore,

\[
0 \geq \int_a^b \int_a^b (f(x) - f(y))(g(x) - g(y)) \, dx \, dy
= 2(b - a) \int_a^b f(x)g(x) \, dx - 2 \left( \int_a^b f(x) \, dx \right) \left( \int_a^b g(x) \, dx \right)
\]

which yields the result upon simplification. \qed

## C.2 Miscellaneous Results: Concentration Bounds and Extensions

### C.2.1 Proof of Lemma 28

As outlined in Section 12.3.2, we choose \( \lambda \) stochastically from a probability space \((\Omega_\lambda, \mathcal{F}_\lambda, P_\lambda)\) such that \( P_\lambda(d\lambda) = \frac{d\lambda}{|\lambda|(\ln \frac{1}{|\lambda|})^2} \) on \( \lambda \in [-e^{-2}, e^{-2}] \setminus \{0\} \).

Take an arbitrary time \( t \geq \tau_0 \). For outcomes within \( A_\delta \), we have the following:

\[
\mathbb{E}^\lambda \left[ X_\lambda^t \right] = \int_{-1/e^2}^{0} X_{\lambda}^\lambda \frac{d\lambda}{-\lambda \left( \ln \frac{1}{|\lambda|} \right)^2} + \int_{0}^{1/e^2} X_{\lambda}^\lambda \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2}
\]

\[
= \int_{-e^{-2}}^{0} e^{\lambda M_t - \frac{1}{2} \lambda^2 U_t} \frac{d\lambda}{-\lambda \left( \ln \frac{1}{|\lambda|} \right)^2} + \int_{0}^{e^{-2}} e^{\lambda M_t - \frac{1}{2} \lambda^2 U_t} \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2}
\]

\[
\geq \int_{0}^{e^{-2}} e^{\lambda |M_t| - \frac{1}{2} \lambda^2 U_t} \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2}
\]

\[
= \exp \left( \frac{M_t^2}{2 U_t} \int_{0}^{e^{-2}} e^{-\frac{1}{2} U_t \left( \lambda - \frac{|M_t|}{U_t} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2} \right)
\]

\[
\geq \exp \left( \frac{M_t^2}{2 U_t} \int_{\frac{|M_t|}{U_t}}^{\frac{|M_t|}{U_t}(1 + \sqrt{\kappa})} e^{-\frac{1}{2} U_t \left( \lambda - \frac{|M_t|}{U_t} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2} \right) \quad \text{(C.16)}
\]

\[
\geq \exp \left( \frac{M_t^2}{2 U_t} \int_{\frac{|M_t|}{U_t}}^{\frac{|M_t|}{U_t}(1 - \sqrt{\kappa})} e^{-\frac{1}{2} U_t \left( \lambda - \frac{|M_t|}{U_t} \right)^2} \frac{d\lambda}{\lambda \left( \ln \frac{1}{|\lambda|} \right)^2} \right) \quad \text{(C.17)}
\]
Simplifying (C.17) further,

\[ \mathbb{E}^\lambda \left[ X^\lambda_t \right] \geq \exp \left( \frac{M_t^2}{2U_t} \right) \exp \left( -\frac{1}{2} k U_t \left( \frac{M_t}{U_t} \right)^2 \right) \int \frac{\exp \left( \frac{M_t}{U_t} (1+\sqrt{k}) \right)}{\lambda \left( \ln \frac{1}{\lambda} \right)^2} d\lambda \]

\[ = \exp \left( \frac{M_t^2}{2U_t} (1-k) \right) \frac{\ln \left( \frac{1+\sqrt{k}}{1-\sqrt{k}} \right)}{\ln \left( \frac{U_t}{|M_t|(1+\sqrt{k})} \right)} \ln \left( \frac{U_t}{|M_t|(1-\sqrt{k})} \right) \]

\[ \geq 2 \exp \left( \frac{M_t^2}{2U_t} (1-k) \right) \frac{1}{\ln \left( \frac{U_t}{|M_t|(1+\sqrt{k})} \right)} \ln \left( \frac{U_t}{|M_t|(1-\sqrt{k})} \right) \]

\[ \geq \frac{2 \exp \left( \frac{M_t^2}{2U_t} (1-k) \right)}{\ln^2 \left( \frac{U_t}{(1-\sqrt{k}) |M_t|} \right)} \quad (C.18) \]

Take \( \tau \) to be any stopping time as in the lemma statement. Then from (C.18),

\[ \mathbb{E}_{A^\delta} \left[ \mathbb{E}^\lambda \left[ X^\lambda_\tau \right] \right] \geq \mathbb{E}_{A^\delta} \left[ \frac{2 \exp \left( \frac{M_t^2}{2U_t} (1-k) \right)}{\ln^2 \left( \frac{U_t}{(1-\sqrt{k}) |M_t|} \right)} \right] \]

finishing the proof.

### C.2.2 Hoeffding and Bernstein Concentration Bounds

In this section, we show that the results of Section 12.2 can be proved through simple extensions of the proof of Theorem 18.

