Classical Improvements to Modern Machine Learning

Shiva Kaul

CMU-CS-24-137

August 2024

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Geoffrey Gordon (Chair)
Zachary Lipton
Aditi Raghunathan
Ryan Tibshirani (U.C. Berkeley)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the Office of Naval Research under the award numbers N000141512365 and N000140911052; the Defense Advanced Research Projects Agency under award numbers FA8702-15-D-0002 and FA8721-05-C-0003; and the National Science Foundation under award numbers CHE1027985, CNS0833882 and CNS-0614679. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: machine learning, conformal prediction, healthcare, linear dynamical systems, orthogonal polynomials, fairness

Abstract

Over the past decade, a large rift has grown between classical and modern machine learning. The predictive performance of modern learning is incomparably better, but it is easier to analyze and guarantee safety, efficiency, fairness, and other properties of classical learning. In this dissertation, I investigate when it is possible to restore such desiderata to modern machine learning by prudently and strategically incorporating classical techniques. I form syntheses between classical and modern learning which can be categorized according to two high-level strategies: (1) wrapping, in which reliable performance guarantees are extracted from modern, opaque models via classical analytic techniques, or (2) swapping, in which some components of a modern model are rebuilt from classical primitives in a way which improves overall efficiency, tractability, and/or expressivity. These efforts lead to new developments in multiple areas of machine learning.

The most important contribution in this dissertation pertains to meta-analysis, a structured form of question-answering which serves as the foundation of evidence-based medicine. Classic meta-analytic techniques are based upon randomized, controlled trials, whose causal validity is trusted; by contrast, modern regression models are trained upon large observational databases whose causal validity is untrusted. I show it is possible by incorporate the untrusted data into meta-analysis without sacrificing validity. This involves basic improvements to full conformal prediction which are of general interest. In a separate, more focused, application to healthcare, I generalize classic, handcrafted heart-rate variability statistics so they can be fine-tuned, via supervised learning, as part of a deep neural network. This leads to more accurate, physiologically-informed models.

I also present foundational computational primitives that can be used within future machine learning models and algorithms. The first is an algorithm to (approximately) run nonlinear RNNs for T steps in just $O(\log T)$ parallel time. A key innovation of this algorithm is replacing nonlinearity across time by nonlinearity along depth through a provably-consistent scheme of local, parallelizable corrections. In this manner, classical linear dynamical systems (also known as state-space models) can be stacked to form fast, nonlinear sequence models. Another new computational primitive is gradient-based optimization over the set of all sequences of orthogonal polynomials. This optimization formulation has connections to many different problems in signal processing and optimization. Finally, I propose fairness criteria that circumvent computational intractability, based upon the geometric notion of margin used throughout learning theory and optimization.

Acknowledgments

My deepest thanks go to my advisor, Geoff Gordon. From day one, he let me explore my interests with a high degree of independence and latitude. Whether these explorations led to electronic structure theory, polynomial optimization, computational learning theory, or the myriad of topics that actually appear in this dissertation, he was both supportive and very capable of keeping me grounded and strongly objective. When I returned to him with the final topic of meta-analysis — and this time, I was sure — he didn't skip a beat. This dissertation reflects just some of his boundless patience and breadth.

I owe Mahadev Satyaranarayan (Satya) for lifting off my academic career. His team — Adam Goode, Jan Harkes and Benjamin Gilbert — were great companions and very supportive during my time with them. Dengyong Zhou and Steve Hanneke are two mentors who inspired me to work on challenging problems. I enjoyed and appreciated my work and interactions with Ryan Tibshirani, Mor Harchol-Balter, Klaus Sutner, and Carlos Guestrin.

My graduate school experience was great thanks to my roommates Jayant Krishnamurthy, Kevin Waugh, Patrick Foley, Remy Marechal, Ligia Nistor, and Sven Stork; my group members Erik Zawadzki, Byron Boots, Ahmed Hefny, and Carlton Downey; my colleagues Pranjal Awasthi and Liu Yang; my summer buddies Jason Franklin, Jonah Sherman, and Nishant Mehta; and my friends Alandra Greenlee, Wolfgang Richter, Matt Stanton, Brendan Meeder, Kristina Sojakova, Abe Othman, Aaditya Ramdas, John Dickerson, Jamie Morgenstern, Ashiqur Khudabukhsh, and John Wright.

Finally, I thank my family (Dad, Mom, and Dhruva) to whom I owe the most.

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 This Dissertation . . . 4
- 1.3 Related Research Directions . . . 6
- 1.4 Organization and Interpretation . . . 7

2 Meta-Analysis with Untrusted Data **9**
- 2.1 Introduction . . . 12
  - 2.1.1 Our Contributions . . . 13
- 2.2 Preliminaries . . . 14
  - 2.2.1 Related Work . . . 15
- 2.3 Predicting Trials with Idiocentric Linear Smoothers . . . 20
  - 2.3.1 Idiocentricity and its Consequences . . . 22
- 2.4 Predicting Effects . . . 23
  - 2.4.1 Understanding Conformal Effect Prediction . . . 25
- 2.5 Simulations . . . 26
- 2.6 Case Study: Amiodarone . . . 27
- 2.7 Appendix . . . 30
  - 2.7.1 Background for Meta-Analysis . . . 30
  - 2.7.2 Computations for KRR . . . 38
  - 2.7.3 Proof of Theorem 3 . . . 40
  - 2.7.4 Proof of Lemma 2 . . . 42
  - 2.7.5 Proof of Lemma 3 . . . 42
  - 2.7.6 Proof of Lemma 4 . . . 43
  - 2.7.7 Predicting Effects with Robust Optimization . . . 44
  - 2.7.8 Simulation Details and Full Results . . . 46
  - 2.7.9 Case Study Details . . . 47
- 2.8 Discussion . . . 54

3 Differentiating Through Orthogonal Polynomial Transforms **55**
- 3.1 Introduction . . . 58
- 3.2 Preliminaries . . . 59
- 3.3 Vector-Jacobian Product Algorithms . . . 62
- 3.4 Learned Polynomial Transforms . . . 64

3.4.1 Learned JPEG . . . 65
3.5 Minimal Values of General Optimization Problems . . . 66
3.5.1 Background . . . 66
3.5.2 Proposed Approach . . . 67
3.5.3 Basic Empirical Evaluation . . . 71
3.6 Related Work . . . 72
3.7 Appendix . . . 75
3.7.1 Numerical Stability of Interpolate . . . 75
3.7.2 Vector-Jacobian Products . . . 77
3.8 Discussion . . . 84

4 Linear Dynamical Systems for Sequence Modeling 87
4.1 Introduction . . . 90
4.2 Linear Dynamical Systems . . . 91
4.2.1 SIMO Canonical Form . . . 92
4.2.2 Diagonalization . . . 93
4.2.3 MIMO Luenberger Form . . . 94
4.3 SIMO LDS in $O(n \log T)$ Parallel Time and n Parameters . . . 94
4.4 Approximating MIMO LDS by SIMO LDS . . . 97
4.4.1 Improper Learning: Random Projection . . . 97
4.4.2 Proper Learning: Perturbed Luenberger Form . . . 98
4.5 Approximating Nonlinear RNNs by Stacked LDS . . . 100
4.6 Experiments . . . 103
4.7 Appendix . . . 105
4.7.1 Proof of Lemma 7 . . . 105
4.7.2 Proof of Lemma 8 . . . 106
4.7.3 Proof of Proposition 8 . . . 107
4.7.4 Proof of Proposition 9 . . . 109
4.7.5 Approximation of Nonlinear Systems by Time-Varying LDS . . . 109
4.7.6 Additional Experiment Details . . . 113
4.8 Discussion . . . 113

5 Interpretable Deep Learning in Healthcare 115
5.1 Introduction . . . 118
5.1.1 Novel Contribution . . . 120
5.1.2 Outline . . . 122
5.2 Study Design . . . 122
5.2.1 Cohort Selection . . . 122
5.2.2 Data Collection . . . 123
5.3 Basic Data Preprocessing and Analysis . . . 126
5.3.1 Noise . . . 127
5.3.2 Heart Rate Metrics and ΔAmylase . . . 127
5.4 Machine Learning with a Structured Model . . . 129
5.4.1 Problem Formulation . . . 129

5.4.2 Key Intuitions . . . 130
5.4.3 Pretrained Parasympathetic Layer . . . 131
5.4.4 Sympathetic Layer . . . 133
5.4.5 Implementation Details . . . 133
5.4.6 Related Work . . . 134
5.5 Machine Learning Results . . . 135
5.6 Discussion . . . 137

6 Towards Computationally-Tractable Multi-Group Fairness 139
6.1 (Social) Mobility . . . 143
6.2 Contrast . . . 146
6.3 The Average Vector . . . 148
6.3.1 Theoretical Support . . . 149
6.4 Experimental Validation . . . 150
6.5 Remarks . . . 151
6.6 Discussion . . . 152

7 Conclusion and Future Work 155
7.1 Review and Subsequent Developments . . . 155
7.2 Thesis Assessment . . . 160
7.3 Future Work . . . 161
7.3.1 Meta-Analysis . . . 161
7.3.2 Other Ideas . . . 165

Bibliography 169

Chapter 1

Introduction

"The past is never dead. It's not even past." - Faulkner [1951]

1.1 Background

Over the past decade, a considerable rift has developed between classical and modern machine learning. Traditional machine learning involves minimizing a convex loss function on the training data, whose variables represent a shallow model and are relatively few in number, with the overall intent of solving a specific, focused prediction task. Over a series of empirical breakthroughs, modern machine learning reversed each of these traditional design decisions. First, the use of nonconvex optimization to train neural networks [Hinton, 2007, LeCun, 2007, Vincent et al., 2008]; then, increasing the depth of these networks to efficiently expand their representational power [Bengio et al., 2009, Krizhevsky et al., 2012]; the use of more parameters than training data ("overparameterization") to further enhance expressiveness while (interestingly) avoiding overfitting [He et al., 2016a, Simonyan and Zisserman, 2014, Zhang et al., 2017]; the development of self-attention and transformers to enable parallel utilization of burgeoning hardware resources, especially GPUs [Vaswani et al., 2017]; finally, the development of foundation models, most notably large language models, to simultaneously address an extremely broad array of prediction tasks [Dai and Le, 2015, Devlin et al., 2018, Radford et al., 2018].

As a consequence of these breakthroughs, modern machine learning has not only obtained state-of-the-art accuracy results far beyond classical machine learning: it has challenged broadly-

held presumptions about the relationship between artificial and human intelligence [Bubeck et al., 2023], and established itself as a crucial engine of global economic growth. But as machine learning's role in society and the economy have expanded, so too have the apprehensions surrounding it, as well as the responsibilities imposed upon it. The expansion of these concerns has led, in turn, to a bifurcation between classical and modern machine learning theory. In classical learning theory, the core goal is to prove that a learning algorithm efficiently, reliably obtains a high-accuracy model under a broad set of circumstances [Valiant, 1984]. For example, noise-tolerant learning algorithms work even when the data are randomly corrupted or even adversarially manipulated [Blum et al., 2003b, Kearns, 1998]. This traditional goal remains an important, challenging research topic, especially in the context of generative adversarial networks, reinforcement learning, and other departures from "plain" supervised learning. But in modern practice, it is straightforward to try a learning algorithm and see if it works — which it very often does. Newfound concerns relate to the potential downstream consequences of using a model with purportedly high test accuracy.

- efficiency: is it possible to make large, overparameterized models less resource intensive during training and/or inference?
- confidence: does the model merely report point predictions, or does it reliably convey its uncertainty with (e.g.) prediction sets that, with high probability, include the truth?
- (analytic) tractability: can the model's behavior be mathematically discerned and rigorously guaranteed?
- interpretability: can we understand why the model makes its decisions? Or have engineering tradeoffs, used to obtain high accuracy and efficiency, obscured its inner workings?
- fairness: do the model's predictions inadvertently harm one or more groups?

These concerns are studied under the umbrella of AI safety or trustworthiness. A generally-held intuition is that classical machine learning is safer than modern learning. It is important to recognize that this intuition is sometimes false: depending on the meaning of "safe", modern learning can be demonstrably safer, for rather straightforward reasons. For example, it can be computationally challenging to train small, classical models in the presence of noise [Blum et al.,

Problem Domain	Classic Approach	Modern Approach	Novel Synthesis	Benefits
Meta-analysis	Averaging (trusted) data only	(Untrusted) LLMs and databases	Conformal meta-analysis	Rigorous, tight intervals
Function approximation	Static polynomial transforms	Gradient-based optimization	Learned polynomial transforms	Expressiveness
Sequence modeling	Linear systems	Transformers and RNNs	Stacks of linear systems	Speed and tractability
Electrocardiology	Static HRV statistics	Deep learning	Learned HRV statistics	Useful in more settings
Fairness	Averages	Combinatorial definitions	Margin-based definitions	Tractability

Figure 1.1: A summary of the contributions of this thesis, emphasizing how a synthesis between classical and modern approaches can yield benefits over either approach taken in isolation.

2003a, Feldman et al., 2006, Kalai et al., 2008]. Depending on the type of noise, these difficulties can vanish when a larger (perhaps overparameterized) model is trained instead [Li et al., 2020, Montasser et al., 2019, Shalev-Shwartz et al., 2011]. Furthermore, modern machine learning can be safened by novel techniques which do not have roots in classical learning. For example, sharpness-aware minimization (SAM) is an alternative to gradient descent which improves generalization and noise-resilience for deep neural networks [Foret et al., 2020]. The study of loss landscapes and sharpness is rather peculiar to modern nonconvex learning, and the rationale for SAM differs for linear and nonlinear models [Baek et al., 2024].

Nevertheless, despite strong research efforts, classical machine learning often retains a strong advantage in some kinds of safety. This is especially common for safety definitions which are challenging to empirically verify, and are instead guaranteed through mathematical proof. For example, the worst-case running time of a learning algorithm or the worst-case coverage guarantee of a prediction set are mathematically, not empirically, guaranteed. It is easier to prove such guarantees for classical machine learning because it adheres more closely to mathematically tractable foundations such as linearity, convexity, and normality.

1.2 This Dissertation

This dissertation forms *syntheses* between classical and modern machine learning techniques, with the goal of retaining the safety, trustworthiness, and theoretical clarity of classical learning, while also obtaining the accuracy, flexibility, and practical benefits of modern learning. Different syntheses are developed in multiple different contexts, where the notions of "classical learning", "modern learning", and "trustworthiness" all have different concrete meanings. Nonetheless, the high-level recipe is similar among most of these efforts:

1. Start with a mathematically tractable model class from classical machine learning, such as linear dynamical systems, sequences of orthogonal polynomials, or Gaussian processes.
2. Imbue it with some empirically successful aspect of modern machine learning, such as depth, stochastic gradient-based optimization, or the involvement of a pretrained foundation model.
3. Demonstrate that the modern enhancement solves some limitation of the classical model (usually representational power) without destroying the original benefits.
4. Elaborate on how this synthesis can lead to practical benefits and novel applications.

These are the specific contexts in which this research strategy is applied:

Chapter 2 (Meta-Analysis with Untrusted Data): Large databases of informally-collected observational data, and foundation models trained upon them, could greatly enhance our understanding of causal interventions, particularly in healthcare. However, evidence-based medicine has been averse to involving such data, due to possible confounding; instead, it restricts its analysis to a relatively small amount of "trusted" data, typically randomized controlled trials. This chapter presents a new algorithm for meta-analysis — predicting the causal effects of interventions from study-level, rather than individual-level, data — which safely uses such observational data: it delivers predictions that are always valid, and are tight when the observational and trusted data align. The new algorithm adapts full conformal prediction to latent observations corrupted by heteroscedastic noise. Aside from their pivotal role in evidence-based medicine, systematic review and meta-analysis can be viewed as a reliable, unbiased, manual form of question answering — as language models could ideally perform. This chapter shows that, by wrapping

such foundation models in conformal prediction, they can be used to reliably answer important, quantitative, causal questions.

Chapter 3 (Differentiating Through Orthogonal Polynomial Transforms): This chapter enables sequences of orthogonal polynomials, a classical mathematical tool, to be efficiently used as a layer within deep neural networks. More specifically, this chapter develops efficient vector-Jacobian products for orthogonal polynomial evaluation and interpolation, thereby enabling efficient gradient-based optimization over the set of orthogonal polynomials.

Chapter 4 (Linear Dynamical Systems for Sequence Modeling): The original motivation behind transformers is that they enabled parallel computation along sequence length, whereas RNNs suffered from a sequential bottleneck. This chapter proves that RNNs can also be parallelized in a similar fashion, by rebuilding them from linear dynamical systems (LDSs), which are also referred to as state space models (SSMs). While SSMs alone cannot express nonlinearities across time, these can be approximated (to any accuracy) by using depth, stacking multiple SSMs with interposed nonlinearities. Thus, SSMs imbued with nonlinearity along depth can be computationally competitive with transformers. Furthermore, unlike transformers, such SSM constructions are amenable to advanced control-theoretic analysis. In recent years, there have been dramatic advances in SSM sequence modeling, with new architectures achieving performance competitive with Transformers [Gu and Dao, 2023, Gu et al., 2021a, 2022, Smith et al., 2022]; this chapter contributes a technique — rigorously replacing nonlinearity along the sequence axis by nonlinearity along a depth axis — that can be employed in future architectures.

Chapter 5 (Interpretable Deep Learning in Healthcare): This is the most applied chapter of the dissertation, attempting to predict the sympathetic nervous system's response to a bout of exercise, based on easily-measured heartrate data. Traditional statistics used to quantify autonomic nervous system activity are based on heart-rate variability (HRV) metrics. To modernize such statistics, this chapter parameterizes and generalizes them within a deep neural network. This results in novel statistics which capture different aspects of nervous system activity, while retaining some interpretability of traditional HRV metrics. This chapter illustrates a domain-specific approach to achieving interpretability, which can be challenging to define and achieve in general [Lipton, 2018].

Chapter 6 (Towards Computationally-Tractable Multi-Group Fairness): This is the most abstract chapter of the dissertation, exploring new fairness definitions rather than solving concrete learning problems. Circa 2016, fairness enjoyed a renaissance in machine learning research [Chouldechova, 2017, Dwork et al., 2012, Hardt et al., 2016c]. Many new fairness definitions were proposed, and most were binary or discrete, which led to computational challenges or even intractability. Furthermore, most definitions did not adequately address multi-group fairness. Inspired by the real-valued (i.e. "scale-sensitive" or "margin-based") formulations in learning theory and optimization, this chapter proposes and explores real-valued, multi-group definitions of fairness that can be satisfied by a very simple algorithm. As a more abstract research effort, this chapter uses lessons from classical learning theory to address modern difficulties in fair learning. Connections between these areas have been explored in subsequent lines of work [Dwork et al., 2021, Gopalan et al., 2023].

1.3 Related Research Directions

This dissertation represents a middle ground between two more extreme research agendas. The first agenda is to develop a fresh theoretical understanding of modern machine learning, without retaining the baggage of traditional machine learning. This is a flourishing and important research direction, and has delivered novel, satisfying explanations of the success of nonconvex optimization, deep learning and overparameterized models [Allen-Zhu et al., 2019b, Belkin et al., 2018, Bubeck and Sellke, 2021, Simon et al., 2024]. Work is currently underway on understanding the power of transformers and the emergent abilities of large language models. Another research approach is to advance traditional learning in its own right, charting a new course without any particular deference to modern learning. For example, mixed-integer programming can be used to learn optimal decision trees [Bertsimas and Dunn, 2017]. Small, interpretable, yet performant linear models for healthcare can be trained using integer programming [Angelino et al., 2018].

While all these research agendas are valuable in their own right, we offer some insight for why the middle ground can be appealing. Over long periods of time, it is not always the case that

research proceeds linearly, with old ideas being completely deprecated and forgotten in favor of new ones. Instead, research often proceeds cyclically, with tradeoffs between old and new ideas being continually reexamined as underlying factors change (e.g. hardware advances and application changes). For example, this dynamic is frequently observed in computer systems research, which involves long-running contentions between RISC vs. CISC microarchitecture, microkernels vs. monolithic kernels, microservices vs. monolithic services, and other such design decisions.

1.4 Organization and Interpretation

The chapters of this dissertation can be read and understood as separate contributions to different areas of machine learning. All technical results, including mathematical notation and proofs, remain self-contained within each chapter. Despite their technical independence, the chapters are not independent: they are sequential iterations of the following overarching research strategy, which constitutes the thesis statement of this dissertation.

> **Thesis**: It is often possible to restore aspects of safety, efficiency, and tractability to modern machine learning by prudently incorporating classical techniques.

Thus, besides making specific, technical contributions, this dissertation has a higher-level, less formal goal: to shed light on *when* it is likely for this research strategy to succeed. To help answer this high-level question, the thesis is organized as follows. At the end of each chapter, there is a Discussion section which rephrases the research as a classical-modern synthesis. It emphasizes the benefit that was obtained by pursuing a synthesis rather than either approach in isolation. Content relevant to the dissertation's final contributions will be marked as follows:

> *These interjections will highlight noteworthy developments in the dissertation.*

The concluding chapter of the dissertation critically, retrospectively evaluates the success of each chapter, by examining how their respective areas of machine learning subsequently evolved. The efforts of each chapter are taxonomized, and factors which possibly led to their success (or failure) are analyzed. This analysis informs recommendations for future work.

Chapter 2

Meta-Analysis with Untrusted Data

Abstract

Meta-analysis is a crucial tool for answering scientific questions. It is usually conducted on a relatively small amount of "trusted" data — ideally from randomized, controlled trials — which allow causal effects to be reliably estimated with minimal assumptions. This chapter shows how to answer causal questions much more precisely by making two changes. First, it incorporates untrusted data drawn from large observational databases, related scientific literature and practical experience — without sacrificing rigor or introducing strong assumptions. Second, it trains richer models capable of handling heterogeneous trials, addressing a long-standing challenge in meta-analysis. This chapter's approach is based on conformal prediction, which fundamentally produces rigorous prediction intervals, but doesn't handle indirect observations: in meta-analysis, we observe only noisy effects due to the limited number of participants in each trial. To handle noise, this chapter develops a simple, efficient version of fully-conformal kernel ridge regression, based on a novel condition called idiocentricity. It introduce noise-correcting terms in the residuals and analyze their interaction with a "variance shaving" technique. In multiple experiments on healthcare datasets, the proposed algorithms deliver tighter, sounder intervals than traditional ones. This chapter charts a new course for meta-analysis and evidence-based medicine, where heterogeneity and untrusted data are embraced for more nuanced and precise predictions.

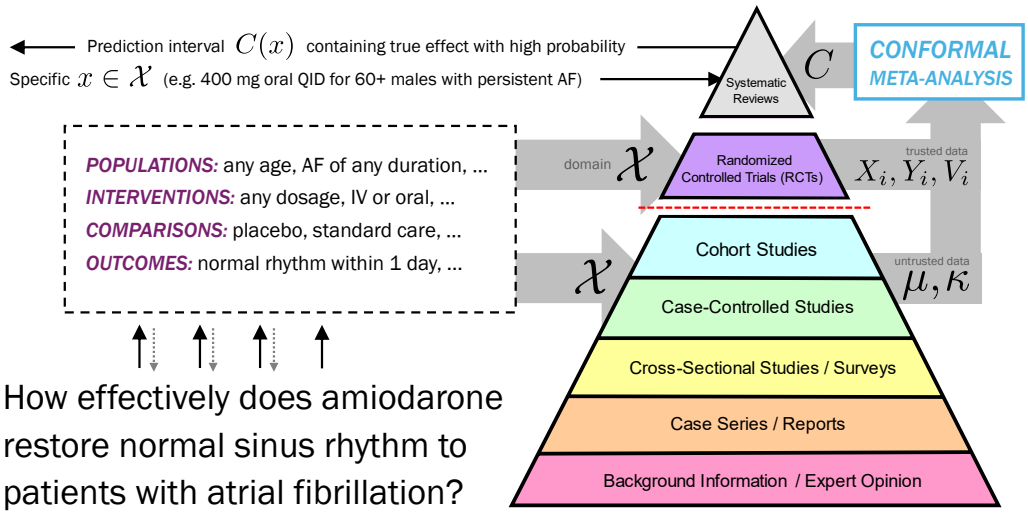


Figure 2.1: We propose changing how meta-analysis answers scientific questions. First, a relatively broad domain $\mathcal{X}$ for the meta-analysis is determined, possibly through further interaction with the user. This allows more expansive questions which include more data. Next, both trusted and untrusted data relevant to $\mathcal{X}$ are retrieved. Conformal meta-analysis takes these and produces not just a single interval, but a predictive model C. Given specific treatment circumstances x, the model predicts $C(x)$ which, under standard assumptions, contains the true effect with high probability.