That proof is the subject of Section 12.3. It applies to the Rademacher random walk, but uses the i.i.d. Rademacher assumption only through an exponential supermartingale construction (Lemma 24). Theorem 18 can be generalized significantly beyond the Rademacher random walk by simply replacing the construction with other similar exponential constructions, leaving the remainder of the proof essentially intact as presented in Section 12.3.
To be specific, the rest of that proof works unchanged if the construction has the following properties:

1. The construction should be of the same form as Lemma 24:

\[ X^\lambda_t = \exp \left( \lambda M_t - \frac{\lambda^2}{2} U_t \right) \]

for some nondecreasing process \( U_t \). (The proof of Theorem 18 sets \( U_t = t \).)

2. \( X^\lambda_t \) should be a supermartingale for \( \lambda \in \left( -\frac{1}{e^2}, \frac{1}{e^2} \right) \setminus \{0\} \).

The second condition – the supermartingale construction – is true whenever the martingale differences are conditionally sub-exponential; given the past, they have exponentially decaying tails of probability mass (see [BLM13a] for details).

Now we give two standard exponential supermartingale constructions with these properties. We first give a construction leading directly to Theorem 21, when it is used to replace Lemma 24 in the proof of Theorem 18.

**Lemma 59.** Suppose the difference sequence is uniformly bounded, i.e. \( |\xi_t| \leq e^2 \) a.s. for all \( t \). Then the process \( X^\lambda_t := \exp \left( \lambda M_t - \lambda^2 (e - 2) V_t \right) \) is a supermartingale for any \( \lambda \in \left[ -\frac{1}{e^2}, \frac{1}{e^2} \right] \).

**Proof of Lemma 59.** It can be checked that \( e^x \leq 1 + x + (e - 2)x^2 \) for \( x \leq 1 \). Then for any \( \lambda \in \left[ -\frac{1}{e^2}, \frac{1}{e^2} \right] \) and \( t \geq 1 \),

\[
\mathbb{E} \left[ \exp (\lambda \xi_t) \mid \mathcal{F}_{t-1} \right] \leq 1 + \lambda \mathbb{E} [\xi_t \mid \mathcal{F}_{t-1}] + \lambda^2 (e - 2) \mathbb{E} [\xi_t^2 \mid \mathcal{F}_{t-1}]
\]

\[
= 1 + \lambda^2 (e - 2) \mathbb{E} [\xi_t^2 \mid \mathcal{F}_{t-1}] \leq \exp \left( \lambda^2 (e - 2) \mathbb{E} [\xi_t^2 \mid \mathcal{F}_{t-1}] \right)
\]

using the martingale property on \( \mathbb{E} [\xi_t \mid \mathcal{F}_{t-1}] \).
Therefore, \( \mathbb{E} \left[ \exp \left( \lambda \xi_t - \lambda^2 (e - 2) \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \right) \mid \mathcal{F}_{t-1} \right] \leq 1 \), so

\[
\mathbb{E} \left[ X_t^\lambda \mid \mathcal{F}_{t-1} \right] \leq X_{t-1}^\lambda
\]

\[\square\]

The second construction leads similarly to Theorem 20, when combined with the theorem’s assumption of a uniformly bounded difference sequence; it has better constants than Lemma 59.

**Lemma 60.** The process \( X_t^\lambda := \exp \left( \lambda M_t - \lambda^2 \frac{2}{3} \right) \left( 2V_t + Q_t \right) \) is a supermartingale for any \( \lambda \in \mathbb{R} \).

**Proof of Lemma 60.** Consider the following inequality: for all real \( x \),

\[
\exp \left( x - \frac{1}{6} x^2 \right) \leq 1 + x + \frac{1}{3} x^2 \tag{C.19}
\]

Suppose (C.19) holds. Then for any \( \lambda \in \mathbb{R} \) and \( t \geq 1 \),

\[
\mathbb{E} \left[ \exp \left( \lambda \xi_t - \lambda^2 \frac{2}{3} \xi_t^2 \right) \mid \mathcal{F}_{t-1} \right] \leq 1 + \lambda \mathbb{E} \left[ \xi_t \mid \mathcal{F}_{t-1} \right] + \frac{\lambda^2}{3} \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] = 1 + \lambda \mathbb{E} \left[ \xi_t \mid \mathcal{F}_{t-1} \right] \leq \exp \left( \frac{\lambda^2}{3} \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \right),
\]

by using the martingale property on \( \mathbb{E} \left[ \xi_t \mid \mathcal{F}_{t-1} \right] \). Therefore, we have that

\[
\mathbb{E} \left[ \exp \left( \lambda \xi_t - \lambda^2 \frac{2}{3} \xi_t^2 - \frac{\lambda^2}{3} \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \right) \mid \mathcal{F}_{t-1} \right] \leq 1
\]

so \( \mathbb{E} \left[ X_t^\lambda \mid \mathcal{F}_{t-1} \right] \leq X_{t-1}^\lambda \) and the result is shown.

It only remains to prove (C.19), which is equivalent to showing that the function

\[
f(x) = \exp \left( x - \frac{1}{6} x^2 \right) - 1 - x - \frac{1}{3} x^2 \leq 0.
\]

This is done by examining derivatives. Note that

\[
f'(x) = \left( 1 - \frac{1}{3} \right) \exp \left( x - \frac{1}{6} x^2 \right) - 1 - \frac{2}{3} x, \quad \text{and} \quad f''(x) = \left( -\frac{1}{3} + \left( 1 - \frac{1}{3} \right)^2 \right) \exp \left( x - \frac{1}{6} x^2 \right) - \frac{2}{3} = \frac{2}{3} (e^y (1 - y) - 1) \text{ where } y := x - \frac{1}{6} x^2.
\]

Here \( e^y \leq \frac{1}{1 - y} \) for \( y < 1 \), and \( e^y (1 - y) \leq 0 \) for \( y \geq 1 \), so \( f''(x) \leq 0 \) for all \( x \). Since \( f'(0) = f(0) = 0 \), the function \( f \) attains a maximum.
of zero over its domain, proving (C.19) and the result.

In order for $X_t^\lambda$ to satisfy the conditions at the beginning of this section, the differences $\xi_t$ need not be uniformly bounded, but rather can simply satisfy conditions on their higher moments. For completeness, here is an example of much weaker sufficient conditions; this is a direct adaptation of the method conventionally used to prove a general version of Bernstein’s inequality ([BLM13a], Thm. 2.10).

Lemma 61. Suppose that for all $k \geq 3$ and all $i$,

$$
E \left[ \xi_i^k \mid \mathcal{F}_{t-1} \right] \leq \frac{1}{2} k! (e/\sqrt{2})^{2(k-2)} E \left[ \xi_i^2 \mid \mathcal{F}_{t-1} \right]
$$

Then for any $\lambda \in \left( -\frac{1}{e^2}, \frac{1}{e^2} \right)$, the process

$$
X_t^\lambda := \exp \left( \lambda M_t - \lambda^2 V_t \right)
$$

is a supermartingale.

Proof of Lemma 61. Taking the Taylor expansion of the exponential function,

$$
E \left[ \exp(\lambda \xi_t) \mid \mathcal{F}_{t-1} \right] = 1 + \lambda E \left[ \xi_t \mid \mathcal{F}_{t-1} \right] + E \left[ \sum_{k=2}^{\infty} \frac{\lambda^k}{k!} \xi_t^k \mid \mathcal{F}_{t-1} \right]
$$

$$
= 1 + \frac{\lambda^2}{2} E \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] + \sum_{k=3}^{\infty} \frac{\lambda^k}{k!} E \left[ \xi_t^k \mid \mathcal{F}_{t-1} \right]
$$

$$
\leq 1 + \frac{\lambda^2}{2} E \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] + \frac{1}{2} \sum_{k=3}^{\infty} \lambda^k (e/\sqrt{2})^{2(k-2)}
$$

$$
\leq 1 + \frac{\lambda^2}{2} E \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \sum_{k=2}^{\infty} (|\lambda|/2 e^2)^{k-2}
$$

using the martingale property on $E \left[ \xi_t \mid \mathcal{F}_{t-1} \right]$, monotone convergence, and the lemma
assumption. For all \(|\lambda| < \frac{2}{e^2}\), the infinite geometric series is summable, giving

\[
\mathbb{E} \left[ \exp (\lambda \xi_t) \mid \mathcal{F}_{t-1} \right] \leq 1 + \frac{\lambda^2}{2(1 - |\lambda|/2)e^2} \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right]
\]

So for all \(|\lambda| < \frac{1}{e^2}\),

\[
\mathbb{E} \left[ \exp (\lambda \xi_t) \mid \mathcal{F}_{t-1} \right] \leq 1 + \frac{\lambda^2}{2(1 - (1/2))} \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \leq \exp \left( \lambda^2 \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \right)
\]

Therefore, \(\mathbb{E} \left[ X_t^\lambda \mid \mathcal{F}_{t-1} \right] \leq X_{t-1}^\lambda \mathbb{E} \left[ \exp \left( \lambda \xi_t - \lambda^2 \mathbb{E} \left[ \xi_t^2 \mid \mathcal{F}_{t-1} \right] \right) \mid \mathcal{F}_{t-1} \right] \leq X_{t-1}^\lambda. \)

**C.2.3 Initial Time Conditions**

As discussed in Section 12.2.2 (in Remark 3), this section outlines how to remove the initial time condition for our martingale concentration bounds, using the proof technique of this chapter. To demonstrate, we extend Theorem 22 here to hold over all times, using the placeholder variance process \(U_t\) and other notation as in Section 12.3 (e.g. \(\lambda_0\)). The Hoeffding- and Bernstein-style results of Section 12.2 can be extended in exactly the same way, by redefining \(U_t\).

It suffices to show a uniform concentration bound for \(|M_t|\) over \(t < \tau_2(\delta) := \min \left\{ s : U_s \geq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right) \right\}\).

**Theorem 62.** Fix any \(\delta > 0\). With probability \(\geq 1 - \delta\), for all \(t < \tau_2(\delta)\) simultaneously,

\[
|M_t| \leq \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right)
\]

**Proof.** Write \(\tau_2(\delta)\) as \(\tau_2\) for convenience. Define the stopping time

\[
\tau = \min \left\{ t \leq \tau_2 : |M_t| > \frac{2}{\lambda_0} \ln \left( \frac{2}{\delta} \right) \right\}
\]
Then it suffices to prove that $P(\tau < \tau_2) \leq \delta$.