2.1 Introduction

A systematic review of a scientific question formally collects relevant, reliable evidence and answers the question as precisely as the evidence allows. Roughly 30,000 systematic reviews are published every year, either as standalone scientific papers or as part of clinical practice guidelines, by thousands of academic, professional, and regulatory organizations [Hoffmann et al., 2021]. Systematic reviews adhere to highly-scrutinized methodology [Higgins et al., 2019, Page et al., 2021, Schünemann et al., 2013] and are widely considered to be the pinnacle of empirical evidence [Guyatt et al., 1995, Murad et al., 2016]. They have a decisively influential role in healthcare and related fields, especially in contentious situations where different parties disagree or have competing interests. This is because systematic reviews are designed to be rigorous and unbiased, in a broad sense [Sackett, 1979]: they should yield reliably correct answers, un-

blemished by personal opinions, conflicts of interest, unproven assumptions, or confounding of causation by correlation.

Meta-analysis is the statistical core of most systematic reviews. A key goal of meta-analysis[1] is to learn, from the collected evidence, a predictor C of causal effect: given features x of a treatment, its true effect u should, with high probability, lie within the predicted interval $C(x) \subseteq \mathbb{R}$. As described here, meta-analysis models heterogeneity in the treatment and (in turn) its effect. For example, changing the age of patients or the dosage of a drug corresponds to a change in x, which would lead to a possibly different prediction $C(x)$ of a different u. Unfortunately, prevalent meta-analysis algorithms do not model heterogeneity in x, treating its consequences as inexplicable random noise in u. This is because the complexity of meta-analysis is profoundly constrained by the stringent expectations placed upon systematic reviews: only a limited fraction of "trusted" evidence is allowed in meta-analysis, leaving little hope of learning the complex relationship between x and u.

Specifically, to avoid the confounding biases of observational studies, meta-analysis is (ideally) based solely upon well-conducted, randomized, controlled trials. These allow causal questions (e.g. "what is the effect of administering this drug?") to be reliably answered. On average, about 10-20 RCTs are included in the meta-analysis of a systematic review [Hoffmann et al., 2021], with up to 500 on the upper end [Cipriani et al., 2018]. This ignores the vast majority of accumulated experience with the empirical phenomena of interest. As discussed in Section 2.7.1, this bulk of unused, untrusted data may be formalized as a prior distribution over relationships between x and u.

2.1.1 Our Contributions

This chapter demonstrates that untrusted data — with all its possible confounding, biases, and even outright errors — can be incorporated into meta-analysis while remaining rigorous and unbiased. In fact, this chapter offers stronger, provable guarantees while weakening the assumptions traditionally employed in meta-analysis. The solution is based upon *conformal prediction* [Lei and Wasserman, 2014, Shafer and Vovk, 2008, Vovk et al., 2005]. While conformal pre-

[1]Meta-analysis also involves estimating parameters with confidence intervals; see Section 2.7.1.

diction aptly manages the inclusion of untrusted data, there are two unresolved challenges when applying it to meta-analysis. The first challenge is noise: though we aim to predict true effects u, the observed effects $Y_i \sim N(U_i, V_i)$ are blurred by limited trial sizes. This noise is curiously challenging to manage, since small (high noise) studies can differ fundamentally from large (low noise) studies. This reflects difficulties in clinical practice, where large-scale trials routinely fail to confirm the results of smaller ones [Ioannidis, 2005, Komajda et al., 2010, Manson et al., 2019]. The second challenge arises from limited sample ($n \leq 500$) of included trials. This essentially mandates the use of full (rather than split) conformal prediction, which poses a computational burden, and complicates efforts to handle noise.

We resolve the aforementioned challenges of applying conformal prediction, giving rise to *conformal meta-analysis*. This approach consists of the following layers: (1) kernel ridge regression learning a posterior from the prior and trials, (2) a fast, simple implementation of full conformal prediction of y, based on residuals produced by KRR, and (3) a strategy for predicting u, exploiting the simplicity of the conformal intervals for y. We show that sufficiently high regularization makes KRR *idiocentric*: as y varies, the residual for (x, y) changes more than the other residuals. Under this condition, fully-conformal KRR can be simplified to computing quantiles in two lists. Its simplicity allows us to prove — through an analysis we call "variance shaving" — that its prediction intervals for y typically contain the true effects u as well, with just a slight loss in confidence.

Our experiments have two goals: (1) to quantify how much conformal meta-analysis could improve predictions when used, as intended, with large amounts of untrusted data, and (2) to more qualitatively assess, before such data are available, how it would impact the experience of producing and consuming systematic reviews. At a high level, we find that conformal meta-analysis could improve how the medical community interacts with evidence.

2.2 Preliminaries

These are the predictive goals of meta-analysis.

Predicting Effects. *Let* $(X_1, U_1, V_1), \ldots, (X_n, U_n, V_n), (x, u, v)$ *be exchangeable random vari-*

ables, where $X_i, x \in \mathcal{X}$ are features, $U_i, u \in \mathbb{R}$ are effects, and $V_i, v > 0$ are variances. Let $Y_i = U_i + \mathcal{E}_i$, where independently $\mathcal{E}_i | V_i \sim N(0, V_i)$. Let $\mu : \mathcal{X} \to \mathbb{R}$ and $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be fixed mean and positive-definite kernel functions, respectively. From (μ, κ), the (X_i, Y_i, V_i), and x, for a desired confidence level $\alpha \in (0, 1)$, produce an interval $C(x)$ such that $\mathbb{P}(u \in C(x)) \geq 1 - \alpha$, where the probability is over all the random variables.

Predicting Trials. *Same as above, except C also takes v, and should satisfy $\mathbb{P}(y \in C(x, v)) \geq 1 - \alpha$, where $y = u + \epsilon$ for independent $\epsilon \sim N(0, v)$.*

> *This is the first time meta-analysis is introduced to the machine learning community as a regression problem of major algorithmic interest.*

The first task is more practically useful and technically involved. However, since u is not observable, but y is, the second task is more easily verifiable. It is not immediately clear which task is more challenging, in the sense of needing wider intervals. On one hand, y has inherently more variance than u. On the other, the prediction of u is made without knowing v, which might otherwise distinguish between small and large trials having characteristically different u. Section 2.7.1 thoroughly describes the origin and purpose of these tasks.

Comment on notation. This chapter uses capital letters (e.g. X) to denote data from the n training trials, and lowercase letters (e.g. x) for the test trial. In the conformal prediction literature, it is more common to use capitals to refer to all $n + 1$ data. The training data are then indexed as $X_1, \ldots, X_n$ (or $X_{:n}$), and the testing data as X_{n+1} (or X_{test}). This notation emphasizes the exchangeability of all the data, and reserves lowercase letters for fixed constants. Both of these conventions are helpful for statistical analysis. This chapter, which is more computational in nature, involves algebraic expressions and code which reference the training and test data separately, where the usual indexing would become cumbersome. We lament this notational incompatibility, but feel it preserves the clarity of some aspects of our presentation.

2.2.1 Related Work

Causal inference from observational data. Performing randomized, controlled trials is not the only way to estimate causal effects. After making appropriate assumptions, causal inferences can

be extracted from observational data [Imbens and Rubin, 2015, Pearl, 2009, Spirtes et al., 2001]. This is an extensive research endeavor encompassing many fields; we mention some of the most relevant work here. The survey by Colnet et al. [2024] discusses various approaches to integrating RCTs with observational data. To estimate the CATE, causal forests [Wager and Athey, 2018] and metalearners [Künzel et al., 2019] combine machine learning techniques with causal reasoning. The most widespread assumption of such methods is ignorability, or unconfoundedness. It requires that, having observed the features x, the treatment assigned to a participant is independent of their potential outcomes $\rho(0)$ and $\rho(1)$. That is, there are no unmeasured variables outside of x that could bias treatment towards different participants. Another widespread assumption is positivity, or overlap: for every x, both the treatment and the comparison have a chance of being assigned.

Such strong, unproven assumptions are plausible in many circumstances, but they are not appropriate for systematic reviews. At some point, assumptions must be tested; systematic reviews, more confirmatory than exploratory in nature, often serve this crucial purpose. Nevertheless, conformal meta-analysis allows these methods to be (indirectly) used in systematic reviews, without any concerns about their unproven assumptions. These methods can ideally be used to extract better μ and κ from the untrusted data. Thus, conformal meta-analysis doesn't replace these methods; rather, it expands their domain of application to more scientific settings.

Conformal prediction of latent variables. Previous works have examined how to conformally predict an underlying u while observing only noisy $Y_1, \ldots, Y_n$. It is often empirically observed that conformal prediction can be obliviously robust to label noise, in the sense that $C(x)$, without any involvement of V or v, manages to covers u without any loss in confidence. However, provable guarantees remain elusive. Feldman et al. [2023] show that if $C(x)$ always contains the median of $u \mid x$, then $C(x)$ covers u with no loss in confidence. This is a very strong assumption in meta-analysis, as it essentially posits that the relationship between x and u has been globally determined, and the main difficulty of conformal prediction is to account for the uncertainty driven by the unobserved variables ξ. Most approaches to (non-obliviously) handling noise involve some modification to split conformal prediction. In classification, the (discrete) labels may be noisy because they are the majority vote from some underlying proba-

bility distribution, which reflects uncertainty over the true class. Stutz et al. [2023] adapt split conformal prediction to account for this uncertainty by sampling multiple labels from the underlying distribution. Sesia et al. [2023] and Penso and Goldberger [2024] modify split conformal prediction to estimate the amount of over (or under) coverage of $C(x)$. Unfortunately, splitting the data is not feasible in meta-analysis, where n is small. Label noise should be distinguished from label shift, when the training $Y_1, \ldots, Y_n$ are sampled from a different distribution than the test y [Podkopaev and Ramdas, 2021].

Meta-regression. A meta-regression fits the observed effects Y_i as a (typically linear) function of the features X_i [Stanley and Jarrell, 1989]. Meta-regression is usually conducted to diagnose which features are responsible for heterogeneity. It can also generate useful hypotheses for future research, by identifying which features are associated with higher or lower effects. While meta-regression and conformal-meta-analysis are similar in form, there are a number of crucial differences. Most importantly, unlike conformal-meta analysis, meta-regression does not offer predictive guarantees for new x; the fit to the data is post-hoc and interpretive [Baker et al., 2009, Thompson and Higgins, 2002]. The (non-predictive) statistical task in meta-regression is to determine which features have a statistically significant relationship with the effect [Huizenga et al., 2011]. To limit spurious findings, meta-regression is typically performed on a small number of prespecified features. By contrast, conformal meta-analysis fits powerful, nonlinear models on a potentially large number of features. In conformal meta-analysis, the regression, as embodied by the prediction band C, is presented as the main result, not just an adjunct diagnostic.

Individual treatment effects. This paper improves predictions by tailoring them to specific patient populations described by x. However, it still averages over individuals within those populations. There are multiple approaches to accounting for this heterogeneity by predicting individual treatment effects. One approach is to perform n-of-1 trials, where a single individual serves as both the treatment and control by applying the treatment at different times [Guyatt et al., 1986, Liang and Recht, 2023]. Another approach is to conduct causal inference, under stronger assumptions, on individual-level data from randomized and/or observational studies [Bica et al., 2021]. As part of this approach, conformal prediction has been employed to obtain prediction intervals for potential outcomes [Lei and Candès, 2021], possibly as a function of a parameter Γ

bounding the amount of unobserved confounding [Jin et al., 2023, Yin et al., 2024]. These approaches require individual-level data, different experimental designs, or stronger assumptions, which are worth pursuing primarily when individual (within-trial) variation is significant relative to between-trial variation. Whether this occurs depends on the nature of the treatment as well as the granularity of $\mathcal{X}$.

Bayesian priors. Conformal meta-analysis takes a prior probability distribution, along with trial data, and makes predictions from a posterior distribution — a process that mirrors Bayesian inference [Gelman et al., 1995]. The choice of prior can substantially influence Bayesian inference, sometimes for the better: for example, informative, data-driven priors for the heterogeneity variance ν can mitigate excessive posterior uncertainty [Lilienthal et al., 2024, Rhodes et al., 2016]. However, in a Bayesian meta-analysis, the prior can potentially hurt the empirical coverage of the reported intervals. As a simple example, even if all the trial data indicate a large treatment effect, a prior which heavily concentrates on zero effect would nonetheless result in tight posterior intervals around zero. Such behavior is inappropriate for systematic reviews, which are meant to resolve collective uncertainty among parties who do not necessarily share the same prior beliefs. One attempt to address this problem is to use uninformative priors. However, even such choices can seriously impact the empirical validity of a Bayesian meta-analysis [Hamaguchi et al., 2021]. In conformal meta-analysis, by contrast, even strong beliefs can be safely encoded into the prior without breaking empirical coverage guarantees. In the aforementioned example of a concentrated, incorrect prior, conformal meta-analysis would merely yield loose intervals.

Uniform confidence bands. Prediction intervals also should not be confused with uniform confidence bands, which offer the following stronger guarantee, and do not involve unobserved ξ:

$$\mathbb{P}_C\left(\text{for all } x \in \mathcal{X},\ u(x) \in C(x)\right) \geq 1 - \alpha$$

Such bands have been developed for Gaussian process regression in the context of online optimization, where new points x are sequentially, adaptively chosen to minimize uncertainty about u [Chowdhury and Gopalan, 2017, Fiedler et al., 2021, Neiswanger and Ramdas, 2021, Srinivas

et al., 2009]. Since subsequent x are chosen adaptively using the band, it is essential for the band to hold for arbitrary x rather than just randomly-sampled x. Strictly speaking, these bands are correct for arbitrary μ and κ. However, their widths depend on the smoothness of u, as quantified by its norm in the reproducing kernel Hilbert space induced by κ. Since u is unknown, this quantity is also unknown. As a practical matter, when μ and κ can range from very good to very poor, the band is either very wide or unknown. Though conformal meta-analysis only offers prediction intervals with marginal coverage guarantees, their width and coverage do not depend on unknown quantities.

Utilizing unlabeled data. Trusted labels are generally considered a scarce resource in machine learning, especially compared to unlabeled data (i.e. x sampled from the marginal distribution of $\mathbb{P}$). Unlabeled data are commonly used to pretrain large foundation models [Dahl et al., 2011, Dai and Le, 2015]. Semi-supervised learning studies how to rigorously use unlabeled data to improve predictions [Balcan and Blum, 2010]. Angelopoulos et al. [2023] recently proposed prediction-powered inference as an approach to safely tighten confidence intervals by using unlabeled data along with a prior derived from separate, untrusted data. In this approach, (1) the unlabeled data and prior (which is temporarily treated as correct) are used to estimate the parameter, (2) concentration inequalities are applied to bound the estimation error arising from limited unlabeled data, and (3) the labeled data are used to correct the estimation error due to inaccuracy of the prior. Subsequently, Zrnic and Candès [2024] proposed cross-prediction-powered inference, which has similar goals but does not utilize untrusted data. Instead, it splits the data (as in cross-validation) to train a prior. Such methods have been used to improve out-of-distribution causal inference [Demirel et al., 2024]. However, these methods are not directly applicable to predictive meta-analysis, in which there are no available unlabeled data. Furthermore, these methods are designed to produce confidence intervals rather than prediction intervals.

Safely using untrusted data. Various endeavors in statistics and machine learning involve making predictions that are rigorously guaranteed, even though they use untrusted data. To some extent, all these techniques manage to circumvent the "garbage-in, garbage-out" principle. PAC-Bayesian generalization theory formalizes inductive bias as an (untrusted) prior probability distribution [McAllester, 1998, Seeger, 2002, Shawe-Taylor and Williamson, 1997]. Its general-

ization bounds are tight when the prior and data align, so that a learning algorithm (producing a posterior distribution) can fit the data without diverging far from the prior. While PAC-Bayes is a very useful theoretical tool, conformal prediction bounds are quantitatively tighter, especially when n is small. In statistics, an untrusted prior distribution can be used to define an e-value, a nonnegative statistic whose mean is at most one [Neiswanger and Ramdas, 2021]. Using its reciprocal as an unnormalized density leads to e-posteriors, which can be used as the basis for valid inferences and decisions [Grünwald, 2023]. To derive confidence intervals with conditional coverage guarantees, likelihood-free inference methods can exploit untrusted prior information [Masserano et al., 2023]. In computer science, algorithms can be infused with untrusted predictions, also called side information, advice, or hints [Mitzenmacher and Vassilvitskii, 2022]. When the predictions are good, the algorithms run faster; when the predictions are bad, the algorithms retain acceptable worst-case performance. A prototypical example is binary search, which can be modified to run in $O(1)$ time given a good prediction of the target's index, and in $O(\log n)$ time no matter how bad the prediction was.

2.3 Predicting Trials with Idiocentric Linear Smoothers

The main result of this section is the following algorithm for Predicting Trials. While useful in its own right, it is also the basis of our subsequent algorithm for Predicting Effects. The parameter $\eta > 0$ adjusts noise correction in the residuals; larger η induces more correction.

Theorem 1 (Conformal Trial Prediction)**.** *Let $\eta > 0$. Under the assumptions for Predicting Trials, Algorithm 2.1 returns $[y_-, y_+]$ satisfying $\mathbb{P}(y \in [y_-, y_+]) \geq 1 - \alpha$.*

The following observation is the basis of full conformal prediction.

Proposition 1 (Full Conformal Prediction)**.** *Let $(X_i, Y_i, V_i) \sim \mathbb{P}$ (for $i = 1, \ldots, n$) as well as $(x, y^*, v) \sim \mathbb{P}$ be exchangeable. Let $[R; r]$ be the residuals of a symmetric (i.e. invariant to permutations of the training data) learning algorithm on $[X; x]$, $[Y; y]$ and $[V; v]$. Given any $\alpha \in (0, 1)$, let $\tau = \lceil (1 - \alpha)(n + 1) \rceil$ and:*

$$C(x, v) = \{y : r \text{ is among the } \tau \text{ smallest of } R_1, \ldots, R_n\} \tag{2.1}$$

Then $\mathbb{P}(y^* \in C(x,v)) \geq 1-\alpha$. *[Vovk et al., 2005]*

```python
# capital arguments for training trials, lowercase are for test
def predict_trial(M, K, Y, V, m, k, k_o, v, alpha, eta):
  # linear algebra for KRR; see appendix
  Q,q,A,a,B,b,D,d,S2,s2,tau = precomputations(M,K,Y,m,k,k_o,alpha)

  if tau <= n: # enough training trials for conformal prediction
    # compute interval L_i = G_i±H_i for each training trial
    # y in L_i corresponds to r <= R_i for residuals
    a2A2 = a**2*S2 - A**2*s2
    rho = eta*(D*s2 - d*S2 - a2A2*v)
    G = (A*B*s2 -a*b*S2) / a2A2
    H = sqrt(maximum(0, s2*S2*(A*b - a*B)**2 - rho*a2A2)) / a2A2
    L_n, L_p = G-H, G+H
    # return quantiles of L_i's upper/lower endpoints
    y_p = sort(L_p)[tau-1]
    y_n = flip(sort(L_n))[tau-1]
    return y_n, y_p
  else: # not enough training trials
    return -inf, inf
```

Algorithm 2.1: Python/NumPy code for conformal prediction of trials. Import statements are omitted. Section 2.7.2 describes the arguments as well as the subroutine for precomputations.

Our residuals are based on KRR, and use the variances $[V;v]$ to correct for the noise present in $[Y;y]$. Given a parameter $\lambda \in \mathbb{R}$, prior (μ,κ), and data $([X;x],[Y;y])$, KRR learns a posterior $(\hat{\mu},\hat{\kappa})$. Let the posterior mean on $[X;x]$ be $[\widehat{M};\hat{m}]$. Let the diagonal of the posterior kernel matrix be $[S^2;s^2]$. Let $Z_i = \mathbb{E}_\xi(\widehat{M}_i - Y_i)^2 - (\widehat{M}_i - U_i)^2 \geq 0$ and $z = \mathbb{E}_\epsilon(\hat{m}-y)^2 - (\hat{m}-u)^2 \geq 0$ be the expected impact of the noise (ξ and ϵ). The residuals are:

$$R_i = \left((\widehat{M}_i - Y_i)^2 - \eta Z_i\right)/S_i^2 \qquad r = \left((\hat{m}-y)^2 - \eta z\right)/s^2$$

Subtracting (an η fraction of) Z_i and z effectively reduces the importance of smaller trials. Deriving these residuals for KRR, with independent Gaussian noise in Y, is basic linear algebra and probability. As Section 2.7.2 shows, the residuals are squares of affine functions in y. That is,

for some A_i, B_i, a, and b:

$$R_i = \left((A_i y + B_i)^2 - \eta Z_i\right)/S_i^2 \qquad r = \left((ay+b)^2 - \eta z\right)/s^2 \tag{2.2}$$

It is easy to see that residuals of this form are shared by any learning algorithm where $[\widehat{M}; \hat{m}]$ are linear in $[Y; y]$, albeit nonlinear in $[X; x]$. These are known as *linear smoothers*, including methods such as k-nearest neighbors, Nadaraya-Watson kernel regression, and smoothing splines [Buja et al., 1989]. Our techniques conceptually extend to all linear smoothers; we use KRR primarily because it gracefully incorporates the prior (μ, κ).

2.3.1 Idiocentricity and its Consequences

Burnaev and Nazarov [2016], building upon Nouretdinov et al. [2001], derived an algorithm for computing KRR's $C(x)$. Though their algorithm is computationally efficient, it returns a general prediction set (a union of disjoint intervals and isolated singletons) which isn't amenable to analytic reasoning. We substantially simplify the algorithm under the following condition.

Definition 1 (Idiocentricity)**.** *The residuals R_i, r are idiocentric if $\frac{|a|}{s} > \frac{|A_i|}{S_i}$ for all i.*

This condition means that changing the test example's y changes its own residual more than it changes the residuals of other examples. For other learning algorithms, it can be generalized in terms of derivatives.

Definition 2 (Idiocentricity)**.** *The residuals $R_1, \ldots, R_n, r$ are idiocentric if:*

$$\lim_{y \to \pm\infty} \frac{|\partial r / dy|}{|\partial R_i / dy|} > 1 \quad \textit{for all } i = 1, \ldots, n$$

This definition reduces to the previous one in the case of linear smoothers. Let us show how idiocentricity simplifies $C(x)$.

Theorem 2. *For $i = 1, \ldots, n$, let $\rho_i = \eta(Z_i s^2 - z S_i^2)$. Define intervals $L_i = G_i \pm H_i$, where:*

$$G_i = \frac{A_i B_i s^2 - ab S_i^2}{(aS_i)^2 - (A_i s)^2} \textit{ and } H_i = \frac{\sqrt{\max\left(0,\ s^2 S_i^2 (A_i b - a B_i)^2 - \rho_i((aS_i)^2 - (A_i s)^2)\right)}}{(aS_i)^2 - (A_i s)^2}$$

If KRR is idiocentric, then its full conformal prediction set (2.1) simplifies to:

$$C(x, v) = \{y : y \textit{ is inside more than } n - \tau \textit{ of the } L_1, \ldots, L_n\}$$

Proof. Since the residuals defined in (2.2) are squared, we can flip the signs of b and B_i to standardize on $a, A_i \geq 0$. $r \leq R_i$ rewrites to $S_i^2(ay + b)^2 + \rho_i \leq s^2(A_i y + B_i)^2$. Under the condition $a/s > A_i/S_i \geq 0$, this is equivalent to $y \in L_i$. □

> *Making fully-conformal prediction fast for a large class of learning algorithms is of general interest in statistics and machine learning.*

We slightly loosen the defining condition of $C(x)$ to obtain an even simpler algorithm.

Lemma 1. *In the notation of Theorem 2, let y_+ be above τ of the upper endpoints of the L_i, and let y_- be below τ of the lower endpoints of the L_i. Then $C(x, v) \subseteq [y_-, y_+]$.*

Proof. The upper endpoint y_+ is met when, for τ of the $i \in \{1, \ldots, n\}$, we have $y_+ \leq L_i$ or $y_+ \geq L_i$. Ignore the first possibility, which becomes more unlikely as y_+ increases, for a potentially looser but nonetheless valid interval. A similar argument justifies y_-. □

KRR is idiocentric when λ is set sufficiently large. The following upper bound is sometimes loose, but works well throughout this chapter. We note that the optimal setting of λ for regression may not coincide with the optimal setting for conformal prediction. For example, $\lambda = 0$ (known as interpolation or ridgeless regression) can be a good learning algorithm [Hastie et al., 2022, Liang and Rakhlin, 2020], but it is useless for full conformal prediction, since its residuals are all zero.

Theorem 3. *KRR is idiocentric if $\lambda \geq \max\{\kappa(X_1, X_1), \ldots, \kappa(X_n, X_n), \kappa(x, x)\}$.*

To prove Theorem 1, use the λ of Theorem 3 to earn the simplified interval of Theorem 2, which is supported by the coverage guarantee of Proposition 1.

2.4 Predicting Effects

The culmination of this chapter is the following algorithm for predicting causal effects.

```python
def predict_effect(M, K, Y, V, m, k, k_o, alpha, eta):
  return predict_trial(M, K, Y, V, m, k, k_o, 0, alpha, eta)
```

Algorithm 2.2: Python code for conformal prediction of effects, deferring entirely to Algorithm 2.1 with $v = 0$.

Theorem 4 (Conformal Effect Prediction)**.** *Let $\eta > 0$. Under the assumptions for Predicting Effects, Algorithm 2.2 returns $[u_-, u_+]$ satisfying $\mathbb{P}(u \in [u_-, u_+]) \geq 1 - \frac{\alpha}{(1-\alpha)\text{erfc}\sqrt{\eta/2}}$.*

Setting $\eta = 0$ (i.e. disabling noise correction) obtains confidence $\frac{1-2\alpha}{1-\alpha}$, which is just a slight loss from $1 - \alpha$ when $\alpha \approx 0$. (For example, 0.95 confidence drops to 0.9473, which probably doesn't change $\tau = \lceil (1-\alpha)(n+1) \rceil$). This setting is appropriate when $V \approx 0$, i.e. the trials all have a large number of participants. By setting $\eta = 2 \cdot \text{inverfc}(\frac{1}{c(1-\alpha)})^2$, the confidence drops to $1 - c \cdot \alpha$. More noise correction is conceptually more appropriate when analyzing mixtures of small and large trials. However, the loss of confidence means larger n is needed, which may not be a worthwhile tradeoff. Conformal prediction is usable only when $\tau \leq n$; with $c = 2$, a final confidence of 95% requires $n \geq 40$. This is twice the n needed for $\eta = 0$.

While the overhead at $\eta = 0$ is not practically important, it indicates either the algorithm or its analysis are suboptimal. When meta-analysis is very close to regression ($V \approx 0$), the original $1 - \alpha$ coverage should be smoothly recovered. In the appendix, we present another approach which behaves correctly in this regard. It is based on fundamentally different techniques which can be extended to non-normal or even adversarial noise. It determines a probability $1 - \delta$ region $\mathcal{U}$ for the true effects U. Then, it formulates an optimization problem to bound all the intervals which could have been generated by $\widehat{U} \in \mathcal{U}$.

This is a general strategy to make conformal prediction robust to label noise.

Theorem 5 (Conformal Effect Prediction via Robust Optimization)**.** *Let $\delta > 0$. Under the assumptions for Predicting Effects, the respective solutions u_- and u_- to (2.10) and (2.11) in Section 2.7.7 satisfy $\mathbb{P}(u \in [u_-, u_+]) \geq (1 - \alpha)(1 - \delta)$.*

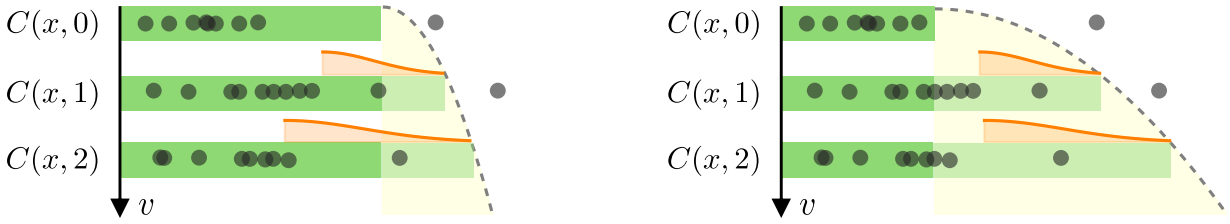


Figure 2.2: A high-level sketch of $C(x,0)$'s coverage of u, when η is small enough (left) versus too large (right). The gray dots are u, and its distributions conditioned on various v are shown. $C(x,0)$ is the dark green bar; as v increases, $C(x,v)$ increases by $\sqrt{\eta v}$, and that growth (in yellow) is shaved. The orange curves convey the spread of $|N(0,v)|$. With good η (left), $C(x,v)$ grows slowly compared to $|N(0,v)|$, which naturally pushes in the u (on average) as v increases. Thus, $C(x,0)$ is wide enough to contain most of the u, no matter what v is. On the right, when η is large, $C(x,v)$ adapts more dynamically to v, so $C(x,0)$ is smaller. Too many u in the yellow region are shaved.

2.4.1 Understanding Conformal Effect Prediction

Theorem 1 guarantees that $C(x,v)$ usually covers $y \sim N(u,v)$. We will use this guarantee to derive intervals $C(x)$ that usually cover u. We don't have a v to plug into $C(x,v)$, so we have to dig into how $C(x,v)$ works. The claim of Theorem 4 is that $C(x,0)$ covers u just slightly less often than it covers y, so long as the level of noise correction η is not too high. This holds because of two counterbalancing properties of $C(x,v)$ that hold for all $v \geq 0$.

The first property is that most of the spread of $|N(0,v)|$ can be shaved from the edges of $C(x,v)$ without losing too many u. This is possible because, in meta-analysis, we care only about small α, ideally around 0.05. Since $C(x,v)$ covers y with high probability, there are only a few u closer than $|N(0,v)|$ to the ends of $C(x,v)$ — otherwise, bad flips of the noise could push too many y out of the interval, which would violate the coverage guarantee of $C(x,v)$. While this logic indicates shaving is a conceptually feasible strategy, it remains an abstract possibility, since we don't know v, and don't know how much to shave. (It should intuitively be $O(\sqrt{v})$, but constants matter).

The second property is that making η smaller limits the growth of $C(x,v)$. We mean this in a completely formulaic sense — we have reasonably concrete expressions for the endpoints of $C(x,v)$, and the following Lemma 2 shows they widen by $\sqrt{\eta v}$. When $\eta = 0$, $C(x,v)$ doesn't depend on v at all. In other words, when noise correction is disabled, $C(x,0)$ must

completely internalize the impact of noise, yielding a relatively wide interval. Larger settings of η allow $C(x, v)$ to grow more with v, allowing (relatively) thin intervals at small v. To concretely realize the shaving strategy, we just need to set η small enough so that, as a function of v, *the shaveable region within* $C(x, v)$ *grows as fast as* $C(x, v)$ *itself*. This allows us to obliviously use the baseline $C(x, 0)$. The conditional distribution $v \mid x$ is arbitrary and unknown, but any probability mass on $v > 0$ simply pushes more u within $C(x, 0)$.

The fact that $C(x, v)$ grows proportionally to $\sqrt{v}$ to capture the noise is not only intuitive, it is necessary. Most well-behaved learning algorithms should yield conformal intervals which grow (on average) at roughly this rate. Our ability to prove an exact growth rate, in the next lemma, relies on the simplicity of full conformal prediction for idiocentric linear smoothers.

Lemma 2 (Normal Interval Growth)**.** *Let* $C(x, v)$ *be the interval from Theorem 2. For all* $\eta \geq 0$ *and* $v > 0$*,* $C(x, v) \subseteq C(x, 0) \pm \sqrt{\eta v}$.

The rest of the proof of Theorem 4 doesn't depend on either idiocentricity or linear smoothers. Lemma 3 formalizes the first property described above: most u are contained within $C(x, v)$ by a margin that grows with v. Finally, Lemma 4 shows that $C(x, v)$ can be shaved down to $C(x, 0)$, with η determining the loss in coverage of u.

Lemma 3 (Pay For Room)**.** *Recall* $y = u + \epsilon$ *for* $\epsilon \sim N(0, v)$*. Let* $w = [u - \epsilon, u + \epsilon]$*, with possibly unsorted endpoints. If* $\mathbb{P}(y \in C(x, v)) \geq 1 - \alpha$*, then* $\mathbb{P}(w \subseteq C(x, v)) \geq (1 - 2\alpha)/(1 - \alpha)$.

Lemma 4 (Shaving)**.** *If* $\mathbb{P}(w \subseteq C(x, v)) \geq \frac{1-2\alpha}{1-\alpha}$*, then* $\mathbb{P}(u \in C(x, 0)) \geq 1 - \frac{\alpha}{(1-\alpha)\text{erfc}\sqrt{\eta/2}}$.

2.5 Simulations

We performed four types of simulations on three biomedical datasets from the Penn Machine Learning Benchmark [Olson et al., 2017]. These regression datasets define K and Y; we generated synthetic M and V according to parameters prior error ≥ 0 and effect noise ≥ 0, respectively. We use Algorithm 2.2 with $\eta = 0$. We compare it to the state-of-the-art HKSJ method, which is described in Section 2.7.1.

Simulation 1: This investigates when conformal meta-analysis is superior to traditional meta-analysis. For different settings of prior error, we compare the widths of the intervals

obtained by different meta-analysis algorithms. The only situation in which HKSJ is competitive with conformal meta-analysis is when the prior is bad and the number of trials is small/moderate. Otherwise, conformal meta-analysis is superior, sometimes achieving intervals that are dramatically thinner than those of HKSJ.

Simulation 2: This experiment checks whether the desired 95% confidence level is still achieved as effect noise increases. Conformal meta-analysis succeeds, whereas HKSJ fails badly. On the other datasets (see appendix), HKSJ sometimes drops below 80% confidence. This deficiency is present at all settings of effect noise, though it aggravates at higher values. This simulation shows that conformal meta-analysis has a rigorous coverage guarantee, and HKSJ does not. It should be noted that HKSJ was developed to improve the coverage guarantee of the more prevalent Higgins-Thompson-Spiegelhalter method.

Simulation 3: This experiment compares different instantiations of Algorithm 2.2: one with $\eta = 0$, and the other with $\eta = 0.4015$, with α adjusted so both ultimately seek a 90% confidence level. With the higher setting of η, over-coverage is consistently demonstrated. This suggests that the analysis of Section 2.4 can be improved, at least in some settings.

Simulation 4: Our approach assumes that, in many fields, it should be possible to develop good priors from large volumes of untrusted data. However, if these priors are indeed very accurate, it is unclear whether using KRR (upon just n trials) is worth the complexity, and possible statistical overhead, over just using the prior as a fixed predictor. (This is conceptually equivalent to using an extremely large ridge parameter λ, or performing split conformal using all the training data for calibration.) This simulation indicates there is no such overhead: our fully-conformal intervals are strictly superior to those derived from a fixed prior. Thus, unless assumptions stronger than exchangeability are used to derive prediction intervals, learning is superior to mere validation. Note that, when prior error is high, HKSJ becomes superior.

2.6 Case Study: Amiodarone

We revisit the systematic review of Letelier et al. [2003], which assessed the effectiveness of amiodarone for atrial fibrillation (AF) patients. Its outcome measure is the relative risk of normal

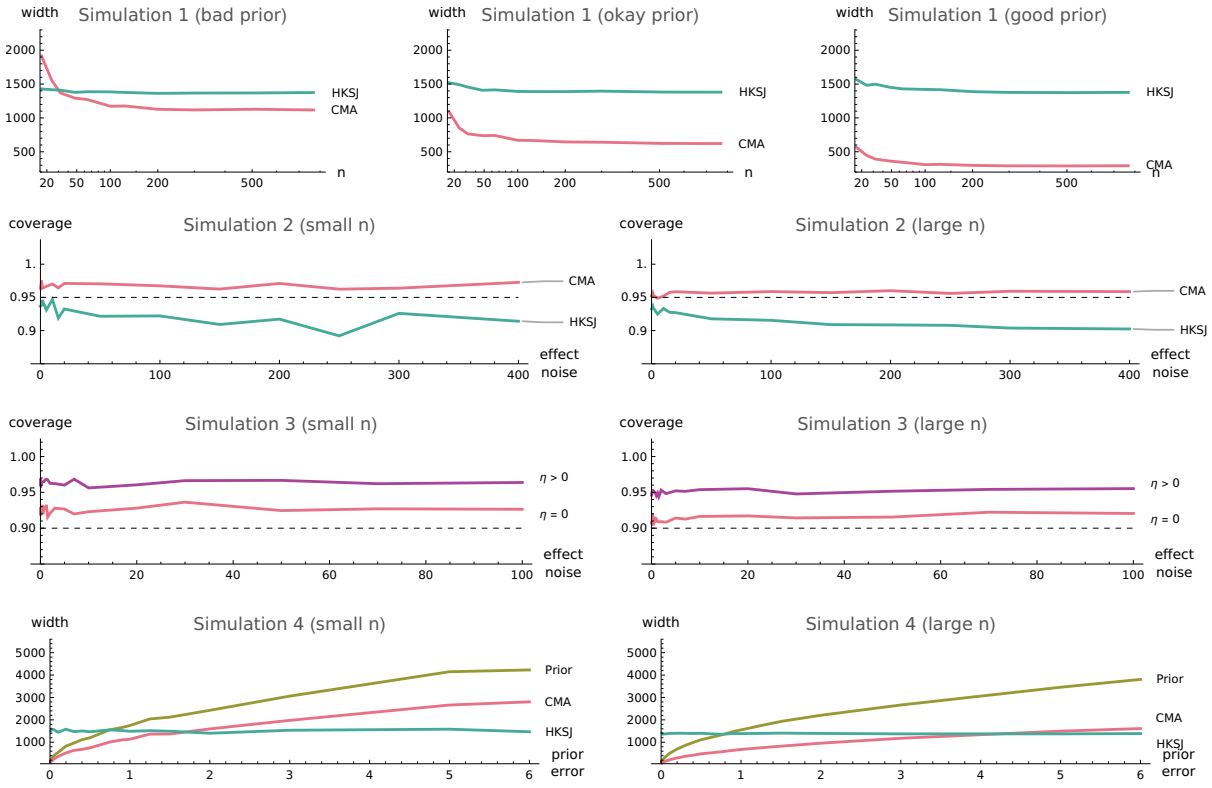


Figure 2.3: Results of all simulations on a single exemplar dataset. See Section 2.7.8 for congruent results on the other datasets, as well as precise descriptions of the effect noise and prior error parameters. Overall, conformal meta-analysis can deliver much tighter intervals than traditional methods (Simulation 1), even though traditional methods have weak coverage guarantees (Simulation 2), whereas our algorithms, or their analyses, have (overly) strong guarantees (Simulation 3). Our algorithms, not just good priors, are essential to this performance (Simulation 4).

sinus rhythm; that is, the probability of restoring normal rhythm when administered amiodarone, divided by the probability of restoration with placebo. The review involved $n = 21$ trials, which we use as training data. For test data, we identify 4 trials that were published after the review, but would have met its inclusion criteria [Balla et al., 2011, Karaçağlar et al., 2019, Kochiadakis et al., 2007, Thomas et al., 2004]. Per the Predicting Trials task, we compare traditional meta-analysis (the Bayesian algorithm of Proposition 4, described in Section 2.7.1) with conformal meta-analysis (Algorithm 2.1, with $\eta = 1$).

Our goal is not to make any scientific claims about amiodarone, nor to reassess its evidence base; that would require following a formal, preregistered protocol. Though we temper our quantitative findings (depicted in Figure 2.4), we find them qualitatively interesting. Conformal

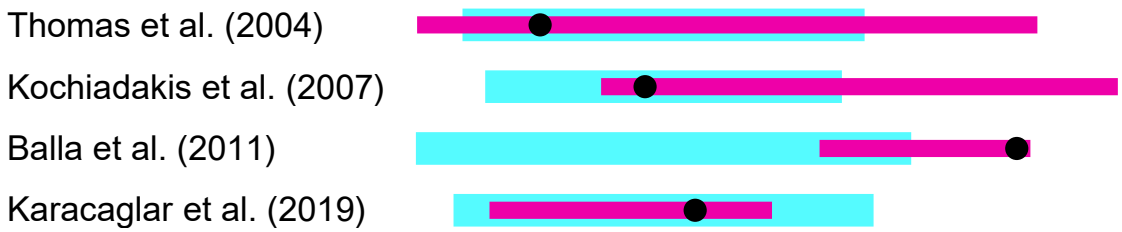


Figure 2.4: Prediction intervals for new observed effects y (black dots) produced by traditional meta-analysis (light blue) and conformal meta-analysis (magenta, thin). On average, they are comparable in width (1.34 and 1.31, respectively). Conformal meta-analysis manages to cover the discrepant trial of Balla et al. [2011]. Note that the prior for conformal meta-analysis was produced **post-hoc**, having already seen the analysis of Letelier et al. [2003] and the results therein. Thus, these intervals should not be interpreted as quantitative evidence, but merely as qualitative illustrations of the behavior of conformal meta-analysis.

meta-analysis manages to correctly predict all 4 trials, whereas traditional meta-analysis suffers a misprediction. This is not statistically convincing, but it aligns with the fact that conformal meta-analysis has a rigorous coverage guarantee, whereas traditional algorithms do not. (See Section 2.7.1 for more details.) It is interesting to observe that not all of the conformal intervals overlap; by contrast, traditional intervals all inherently overlap. This suggests users of conformal meta-analysis could enjoy predictions that are meaningfully responsive to the details of their proposed treatment, perhaps distinguishing between effective and ineffective ones.

Section 2.7.9 describes how we conducted the conformal meta-analysis. We highlight some ways it differed from the usual process. The first change is training a prior on helpful data that would otherwise be ignored. We identify 8 trials that did not meet the inclusion criteria, since they were not placebo controlled. To generate pseudo-effects for these trials, we need to understand the placebo effect. This leads to the second major change, which is holistically including the perspectives of practitioners. The critique of Slavik and Zed [2004], written by two doctors of pharmacy, gave estimates for the placebo effect on sinus rhythm (i.e. spontaneous conversion) in different circumstances. We use these estimates to generate the pseudo-effects. Finally, arguably the biggest change involves LLMs. In order to extract features from trials, we give their published PDFs to LLMs (specifically, GPT-4 and Claude) along with a prompt including example output. Next, parsing code (also written by LLMs) converts the textual features to nu-

merical (x, y, v). Thus, LLMs can be used to aid meta-analysis, much as meta-analysis serves as a question-answering system. This experience, and the results of the chapter overall, reflect positively on the following dilemma: *can language models be used to rigorously answer scientific questions?*

> *Meta-analysis can be viewed as a structured, quantitative, yet natural form of question answering. It could be used as a testbed for studying LLM safety.*

2.7 Appendix

2.7.1 Background for Meta-Analysis

Outcomes and Effects

Let x be a features describing a treatment. This consists of the prospectively-set criteria of its population, intervention, comparison, and measure of outcome, commonly abbreviated as PICO [Richardson et al., 1995]. For example, x may include the duration of an exercise program and the minimum age of its participants. It may also include auxiliary information that was collected passively and retrospectively, though (as described in the next section) this may complicate the interpretation of the meta-analysis. x does not have to be numerical; it can be, for example, a published document describing a clinical trial. The number of participants in such a trial should not be intentionally encoded in x, since a treatment should be applicable to any number of people. However, avoiding implicit, unintentional correlations between trial design and trial size may be difficult or impossible. Let ξ encode factors which influence the treatment, but are neither controlled nor observed. For example, the effect of an exercise program may surreptitiously depend on the altitude of the training facility or the jobs of the participants.

In the Neyman-Rubin framework of potential outcomes [Neyman, 1923, Rubin, 1974], for a single participant denoted by ρ, $\rho(1) \in \mathbb{R}$ is the outcome when assigned the treatment, and $\rho(0) \in \mathbb{R}$ is the outcome when assigned the comparison. Each outcome may be a final measurement (such as the amount of strength gained after training), or its change from a baseline measurement, or the logarithm of the ratio of final to baseline. The difference $\rho(1) - \rho(0)$ is the individual

effect of the treatment. The potential outcomes framework is challenging because we cannot observe both terms in $\rho(1) - \rho(0)$, since each participant is assigned to either the treatment or the comparison. The conditional average treatment effect (CATE), denoted by u, quantifies the expected difference between the treatment and comparison for a new participant:

$$u(x, \xi) = \mathbb{E}_\rho \left(\rho(1) - \rho(0) \mid x, \xi \right) \tag{2.3}$$

The CATE is usually defined solely in terms of the observed variables x. We include ξ to emphasize the influence of unobserved variables, which are sometimes ignored in causal inference.

Different Goals of Meta-Analysis

The CATE is the predictive target of meta-analysis. With high probability (typically 95%, with $\alpha = 0.05$), the CATE should lie within the predicted interval:

$$\mathbb{P}_{C,x,\xi} \left(u(x, \xi) \in C(x) \right) \geq 1 - \alpha \tag{2.4}$$

Rather than predicting relatively specific, tangible effects, meta-analysis often focuses on estimating more abstract, harder-to-verify quantities. Meta-analyses usually report a confidence interval CI $\subset \mathbb{R}$ which, with high probability, should contain the average treatment effect (ATE, also known as the summary effect or grand mean):

$$\mathbb{P}_{\text{CI}} \left(\text{ATE} \in \text{CI} \right) \geq 1 - \alpha \;\; \text{where} \;\; \text{ATE} = \mathbb{E}_{x,\xi} \, u(x, \xi)$$

Whereas the confidence interval merely needs to capture the ATE, the prediction interval must capture most of the dispersion around it. (Formally, a prediction interval covers a random variable, and its coverage probability must also account for the randomness of that variable, whereas a confidence interval covers a fixed value). In the presence of significant heterogeneity, the confidence interval is much tighter than the prediction interval, and has little chance of capturing the effect of a future treatment. Due to this potentially unintuitive behavior, and the possibility of instilling overconfidence in evidence about the treatment, many prominent researchers encourage

systematic reviews to report prediction intervals [Borenstein, 2024, IntHout et al., 2016, Riley et al., 2011]. According to some researchers, the relative ease of corroborating (or refuting) predictions makes them essential for scientific rigor and reproducibility [Billheimer, 2019].

These problems are exacerbated by the introduction of features (x) and larger numbers of trials (n), as proposed in this chapter. Since confidence intervals are tighter than prediction intervals, it may be technically tempting to use untrusted priors to analogously tighten intervals for ATE. However, when considering many trials with substantially different features, ATE becomes a useless quantity [Feinstein, 1995, Gould, 2010, Simonsohn et al., 2022, Subramanian et al., 2018]. It is arguably misleading to use features within a statistical analysis but to simultaneously obfuscate their existence in the reported statistic. This is why prediction intervals are presently the preferred solution concept.

While prediction intervals avoid some of the unintuitive pitfalls of confidence intervals, it is important to note that the predictive guarantee (2.4) has subtleties of its own. It is a mixed observational-causal guarantee: coverage does not hold for all x, just marginally (on average) over x. For example, if $\alpha = 0.05$, then it is possible for coverage to be 99% for patients younger than 60 and only 80% for patients between 60 and 70, so long as the average is at least 95%. Achieving conditional coverage guarantees (i.e. without averaging over x) is not possible without further assumptions [Lei and Wasserman, 2014]. Since prevalent meta-analysis algorithms do not involve x, their guarantees are of course marginal over x.

The guarantee (2.4) is most reliable when the distribution over x is explicitly specified by a generative model. If trial designs are actually chosen according to this distribution, and x consists solely of prospectively-set, controllable variables, then it is easy to sample future x for which the coverage guarantee holds. If x includes retrospectively-collected information, or the trials are designed according to unspecified criteria, then the guarantee becomes less meaningful.

Randomized Controlled Trials (RCTs)

An RCT enrolls m participants with potential outcomes $\rho_1, \ldots, \rho_m$. Uniformly at random, it assigns m_0 of them to group 0 (the comparison), and the remaining m_1 to group 1 (the treatment). Most RCTs do not report individual outcomes. Rather, they report the mean and (corrected)

variance of the comparison outcomes as $y^{(0)}$ and $v^{(0)}$. The same statistics are reported for the treatment outcomes as $y^{(1)}$ and $v^{(1)}$. These are combined into y, the difference in means, and v, a sum of the observed standard errors [Deeks and Higgins, 2010]. These statistics are defined as:

$$y^{(g)} = \frac{1}{m_g} \sum_{i \text{ in group } g} \rho_i(g) \qquad\qquad y = y^{(1)} - y^{(0)}$$

$$v^{(g)} = \frac{1}{m_g - 1} \sum_{i \text{ in group } g} (\rho_i(g) - y^{(g)})^2 \qquad\qquad v = \frac{v^{(0)}}{m_0} + \frac{v^{(1)}}{m_1}$$

Condensing the data into y and v has the following rationale. It can be shown that y is an unbiased estimate of the CATE:

$$\mathbb{E}(y \mid x, \xi) = \mathbb{E}(u \mid x, \xi)$$

Thus, as the RCT enrolls a very large number of participants, y converges to u, regardless of x and ξ. This is the primary reason why RCTs are so valuable. v is an estimate of y's variance around u, under conditions discussed in the next section.