On the event $\{\tau < \tau_2\}$, we have $|M_\tau| > \frac{2}{\lambda_0} \ln\left(\frac{\tau}{\delta}\right)$ by definition of $\tau$. Therefore, using Lemma 25,

$$2 \geq \mathbb{E}\left[\exp\left(\lambda_0 |M_\tau| - \frac{\lambda_0^2}{2} U_\tau\right)\right] \geq \mathbb{E}\left[\exp\left(\lambda_0 |M_\tau| - \frac{\lambda_0^2}{2} U_\tau\right) | \tau < \tau_2\right] P(\tau < \tau_2)$$

$$\overset{(a)}{\geq} \mathbb{E}\left[\exp\left(\lambda_0 \left(\frac{2}{\lambda_0} \ln\left(\frac{\tau}{\delta}\right)\right) - \frac{\lambda_0^2}{2} \left(\frac{2}{\lambda_0^2} \ln\left(\frac{\tau}{\delta}\right)\right)\right) | \tau < \tau_2\right] P(\tau < \tau_2)$$

$$= \mathbb{E}\left[\exp\left(\ln\left(\frac{2}{\delta}\right)\right)\right] P(\tau < \tau_2) = \frac{2}{\delta} \mathbb{P}(\tau < \tau_2)$$

where $(a)$ uses that $|M_\tau| > \frac{2}{\lambda_0} \ln\left(\frac{\tau}{\delta}\right)$ when $\tau < \tau_2$, and that $U_\tau \leq \frac{2}{\lambda_0^2} \ln\left(\frac{\tau}{\delta}\right)$ since $\tau < \tau_2$. Therefore, $P(\tau < \tau_2) \leq \delta$, as desired.

Taking a union bound of Theorem 62 with Theorem 22 gives that w.h.p., for all $t$,

$$|M_t| \leq \mathcal{O}\left(\sqrt{U_t \left(\ln \ln U_t + \ln \left(\frac{1}{\delta}\right)\right)} + \ln \left(\frac{1}{\delta}\right)\right)$$

This matches the rate of Bennett/Bernstein inequalities which hold for a fixed time ([BLM13a]), except for an extra $\sqrt{\ln \ln U_t}$ factor on the Gaussian-regime term that accounts for the uniformity over time.
Appendix D

Deferred Material from Chapter 14

D.1 Proof of Theorem 40

Proof. Write $K$ as a placeholder absolute constant in the sense of Sec. 14.2. Then for any sufficiently high $n$, our definitions for $q_n$ and $p_n$ tell us that

$$
P_{H_1} (\tau \geq n) = P_{H_1} (\forall t \leq n : S_n \leq q_n) \leq P_{H_1} (S_n \leq q_n)$$

$$= P_{H_1} (S_n - n\delta \leq q_n - n\delta)$$

$$\leq P_{H_1} (S_n - n\delta \leq -Kn\delta)$$

$$\leq \beta$$  (D.1)

for $n \geq n^*$, from (14.5) and the definition of $n^*$. Also, using a Hoeffding bound on (D.1), we see that $P_{H_1} (\tau \geq n) \leq e^{-Kn\delta^2}$. So for any $\delta$ and $\beta$,

$$\mathbb{E}_{H_1} [\tau] = \sum_{n=1}^{\infty} P_{H_1} (\tau \geq n) \leq n^* + \sum_{n=n^*}^{\infty} P_{H_1} (\tau \geq n)$$

$$\leq n^* + \sum_{n=n^*}^{\infty} e^{-Kn\delta^2} \leq n^* + \frac{\beta^K}{1 - e^{-K\delta^2}}$$  (D.3)

$$\leq n^* + \beta C \left( \frac{1}{K\delta^2} + 1 \right) \leq \left( 1 + \frac{K_1 \beta K_2}{\ln \frac{1}{\beta}} \right) n^*$$  (D.4)
Here (D.3) sums the infinite geometric series with initial term \((e^{-n\delta^2})^K \leq \beta^K\), and (D.4) uses the inequality \(\frac{1}{1-e^{-x}} \leq \frac{1}{x} + 1\) as well as \(n^*_\beta(\delta) \geq \frac{K\ln\frac{1}{\delta^2}}{\delta^2}\). \(\square\)

### D.2 Proof of Proposition 23

**Proof of Proposition 23.** Since \(x, x', y, y'\) are all independent,

\[
\mathbb{E}[h] = (\mathbb{E}[x] - \mathbb{E}[y])^\top (\mathbb{E}[x'] - \mathbb{E}[y']) = \delta^\top \delta
\]

Next,

\[
\mathbb{E}[h^2] = \mathbb{E}[(x - y)^\top (x' - y')^2] = \mathbb{E}[(x - y)^\top (x' - y')(x' - y')^\top (x - y)] = \mathbb{E}[\text{tr}((x - y)(x - y)^\top (x' - y')(x' - y')^\top))] = \text{tr}(\mathbb{E}[(x - y)(x - y)^\top]) \mathbb{E}[(x' - y')(x' - y')^\top])
\]