Random-Effects Model of the Data

Meta-analysis is conducted upon n trials, each with data $X_i \in \mathcal{X}$, $Y_i \in \mathbb{R}$ and $V_i > 0$ for $i = 1, \ldots, n$. As discussed above, each trial's Y_i is centered around U_i, but varies around it due to its limited number of participants. Because Y_i is a sample average, by the central limit theorem, it is asymptotically normally distributed around U_i. The random-effects model of meta-analysis [DerSimonian and Laird, 1986, Higgins et al., 2009] asserts, as a simplifying assumption, that Y_i is exactly (not just asymptotically) normally distributed around U_i with true variance equal to the observed one. That is, $Y_i \sim N(U_i, V_i)$. This can be written in a way that highlights a key difference between the standard random-effects model and this chapter's model:

$$Y_i(X_i, \xi_i) = \text{ATE} + \underbrace{U_i(X_i, \xi_i) - \text{ATE}}_{\text{between-trial heterogeneity}} + \underbrace{N(0, V_i)}_{\text{within-trial variation}} \tag{2.5}$$

The first and last terms are the same in both models. The random-effects model asserts that the middle term $U_i - \text{ATE} \sim N(0, \nu)$ where ν (often denoted by τ^2) is called the heterogeneity variance. By contrast, in this chapter, U_i depends on the covariates X_i, and may also involve arbitrary (non-Gaussian) noise through ξ_i. Thus, this chapter eliminates a normality assumption which is viewed as dubious in practice [Liu et al., 2023b]. The normality of within-trial variation, though less controversial, may be tenuous for small trials [Jackson and White, 2018].

Untrusted Data as a Probability Distribution

Independently of RCTs, practitioners and researchers often possess deep intuitions about the CATE. These intuitions arise from the lower levels of the evidence hierarchy: observational studies, individually-published cases, hands-on experience, and personal belief [Murad et al., 2016]. It is difficult to rigorously infer causation from such untrusted (or "real-world") data, since they are observational and may have deeply-embedded biases. Nonetheless, it is often found that untrusted data agree with RCTs [Benson and Hartz, 2000, Concato et al., 2000]. Retrospectively, Toews et al. [2024] found the ratio of risk-ratios between RCTs and observational studies to be approximately 1.08. The prospective RCT-DUPLICATE trial found their Pearson correlation to be 0.82 [Wang et al., 2023], with much of the discrepancy attributable to readily-identified factors [Heyard et al., 2024]. For example, observational claims data do not typically record whether treatment was initiated in a hospital, but this may affect the outcomes of RCTs.

Since untrusted data originates from different kinds of sources and experiences, it does not share the form of RCTs. A modern approach to capturing large, disparate quantities of knowledge is to (pre)train foundation models. Such models are already being developed for healthcare [Moor et al., 2023, Singhal et al., 2023, Tu et al., 2024]. This approach involves learning an embedding $\phi(x)$ which maps features x into a Euclidean space having inner product $\kappa(x, x') = \phi(x)^T \phi(x')$. On top of this embedding, a linear predictor of the CATE can be trained as $\mu(x) = w^T \phi(x)$. Practically, this representation (μ, κ) encompasses nearly every useful way of predicting the CATE. Mathematically, this representation constructs a Gaussian process, a probability distribution over functions $f : \mathcal{X} \mapsto \mathbb{R}$, with higher probability placed on f which could plausibly approximate the CATE [Kanagawa et al., 2018, Williams and Rasmussen, 2006].

In this probabilistic perspective, $\mu(x) = \mathbf{E}_f f(x)$ and $\kappa(x, x') = \mathbf{E}_f(f(x)-\mu(x))(f(x')-\mu(x'))$. Gaussian processes are often used as prior probability distributions in Bayesian inference [Gelman et al., 1995]. (See Section 2.2.1 for further comparison to Bayesian inference).

A significant restriction is that μ and κ are fixed relative to the data. In practical terms, this means the outcomes of the trials are not reincorporated into the prior. Otherwise, the trials could trivially, erroneously serve as their own reality check. Thus, although μ and κ are completely untrusted in terms of their veracity and utility, their provenance (especially the data used to generate them) must be clearly understood. Practices such as preregistration and data transparency can facilitate this understanding [Munafò et al., 2017]. Importantly, this assumption is about the processes used to include data, which are under our control. It is not about the complex phenomena which generate the data itself. In this sense, it is much weaker than the assumptions of ignorability and positivity which are made in causal inference.

The assumption of fixed μ and κ is technically stronger than necessary. The following task description more precisely specifies the exchangeability requirement which is required for our techniques to apply.

Predicting Effects (Technical). *Let $\mu : \mathcal{X} \to \mathbb{R}$ and $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Let $\bar{X} = [X; x] \in \mathcal{X}^{n+1}$, $\bar{U} = [U; u] \in \mathbb{R}^{n+1}$, $\bar{V} = [V; v] \in \mathbb{R}_+^{n+1}$, $\bar{M} = [\mu(\bar{X}_i)]_i$, and $\bar{K} = [\kappa(\bar{X}_i, \bar{X}_j)]_{i,j} \succ 0$ be random variables. Suppose, for any permutation σ of $\{1, \ldots, n+1\}$, the joint distribution of the $(\bar{X}_i, \bar{U}_i, \bar{V}_i, \bar{M}_i, [\bar{K}_{i,j}]_j)$ equals that of the $(\bar{X}_{\sigma(i)}, \bar{U}_{\sigma(i)}, \bar{V}_{\sigma(i)}, \bar{M}_{\sigma(i)}, [\bar{K}_{\sigma(i),\sigma(j)}]_j)$. Let $Y_i = U_i + \mathcal{E}_i$, where independently $\mathcal{E}_i | V_i \sim N(0, V_i)$. From $(\bar{X}, Y, V, \bar{M}, \bar{K})$, for a desired confidence level $\alpha \in (0, 1)$, produce an interval $C(x)$ such that $\mathbb{P}(u \in C(x)) \geq 1 - \alpha$, where the probability is over all the random variables.*

An advantage of this more technical formulation is that its underlying exchangeability assumption can be tested [Vovk, 2021]. Thus, even when the prior has unknown provenance, a diagnostic hypothesis test can potentially check if its involvement in the meta-analysis is valid.

Standard Meta-Analysis Algorithms

As previously mentioned, prevalent algorithms for meta-analysis ignore the covariates x; in the parlance of the field, they perform mean-effect prediction rather than meta-regression. Thus,

they simply return a single prediction interval $C \subset \mathbb{R}$ rather than a prediction band. Because the model (2.5) is not analytically solvable, there is no exact, rigorous frequentist prediction interval. Instead, there are many different formulae [Nagashima et al., 2021, Veroniki et al., 2019], each involving approximations which hold only as $n \to \infty$. Most of the prediction intervals have this form:

$$C = \widehat{\text{ATE}} \pm t\sqrt{\hat{\nu} + \widehat{\text{Var}}(\widehat{\text{ATE}})} \tag{2.6}$$

In this expression, the variance estimates $\hat{\nu}$ and $\widehat{\text{Var}}(\widehat{\text{ATE}})$ are usually algorithm-specific. More generally, t is the $1 - \frac{\alpha}{2}$ quantile of a Student t distribution with either $n-1$ or $n-2$ degrees of freedom. $\widehat{\text{ATE}}$ is an estimate of ATE, usually based upon inverse-variance weighting:

$$\widehat{\text{ATE}} = \sum_i w_i Y_i \Big/ \sum_i w_i \qquad \text{where } w_i = \frac{1}{V_i + \hat{\nu}} \text{ for each } i = 1, \ldots, n \tag{2.7}$$

In practice, the most widely-used prediction interval is based on the classical heterogeneity estimator $\hat{\nu}$ of DerSimonian and Laird [1986], and an estimator $\widehat{\text{Var}}(\widehat{\text{ATE}})$ proposed by Higgins et al. [2009]. When n is small, experimental evidence indicates this interval is too small to satisfy (2.4) with the desired probability $1 - \alpha$. To the best of our knowledge, this method does not have a proven coverage guarantee, so the following result is stated imprecisely.

Proposition 2 (Classical Prediction Interval)**.** *Assume the model (2.5) with $U_i \sim N(ATE, \nu)$. Define the following quantities within (2.6):*

$$\hat{\nu} = \frac{Q - (n-1)}{S_1 + S_2/S_1} \quad \widehat{Var}(\widehat{ATE}) = (\sum_i w_i)^2 \quad \bar{Y} = \frac{\sum_{i=1}^n V_i^{-1} Y_i}{\sum_{i=1}^n V_i^{-1}} \quad Q = \sum_{i=1}^n V_i^{-1}(Y_i - \bar{Y})^2 \quad S_r = \sum_{i=1}^n V_i^{-r}$$

Then C, as defined in (2.6), approximately satisfies (2.4) as $n \to \infty$.

Partlett and Riley [2017] proposed an alternative prediction interval based upon restricted maximum likelihood (REML) and Hartung-Knapp-Sidik-Jonkman (HKSJ) estimators [Nagashima et al., 2021]. REML obtains $\hat{\nu}$ and $\widehat{\text{ATE}}$ as the maximizers of a log-likelihood function $\ell(\hat{\nu}, \widehat{\text{ATE}})$ which is filtered to remove influences from irrelevant variables [Viechtbauer, 2005]. It is not concave, so it cannot be maximized by standard algorithms. However, its stationary points

$\partial \ell \ / \ d\hat{\nu} = 0$ (for fixed $\widehat{\text{ATE}}$) and $\partial \ell \ / \ d\widehat{\text{ATE}} = 0$ (for fixed $\hat{\nu}$) have closed-form expressions, so it is amenable to alternating maximization. The following estimator $\widehat{\text{Var}}(\widehat{\text{ATE}})$ was developed independently by Hartung and Knapp [2001] and Sidik and Jonkman [2003]. Cochrane Statistical Methods and other groups endorse the use of HKSJ [IntHout et al., 2014, Veroniki, 2022, Veroniki et al., 2019]. This method also does not have a proven coverage guarantee.

Proposition 3 (REML+HKSJ Prediction Interval)**.** *Assume the model (2.5) with $U_i \sim N(ATE, \nu)$. Initialize $\hat{\nu} = 0$. Alternate the updates to $\widehat{ATE}$ and w in (2.7) with the following update of $\hat{\nu}$, until a fixed point is approximately reached:*

$$\hat{\nu} \leftarrow \frac{\sum_{i=1}^n w_i^2((Y_i - \widehat{ATE})^2 - V_i)}{\sum_{i=1}^n w_i^2} + \frac{1}{\sum_{i=1}^n w_i} \qquad \widehat{Var}(\widehat{ATE}) = \sum_{i=1}^n \frac{(Y_i - \widehat{ATE})^2 w_i}{(n-1)\sum_j w_j}$$

Then C, as defined in (2.6), approximately satisfies (2.4) as $n \to \infty$.

In addition to these frequentist intervals, Bayesian intervals for u can also be obtained [Gelman et al., 1995, Smith et al., 1995]. These begin with prior distributions over ATE and ν. Improper (i.e. unnormalized) uniform priors are a default uninformative choice [Röver, 2017]. Using the random-effects model as a likelihood, Bayes' theorem obtains the posterior distribution over ATE and ν, which induces a (normal) posterior distribution over u. From this posterior distribution, a prediction interval for u can be derived. Such intervals can be highly sensitive to the choice of uninformative prior, which is partially why Bayesian methods are less common in systematic reviews [Hamaguchi et al., 2021]. Nonetheless, there are some circumstances where the flexibility of Bayesian methods is desirable. For example, the Bayesian approach can be extended to predicting trials. The posterior distribution for future $y \sim N(u, v)$ is just u's posterior with v more variance.

Proposition 4 (Bayesian Trial Prediction)**.** *Let the prior distribution over ATE be improper uniform. Assume the likelihood (2.5) with $U_i \mid ATE, \nu \sim N(ATE, \nu)$. Then, recalling (2.7), the posterior predictive distribution conditioned on ν is $y \mid \nu = \hat{\nu} \sim N\left(\widehat{ATE}, (\sum_i w_i)^{-1} + \hat{\nu} + v\right)$. [Röver, 2017]*

The Ethics of Meta-Analysis

Healthcare is important, uncertain, and sometimes controversial. Evidence-based medicine was introduced to help resolve some of these issues, but it involves controversy of its own. It unavoidably privileges certain kinds of experiences and opinions over others. This paper does not introduce these problems, but it does operate in their midst. Let us examine how these problems could be ameliorated or aggravated by our approach.

Currently, meta-analysis in evidence-based medicine is highly exclusionary. The "lower levels" of the evidence hierarchy are deprecated in favor of RCTs in an effort to preserve rigor and eliminate bias. However, this introduces some bias of its own. For example, RCTs are expensive to conduct. Any methodology that substantially prefers RCTs may be substantially influenced by funding agencies and associated institutions [Lundh et al., 2017]. Furthermore, RCTs are not ethical to conduct in many situations [Morris and Nelson, 2007]. Conformal meta-analysis recognizes that RCTs are especially valuable, but it holistically incorporates data of less rarified origin. Even when our methods do not lead to quantitative improvements, they are arguably more fair, inclusive, and comprehensive. They could ameliorate concerns that evidence-based medicine limits the autonomy of healthcare professionals [Armstrong, 2007].

However, conformal meta-analysis introduces additional computational and statistical complexity into the process of meta-analysis. This complexity could be exploited by bad actors, with negative societal consequences. For example, a malicious meta-analyst could sneak trial data into their prior to arrive at intentionally biased conclusions. To prevent such harms from occurring, any rigorous conclusions derived from conformal meta-analysis need to be accompanied by safeguards on the handling of data.

2.7.2 Computations for KRR

Let M and K be the mean and kernel function applied to the training features:

$$M = [\mu(X_1), \ldots, \mu(X_n)]^T \in \mathbb{R}^n \qquad K = [\kappa(X_i, X_j)]_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$$

Given a parameter $\lambda \in \mathbb{R}$ and observations $U \in \mathbb{R}^n$, KRR learns the following posterior on the

training features:

$$\widehat{M} = (\widehat{K}/\lambda)U + (K/\lambda + I)^{-1}M \qquad\qquad \widehat{K} = \lambda(K + \lambda I)^{-1}K$$

In full conformal prediction, KRR is applied to the training set (X, U) augmented by (x, u). We will use bars to denote this augmentation, so $\bar{X} = [X; x]$, $\bar{U} = [U; u]$. Let $m = \mu(x)$, $k = [\kappa(X_1, x), \ldots, \kappa(X_n, x)]^T$, $k_0 = \kappa(x, x)$, and:

$$\bar{I} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \qquad \bar{K} = \begin{bmatrix} K & k \\ k^T & k_0 \end{bmatrix} \qquad \bar{Q} := (\bar{K} + \lambda\bar{I})^{-1}\bar{K} = \begin{bmatrix} Q & q \\ q^T & q_0 \end{bmatrix}$$

Then, the augmented posterior mean is:

$$\begin{bmatrix} \widehat{M} \\ \hat{m} \end{bmatrix} = \bar{Q}\begin{bmatrix} U \\ u \end{bmatrix} + \overbrace{(\bar{K}/\lambda + \bar{I})^{-1}\begin{bmatrix} M \\ m \end{bmatrix}}^{\bar{t}}$$

So the differences between the observations and posterior means are:

$$\begin{bmatrix} U - \widehat{M} \\ u - \hat{m} \end{bmatrix} = (\bar{I} - \bar{Q})\begin{bmatrix} U \\ u \end{bmatrix} - \bar{t} = \begin{bmatrix} (I - Q)U - qu \\ -q^T U + (1 - q_0)u \end{bmatrix} - \bar{t} = \begin{bmatrix} Au + B \\ au + b \end{bmatrix}$$

with the abbreviations:

$$\begin{bmatrix} A \\ a \end{bmatrix} = \begin{bmatrix} -q \\ 1 - q_0 \end{bmatrix} \qquad\qquad \begin{bmatrix} B \\ b \end{bmatrix} = \begin{bmatrix} I - Q \\ -q^T \end{bmatrix} U - \bar{t}$$

The augmented posterior kernel matrix is $\lambda\bar{Q}$. Thus, $S_i = \sqrt{\lambda Q_{ii}}$ and $s = \sqrt{\lambda q_0}$. To determine Z_i and z, decompose the differences between the observations and the posterior means. As

before, denote augmentation with overlines, as in $\bar{\mathcal{E}} = [\mathcal{E}; \epsilon]$.

$$\begin{bmatrix} Y - \widehat{M} \\ y - \hat{m} \end{bmatrix} = (\bar{I} - \bar{Q})(\bar{U} + \bar{\mathcal{E}} - \bar{M}) - \bar{z}$$

$$= \begin{bmatrix} U - \widehat{M} \\ u - \hat{m} \end{bmatrix} - (\bar{I} + \bar{Q})\bar{\mathcal{E}} = \begin{bmatrix} U - \widehat{M} \\ u - \hat{m} \end{bmatrix} + \begin{bmatrix} (I - Q)\mathcal{E} - q\epsilon \\ -q^T\mathcal{E} + (1 - q_0)\epsilon \end{bmatrix}$$

Now, calculate the mean squared error with respect to $\mathcal{E}_i \sim N(0, V_i)$ and $\epsilon \sim N(0, v)$:

$$\mathbb{E}\,(Y_i - \widehat{M}_i)^2 = \mathbb{E}\,(U_i - \widehat{M}_i + (e_i - Q_i)^T\mathcal{E} - q_i\epsilon)^2$$

$$= (U_i - \widehat{M}_i)^2 + \mathbb{E}\left((1 - Q_{ii})\mathcal{E}_i - \sum_{j \neq i} Q_{i,j}\mathcal{E}_j - q_i\epsilon\right)^2$$

$$= (U_i - \widehat{M}_i)^2 + \overbrace{\underbrace{(1 - Q_{ii})^2 V_i + \sum_{j \neq i} Q_{i,j}^2 V_j}_{D_i} + \underbrace{q_i^2}_{A_i^2} v}^{Z_i}$$

$$\mathbb{E}\,(y - \hat{m})^2 = \mathbb{E}\,(u - \hat{m} - q^T\mathcal{E} + (1 - q_0)\epsilon)^2 = (u - \hat{m})^2 + \overbrace{\underbrace{\sum_j q_j^2 V_j}_{d} + \underbrace{(1 - q_0)^2}_{a^2} v}^{z}$$

2.7.3 Proof of Theorem 3

Recalling Definition 1 and the computations in Section 2.7.2, we seek to prove:

$$\frac{|q_i|}{\sqrt{Q_{ii}}} < \frac{|1 - q_0|}{\sqrt{q_0}} \iff \frac{|q_i|}{\sqrt{Q_{ii} \cdot q_0}} < \frac{|1 - q_0|}{q_0}$$

Since $\bar{Q}$ is positive definite, its entries are the inner products among some vectors $f_0, \ldots, f_n$. In particular, $q_i = \langle f_i, f_0 \rangle$. Thus, by the Cauchy-Schwartz inequality:

$$|q_i| = |\langle f_i, f_0 \rangle| \leq ||f_i|| \cdot ||f_0|| = \sqrt{||f_i||^2 \cdot ||f_0||^2} = \sqrt{Q_{ii} \cdot q_0}$$

Thus, it suffices to show that $1 < \frac{1 - q_0}{q_0}$, that is, $0 < q_0 < \frac{1}{2}$. Since $\bar{Q}$ is positive definite, $q_0 > 0$ is obvious. To establish $q_0 < \frac{1}{2}$, let us examine the constraints on the last row of $\bar{Q}$. By the original

definition of $\bar{Q}$, taking just the last column of $\bar{K}$:

$$\begin{bmatrix} q \\ q_0 \end{bmatrix} = (\bar{K} + \lambda \bar{I})^{-1} \begin{bmatrix} k \\ w \end{bmatrix}$$

Expanding and multiplying by both sides:

$$\left(\begin{bmatrix} K & k \\ k^T & k_0 \end{bmatrix} + \lambda \bar{I} \right) \begin{bmatrix} q \\ q_0 \end{bmatrix} = \begin{bmatrix} k \\ k_0 \end{bmatrix}$$

Expanding again:

$$\begin{bmatrix} K \\ k^T \end{bmatrix} q + \begin{bmatrix} k \\ k_0 \end{bmatrix} q_0 + \lambda \begin{bmatrix} q \\ q_0 \end{bmatrix} = \begin{bmatrix} k \\ k_0 \end{bmatrix}$$

This finally leads to the constraints:

$$(K + \lambda I)q = (1 - q_0)k$$
$$k^T q + \lambda q_0 = (1 - q_0)k_0$$

Inverting the first equation to solve for $q = (1 - q_0)(K + \lambda I)^{-1}k$ and plugging into the second yields:

$$(1 - q_0)k^T(K + \lambda I)^{-1}k + \lambda q_0 = (1 - q_0)k_0$$

If we take $\lambda = k_0$ then:

$$(1 - q_0)k^T(K + k_0 I)^{-1}k = (1 - 2q_0)k_0$$
$$\sum_{i=1}^{n} \frac{\tilde{k}_i^2}{\lambda_i + k_0} = \frac{1 - 2q_0}{1 - q_0} k_0$$

The left hand side is positive, so in order for the right hand to be positive, it is necessary that $q_0 < \frac{1}{2}$, as originally desired. To ensure λ (and KRR overall) remain symmetric, this analysis

must be applied to any permutation of the data. Thus, λ should be larger than any diagonal entry of $\bar{K}$, not just k_0.

2.7.4 Proof of Lemma 2

The interval for y depends on v only through ρ_i:

$$\frac{1}{\eta}\rho_i = Z_i s^2 - z S_i^2 = D_i s^2 - d S_i^2 - \overbrace{\left((aS_i)^2 - (A_i s)^2\right)}\, v$$

Under idiocentricity, $a/s > A_i/S_i$. Thus, the bracketed term above is positive, ρ_i decreases with v, the square-root radius in L_i (which subtracts ρ_i) increases with v, and the denominator in L_i is positive. Dividing by the denominator, the radius H_i is of the form $\sqrt{\ldots + \eta v} \leq \sqrt{\ldots} + \sqrt{\eta v}$. Neither the center G_i of L_i nor the other elided terms in the radius depend on v; the $\sqrt{\eta v}$ term is the only one which involves v.

2.7.5 Proof of Lemma 3

Abbreviate $C(x, v) = C$. The key property we repeatedly use is that y is one of the endpoints of w chosen uniformly at random, conditionally independent of the other data. If $w \not\subseteq C$, then either both of its endpoints are not in C, or exactly one of them isn't. In the former case, y clearly isn't in C; in the latter, it isn't with probability $\frac{1}{2}$. Let gray be the event that exactly one of w's endpoints is outside of C. First, we prove that:

$$\mathbb{P}(\mathsf{gray}) \leq 2\alpha \tag{2.8}$$

Let near denote both of w's endpoints are in C, and far that neither are in C, so that $\mathsf{near}, \mathsf{gray}, \mathsf{far}$ partition the probability space. By total probability, and the aforementioned reasoning about y:

$$\begin{aligned}
\mathbb{P}(y \in C) &= (1 - \mathbb{P}(\mathsf{gray}) - \mathbb{P}(\mathsf{far}))\mathbb{P}(y \in C \mid \mathsf{near}) + \mathbb{P}(\mathsf{far})\mathbb{P}(y \in C \mid \mathsf{far}) + \mathbb{P}(\mathsf{gray})\mathbb{P}(y \in C \mid \mathsf{gray}) \\
&= (1 - \mathbb{P}(\mathsf{gray}) - \mathbb{P}(\mathsf{far}))(1) + \mathbb{P}(\mathsf{far})(0) + \mathbb{P}(\mathsf{gray})\frac{1}{2} \\
&\leq 1 - \mathbb{P}(\mathsf{gray}) + \mathbb{P}(\mathsf{gray})\frac{1}{2}
\end{aligned}$$

Combining this with the assumption yields (2.8). Next:

$$\begin{aligned}
\mathbb{P}(y \in C \mid w \not\subseteq C) &= \mathbb{P}(\textsf{gray})\mathbb{P}(y \in C \mid \textsf{gray}) && \text{(only nonzero case)} \\
&= \mathbb{P}(\textsf{gray})\frac{1}{2} && \text{(symmetry)} \\
&\leq \alpha && \text{(2.8)}
\end{aligned}$$

With this inequality, the original claim follows from:

$$\begin{aligned}
1 - \alpha &\leq \mathbb{P}(y \in C) && \text{(assumption)} \\
&= \mathbb{P}(w \subseteq C, y \in C) + (1 - \mathbb{P}(w \subseteq C))\mathbb{P}(y \in C \mid w \not\subseteq C) && \text{(total probability)} \\
&\leq \mathbb{P}(w \subseteq C, y \in C) + (1 - \mathbb{P}(w \subseteq C))\alpha && \text{(proved above)} \\
&= \mathbb{P}(w \subseteq C) + (1 - \mathbb{P}(w \subseteq C))\alpha && (y \in w)
\end{aligned}$$

Note this proof required ϵ to be symmetric, zero mean, and conditionally independent given its variance v, but not necessarily normally distributed.