Since \(\mathbb{E}[(x - y)(x - y)^\top] = \Sigma_1 + \Sigma_2 + \delta \delta^\top = 2\Sigma + \delta \delta^\top\), we have

\[
\text{var}(h) = \mathbb{E}[h^2] - (\mathbb{E}h)^2 = \text{tr}((2\Sigma + \delta \delta^\top)^2) - \|\delta\|^4 = 4\text{tr}(\Sigma^2) + 4\delta^\top \Sigma \delta
\]

from which the result is immediate. \(\square\)

### D.3 Proof of Theorem 41

We rely upon a variance-dependent form of the LIL. Upon noting that \(\mathbb{E}[T_n - T_{n-1}] = 0\) and \(\mathbb{E}[(T_n - T_{n-1})^2] = V_0\), it is an instance of a general martingale
concentration inequality proved in Ch. 12.

**Theorem 63 (Uniform Bernstein Bound (Instantiation of Thm. 21)).** Suppose that $|T_n - T_{n-1}| \leq 1$ w.p. 1 for all $n \geq 1$. Fix any $\xi < 1$ and define

$$
\tau_0(\xi) = \min \left\{ s : sV_0 \geq \frac{\left(1 + \sqrt{1/3}\right)^2}{e-2} \ln \left(\frac{4}{\xi}\right) \right\}
$$

Then with probability $\geq 1 - \xi$, for all $n \geq \tau_0$ simultaneously, $|T_n| \leq \frac{2(e-2)}{(1+\sqrt{1/3})}nV_0$ and

$$
|T_n| \leq \sqrt{6(e-2)nV_0 \left(2 \ln \left(\frac{3(e-2)e^2nV_0}{|T_n|}\right) + \ln \left(\frac{2}{\xi}\right)\right)}
$$

In principle this tight control by the second moment is enough to achieve our goals, just as the second-moment Bernstein inequality for random variables suffices for proving empirical Bernstein inequalities.

However, the version we use for our empirical Bernstein bound is a more convenient though looser restatement of Theorem 63. To derive it, we reproduce the following result from the appendices of Ch. 12 concerning initial time conditions:

**Proposition 64 (Theorem 62).** Take any $\xi > 0$, and define $T_n$ and $\tau_0(\xi)$ as in Theorem 63. With probability $\geq 1 - \xi$, for all $n < \tau_0(\xi)$ simultaneously,

$$
|T_n| \leq 2 \left(1 + \sqrt{1/3}\right) \ln \left(\frac{4}{\xi}\right)
$$

Theorem 41 follows by loosely combining the above two uniform bounds.

**Proof of Theorem 41.** Recall $V_n := nV_0$. Theorem 63 gives that w.p. $1 - \frac{\xi}{2}$, for all
\[ n \geq \tau_0(\xi/2), \quad |T_n| \leq \frac{2(e-2)}{1+\sqrt{1/3}} V_n \text{ and} \]

\[ |T_n| \leq \max \left( 3(e-2)e^2, \sqrt{2C_1V_n \ln \ln V_n + C_1V_n \ln \left( \frac{4}{\xi} \right)} \right) \quad (D.5) \]

Taking a union bound of (D.5) with Prop. 64 gives that w.p. \( \geq 1 - \xi \), the following is true for all \( n \) simultaneously:

\[
|T_n| \leq \begin{cases} 
2 \left( 1 + \sqrt{\frac{1}{3}} \right) \ln \left( \frac{8}{\xi} \right), & \text{if } n < \tau_0(\xi/2) \\
\frac{2(e-2)}{1+\sqrt{1/3}} V_n \text{ and } \max \left( 3(e-2)e^2, \sqrt{2C_1V_n \ln \ln V_n + C_1V_n \ln \left( \frac{4}{\xi} \right)} \right) & \text{if } n \geq \tau_0(\xi/2)
\end{cases}
\]

For all \( n \) we have \( |T_n| \) bounded by the maximum of the two cases above. The result can be seen to follow, by relaxing the explicit bound \( |T_n| \leq \frac{2(e-2)}{1+\sqrt{1/3}} V_n \) to instead transform \( \ln \ln \) into \([\ln \ln]_+\).

\[ \square \]

### D.4 Proportionality Constants and Guaranteed Correctness

After observing the first few samples, regardless of how many, it is impossible to empirically conclude with certainty that the type I error of a sequential test (Fig. 14.2) has ever leveled off. And although our theory can guarantee type I error control, it is reasonable to question whether our empirically recommended prescription \( C = \sqrt{2} \) is actually sound, even in the hypothetical case \( n \to \infty \).

In fact, we can show that it is unsound. Consider first the biased coin example of
Sec. 14.2. If $S_n$ is the test statistic, the number of type I error violators under the null is

$$\sup_{n \geq 1} \frac{S_n}{\sqrt{n \ln \ln n}} \geq \inf \sup_{k \geq 1, n \geq k} \frac{S_n}{\sqrt{n \ln \ln n}}$$

$$= \lim \sup_{k \to \infty, n \geq k} \frac{S_n}{\sqrt{n \ln \ln n}}$$

$$= \lim \sup_{n \to \infty} \frac{S_n}{\sqrt{n \ln \ln n}} = \sqrt{2}$$

w.p. 1, from the asymptotic LIL of Thm. 39.