2.7.6 Proof of Lemma 4

Abbreviate $C = C(x, v)$ and $\widetilde{C} = C(x, 0)$. For the first inequality of the following block, the worst case is obtained when u is exactly one of the endpoints of $\widetilde{C}$ (say, the upper endpoint $\tilde{c}_+$), since that maximizes the distance from the endpoint of C (say, c_+), and therefore maximizes probability that w will still remain within C.

$$\begin{aligned}
\mathbb{P}(w \subseteq C \mid u \notin \tilde{C}) &\leq \mathbb{P}(\tilde{c}_+ + |\epsilon| \leq c_+) \\
&= \mathbb{P}(|\epsilon| \leq \sqrt{\eta v}) && \text{(Lemma 2)} \\
&= \operatorname{erf}\sqrt{\frac{\eta}{2}} && \text{(normal distribution)}
\end{aligned}$$

Thus, the desired claim follows from total probability and some rearranging:

$$\begin{aligned}
\frac{1-2\alpha}{1-\alpha} &\leq \mathbb{P}(w \subseteq C) && \text{(assumption)} \\
&= \mathbb{P}(u \in \tilde{C})\mathbb{P}(w \subseteq C \mid u \in \tilde{C}) + \mathbb{P}(u \notin \tilde{C})\mathbb{P}(w \subseteq C \mid u \notin \tilde{C}) && \text{(total probability)} \\
&\leq \mathbb{P}(u \in \tilde{C}) + (1 - \mathbb{P}(u \in \tilde{C}))\text{erf}\sqrt{\eta/2} && \text{(proved above)}
\end{aligned}$$

2.7.7 Predicting Effects with Robust Optimization

If we had observed true effects U rather than noisy Y, then straightforward conformal prediction would yield a satisfactory interval.

Proposition 5 (Full Conformal Prediction)**.** *Let $(X_i, U_i) \sim \mathbb{P}$ (for $i = 1, \ldots, n$) as well as $(x, u^*) \sim \mathbb{P}$ be exchangeable. Let $[R; r]$ be the residuals of a symmetric learning algorithm on $[X; x]$ and $[U; u]$. Given any $\alpha \in (0, 1)$, let:*

$$C(x) = \{u : r \text{ is among the } \tau \text{ smallest of } R_1, \ldots, R_n\} \qquad \text{for } \tau = \lceil (1-\alpha)(n+1) \rceil \tag{2.9}$$

Then $\mathbb{P}(u^* \in C(x)) \geq 1 - \alpha$. [Vovk et al., 2005]

Let $C(x; \widehat{U})$ denote the prediction interval when $\widehat{U}$ is given as training data. Suppose we know a set $\mathcal{U}$ which contains the true U. If the outer interval $\widehat{C}(x)$ contains all $C(x; \widehat{U})$ over $\mathcal{U}$, then of course $\widehat{C}(x)$ contains $C(x; U)$ and inherits its coverage. Lemma 5 shows the uncertainty over U falling in that plausible set separates from fully-conformal KRR's uncertainty over u, given U. This is because $\mathcal{E}$ is independent from all else, given V.

Lemma 5 (Cover All Possibilities)**.** *Let $\widehat{C}(x)$ contain all intervals induced by the ellipsoid $\mathbf{E}$:*

$$\mathbf{E} = \left\{ E : \sum_{i=1}^{n} \frac{E_i^2}{V_i} \leq \rho \right\} \qquad\qquad \widehat{C}(x) = \bigcup_{E \in \mathbf{E}} C(x; \underbrace{U + \mathcal{E}}_{Y} - E)$$

Let $\rho > 0$ be chosen so that $\mathbb{P}_{\mathcal{E}}(\mathcal{E} \in \mathbf{E} \mid V) \geq 1 - \delta$. Then $\mathbb{P}(u \in \widehat{C}(x)) \geq (1-\alpha)(1-\delta)$.

Proof. Let $C(x; U) = C(x)$ be the interval from Proposition 5 when computed on the true U. In

the following, let rest denote X, U, x, u.

$$\begin{aligned}
\mathbb{P}(u \in \widehat{C}(x)) &\geq \mathbb{P}(u \in C(x), C(x) \subseteq \widehat{C}(x)) && \text{(partial probability)} \\
&= \mathbb{E}_V \mathbb{E}_{\mathsf{rest}} \left(1(u \in C(x)) \cdot \mathbb{P}_{\mathcal{E}}(C(x) \subseteq \widehat{C}(x) \mid V, \mathsf{rest})\right) && \text{(total probability)} \\
&\geq \mathbb{E}_V \mathbb{E}_{\mathsf{rest}} \left(1(u \in C(x)) \cdot (1-\delta)\right) && \text{(see below)} \\
&= (1-\delta)\mathbb{P}_{\mathsf{rest},V}(u \in C(x)) && \text{(total probability)} \\
&\geq (1-\delta)(1-\alpha) && \text{(conformal prediction)}
\end{aligned}$$

A sufficient condition for $C(x;U) \subseteq \widehat{C}(x)$ is that $\mathcal{E} = E$ for some $E \in \mathbf{E}$, i.e. that $\mathcal{E}$ belongs to the ellipsoid. Note that $\widehat{C}(x)$ depends on U but this condition does not. Thus:

$$\begin{aligned}
\mathbb{P}_{\mathcal{E}}(C(x) \subseteq \widehat{C}(x) \mid V, \mathsf{rest}) &\geq \mathbb{P}_{\mathcal{E}}(\mathcal{E} \in \mathbf{E} \mid V, \mathsf{rest}) && \text{(sufficient condition)} \\
&= \mathbb{P}_{\mathcal{E}}(\mathcal{E} \in \mathbf{E} \mid V) && \text{(conditional independence)} \\
&\geq 1-\delta && \text{(assumption)}
\end{aligned}$$ □

This lemma doesn't make any smoothness assumptions on how $C(x;\widehat{U})$ changes as $\widehat{U}$ varies away from U; it relies on the coverage of exactly $C(x;U)$, but not of any slight perturbation $C(x;\widehat{U})$. Furthermore, the lemma does not depend specifically on the normal distribution of $\mathcal{E}$, just that we know a set $\mathbf{E}$ which captures it with probability $1-\delta$. For Gaussian noise, this is an ellipsoid of appropriate scale. This proof does not depend on the geometry of $\mathbf{E}$, just the fact that it contains $\mathcal{E}$ with high probability, and can be computed from Y and V. Thus, this overall strategy can be extended to handle non-Gaussian noise.

The previous lemma converts the statistical problem of covering u into the purely computational problem of determining the endpoints of $\widehat{C}(x)$. When $\eta = 0$, and if U is provided in lieue of Y, Algorithm 2.2 computes the interval $C(x)$ specified in Proposition 5. This allows us to concretely bound the endpoints of $\widehat{C}(x)$ as the following two optimization problems:

$$u_- := \min_{E \in \mathbf{E}} \max\{\text{bottom } n-\tau+1 \text{ lower endpoints of } L_1, \ldots, L_n\} \tag{2.10}$$

$$u_+ := \max_{E \in \mathbf{E}} \min\{\text{top } n-\tau+1 \text{ upper endpoints of } L_1, \ldots, L_n\} \tag{2.11}$$

Though this a nonconvex optimization problem, it has useful structure. From Theorem 2, recall that the endpoints equal $G_i \pm H_i$, where $H_i = |\Delta_i|$. We are using $\eta = 0$, which implies $\rho_i = 0$, and in turn simplifies the equations for these variables. Recalling the equations from Section 2.7.2 and Theorem 2, the conformal idiocentric KRR equations are a set of constraints in the variables $B, b, \widehat{U}$ and E, involving constants $a, A, s, S, Q, q, \bar{t}, Y$ and V:

$$G_i = \frac{A_i B_i s^2 - abS_i^2}{(aS_i)^2 - (A_i s)^2} \qquad \Delta_i = sS_i \frac{A_i b - aB_i}{(aS_i)^2 - (A_i s)^2} \tag{2.12}$$
$$\begin{bmatrix} B \\ b \end{bmatrix} = \begin{bmatrix} I - Q \\ -q^T \end{bmatrix} \widehat{U} - \bar{t} \qquad \widehat{U} = Y - E \qquad \sum_{i=1}^{n} \frac{E_i^2}{V_i} \leq \rho$$

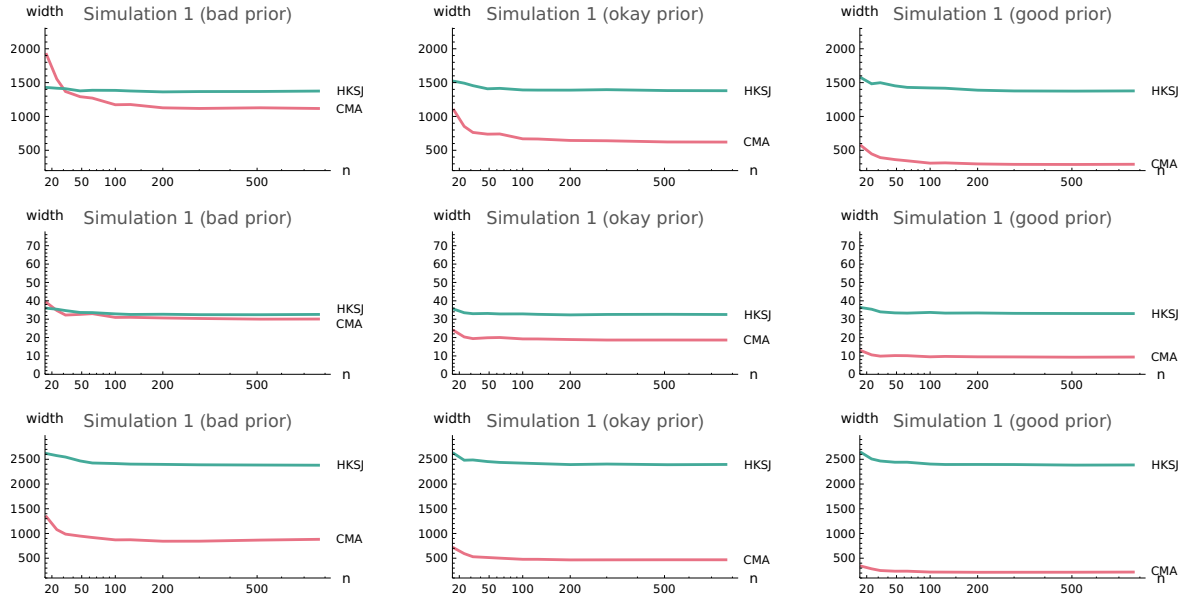
The four constraints on the left are linear. The quadratic constraint is convex, since $V_i > 0$. Thus, the constraints (2.12) are convex. Thus, despite the nonconvexity of the objective, the problems (2.10) and (2.11) may be amenable to semidefinite programming relaxations, robust optimization, and/or nonconvex optimization. We leave such investigation to separate work.

2.7.8 Simulation Details and Full Results

The simulations were performed using three partially-synthetic biomedical datasets from the Penn Machine Learning Benchmark [Olson et al., 2017]: 1196_BNG_pharynx, 1201_BNG_breastTumor, and 1193_BNG_lowbwt. We randomly subsample training data (X, U) as well as test data (x, u). The kernel matrix K is generated using either the Gaussian or Laplace kernel as κ. For consistency across datasets having different scales, a parameter effect noise > 0 is introduced, and the distribution of V is constructed to satisfy $\mathbb{E}(V_i) =$ effect noise $\cdot \sqrt{\mathbb{E}|U_i|}$. Specifically $V_i \sim \text{Exp}(1) \cdot \sqrt{\text{effect noise} \cdot \mathbb{E}|U_i|}$. Similarly, to produce prior means M of varying quality, a parameter prior error > 0 is introduced, and the distribution of M satisfies MSE$(M, U) =$ prior error $\cdot \mathbb{V}(U)$. Furthermore, the difference between M and U should not be purely random — otherwise, using KRR to explain this difference would be hopeless. Instead, we generate a random offset function $\tilde{f}(x) = \sum_i g_i \kappa(\tilde{x}_i, x)$ for random held-out data $\tilde{x}_i$ and $g_i \sim N(0, 1)$. Since $\tilde{f}$ is an RKHS element generated from random data, there is some hope in approximating it using the training data. Letting $\tilde{F}$ be $\tilde{f}$ applied to the training features, we

generate $M = p\tilde{F} + (1-p)U$ where $p = \sqrt{\text{prior error} \cdot \mathbb{V}(U)/\text{MSE}(U, \tilde{F})}$.

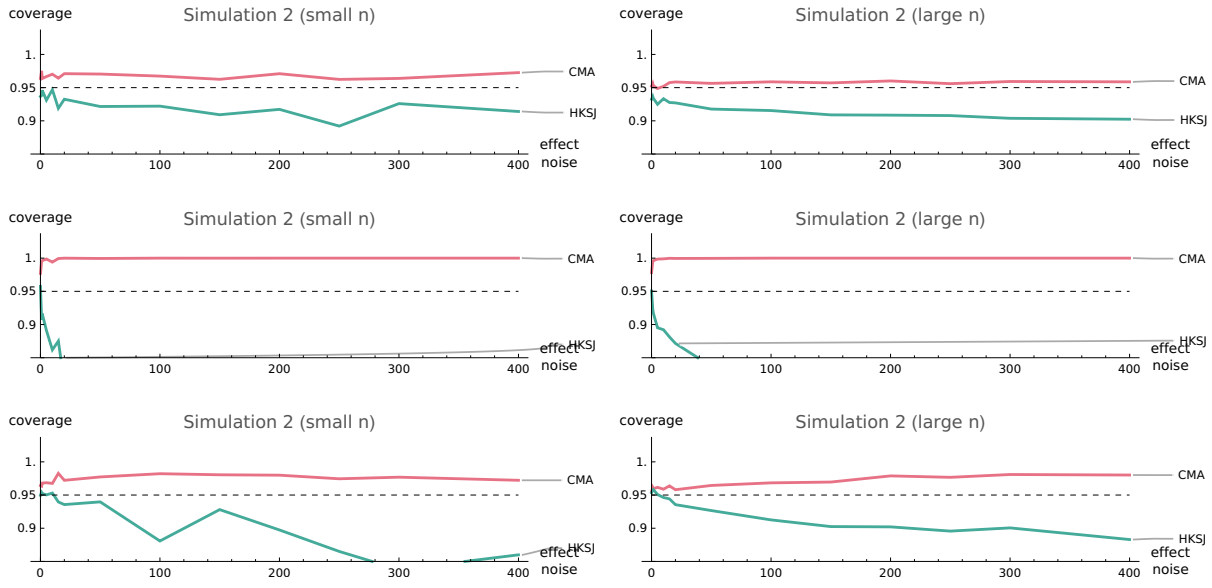
All simulations are averaged over 32 random splits. Intervals are computed for between 256 and 768 test data in each run. Due to the efficiency of our proposed algorithms, all experiments are capable of running on a free Google Colab instance.



Simulation 2.1: Rows are different datasets; the different columns, from left to right, set prior error equal to 3.0, 0.9, and 0.2, respectively. $\alpha = 0.1$ and effect noise $= 0.5$ were used.

2.7.9 Case Study Details

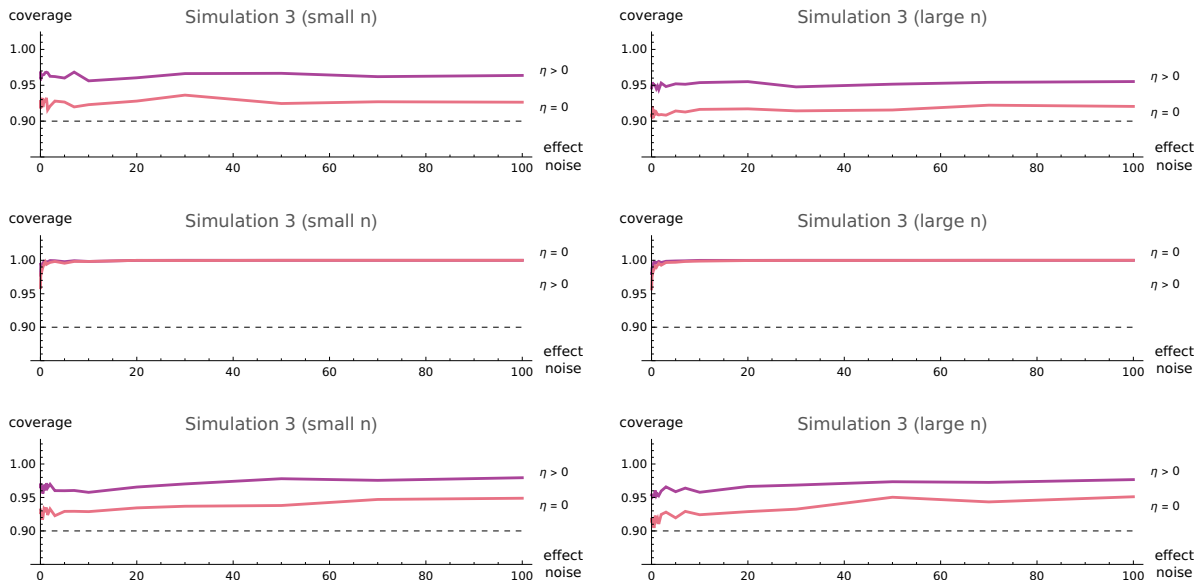
We follow the meta-analysis process illustrated in Figure 2.1. First, we determine the domain $\mathcal{X}$ of x. Helpfully, Letelier et al. [2003] identified 10 potentially-relevant features, such as mean age, mean AF duration, and amiodarone therapy protocol (e.g. "IV, 5 mg/kg in 30 min + 10 mg/kg in 20 h" or "Oral, 600 mg/d for 3 wk"). In order to extract these features from the trial, we give their published PDFs to a publicly-available language model, along with a prompt including example output. This extraction is fairly reliable, echoing the experience of Yun et al. [2024]. Next, parsing code (also written by the language model) converts the extracted textual features to numerical vectors x. As exemplified in Figure 2.7, this parsing can be tedious and error-prone, even with a state-of-the-art LLM. Our final predictions end up relying on just three features:



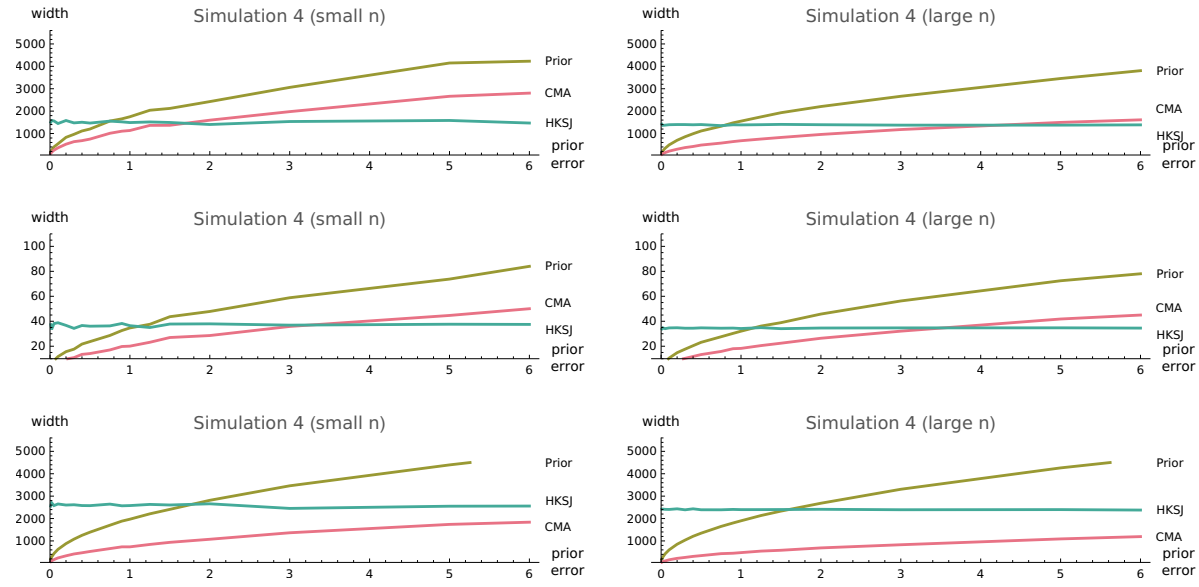
Simulation 2.2: Rows are different datasets. $n = 50$ and $n = 200$ are used in the left and right columns, respectively. prior error is set low to 0.2.

total amiodarone dosage in the first 24 hours, whether mean AF duration was above or below 48 hours, and the number of patients (which is a sensible feature when predicting trials rather than effects).

In lieue of a powerful pretrained foundation model, we base μ and κ on the critique of Slavik and Zed [2004]. They describe how multiple sources of heterogeneity, such as dosage, could impact the effect of amiodarone. Most importantly, amiodarone has a relatively slow course of action, whereas patients with recent-onset AF (usually defined as an AF duration of less than 48 hours) have a high chance of spontaneously reverting to normal sinus rhythm. (Letelier et al. [2003] also noted this pattern). With recent-onset AF, median spontaneous conversion rates are "11% at 2 hours after admission, 18% at 3 hours, 25% at 4 hours, 31% at 6 hours, 39% at 8 hours, 38% at 12 hours, 58% at 24 hours, and 67% at 48 hours.". This compares to only 0–8% within the first 72 hours for patients with persistent AF. We identify 8 further trials which compared amiodarone to an active comparison. We compute pseudo-effects (as relative risk) by taking the ratio of the observed probability of conversion under amiodarone, over the aforementioned estimated probability of spontaneous conversion over time. Such indirect comparison is reminiscent of how network meta-analysis works [Cipriani et al., 2013].



Simulation 2.3: Rows are different datasets; $n = 50$ and $n = 200$ are used in the left and right columns, respectively. $\alpha = 0.1$ and prior error $= 0.1$ were used.



Simulation 2.4: Rows are different datasets; $n = 16$ and $n = 200$ are used in the left and right columns, respectively. A low effect noise $= 0.02$ was set, along with $\alpha = 0.1$.

```
def precomputations(M, K, U, m, k, k_o, alpha):
  n = len(M)
  I = eye(n)
  I_ = eye(n+1)
  M_ = append(M,m)
  K_ = block([[K, k[:, newaxis]], [k, k_o]])
  lambda = amax(diag(K_))
  t_ = solve(K_/lambda + I_, M_)
  Q_ = solve(K_+lambda*I_, K_)
  Q = Q_[:-1,:-1]
  q = Q_[-1,:-1]
  q_o = Q_[-1,-1]

  A = -q
  a = 1-q_o
  B = U - Q@U - t_[:-1]
  b = -q@U - t_[-1]
  # a is already positive; flip signs (wlog) so that a,A_i >= 0
  B *= sign(A) + (A == 0)
  A *= sign(A)
  S2 = lambda*diag(Q)
  s2 = lambda*q_o
  D = square(I-Q) @ V
  d = square(q) @ V

  tau = ceil((1-alpha)*(n+1)).astype(int32)

  return Q,q,A,a,B,b,D,d,S2,s2,tau
```

Algorithm 2.3: Python / NumPy code for common linear-algebraic computations described in Section 2.3. In this code, and the code throughout the paper, some elisions and deoptimizations are made for readability. In particular, import statements are omitted.

```
Can you extract the following features from the attached PDF
paper? I gave example values, from another paper, which should
be replaced with the actual values in this paper. The only
relevant outcome is conversion to normal sinus rhythm. Also,
create a new key like "Results": [a, b, c, d] where a is the
number of amiodarone patients converted to sinus rhythm, b is
the total number of amiodarone patients, c is the number of
comparison patients converted to sinus rhythm, and d is the
total number of comparison patients. Answer as JSON.

{"Name": "Villani et al.11 (Italy) 2000", "Features": { "
Amiodarone Therapy Protocol": "Oral, 400 mg/d for 1 mo", "
Comparison Treatment": "Oral digoxin, 0.25 mg/d or oral
diltiazem hydrochloride 180- 360 mg/d for 1 mo", "Time to
Outcome Measure": "1 mo", "Number of Amiodarone Patients":
"44", "Number of Control Patients": "30", "Fraction with CV
Disease": "47", "Mean Left Atrium Size, mm": "50", "Mean AF
Duration": "17 wk", "Mean Age": "58", "Fraction Male": "67", "
Adequate Concealment of Treatment": "No", "Follow-up Fraction":
 "100", "Masked Patients": "Yes", "Masked Caregiver": "no", "
Masked Assessor": "no" }}
```

Figure 2.5: Prompt used to extract relevant data from trial PDFs.

```
In the attached JSON list, each element represents a study
described by the "Features" attribute. Convert these features
to real numbers so they can be provided to a learning algorithm
.

* amiodarone treatment should be the total dosage, in
milligrams, which is given over the first 24 hours. If the
dosage is specified per kg bodyweight, then take into account
the average bodyweight of the patients.

* comparison treatment should be converted to [0,1], where 0
denotes placebo and 1 an intensive, high dose comparison
regimen.

* if the fraction of male patients is unknown, just assume it
is 0.5.

* fraction with CV disease and followup fraction were reported
as integers, so for example 78 should be converted to 0.78.

* number of control and amiodarone patients should be just
copied over as integers

* mean AF duration and time to outcome measure should be
converted to -1 for <= 48 hours and 1 for > 48 hours

* mean left atrium size and mean age should be rescaled to
\[-1,1\] where 0 is the average of the feature, -1 is the
minimum, and 1 is the maximum

* the boolean features should be rescaled to \[-1, 1\], where
-1 means false, 1 means true, and 0 means not present or not
confident.

* include the same keys for all the studies, using the original
 key names.

Answer as JSON; no further explanation is necessary.
```

Figure 2.6: Prompt used to convert extracted data to numerical features.

```python
def parse_dosing_protocol(protocol):
    if protocol is None or protocol.lower() == 'not specified':
        return 0

    weight = 70  # Average body weight in kg
    total_mg = 0  # Initialize total milligrams

    # Normalize and break down the protocol into components
    protocol = protocol.lower().replace('over', 'in').replace('plus', ',')
    phases = protocol.split('+')

    for phase in phases:
        parts = phase.split(',')
        for part in parts:
            part = part.strip()
            tokens = part.split()
            dose = 0
            rate_based = False
            duration = 24  # Default duration is 24 hours unless specified

            # Parse the dose and units
            for i, token in enumerate(tokens):
                try:
                    # Attempt to convert token to float to find numeric values
                    potential_dose = float(token)

                    # Check for units immediately following the numeric value
                    if i + 1 < len(tokens):
                        unit = tokens[i + 1]
                        if 'g' in unit and 'mg' not in unit:
                            potential_dose *= 1000  # Convert grams to milligrams
                        elif 'mg/kg' in unit:
                            potential_dose *= weight  # Convert to total mg for given weight

                        # Determine if the dose is time-bound
                        if 'hour' in unit or 'h' in unit or 'min' in unit:
                            rate_based = True  # The dose is a rate per time
                            duration = extract_duration(part)
                            if 'min' in unit:
                                duration /= 60  # Convert minutes to hours
                    dose = potential_dose
                    break
                except ValueError:
                    continue  # Not a number, move to next token

            # Apply the dose calculation based on the duration and whether it's rate-based
            if rate_based:
                total_mg += min(duration, 24) * dose  # Apply the rate up to 24 hours
            elif 'day' in part:
                if 'first' in part or '1 day' in part or '1 week' in part:
                    total_mg += dose  # Apply if it specifies the first day or week
            else:
                total_mg += dose  # Single dose or calculated for the duration

    return total_mg
```

Figure 2.7: Python code generated by GPT-4 to parse and convert amiodarone therapy protocols. Generating this code required multiple rounds of interaction with the language model. This code still has mild bugs, which were intentionally left untouched.

2.8 Discussion

This chapter explores a synthesis between modern regression models trained on large quantities of untrusted data, and rigorous estimates of causal effect based on small amounts of trusted data. To researchers in machine learning, it is somewhat unsurprising that prior beliefs or inductive bias can be safely incorporated into learning algorithms; as discussed earlier in the chapter, a variety of statistical techniques enable this combination. In evidence-based medicine, however, such a synthesis between observational data and rigorous causal inference is both counterintuitive and remarkable. To successfully apply conformal prediction to this field, this chapter fundamentally advanced some core methodology in (full) conformal prediction. In particular, it shows that full conformal prediction can be fast and simple for a wide class of learning algorithms.

In my opinion, the results of this chapter are the most significant of this dissertation. In their seminal paper on random-effects meta-analysis, DerSimonian and Laird [1986] expressed hope for resolving heterogeneity by using features x. 35 years later, Bryan et al. [2021] declared that such a "heterogeneity revolution" had still not occurred. Conformal meta-analysis could help spark this revolution, but much further research is warranted. Improvements to our algorithms and their analyses seem possible. It should be possible to further relax our statistical assumptions, such as exchangeability [Barber et al., 2023]. By pairing our techniques with foundation models, we hope to answer important questions.

At a high level, this chapter follows a "wrapping" strategy: it ensconces a large, difficult-to-scrutinize model within an algorithm guaranteed to extract rigorous predictions. The following three chapters employ a different strategy: they attempt to improve large models from within, "swapping" out internal components to improve tractability, speed, and other factors.

Chapter 3

Differentiating Through Orthogonal Polynomial Transforms

Abstract

Every length-$(n+1)$ sequence of orthogonal polynomials is uniquely represented by two length-$(n+1)$ sequences of coefficients α and β. This chapter makes this representation learnable by gradient-based methods. This amounts to implementing differentiation algorithms for orthogonal polynomial operations. Automatic differentiation may be applied to such operations, but this uses $O(n^2)$ memory, is very slow in practice, and does not facilitate development of custom descent algorithms based on approximate gradients. By exploiting reversibility, we derive differentiation algorithms which use $O(n)$ memory and are much faster in practice. Using these algorithms, fixed polynomial transforms (e.g. discrete cosine transforms) can be replaced by learnable layers. These are more expressive, but they retain the computational efficiency and analytic tractability of orthogonal polynomials.

As another application, this chapter presents an algorithm for approximating the minimal value $f(w^*)$ of a general nonconvex objective f, without finding the minimizer w^*. It follows a scheme recently proposed by Lasserre [2020], whose core algorithmic problem is to find the sequence of polynomials orthogonal on a given probability distribution. Despite the general intractability of this problem, encouraging initial results are observed on some test cases. The fulcrum of this application is that positive measures, moment sequences, and orthogonal polynomial sequences correspond to one another; this chapter is the first work to explore the orthogonal polynomials as a parameterization for optimization, which may have computational advantages in future applications.

3.1 Introduction

Sequences $p_0, p_1, \ldots, p_n$ of univariate orthogonal polynomials — such as the Chebyshev, Laguerre, and Hermite sequences — are fundamental in many areas of scientific computing. Their core operation is evaluation (or transformation): given a polynomial f, typically represented as coefficients $c_0, \ldots, c_n$ in the basis $p_0, \ldots, p_n$, compute $f(x_i)$ at $n+1$ points $x = [x_0, \ldots, x_n]$.[1] The inverse operation is interpolation: given distinct points x_i and corresponding outputs y_i, compute the unique f satisfying $f(x_i) = y_i$. A less well-known operation, computing the Jacobian determinant $|\partial f(x)/\partial c|$, is useful in applications such as normalizing flows [Rezende and Mohamed, 2015].

Orthogonal polynomial sequences can be written recursively in terms of two length-$(n+1)$ sequences α and β, where both sequences are real, and the latter sequence is positive:

$$p_{-1}(x_i) = 0; \quad p_0(x_i) = 1; \qquad p_{j+1}(x_i) = (x_i - \alpha_j)p_j(x_i) - \beta_j p_{j-1}(x_i) \quad \text{for } \beta_j > 0 \tag{3.1}$$

The correspondence between the orthogonal polynomial sequences p and coefficients (α, β) is bijective. (A more formal statement is given in Section 3.2.) This chapter enables gradient-based optimization over the set of orthogonal polynomial sequences. We make (α, β) a learnable representation by deriving the gradients, with respect to (α, β), of the evaluation and interpolation operations. Compared to naive algorithms obtained by automatic differentiation, the hand-derived algorithms use $O(n)$ (rather than $O(n^2)$) memory, are much faster in practice, and enable development of approximate-gradient descent algorithms. This chapter explores the following two applications of this new optimization capability.

Learned polynomial transforms. Currently, orthogonal polynomial transforms are manually chosen based on intuitions about their suitability. Informally, Chebyshev polynomials resemble cosines; Laguerre polynomials are similar to cosines multiplied by exponentials; Hermite polynomials are roughly cosines divided by Gaussians [Valiant, 2016]. They are orthogonal with respect to different distributions μ over their inputs x; see Figure 3.1. All these transforms are generalized by the proposed learnable layer, which is called DXT. It has fast algorithms for its

[1] Zero-indexing vectors of length $n+1$ is a standard convention for polynomial transforms.

forward, backward, inverse, and log-determinant passes. DXT is empirically evaluated as a drop-in replacement for the DCT and IDCT within JPEG compression, where it is trained by gradient descent to improve image quality. On the CLIC dataset, this obtains better tradeoffs between visual quality and compression over standard JPEG. It may be possible to unobtrusively improve many signal processing pipelines in this manner.

Minimal values of optimization problems. This chapter presents an algorithm, called Mop, for approximating the minimal value $f(w^*)$ of a continuous function f, without finding its minimizer w^*. In recent work, Lasserre [2020] reduced this difficult problem to the following one: given sampling access to a distribution μ over $\mathbb{R}$, find the sequence of orthogonal polynomials (represented as coefficients α and β) which are orthogonal with respect to μ. We formulate this as an optimization problem which is amenable to stochastic gradient descent.

This chapter's overall contributions are: (1) memory-efficient algorithms, with fast CUDA implementations, for computing the vector-Jacobian products of evaluation and interpolation, (2) the learnable DXT layer, which demonstrates reduced error when applied to image compression, and (3) Mop, an algorithm for estimating the minimal value f^* of a continuous function f.

3.2 Preliminaries

The following background material is found in references on orthogonal polynomials [Gautschi, 2004, 2005, Ismail et al., 2005] and numerical linear algebra [Higham, 2002].

Measures and moments. Let μ be a measure over numbers $z \in \mathbb{R}$. (We will typically take z to be one of the coordinates x_i of the length-$(n+1)$ vector x.) The measure μ is positive if $\mu(Z) > 0$ for all nonempty open sets Z. Its moments are $m_k = \int z^k d\mu(z)$ for $k \geq 0$. These define a linear functional $L(p_j)$ over polynomials p_j via $L(z^k) = m_k$. Positive measures μ, moment sequences m whose Hankel matrices $[m_{i+j}]_{i,j}$ are positive definite, and positive linear functionals (satisfying $L(p_j) > 0$ for all nonnegative polynomials p_j) uniquely correspond to one another.

Orthogonal polynomials. A measure μ defines an inner product $\langle p_i, p_j \rangle = \int p_i(z) p_j(z) d\mu(z)$ over polynomials. p_i and p_j are μ-orthogonal if $\langle p_i, p_j \rangle = 0$. A sequence $[p_0, p_1, \ldots, p_n] = p$

Transform	x_i	α_k	$\beta = \gamma^2$	μ
Cosine-III	$\cos(\frac{\pi}{n+1})(i+\frac{1}{2})$	0	$\beta_0 = \pi, \beta_1 = \frac{1}{2}, \beta_k = \frac{1}{4}$	$2 \cdot \text{Beta}(\frac{1}{2}, \frac{1}{2}) - 1$
Legendre	[Bogaert, 2014]	0	$\beta_0 = 2, \beta_k = k^2/(4k^2-1)$	$\text{Uniform}(-1, 1)$
Hermite	[Press et al., 1992]	0	$\beta_0 = \sqrt{\pi}, \beta_k = k/2$	$\text{Normal}(0, \frac{1}{2})$
Laguerre	[Press et al., 1992]	$2k+1$	$\beta_0 = 1, \beta_k = k^2$	Exponential(1)

Figure 3.1: Parameters of classic orthonormal polynomial transforms, up to diagonal scaling. Cosine-III (short for the Discrete Cosine Transform, type III) is formed from the Chebyshev polynomials. The evaluation points x_i are taken to be the roots of p_{n+1}. They may not have a closed form, but can be calculated by the cited algorithms. Coefficient sequences may be obtained from Gautschi [2005], Leibon et al. [2008] and Chapter 4.5 of Press et al. [1992]. μ is the probability distribution which renders the polynomial sequence μ-orthogonal.

of polynomials is μ-orthogonal if p_i and p_j are μ-orthogonal when $i \neq j$. A polynomial of z is monic if its leading coefficient on z^n is 1. A polynomial q_i is orthonormal if its norm $||q_i|| = \sqrt{\langle q_i, q_i \rangle} = 1$. q is an orthonormal polynomial sequence if each q_i is orthonormal.

Three-term recurrence. For every positive μ, the sequence of μ-orthogonal monic polynomials satisfies the three-term recurrence (3.1) for some α and β. The orthonormal polynomial sequence $q_0, q_1, \ldots$ satisfies the following similar three-term recurrence, involving the coefficient sequences (α, γ):

$$q_{-1}(x_i) = 0; \quad q_0(x_i) = \gamma_0^{-1} \qquad \gamma_{j+1} q_{j+1}(x_i) = (x_i - \alpha_j) q_j(x_i) - \gamma_j q_{j-1}(x_i) \text{ for } \gamma_j > 0 \quad (3.2)$$

The orthonormal polynomial sequence q defined by (α, γ) corresponds to the monic polynomial sequence p defined by (α, β) where $\gamma_j = \sqrt{\beta_j}$ for $j > 0$. The Jacobi matrix, truncated to order n, organizes the coefficients (α, β) in the following $n \times n$ symmetric tridiagonal matrix:

$$J_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & 0 & 0 & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & 0 & 0 \\ 0 & \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix} \quad (3.3)$$

Favard's theorem.[2] Given any sequence of monic polynomials p defined by α and β in the recurrence 3.1, define a linear functional L over polynomials as follows: $L(p_0) = \beta_0$ and $L(p_j) = 0$ for $j > 0$. Then L is positive by construction. The theorem states that L corresponds to a positive measure μ for which p are μ-orthogonal, and that $L(p_j) = \mathbb{E}_{z\sim\mu} p_j(z)$.

Normalization. The squared Euclidean norm of a degree-k monic polynomial is $||p_k||^2 = \langle p_k, p_k\rangle = \prod_{j=0}^{k} \beta_j$. Neither the three-term recurrence (3.1) nor the Jacobi matrix involve β_0. This is because $\beta_0 = ||p_0||^2 = \int 1 d\mu(x)$ is the normalizing constant of μ. By fixing $\beta_0 = 1$, we restrict attention to probability distributions.

Basic operations and Vandermonde systems. The fundamental computational operations for orthogonal polynomials are listed in Figure 3.2. Evaluation and interpolation are the most commonly-used operations and are the primary subject of this chapter. Code for these operations is given in Figure 3.3. These operations can be viewed as matrix-vector multiplication and linear system solving, respectively, with a polynomial Vandermonde matrix V, which is defined below. Recalling that c and y are vectors for coefficients and output values, respectively:

$$Vc = y \quad \text{where} \quad V_{i,j} = p_j(x_i) \qquad \text{(indexed from zero)} \tag{3.4}$$

V generalizes the Vandermonde matrix from monomials to orthogonal polynomials. Its determinant is the same as the Vandermonde matrix: $\det(V) = \prod_{1\le i<j\le n}(x_j - x_i)$ [Barnett, 1975]. V is invertible when the x_i are distinct. The derivative Vandermonde matrix has entries $V'_{i,j} = p'_j(x_i)$; it will appear in the algorithms developed in this chapter.

Numerical stability. The Evaluate algorithm in Figure 3.3 (also called Clenshaw's algorithm) is known to be numerically stable [Smoktunowicz, 2002]. By contrast, Interpolate can be unstable for moderate n [Gohberg and Olshevsky, 1997, Higham, 1990]. Fortunately, various mitigations have been developed for this issue. Furthermore, the instability in Interpolate can be resolved entirely using program transformation techniques. This chapter does not encounter such numerical instability, but this important issue is nonetheless discussed further in the appendix.

[2]This is also by Shohat, Stone, etc., so it is also called the spectral theorem of orthogonal polynomials.

Operation	Inputs	Output	$O(n^2)$-time Algorithm	$O(n \log^2 n)$-time Algorithm
Evaluation	x, α, β, c	Vc	[Smith, 1965]	[Potts, 2003]
Interpolation	x, α, β, y	$V^{-1}y$	Dual in Higham [1988]	not available
Derivative Evaluation	x, α, β, c	$V'c$	[Smith, 1965]	not available
Transpose Multiply	x, α, β, c	$V^T c$	Appendix	[Driscoll et al., 1997]
Transpose Solving	x, α, β, y	$V^{-T}y$	Primal in Higham [1988]	[Bostan et al., 2010]
Determinant	x, α, β	$\det(V)$	Formula in Section 3.2	[Gohberg and Olshevsky, 1994]
Eigenvalues	α, β	$\lambda_i(J_n)$	[Dhillon et al., 2006]	[Coakley and Rokhlin, 2013]

Figure 3.2: Algorithms for orthogonal polynomials. All the inputs are length-$(n+1)$ vectors. The $O(n^2)$ algorithms can be parallelized. In particular, a version of Clenshaw's algorithm for evaluation takes $O(n)$ parallel time [Barrio, 2000]. Interpolate (the dual in Higham [1988]) takes $O(n)$ parallel time, since the inner loop over j can be run by $O(n)$ threads, each operating on three entries of c. General-purpose $O(n \log^2 n)$-time algorithms for interpolation and derivative evaluation are not available in the literature, to our knowledge.

3.3 Vector-Jacobian Product Algorithms

Let ℓ be a scalar loss function of some optimization variables (say, x) which is defined by a forward computation graph. In reverse-mode differentiation, also known as backpropagation, we seek to compute the gradient of ℓ with respect to x. To do so, we compute the gradient of ℓ with respect to intermediate expressions (say, y) in the computation graph. In the standard "overbar" notation [Baydin et al., 2017], the gradient of y is denoted by $\bar{y} = \frac{\partial \ell}{\partial y}$. Now, suppose the orthogonal polynomial evaluation operation $y = Vc$ is part of the computation graph. The evaluation points x are (some of) its inputs. During the forward pass, we compute y as usual; up to this point in the backwards pass, we compute $\bar{y}$. Let $\nabla_x y = \frac{\partial y}{\partial x}$ be the Jacobian matrix of y with respect to x. Then (by linearity of differentiation) it holds that $\bar{x} = \bar{y}^T \nabla_x y$. Rather than naively forming the matrix $\nabla_x y$, backpropagation directly implements the operation $(x, v) \mapsto v^T \nabla_x y$. Applying this at $(x, \bar{y})$ yields the desired result. To summarize: efficient reverse-mode differentiation amounts to efficient implementation of the vector-Jacobian product (VJP).

Vector-Jacobian products may be computed automatically from a forward computation graph.

procedure Evaluate(x, α, β, c)
  $u = [c_n, \ldots, c_n]$
  $v = [0, \ldots, 0]$
  for $k \in [n-1, \ldots, 0]$ **do**
    $\tau = u$
    $u = (x - \alpha_k) \cdot u$
      $\quad -\beta_{k+1} v + c_k$
    $v = \tau$
  return u

Figure 3.3: Algorithms for polynomial evaluation and interpolation. All the inputs are vectors in $\mathbb{R}^{n+1}$. Both algorithms return a vector in $\mathbb{R}^{n+1}$. Evaluate is the Clenshaw algorithm. Interpolate is the dual algorithm of Higham [1988]. DivDiffs(x, y) computes the first row of the table of divided differences of y with respect to x; see the appendix for its definition.

procedure Interpolate(x, α, β, y)
  $\delta = \text{DivDiffs}(x, y)$
  return ChangeBasis$(x, \alpha, \beta, \delta)$
procedure DivDiffs(x, y) (naive)
  $\delta = y$
  for $k \in [0, \ldots, n-1]$ **do**
    $\Delta = \delta_k$
    for $j \in [k+1, \ldots, n]$ **do**
      $\tau = \delta_j$
      $\delta_j = (\delta_j - \Delta)/(x_j - x_{j-k-1})$
      $\Delta = \tau$
  return δ
procedure ChangeBasis(x, α, β, δ)
  $c = \delta$
  $c_{n-1} \stackrel{+}{=} (\alpha_0 - x_{n-1}) c_n$
  for $k \in [n-2, \ldots, 0]$ **do**
    $c_k \stackrel{+}{=} (\alpha_0 - x_k) c_{k+1} + \beta_1 c_{k+2}$
    for $j \in [1, \ldots, n-2-k]$ **do**
      $c_{k+j} \stackrel{+}{=} (\alpha_j - x_k) c_{k+j+1} + \beta_{j+1} c_{k+j+2}$
    $c_{n-1} \stackrel{+}{=} (\alpha_{n-k-1} - x_k) c_n$
  return c

---

procedure $\overline{\text{Evaluate}}(x, \alpha, \beta, u, v, \bar{u})$
  $\bar{x} = \bar{u} \circ (V'c)$
  $\bar{c} = V^T \bar{u}$
  $\bar{v} = 0$
  $\bar{\alpha} = \bar{\beta} = [0, \ldots, 0]$
  for $k \in [0, \ldots, n-1]$ **do**
    $w = \frac{1}{\beta_{k+1}}(-u + (x - \alpha_k) \cdot v$
      $\quad + c_k)$
    $\bar{\alpha}_k = -\bar{u}^T v$
    $\bar{\beta}_{k+1} = -\bar{u}^T w$
    $u, v = v, w$
    $\tau = \bar{u}$
    $\bar{u} = \bar{v} + \bar{u} \cdot (x - \alpha_k)$
    $\bar{v} = -\tau \cdot \beta_{k+1}$
  return $\bar{x}, \bar{\alpha}, \bar{\beta}, \bar{c}$

Figure 3.4: These algorithms compute the vector-Jacobian products of Evaluate and Interpolate. In $\overline{\text{Evaluate}}$, u, v are the corresponding values computed during the forward pass, i.e. u is an alias of y.

procedure $\overline{\text{Interpolate}}(x, \alpha, \beta, y, c, \bar{c})$
  $\bar{y} = V^{-T} \bar{c}$
  $\bar{x} = -\bar{y} \circ (V'c)$
  $\bar{\alpha}, \bar{\beta} = [0, \ldots, 0]$
  for $k \in [0, \ldots, n-2]$ **do**
    $c_{n-1} \stackrel{+}{=} -(\alpha_{n-k-1} - x_k) c_n$
    $\bar{\alpha}_{n-k-1} \stackrel{+}{=} \bar{c}_{n-1} \cdot c_n$
    $\bar{c}_n = \bar{c}_{n-1} \cdot (\alpha_{n-k-1} - x_k)$
    for $j \in [n-2-k, \ldots, 1]$ **do**
      $c_{k+j} \stackrel{+}{=} -(\alpha_j - x_k) c_{k+j+1} - \beta_{j+1} c_{k+j+2}$
      $\bar{\alpha}_j \stackrel{+}{=} \bar{c}_{k+j} \cdot c_{k+j+1}$
      $\bar{\beta}_{j+1} \stackrel{+}{=} \bar{c}_{k+j} \cdot c_{k+j+2}$
      $\bar{c}_{k+j+2} \stackrel{+}{=} \bar{c}_{k+j} \cdot \beta_{j+1}$
      $\bar{c}_{k+j+1} \stackrel{+}{=} \bar{c}_{k+j} \cdot (\alpha_j - x_k)$
    $c_k \stackrel{+}{=} -(\alpha_0 - x_k) c_{k+1} - \beta_1 c_{k+2}$
    $\bar{\alpha}_0 \stackrel{+}{=} \bar{c}_k \cdot c_{k+1}$
    $\bar{\beta}_1 \stackrel{+}{=} \bar{c}_k \cdot c_{k+2}$
    $\bar{c}_{k+1} \stackrel{+}{=} \bar{c}_k \cdot (\alpha_0 - x_k)$
    $\bar{c}_{k+2} \stackrel{+}{=} \bar{c}_k \cdot \beta_1$
  $c_{n-1} \stackrel{+}{=} -(\alpha_0 - x_{n-1}) c_n$
  $\bar{\alpha}_0 \stackrel{+}{=} \bar{c}_{n-1} \cdot c_n$
  return $\bar{x}, \bar{\alpha}, \bar{\beta}, \bar{y}$

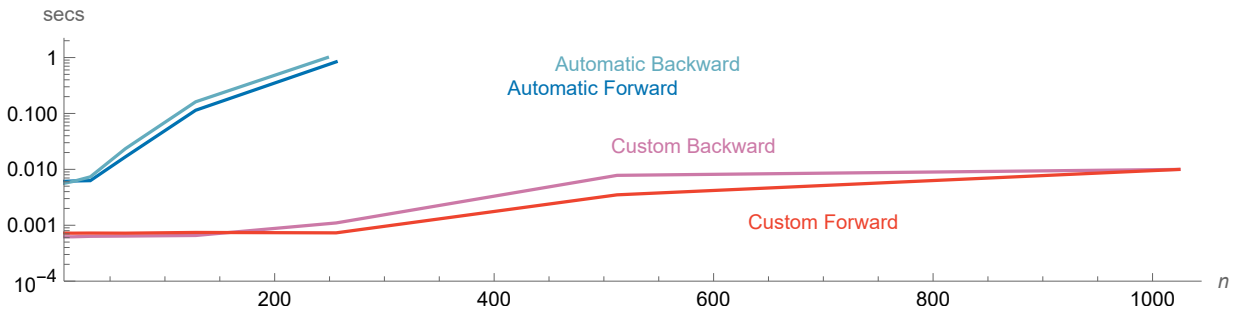


Figure 3.5: Runtime comparisons of automatic and custom VJP implementations. The latter is orders of magnitudes faster than the former, for both the forward and backward passes. The standard implementation has for loops specially expressed as structured control flow; presently, algorithms involving sparse updates within nested loops are an edge case for compilers. Python's `timeit` is used to perform timing, taking the best of 4 runs, each having 2 repetitions. A batch size of 32 is used.