So the sequential test will almost surely reject with $C = 2$, which is very undesirable. We still recommend this for two reasons.

Firstly, it appears not to be an empirical issue, because of $C_0$ and because of our finite $N$ needed in practice to detect the alternative. As evidence of this, we count type I violations under the fair-coin null (the maximally anti-concentrated stochastic process under the random walk) for a very large $N = 10^6$ with $C = 3, \alpha = 0.05, C_0 = \log \frac{1}{\alpha}$, repeatedly using $10^5$ Monte Carlo trials. We see an average of 3-5 type-I-violating sample paths (out of $10^5$) – almost no type I error, because $C$ is relatively high.

Secondly, it is possible to set $C > 2$ and get provable type I error control, at the cost of a somewhat higher stopping time. In the theory, $C$ and $C_0$ can be tightened for sufficiently high sample sizes, as seen in Ch. 12 – the reason is that for sufficiently high $n$, the order of growth of the bound is dominated by $O(\sqrt{n \ln \ln n})$, and all the sources of looseness in the analysis leading to our final uniform empirical Bernstein inequality (Thm. 43) can therefore be bounded by increasing the proportionality constant on the iterated-logarithm term.

These ideas generalize cleanly beyond the biased-coin example to our other tests. Exactly the same argument as above can be used with our two-sample mean test statistic $T_n$, its variance process $V_n$, and the variance-based asymptotic LIL ([Sto70]) to give, w.p.
\[
\sup_{n \geq 1} \frac{T_n}{\sqrt{n \ln \ln n}} \geq \sqrt{2}
\]

So even after taking into account the convergence rate of the empirical variance \( \hat{V}_n \to V_n \), our basic conclusions and recommendations remain the same for all our tests beyond the biased-coin setting.

### D.5 Type II Error Approximation

We argue that the power of the sequential test with maximum runtime \( N \) is approximately lower-bound by the power of the batch test with \( N \) examples (e.g. in (14.3) for the coin example). This argument can be made more exact.

For the coin example, we can work with a more refined approximation via the CLT, when \( N \) is high so that \( N\delta > q_N > p_N \). Defining \( p^x_N = \frac{p_N - N\delta}{\sqrt{N}} \) and \( q^x_N = \frac{q_N - N\delta}{\sqrt{N}} \),

\[
P_{H_1} (\exists n \leq N : S_n > q_N) \geq P_{H_1} (S_N > q_N) \\
= P_{H_1} (S_N > p_N) - P_{H_1} (q_N > S_N \geq p_N) \\
= P_{H_1} (S_N > p_N) - P_{H_1} \left( \frac{q^x_N > S_N - N\delta}{\sqrt{N}} \geq p^x_N \right) \\
\approx P_{H_1} (S_N > p_N) - (\Phi(q^x_N) - \Phi(p^x_N))
\]

When there is an abundance of data, the sequential test would typically be run with very large \( N \), since it would typically stop much sooner (see Appendix D.8). So this CLT approximation is in fact extremely good, and it can be made into a lower bound if necessary with a negligible \( N^{-O(1)} \) deviation term. A similar argument can be made for our more complex tests.
D.6 Proof of Lemma 42

Proof. Here, $v_i := h_i^2 - \mathbb{E}[h_i^2]$ has mean zero by definition. It has a cumulative variance process that is self-bounding:

$$B_n := \sum_{i=1}^{n} \mathbb{E}[v_i] = \sum_{i=1}^{n} \mathbb{E}\left[(h_i^2 - \mathbb{E}[h_i^2])^2\right]$$

$$= \sum_{i=1}^{n} \left(\mathbb{E}[h_i^4] - (\mathbb{E}[h_i^2])^2\right) \leq \sum_{i=1}^{n} \mathbb{E}[h_i^4]$$

$$\leq \sum_{i=1}^{n} \mathbb{E}[h_i^2] = nV_0 := A_n \quad (D.6)$$

where the last inequality (a) uses that $|h_i| \leq 1$, and we define the process $A_n$ for convenience.

Applying Theorem 41 to the mean-zero random walk $\sum_{i=1}^{n} v_i$ gives $(1 - \xi)$-a.s. for all $t$ that:

$$\left|\hat{V}_n - A_n\right| = \left|\sum_{i=1}^{n} (h_i^2 - \mathbb{E}[h_i^2])\right|$$

$$< C_0(\xi) + \sqrt{2C_1B_n[\ln \ln]_+ (B_n) + C_1B_n \ln \left(\frac{4}{\xi}\right)}$$

$$\leq C_0(\xi) + \sqrt{2C_1A_n[\ln \ln]_+ (A_n) + C_1A_n \ln \left(\frac{4}{\xi}\right)}$$

This can be relaxed to

$$A_n - \sqrt{2C_1A_n[\ln \ln]_+ (A_n) + C_1A_n \ln \left(\frac{4}{\xi}\right)} - C_0(\xi) - \hat{V}_n \leq 0 \quad (D.7)$$
Suppose $A_n \geq 108 \ln \left( \frac{4}{\xi} \right)$. Then a straightforward case analysis confirms that

$$A_n \geq 8 \max \left( 2C_1 \ln \ln \left( A_n \right), C_1 \ln \left( \frac{4}{\xi} \right) \right)$$