However, this stores intermediate computations of the forward pass, which requires memory scaling with depth. Evaluate and Interpolate have depth $O(n)$, so $O(n^2)$ memory would be used. Furthermore, they involve sparse updates in nested loops, which are faster in lower-level CPU/GPU code than in higher-level XLA. This is because languages like XLA tend to be organized around bulk operations.

The appendix manually derives efficient VJPs for Evaluate and Interpolate. By exploiting reversibility, it reduces their memory requirements from $O(n^2)$ to $O(n)$. The complete algorithms are displayed in Figure 3.4. As illustrated in Figure 3.5, they are substantially faster than automatic differention, even when memory is not a concern. The appendix similarly derives an efficient VJP for orthonormal polynomial evaluation, called NEvaluate, which will be used in Section 3.5.

3.4 Learned Polynomial Transforms

Now we define the differentiable DXT layer, which encapsulates the Evaluate and Interpolate algorithms. Its arguments (which may be trainable parameters) are the evaluation points $x \in \mathbb{R}^{n+1}$ and the coefficients $\alpha, \beta \in \mathbb{R}^{n+1}$. Its forward pass is Evaluate and its inverse pass is Interpolate. It is invertible when the x_i are distinct. Its Jacobian is V, whose log-determinant is calculated by an algorithm cited in Figure 3.2. Finally, its backwards pass is given in Figure 3.4.

The following subsection applies DXT to the practical problem of image compression.

3.4.1 Learned JPEG

Perhaps the most widely-used orthogonal polynomial transform is the discrete cosine transform (DCT) within JPEG image compression [Wallace, 1992]. At a high level, JPEG converts RGB channels to one luma (brightness channel) and two chroma (color) channels, and operates on each channel separately. It splits the image into 8x8 patches and applies the 2D DCT, which is equivalent to the DCT applied to each column, followed by the DCT applied to each row. The transformed patches are quantized (rounded to the nearest integer) after pointwise division by an 8x8 quantization table, in which larger values correspond to less visually significant components. (The human visual system is less sensitive to high-frequency stimuli. Since the DCT expresses the patch as a combination of different-frequency components, the standard tables can readily discard high-frequency information.) Finally, the sequence of integers is losslessly compressed.

The quantization tables are learnable parameters of JPEG (among other compression methods). They can be learned by proxy objectives [Fung and Parker, 1995] or zero-order methods [Hopkins et al., 2018]. By replacing rounding with a smooth approximation, JPEG becomes differentiable [Shin and Song, 2017], allowing the quantization tables to be learned with first-order methods [Luo et al., 2020]. We investigate the additional benefit of replacing the DCT by DXT. To do so, we use a simple objective which (loosely) captures the tradeoff between visual distortion and compression rate, whose relative importance is set by $\lambda > 0$. To measure the former, we use PSNR, which is a normalized log-squared error. To measure the latter, it is typical to jointly train a measure of entropy, which can be used for the lossless compression step. Since we don't aim to replace this step, we retain the standard entropy measure; the objective linearly penalizes higher values of this measure.

We evaluate the DXT-based compression method on the CLIC 2020 dataset, from the eponymous image compression challenge [Toderici et al., 2020]. Adagrad is used as the optimizer. Learning rates of 0.2 and 2.0 performed best with and without DXT, respectively. The standard train/test split of CLIC is used. The batch size is 1. As shown in Figure 3.6, replacing DXT with the DCT consistently reduces error. These improvements are modest, but there are other

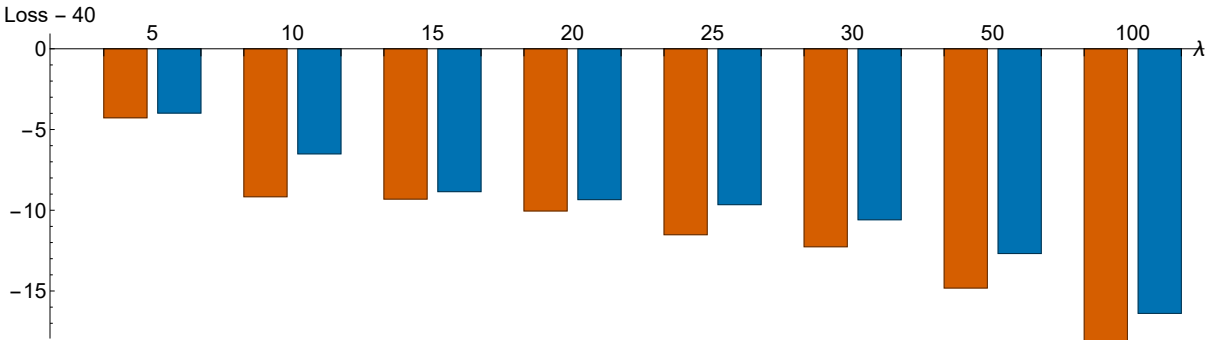


Figure 3.6: Learning both DXT and quantization tables (red) achieves lower loss than learning just the tables (blue). The difference is minimal at lower values of λ, where the target PSNRs are roughly 35-40. This is the regime where the standard JPEG tables are designed to operate. However, as λ increases (and the target PSNR decreases), DXT can express a substantially different transform.

advantages to keeping the compression pipeline mostly intact. The first is interpretability: the explanation of JPEG quantizing visually unimportant components remains valid. Another is ease of training: there are only 24 additional parameters to be trained on a multi-gigabyte dataset.

3.5 Minimal Values of General Optimization Problems

This application is presented merely as a showcase of this chapter's vector-Jacobian algorithms. Nonetheless, it is an ambitious problem with many potential applications.

3.5.1 Background

Let $W \subseteq \mathbb{R}^N$ be a compact set and let $f : W \mapsto \mathbb{R}$ be a continuous function which (for simplicity) attains its minimum over W. $f^* = \min_{w \in W} f(w)$ is that minimum value. Approximating f^* is NP-hard in general, but may be possible for some practically relevant f. (The minimal value f^* should not be confused with the minimizer $w^* \in \mathbb{R}^N$ achieving $f(w^*) = f^*$.) In many situations, such as nonconvex polynomial optimization [Laurent, 2009], determining f^* helps obtain w^*. Knowing f^* can also be useful in of itself. For example, knowing f^* could be practically useful for debugging model training. If a model is trained to have parameters $\tilde{w}$, and its

loss $f(\tilde{w})$ is much larger than f^*, then blame lies with the model's training, rather than its raw expressiveness.

Lasserre [2020] recently proposed the following approximation scheme for f^*. Suppose ρ is some probability distribution on W which (for simplicity) is absolutely continuous with respect to Lebesgue measure. We call it a prior since it is ideally concentrated around w^*. Then $\mu(X) = \rho(f^{-1}(X))$ is the distribution of $f(w)$, where X is a measurable subset of $\mathbb{R}$, and $f^{-1}(X) = \{w \in W : f(w) \in X\}$ is the preimage of X under f. Let α^* and β^* be the (unique) recurrence coefficients of the sequence of polynomials p orthogonal with respect to μ. Let λ be the minimum eigenvalue of J_n (or, equivalently, the smallest root of p_{n+1}). Lasserre [2020] shows that, as $n \to \infty$, $\lambda \to f^*$ from above. Laurent and Slot [2020] prove the rate of convergence to f^* is $O(\log^2 n/n^2)$ if W satisfies a mild geometric condition.

To implement this scheme, Lasserre [2020] observes that the smallest eigenvalue of J_n coincides with the smallest generalized eigenvalue of two $(n+1) \times (n+1)$ matrices formed from the moments $\mathbb{E}f(w)^k = \mathbb{E}x^k$. In particular, λ is the largest value satisfying $[\mathbb{E}x^{i+j+1}]_{i,j=0}^n \succeq \lambda[\mathbb{E}x^{i+j}]_{i,j=0}^n$. In principle, these moments can be computed if ρ can be sampled and f can be evaluated. However, unless f has special structure, the computation is hard for large n and N. In particular, the sample complexity of estimating the moments (i.e. the number of evaluations of f) is exponential in n.

3.5.2 Proposed Approach

The core problem in this scheme is: given sampling access to the distribution μ, find the unique sequence q^* of μ-orthonormal polynomials, represented by the coefficients (α^*, β^*). Let us design an optimization problem whose solution is (α^*, β^*), which we can then (attempt to) solve with gradient descent. By definition, the μ-orthonormal polynomials satisfy $I = \mathbb{E}_{z\sim\mu}[\, q_i^*(z)q_j^*(z)]_{i,j}$. Furthermore, since μ is a probability distribution rather than an unnormalized measure, we have that $q_0^*(z) = 1$, by the normalization discussion in Section 3.2. Recalling (3.2), this is accomplished by setting $\beta_0 = 1$. Let $r = [q_1(z), \ldots, q_n(z)]$ be the 'rest' of $q(z)$. By Favard's theorem (also stated earlier), $\mathbb{E}_{z\sim\mu}\, q_j^*(z) = 0$ for $j > 0$. Let the mean and covariance induced by the parameters (α, β) be $\sigma = \mathbb{E}_{z\sim\mu}\, r$ and $\Sigma = \mathbb{E}(r-\sigma)(r-\sigma)^T$. Then the optimal (α^*, β^*) induces

procedure Mop(f, ρ)
  Initialize α and β
  for $t = 1, \ldots$ **do**
    Sample $w_i \sim \rho$ and compute $x_i = f(w_i)$ for $i = 0, \ldots, n$
    $r_j = \text{NEvaluate}(x, \alpha, \sqrt{\beta}, e_j)$ for $j = 1, \ldots, n$
    Form estimates $\hat{\sigma}$ and $\hat{\Sigma}$ according to (3.5) using $r_1, \ldots, r_n$
    (Optional) Repeat above steps b times and average the estimates $\hat{\sigma}, \hat{\Sigma}$
    Update α, β with a gradient step on $\ell(\hat{\sigma}, \hat{\Sigma})$
    Constrain $\beta > 0$
  $\lambda =$ smallest eigenvalue of J_n as defined in (3.3)
  return λ

Figure 3.7: Mop aims to find the minimal value f^* of the function f over the support of the 'prior' distribution ρ. It does so by stochastically minimizing, over α and β, an objective $\ell(\hat{\sigma}, \hat{\Sigma})$, where σ and $\hat{\Sigma}$ are functions not only of α and β, but of evaluations of f upon draws from ρ. Once α and β are optimized, f^* is estimated by computing an eigenvalue of a matrix defined in terms of α and β. The loss function ℓ, batch size b, sequence length n, and regularization strength $\delta \geq 0$ are all parameters for Mop.

$\sigma = 0$ and $\Sigma = I$. Thus, we pick a loss $\ell(\sigma, \Sigma)$ which is minimized at those values. σ and Σ are expectations over μ, so they cannot be computed exactly, but they may be estimated from batches of evaluations. Consider the following (regularized) estimators:

$$\hat{\sigma} = \hat{\mathbb{E}}\, r \quad \text{and} \quad \hat{\Sigma} = (1 - \delta)\hat{\mathbb{E}}(r - \hat{\sigma})(r - \hat{\sigma})^T + \delta I \tag{3.5}$$

where $\delta > 0$ and $\hat{\mathbb{E}}$ is the batch mean. (The purpose of the regularization is to avoid ill-conditioned covariance estimates; Bessel's correction to the sample covariance may also be applied). When ℓ is convex, $\ell(\sigma, \Sigma) \leq \mathbb{E}\, \ell(\hat{\sigma}, \hat{\Sigma})$. The latter can be minimized by stochastic gradient descent, forming $\hat{\sigma}$ and $\hat{\Sigma}$ from each iteration's batch of x. This yields the Mop algorithm (Figure 3.7), so-named because it reveals the "floor" f^*, or at least abbreviates Minimal value of Optimization Problem.

Implementation. Mop is readily implemented using the evaluation operation for orthonormal polynomials (i.e. NEvaluate, as given by Figure 3.12 in the appendix). Let $x_0, \ldots, x_n \in \mathbb{R}$ be drawn iid from μ, let V be formed from these points, and let e_j be the jth coordinate vector. Then $Ve_j = [q_j(x_k)]_k$. Taking an average over the x_k yields an unbiased estimate $\hat{\sigma}$ of σ, as seen

by comparing to the definitions of r and σ above. Similarly, we can obtain an unbiased estimate $\hat{\Sigma}$ of Σ by averaging over the x_k:

$$\mathbb{E}_{x_k\sim\mu}\frac{1}{n+1}^T((Ve_i)\circ(Ve_j)) = \mathbb{E}_{x_k\sim\mu}\frac{1}{n+1}\sum_{k=0}^{n} q_i(x_k)q_j(x_k) = \mathbb{E}_{z\sim\mu}q_i(z)q_j(z)$$

So, to summarize, $\hat{\sigma}$ and $\hat{\Sigma}$ are obtained by calling NEvaluate, with a size-b batch of x, on $e_1, \ldots, e_n$.

When could Mop be better than estimating moments? As noted by Lasserre [2020], the moment matrix is just the identity matrix when expressed in the basis of orthonormal polynomials q^*. Since every positive-definite Hankel matrix corresponds to moments of a positive measure on $\mathbb{R}$, it is straightforward to constrain moment estimates to exactly the correct set. However, the moment sequence has high-degree, high-variance terms. The moment matrix is known to have condition number that increases exponentially with n. By contrast, NEvaluate is known to be backwards stable [Smoktunowicz, 2002]. (However, the stochasticity and nonconvexity of Mop's optimization may introduce numerical issues of their own.)

> *The high-level idea is to reparameterize an optimization over univariate moment matrices (i.e. symmetric positive-definite Hankel matrices) to an optimization over sequences of orthogonal polynomials, thereby eliminating the positive-definite constraint. This could have other applications.*

When could Mop be better than random sampling? Mop involves evaluating f at many random w_i. In general, Mop cannot be expected to achieve better performance than random sampling. However, it nontrivially ties these samples together by exploiting a crucial property of how α^* and β^* relate to μ. It might be possible for random samples to elucidate this relationship faster than they directly reveal f^*. Furthermore, if additional structure about μ is known, then it may be possible to derive more effective specializations of Mop.

Choice of ℓ. An obvious choice for ℓ is the squared Frobenius norm $\ell_{\mathrm{F}}(\hat{\sigma}, \hat{\Sigma}) = ||\hat{\sigma}||^2 + ||I - \hat{\Sigma}||^2$. In our experience, we encountered more success with $\ell_{\mathrm{KL}}(\hat{\sigma}, \hat{\Sigma}) = ||\hat{\sigma}||^2 + \mathrm{tr}\,\hat{\Sigma} - \log\det\hat{\Sigma} - (n+1)$, which is the Kullback-Leibler divergence between a standard normal distribution and

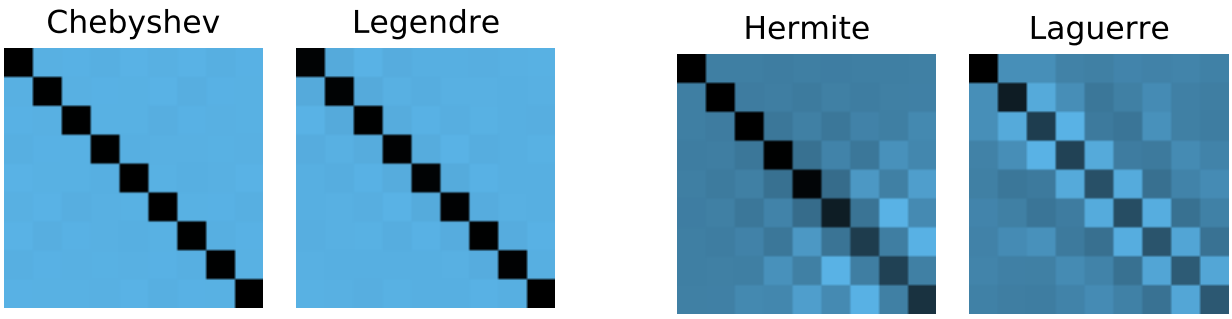


Figure 3.8: Plots of $\hat{\Sigma}$ for classical orthogonal polynomials. These are formed from $b \cdot n = 1024$ samples and the correct α, β, μ listed in Figure 3.1, so $\hat{\Sigma}$ should be close to identity. The Chebyshev and Legendre polynomials behave as desired. The Hermite and Laguerre polynomials suffer in the bottom-right entries involving higher-degree polynomials. This is because Hermite and Laguerre μ have unbounded support, along which high-degree polynomials quickly diverge. The Chebyshev and Legendre μ are supported on $[-1, 1]$. f should ideally be bounded.

$N(\hat{\sigma}, \hat{\Sigma})$. One possible explanation is that the sampling distributions of the estimators are normal. Another possibility is a low Jensen gap in $\mathbb{E}\, \ell_{\mathrm{KL}}(\hat{\Sigma}) = \mathrm{tr}(\Sigma) + \sum_i \mathbb{E} \log \lambda_i(\hat{\Sigma})$.

Stochastic optimization. Our approach extends beyond deterministic objectives f to the stochastic objectives which dominate machine learning [Srebro and Tewari, 2010]. In this setting, $\mathcal{D}$ is a distribution over examples z — for example, input-output pairs in supervised learning. $f_z(w)$ is typically the loss of weights w on example z. A stochastic objective $F(w) = \mathbb{E}_{z \sim \mathcal{D}} f_z(w)$ can be easily handled in Mop by combining the sampling of $z \sim \mathcal{D}$ and $w \sim \rho$ in μ. In this setting, μ would be the posterior loss (or negative log-likelihood), over the data distribution $\mathcal{D}$, of a Bayesian neural network with prior ρ over its parameters w.

Limitations and failure modes. As mentioned, approximating f^* is a computationally intractable problem. Furthermore, Mop is a statistical query algorithm [Kearns, 1998]: it uses gradient estimates computed from batches of data, and does not directly manipulate individual examples. It is therefore subject to stronger (information-theoretic) lower bounds than the previously-mentioned NP-hardness of minimal value approximation [Reyzin, 2020]. Accordingly, there are different ways in which Mop can fail or demand exorbitant resources. It is possible to encounter local minima or saddle points of ℓ with respect to α and β. The approximation of f^* by λ occurs asymptotically as n grows, so a large n may be required for some f. Larger n, in turn, demand more samples of f. $\lambda \to f^*$ from above for J_n defined by the optimal (α^*, β^*);

this is not necessarily true for suboptimal $(\tilde{\alpha}, \tilde{\beta})$ obtained by an inexact algorithm. Furthermore, Lasserre's result does not take finite-sample approximation of moments (or in our setting, means and covariances) into consideration.

3.5.3 Basic Empirical Evaluation

As a preliminary test, we apply the algorithm to f with known f^*. Some of these problems are easy, and Mop is able to solve them. One of them is impossible to solve, so Mop (of course) fails.

1. Recovering classical orthogonal polynomials. Mop defines μ in terms of f and ρ. Before doing that, let us simply take the μ listed in Figure 3.1, and see if minimizing $L(\hat{\sigma}, \hat{\Sigma})$ recovers the listed coefficients (α^*, β^*). As depicted in Figure 3.9, success depends on the choice of L and μ. In all cases, ℓ_{KL} achieved lower error than ℓ_{F}. The Chebyshev and Legendre coefficents were easier to recover than the Hermite and Laguerre ones. This agrees with the intuitions in Figure 3.8. In this experiment, $n = 8$, $b = 512$, and $\delta = 0$; the initialization is $\alpha_i = 0$ and $\beta_i = 1/2$.

2. Test polynomials. Lasserre's scheme has been run on 4 standard test polynomials [Lasserre, 2020]. These are bivariate ($N = 2$) and have minimum value $f^* = 0$ on $W = [-1, 1]^N$. We normalized the polynomials to take values in $[0, 1]$. To distinguish the algorithm's behavior from trivially returning 0, we added constants (either $0.2, 0.4, 0.6$, or 0.8), shifting the f^*. Figure 3.10 shows ℓ_{KL} reasonably approximates f^*, whereas the ℓ_{F} does not. Due to the normalization, our results are quantitatively different than those previously reported, but are qualitatively the same: the Matyas polynomial is the easiest, and the camel polynomial is the hardest [Laurent and Slot, 2020]. Note that λ produced midway during optimization are neither upper nor lower bounds of f^*; it is important to completely optimize α and β. In this experiment, $n = 8$, $b = 1024$, and $\delta = 0$. α was initialized with a Glorot normal and β by random $U(0, 1)$. The optimizer is RMSprop with learning rate 0.05 and gradient clipping.

3. Learning halfspaces / noisy parities. Let x be uniform on the hypercube $\{-1, 1\}^N$. For some noise rate $\eta > 0$, y is the parity of x with probability $1 - \eta$, and is negated with probability η. Let $f^* = \mathrm{argmin}_{w \in \mathbb{R}^N} - \mathbb{E} h_w(x) y$ be the (negated) correlation of the best possible (smooth) halfspace $h_w(x) = \tanh(w^T x)$. Using a statistical query algorithm, it is impossible to distinguish

f^* from zero, by the reduction of Kalai et al. [2008] to learning noisy parities [Blum et al., 1994]. For $N = 16$ and $\eta = 0$ we attempt to use Mop to approximate f^*. For varying values of n, we run Mop samples (x, y). For distinguishment, we run it on $(x, \tilde{y})$ where $\tilde{y}$ are random signs. In Figure 3.11, we see that Mop does not distinguish these distributions, regardless of n. In this experiment, $n = 8$, $b = 512$. Various optimizers (including SGD and RMSprop) were attempted, and none changed the (lack of) results.

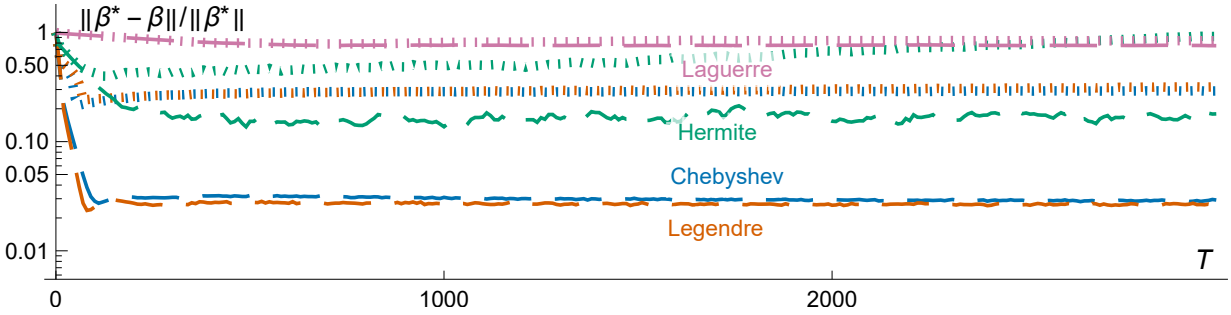


Figure 3.9: Mop recovering β^* for different classical polynomials. Thick dashed lines use ℓ_{KL}. Dotted lines use ℓ_{F}. β_0, β_n, and α_n are not in J_n, and so are not measured as part of the relative error. The error of α is not plotted because it follows the same pattern.