This is precisely the condition needed to invert (D.7) using Lemma 65. Doing this yields that

$$\sqrt{A_n} \leq \sqrt{2C_1 \ln \ln \left( 2C_0(\xi) + 2\hat{V}_n \right) + C_1 \ln \left( \frac{4}{\xi} \right)}$$

$$+ \sqrt{C_0(\xi) + \hat{V}_n}$$

(D.8)

For sufficiently high $\hat{V}_n$ ($O\left( \ln \left( \frac{4}{\xi} \right) \right)$ suffices), the first term on the right-hand side of (D.8) is bounded as

$$\sqrt{2C_1 \ln \ln \left( 2C_0(\xi) + 2\hat{V}_n \right) + C_1 \ln \left( \frac{4}{\xi} \right)} \leq \sqrt{4C_1 \ln \ln \left( 2C_0(\xi) + 2\hat{V}_n \right)}$$

$$\leq \sqrt{8C_1 \left( C_0(\xi) + \hat{V}_n \right)}$$

Resubstituting into (D.8) and squaring both sides yields the result. It remains to check the case $A_n \leq 108 \ln \left( \frac{4}{\xi} \right)$. But this bound clearly holds in the statement of the result, so the proof is finished.

The following lemma is useful to invert inequalities involving the iterated logarithm.

**Lemma 65.** Suppose $b_1, b_2, c$ are positive constants, $x \geq 8 \max \left( b_1 \ln \ln \left( x \right), b_2 \right)$, and

$$x - \sqrt{b_1 x \ln \ln \left( x \right) + b_2 x - c} \leq 0$$

(D.9)
Then

\[ \sqrt{x} \leq \sqrt{b_1 \ln \ln_+ (2c) + b_2 + \sqrt{c}} \]

**Proof.** Suppose \( x \geq 8 \max(b_1 \ln \ln_+ (x), b_2) \). Since \( x \geq 8b_2 \), we have

\[ 0 \leq \frac{x}{8} - b_2 \leq \frac{x}{4} - b_1 \left( \frac{x}{8b_1} \right) - b_2 \quad \implies \]

\[ 0 \leq \frac{x^2}{4} - b_1 x \left( \frac{x}{8b_1} \right) - b_2 x \]

Substituting the assumption \( \frac{x}{8b_1} \geq [\ln \ln_+ (x)] \) gives

\[ 0 \leq \frac{x^2}{4} - b_1 x [\ln \ln_+ (x)] - b_2 x \quad \implies \]

\[ \sqrt{b_1 x [\ln \ln_+ (x)] + b_2 x} \leq \frac{1}{2} x \]

Substituting this into (D.9) gives \( x \leq 2c \). Therefore, again using (D.9),

\[ 0 \geq x - \sqrt{b_1 x [\ln \ln_+ (x)] + b_2 x} - c \]

\[ \geq x - \sqrt{b_1 x [\ln \ln_+ (2c)] + b_2 x} - c \]

This is now a quadratic in \( \sqrt{x} \). Solving it (using \( \sqrt{x} \geq 0 \)) gives

\[ \sqrt{x} \leq \frac{1}{2} \left( \sqrt{b_1 [\ln \ln_+] (2c) + b_2 + \sqrt{b_1 [\ln \ln_+] (2c) + b_2 + 4c}} \right) \]

\[ \leq \sqrt{b_1 [\ln \ln_+] (2c) + b_2 + \sqrt{c}} \]

using the subadditivity of \( \sqrt{c} \).

\[ \square \]
D.7 Proof of Theorem 44

In this proof we use \( <, \leq, >, \geq, \approx \) to denote \( <, \leq, >, \geq, = \) when ignoring constants. Let us first bound \( P(\tau > n) \) for sufficiently large \( n \) such that \( \hat{V}_n \approx V_n \) and that \( \sqrt{2\log(1/\alpha) \ln n} \ll n\|\delta\|^2 \):

\[
P_{H_1}(\tau > n) = 1 - P_{H_1}(\tau \leq n) = 1 - P_{H_1}(\exists n \leq n : T_n > q_n) \\
\leq 1 - P_{H_1}(T_n > q_n)
\]

We have \( q_n \approx p_n \sqrt{\ln \ln V_n} \leq p_n \sqrt{\ln \ln n} \); recalling \( p_n \approx \sqrt{2V_n \log(1/\alpha)} \) from Eq.(14.11), we get

\[
P_{H_1}(\tau > n) \leq 1 - P_{H_1} \left( T_n > \sqrt{2V_n \log(1/\alpha) \ln \ln n} \right) \\
= P_{H_1} \left( \frac{T_n - n\|\delta\|^2}{\sqrt{V_n}} > \sqrt{2\log(1/\alpha) \ln \ln n} - n\|\delta\|^2 \right)
\]

The above expression then corresponds to a tail inequality for the centered standardized random variable on the LHS, which is a sum of bounded random variables, and hence standard sub-Gaussian inequalities yield

\[
P_{H_1}(\tau > n) \leq \exp \left( -n^2\|\delta\|^4 / V_n \right) \\
= \exp \left( -n \frac{\|\delta\|^4}{8Tr(\Sigma^2) + 8\delta^T\Sigma\delta} \right) \quad (D.10)
\]

Recalling from Eq.(14.13) that

\[
n^*_\beta \approx \frac{8Tr(\Sigma^2) + 8\delta^T\Sigma\delta}{\|\delta\|^4} (z_\beta + z_\alpha)^2,
\]

we infer from this and (D.10) that \( P_{H_1}(\tau > n^*_\beta) \) is a small constant bounded away from 1.
Since \( \tau = \sum_n \mathbb{1}(\tau > n) \), we have by summing a geometric series that

\[
\mathbb{E}_{H_1} [\tau] = \sum_{n \geq 1} P_{H_1}(\tau > n)
\]

\[
\leq n^*_\beta + \sum_{n \geq n^*_\beta} P_{H_1}(\tau > n)
\]

\[
\leq n^*_\beta + \frac{P_{H_1}(\tau > n^*_\beta)}{1 - \exp\left(-\frac{\|\delta\|^4}{8Tr(\Sigma^2) + 8\delta^T \Sigma \delta}\right)}
\]

Using the inequality \( 1 - \exp(-x) \leq x \), i.e. \( \exp(-x) \geq 1 - x \), and substituting for \( n^*_\beta \), we get

\[
\mathbb{E}_{H_1} [\tau] \leq n^*_\beta + \frac{8Tr(\Sigma^2) + 8\delta^T \Sigma \delta}{\|\delta\|^4} P_{H_1}(\tau > n^*_\beta)
\]

\[
\approx n^*_\beta + \frac{n^*_\beta}{(z_\beta + z_\alpha)^2} P_{H_1}(\tau > n^*_\beta)
\]

\[
= (1 + O(1)) n^*_\beta
\]

D.8 Experimental Protocol

This section contains some notes on the experiments.

The graphs of Fig. 14.2 are each generated by 10,000 Monte Carlo trials. The remaining graphs all use \( \alpha = 0.05 \).

The graphs of Fig. 14.3 are each generated by 1,000 Monte Carlo trials on the data, and the solid lines are the resulting stopping time distribution of the sequential test. As for the dashed lines, the true power of the batch test is also estimated by 1,000 Monte Carlo trials. Note that these experiments are run with \( N = 50000 \), even though the tests always seem to stop much sooner because the \( \delta \) are sufficiently high. When data are abundant enough to detect any discernible difference between the samples, we suggest
setting $N$ very high, as this gives better power.

The graphs of Fig. 14.4 are each generated by 1,000 Monte Carlo trials.

Evaluating the dependence on dimensionality $d$ is outside our scope in this paper.

The high-dimensional properties of our statistic are further evaluated and discussed in [RRP+15], which shows that it is possible to achieve better high-dimensional power with fewer samples than our test statistic. But our standout contribution is sequential, and we focus on these aspects.

### D.9 Supplemental Graphs

#### Type I Error

In Figure 14.2, we see that the cumulative type I error rate is increasing, not leveling off. To change this, the proportionality constant on the iterated logarithm $C$ must be increased. The result of $C = 2.2$ is plotted to the right of Figure D.1, with the $C = 2$ random walk of Figure 14.2 at left on the same scale for comparison. We see that just a slight increase in $C$ lowers type I violations significantly; at every $\alpha$, the type I error is less than half of the desired tolerance. Extrapolating the linear graphs, we predict that type I error will be controlled up to the huge sample size $\sim e^{25} \approx 7.2 \times 10^{10}$ for every $\alpha$, and further increases in $C$ make it infeasible to run for long enough to break type I error control. This gives some empirical validation for our recommendations.

We can also look at the stopping time under the null with our simulated Gaussians. These show better empirical concentration than the coin, unsurprisingly; so we do not graph them here.

#### Type II Error

For completeness, we give the equivalents to Figs. 14.3 and 14.4 here, as Figs. D.2 and D.3 respectively. Note that $\tau$ is $O(\frac{1}{\delta^2})$ in Fig. D.3, as the theory for the coin
**Figure D.1.** $\Pr_{H_0}(\tau \leq n)$ for different $\alpha$, on biased coin, for $C = 2$ (left) and $C = 2.2$ (right).

**Figure D.2.** Power vs. $\ln(N)$ for different $\delta$, on Gaussians. Dashed lines represent power of batch test with $N$ samples.

predicts.
Figure D.3. Distribution of $\ln(\tau)$ for $\delta \in \{e^{-1} : c \in \{1, 2, \ldots, 5\}\}$, so that the abscissa values $\{\ln(\frac{1}{\delta})\}$ are a unit length apart. Dashed line has slope 2.
**Bibliography**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors,</td>
</tr>
<tr>
<td>[BB08]</td>
<td>Olivier Bousquet and Léon Bottou.</td>
<td>The tradeoffs of large scale learning.</td>
</tr>
</tbody>
</table>


[Kor09] Yehuda Koren. The bellkor solution to the netflix grand prize. 2009.