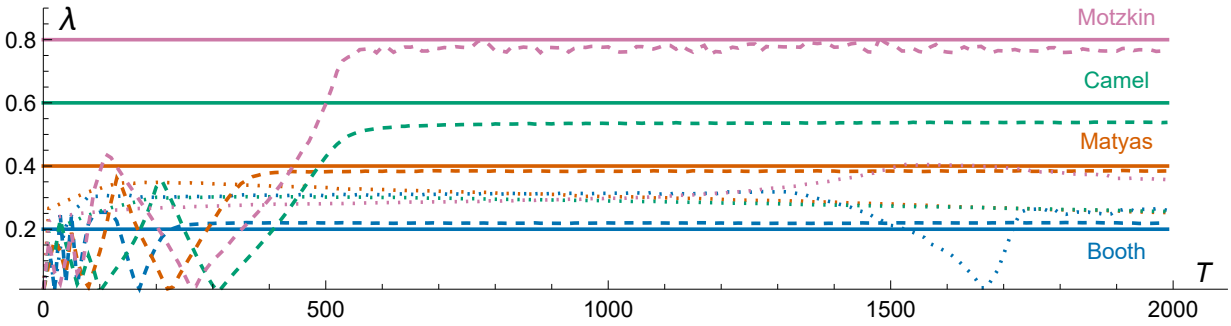


Figure 3.10: Mop on toy polynomials. Solid lines are true f^*, thick dashed lines are Mop using ℓ_{KL}, and dotted lines use ℓ_{F}. Mop using ℓ_{KL} approximates the f^*, but only after optimization is complete.

3.6 Related Work

Reversibility underpins low-memory, reverse-mode automatic differentiation [Gomez et al., 2017]. Checkpointing [Griewank and Walther, 2000] and tensor rematerialization [Jain et al., 2020] recompute selected intermediate values in the reverse pass, which reduces memory use at the cost

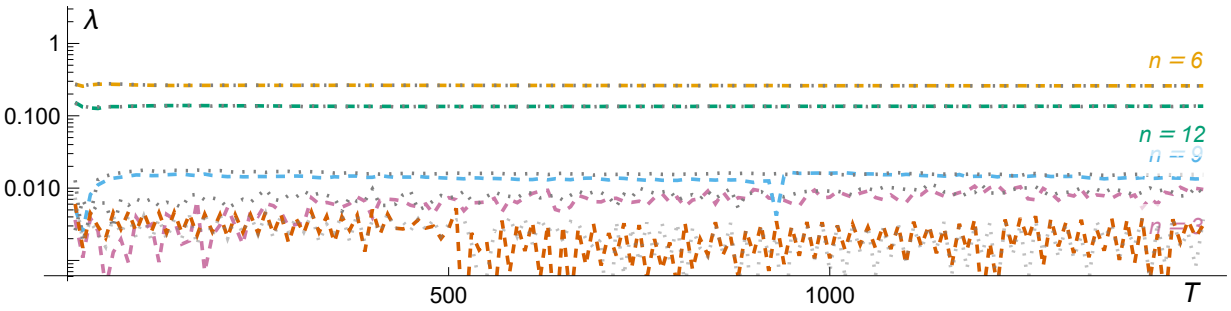


Figure 3.11: Mop fails to distinguish parity data (colored lines) from noise (gray lines), no matter the value of n.

of additional computation. Besides our manual derivation, there are other ways to obtain our VJPs, though they would require comparable effort, and may not be as practical. If an algorithm is written in a reversible programming language — which is not a trivial rewriting — then its VJPs can be computed with low memory overhead [Liu and Zhao, 2020]. Forward mode differentiation uses $O(n)$ memory, but generally requires a factor $O(n)$ more computation, and needs software support to intermix with reverse mode.

Learned image compression algorithms usually replace traditional compression pipelines with neural networks [Jiang, 1999]. They tend to achieve excellent compression results at the expense of computational complexity. Even as machine learning hardware becomes faster, there will likely be a role for fast, simple compression algorithms. This is amply demonstrated by the most powerful GPU ever released, which adds five dedicated cores for JPEG decoding [Lisiecki et al., 2020].

Orthogonal polynomial transforms are subsumed by recently-developed representations of structured linear maps, such as tridiagonal factorizations of low-displacement rank operators [Thomas et al., 2018], and butterfly factorizations culminating in the Kaleidoscope hierarchy [Dao et al., 2019, 2020]. It is appropriate to think of unstructured matrices, Kaleidoscope, DXT, and fixed polynomial transforms as varying tradeoffs between expressive power and tractability.

procedure NEvaluate(x, α, γ, c)
  $u = [c_n, \ldots, c_n]$
  $v = [0, \ldots, 0]$
  for $k \in [n-1, \ldots, 0]$ **do**
    $\tau = u$
    $u = ((x - \alpha_k)/\gamma_{k+1}) \cdot u$
      $-\frac{\gamma_{k+1}}{\gamma_{k+2}} v + c_k$
    $v = \tau$
  $\mu = u/\gamma_0$
  return μ

procedure NInterpolate(x, α, γ, y)
  $\delta = \text{DivDiffs}(x, y)$
  return NChangeBasis($x, \alpha, \gamma, \delta$)

procedure NChangeBasis($x, \alpha, \gamma, \delta$)
  $c = \delta$
  $c_{n-1} \stackrel{+}{=} (\alpha_0 - x_{n-1}) c_n$
  $c_n = \gamma_1 c_n$
  for $k \in [n-2, \ldots, 0]$ **do**
    $c_k \stackrel{+}{=} (\alpha_0 - x_k) c_{k+1} + \gamma_1 c_{k+2}$
    for $j \in [1, \ldots, n-2-k]$ **do**
      $c_{k+j} = \gamma_j c_{k+j} + (\alpha_j - x_k) c_{k+j+1}$
        $+\gamma_{j+1} c_{k+j+2}$
    $c_{n-1} = \gamma_{n-k-1} c_{n-1} + (\alpha_{n-k-1} - x_k) c_n$
    $c_n = \gamma_{n-k} c_n$
  $\sigma = \gamma_0 \cdot c$
  return σ

Figure 3.12: Variants of Evaluate and Interpolate for orthonormal polynomial sequences. As in Interpolate, DivDiffs is the first row of the table of divided differences.

Our results partially extend to the complex domain. The Fast Fourier Transform, perhaps the most well-known orthogonal polynomial transform, involves the monomials ($\alpha = \beta = 0$) of the roots of unity $x_i \in \mathbb{C}$. This can be handled by our algorithms, and indeed the original version of Interpolate [Björck and Pereyra, 1970]. The Szegö polynomials, which are orthogonal with respect to a Hermitian inner product on the unit circle, need a variant of Interpolate [Bella et al., 2007].

The problem of finding the orthogonal polynomials (as α, β) which match the given distribution (as moments m) was first studied by Chebyshev. The map $m \mapsto (\alpha, \beta)$ is ill-conditioned [Gautschi, 1967], so it is preferable to begin with "modified" moments [Gautschi, 2004]. In the context of time series, Gu et al. [2020] choose a measure μ over the past whose corresponding p is known. They approximate the observed history f in the basis of p. Our techniques could allow adaptive choise of μ and p.

procedure $\mathrm{V^T Multiply}(x, \alpha, \beta, c)$
$q = [0, \ldots, 0]$
$p = [1, \ldots, 1]$
$y = [0, \ldots, 0]$
for $i \in [0, \ldots, n]$ **do**
for $j \in [0, \ldots, n]$ **do**
$y_i \stackrel{+}{=} p_j \cdot c_j$
if $i < n$ **then**
$\tau = p_j$
$p_j = (x_j - \alpha_i)p_j - \beta_i q_j$
$q_j = \tau$
return y

procedure $\mathrm{NV^T Multiply}(x, \alpha, \gamma, c)$
$q = [0, \ldots, 0]$
$p = [\gamma_0^{-1}, \ldots, \gamma_0^{-1}]$
$y = [0, \ldots, 0]$
for $i \in [0, \ldots, n]$ **do**
for $j \in [0, \ldots, n]$ **do**
$y_i \stackrel{+}{=} p_j \cdot c_j$
if $i < n$ **then**
$\tau = p_j$
$p_j = ((x_j - \alpha_i)/\gamma_{i+1})p_j - \frac{\gamma_i}{\gamma_{i+1}} q_j$
$q_j = \tau$
return y

Figure 3.13: Transpose Vandermonde multiplication for monic orthogonal (left) and orthonormal (right) polynomial sequences. These use $O(n^2)$ time using $O(n)$ space. They are used in the following algorithms for VJPs.

3.7 Appendix

3.7.1 Numerical Stability of Interpolate

The divided differences of y at x are the following $\frac{1}{2}(n+1)(n+2)$ recursively-defined values:

$$y[x_i] = y_i \quad y[x_0, x_1] = \frac{y[x_1] - y[x_0]}{x_1 - x_0} \quad y[x_j, \ldots, x_k] = \frac{y[x_{j+1}, \ldots, x_k] - y[x_j, \ldots, x_{k-1}]}{x_k - x_j} \tag{3.6}$$

$\mathrm{DivDiffs}(x, y) = [y[x_0], y[x_0, x_1], \ldots, y[x_0, \ldots, x_n]] \in \mathbb{F}^{n+1}$ is called the first row of the table of divided differences. They are also known as the coefficients of the Newton interpolant of the data (x, y) [Berrut and Trefethen, 2004]. Interpolate starts by computing these coefficients δ. Then, it changes the basis of these coefficients to the specified orthogonal polynomial sequence. The blame for its instability lies with the first step: naive calculation of divided differences, according to the definition (3.6), can lead to severe numerical error.

This error can be ameliorated in some simple ways. The most appropriate way depends on the amount of control over the input data. If x can be chosen arbitrarily, then complex nodes can be more stable [Gautschi, 1990]. In particular, consider setting $x_k = e^{-2\pi i z_k}$ where z is

low-discrepancy sequence, such as the van der Corput sequence. Intuitively, such sequences will keep denominators $x_i - x_j$ close to uniform. If x is real, then stability strongly depends on the ordering of $x_0, \ldots, x_n$ [Gohberg and Olshevsky, 1997, Higham, 1990]. Choosing x randomly, as is typical when training a neural network, is a poor choice. If x is in increasing order, then divided differencing is (relatively) stable [Higham, 1987]. If some points are nonnegative, then the Leja ordering explicitly maximizes the relevant denominators $x_i - x_j$ [Higham, 1990]. If all these options are exhausted, iterative refinement can eliminate moderate amounts of error [Higham, 1991].

The aforementioned fixes may work in specific scenarios, but they do not satisfactorily solve the numerical instability of divided differencing. Fortunately, there is a principled, fairly-general solution to this problem. Divided differencing, like differentiation, is a composable program transformation [Reps and Rall, 2003]: given a program for f, a program computing the divided differences $x \mapsto f[x_0, \ldots, x_n]$ can be automatically derived. Existing software packages for machine learning support the implementation of such transformations [Bradbury et al., 2018]. It should be noted, however, that reverse-mode automatic divided differencing may not be as efficient as reverse-mode automatic differentiation. Furthermore, divided differencing is not quite as general as automatic differentiation, since branching conditions are not as easily handled [Vavasis, 2013].

> *Implementing divided differencing as a composable function transformation in Jax or Pytorch could have many other applications.*

Thus, the Interpolate algorithm of Higham [1988] is both simple and worthwhile in the long term. The Parker algorithm, which computes V^{-1} and then dense-multiplies $V^{-1}y$, is known to be more numerically stable [Calvetti and Reichel, 1993, Gohberg and Olshevsky, 1997]. However, it requires $O(n^2)$ memory. Specialized Gaussian elimination with partial pivoting [Kailath and Olshevsky, 1997] takes $O(n^2)$ time and $O(n)$ memory. However, its implementation is involved, and is asymptotically slower than Fourier-based methods. Some of the cited $O(n \log^2 n)$ algorithms in Figure 3.2 may also be numerically unstable; see, for example, Remark 4.2 in Bella et al. [2008].

3.7.2 Vector-Jacobian Products

The following algorithms have been numerically checked against finite differencing. As discussed in Section 3.3, We use standard "overbar" notation for adjoints [Baydin et al., 2017]. Let ℓ be the final scalar loss produced in the entire forward pass. The adjoint of each intermediate variable u_i is $\bar{u}_i = \frac{\partial \ell}{du_i}$.

Evaluate

By linearity, $\bar{c} = \bar{y}^T \nabla_c V c = (\bar{y}^T V)^T = V^T \bar{y}$. By similarly elementary operations:

$$\bar{x} = \bar{y}^T \left[\frac{dVc}{dx_k}\right]_k = \left[\sum_i \bar{y}_i \sum_j 1(i=k) p_j'(x_i) c_j\right]_k = \left[\bar{y}_k \sum_j p_j'(x_k) c_j\right]_k = \bar{y} \circ (V'c)$$

Now we derive the VJPs with respect to α and β. Here, on the left, is Evaluate written with indexed notation for u, which distinguishes the intermediate values.

procedure Evaluate(x, α, β, c)
  $u^{(n)} = [c_n, \ldots, c_n]$
  $u^{(n+1)} = [0, \ldots, 0]$
  for $k \in [n-1, \ldots, 0]$ **do**
    $u^{(k)} = (x - \alpha_k) \cdot u^{(k+1)} - \beta_{k+1} u^{(k+2)} + c_k$
  return $u^{(0)}$

procedure $\overline{\text{Evaluate}}$($x, \alpha, \beta, u^{(0)}, u^{(1)}, \bar{u}^{(0)}, \bar{u}^{(1)}$)
  $\bar{\alpha} = \bar{\beta} = [0, \ldots, 0]$
  for $k \in [0, \ldots, n-1]$ **do**
    $u^{(k+2)} = \frac{1}{\beta_{k+1}}(-u^{(k)} + (x - \alpha_k) \cdot u^{(k+1)} + c_k)$
    $\bar{\alpha}_k = -\bar{u}^{(k)} \cdot u^{(k+1)}$
    $\bar{\beta}_{k+1} = -\bar{u}^{(k)} \cdot u^{(k+2)}$
    $\bar{u}^{(k+1)} \mathrel{+}= \bar{u}^{(k)} \cdot (x - \alpha_k)$
    $\bar{u}^{(k+2)} = \bar{u}^{(k)} \cdot (-\beta_{k+1})$
  return $\bar{\alpha}, \bar{\beta}$

The adjoints are computed via standard reverse accumulation, as follows. Also, exploiting

reversibility, prior values $u^{(k+2)}$ can be computed from later values $u^{(k+1)}$.

$$\begin{aligned}
\bar{u}^{(k+1)} &= \bar{u}^{(k)} \cdot (x - \alpha_k) \\
\bar{u}^{(k+2)} &= \bar{u}^{(k)} \cdot (-\beta_{k+1}) \\
\bar{\alpha}_k &= \bar{u}^{(k)} \cdot (-u^{(k+1)}) \\
\bar{\beta}_{k+1} &= \bar{u}^{(k)} \cdot (-u^{(k+2)}) \\
u^{(k+2)} &= \frac{1}{\beta_{k+1}}(-u^{(k)} + (x - \alpha_k) \cdot u^{(k+1)} + c_k)
\end{aligned}$$

Performing these computations, in the reverse order of Evaluate, yields the VJP for α and β, as given above on the right. Note that $\bar{v} = 0$ since it is computed during the forward pass, but is not part of the output of Evaluate. The full VJP for Evaluate, given in Figure 3.4, includes the computation of $\bar{x}$ and $\bar{c}$, and elides the indexing of intermediate values.

Interpolate

It is known that $\bar{c}^T \nabla_y c$ is the solution $\bar{y}$ to $V^T \bar{y} = \bar{c}$ [Abadi et al., 2015]. Also, $\bar{c}^T \nabla_V c = -\bar{y} c^T$. By the chain rule, $\nabla_x c = \frac{\partial c}{dV} \circ \frac{\partial V}{dx}$. By the definition of V, $\frac{\partial V_{i,j}}{dx_k} = 1(i = k) p'_j(x_k)$. Therefore, similarly to the previous derivation for evaluation:

$$\begin{aligned}
\bar{x} = \bar{c}^T \nabla_x c &= \left[-\bar{y} c^T \circ \frac{dV}{dx_k} \right]_k = \left[\sum_{i,j} -\bar{y}_i c_j 1(i = k) p'_j(x_i) \right]_k = \left[\sum_j -\bar{y}_k c_j p'_j(x_k) \right]_k \\
&= -\bar{y} \circ \left[\sum_j c_j p'_j(x_k) \right]_k = -\bar{y} \circ (V' c)
\end{aligned}$$

For $\bar{\alpha}$ and $\bar{\beta}$, we write ChangeBasis in the indexed notation of Higham [1988].

procedure ChangeBasis(x, α, β, δ)

$c^{(n)} = \delta$

$c_{n-1}^{(n-1)} = c_{n-1}^{(n)} + (\alpha_0 - x_{n-1})c_n^{(n)}$

$c_n^{(n-1)} = c_n^{(n)}$

for $k \in [n-2, \ldots, 0]$ **do**

$c_k^{(k)} = c_k^{(k+1)} + (\alpha_0 - x_k)c_{k+1}^{(k+1)} + \beta_1 c_{k+2}^{(k+1)}$

for $j \in [1, \ldots, n-2-k]$ **do**

$c_{k+j}^{(k)} = c_{k+j}^{(k+1)} + (\alpha_j - x_k)c_{k+j+1}^{(k+1)} + \beta_{j+1}c_{k+j+2}^{(k+1)}$

$c_{n-1}^{(k)} = c_{n-1}^{(k+1)} + (\alpha_{n-k-1} - x_k)c_n^{(k+1)}$

$c_n^{(k)} = c_n^{(k+1)}$

return $c^{(0)}$

We derive the reverse-mode adjoints in the usual manner.

$$
\begin{aligned}
c_n^{(k)} &= c_n^{(k+1)} & \bar{c}_n^{(k+1)} &= \bar{c}_n^{(k)} \\
c_{n-1}^{(k)} &= c_{n-1}^{(k+1)} + (\alpha_{n-k-1} - x_k) c_n^{(k+1)} & \bar{c}_{n-1}^{(k+1)} &= \bar{c}_{n-1}^{(k)} \\
& & \bar{c}_n^{(k+1)} &= \bar{c}_{n-1}^{(k)} \cdot (\alpha_{n-k-1} - x_k) \\
& & \bar{\alpha}_{n-k-1} &\stackrel{+}{=} \bar{c}_{n-1}^{(k)} \cdot c_n^{(k+1)} \\
c_{k+j}^{(k)} &= c_{k+j}^{(k+1)} + (\alpha_j - x_k) c_{k+j+1}^{(k+1)} + \beta_{j+1} c_{k+j+2}^{(k+1)} & \bar{c}_{k+j}^{(k+1)} &= \bar{c}_{k+j}^{(k)} \\
& & \bar{c}_{k+j+1}^{(k+1)} &= \bar{c}_{k+j}^{(k)} \cdot (\alpha_j - x_k) \\
& & \bar{c}_{k+j+2}^{(k+1)} &= \bar{c}_{k+j}^{(k)} \cdot \beta_{j+1} \\
& & \bar{\alpha}_j &\stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot c_{k+j+1}^{(k+1)} \\
& & \bar{\beta}_{j+1} &\stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot c_{k+j+2}^{(k+1)} \\
c_k^{(k)} &= c_k^{(k+1)} + (\alpha_0 - x_k) c_{k+1}^{(k+1)} + \beta_1 c_{k+2}^{(k+1)} & \bar{c}_k^{(k+1)} &= \bar{c}_k^{(k)} \\
& & \bar{c}_{k+1}^{(k+1)} &= \bar{c}_k^{(k)} \cdot (\alpha_0 - x_k) \\
& & \bar{c}_{k+2}^{(k+1)} &= \bar{c}_k^{(k)} \cdot \beta_1 \\
& & \bar{\alpha}_0 &\stackrel{+}{=} \bar{c}_k^{(k)} \cdot c_{k+1}^{(k+1)} \\
& & \bar{\beta}_1 &\stackrel{+}{=} \bar{c}_k^{(k)} \cdot c_{k+2}^{(k+1)} \\
c_n^{(n-1)} &= c_n^{(n)} & \bar{c}_n^{(n)} &= \bar{c}_n^{(n-1)} \\
c_{n-1}^{(n-1)} &= c_{n-1}^{(n)} + (\alpha_0 - x_{n-1}) c_n^{(n)} & \bar{c}_{n-1}^{(n)} &= \bar{c}_{n-1}^{(n-1)} \\
& & \bar{c}_n^{(n)} &= \bar{c}_{n-1}^{(n-1)} \cdot (\alpha_0 - x_{n-1}) \\
& & \bar{\alpha}_0 &\stackrel{+}{=} \bar{c}_{n-1}^{(n-1)} \cdot c_n^{(n)}
\end{aligned}
$$

Observe that the reverse accumulation involves values $c^{(k)}$ for $k > 0$. As with Evaluate, we

can compute $c^{(k+1)}$ from $c^{(k)}$.

$$
\begin{aligned}
c_n^{(k)} = c_n^{(k+1)} &\iff c_n^{(k+1)} = c_n^{(k)} \\
c_{n-1}^{(k)} = c_{n-1}^{(k+1)} + (\alpha_{n-k-1} - x_k) c_n^{(k+1)} &\iff c_{n-1}^{(k+1)} = c_{n-1}^{(k)} - (\alpha_{n-k-1} - x_k) c_n^{(k+1)} \\
c_{k+j}^{(k)} = c_{k+j}^{(k+1)} + (\alpha_j - x_k) c_{k+j+1}^{(k+1)} + \beta_{j+1} c_{k+j+2}^{(k+1)} &\iff c_{k+j}^{(k+1)} = c_{k+j}^{(k)} - (\alpha_j - x_k) c_{k+j+1}^{(k+1)} - \beta_{j+1} c_{k+j+2}^{(k+1)} \\
c_k^{(k)} = c_k^{(k+1)} + (\alpha_0 - x_k) c_{k+1}^{(k+1)} + \beta_1 c_{k+2}^{(k+1)} &\iff c_k^{(k+1)} = c_k^{(k)} - (\alpha_0 - x_k) c_{k+1}^{(k+1)} - \beta_1 c_{k+2}^{(k+1)} \\
c_n^{(n-1)} = c_n^{(n)} &\iff c_n^{(n)} = c_n^{(n-1)} \\
c_{n-1}^{(n-1)} = c_{n-1}^{(n)} + (\alpha_0 - x_{n-1}) c_n^{(n)} &\iff c_{n-1}^{(n)} = c_{n-1}^{(n-1)} - (\alpha_0 - x_{n-1}) c_n^{(n)}
\end{aligned}
$$

Combining the adjoint equations with the reversed computation of $c^{(k)}$, we obtain the VJP with respect to α and β. The full algorithm in Figure 3.4 elides no-ops and the indexed notation.

procedure $\overline{\text{ChangeBasis}}(x, \alpha, \beta, \delta, c^{(0)}, \bar{c}^{(0)})$

$\bar{\alpha}, \bar{\beta} = [0, \ldots, 0]$

for $k \in [0, \ldots, n-2]$ **do**

$c_n^{(k+1)} = c_n^{(k)}$

$c_{n-1}^{(k+1)} = c_{n-1}^{(k)} - (\alpha_{n-k-1} - x_k) c_n^{(k+1)}$

$\bar{\alpha}_{n-k-1} \stackrel{+}{=} \bar{c}_{n-1}^{(k)} \cdot c_n^{(k+1)}$

$\bar{c}_n^{(k+1)} = \bar{c}_{n-1}^{(k)} \cdot (\alpha_{n-k-1} - x_k)$

$\bar{c}_{n-1}^{(k+1)} = \bar{c}_{n-1}^{(k)}$

for $j \in [n-2-k, \ldots, 1]$ **do**

$c_{k+j}^{(k+1)} = c_{k+j}^{(k)} - (\alpha_j - x_k) c_{k+j+1}^{(k+1)} - \beta_{j+1} c_{k+j+2}^{(k+1)}$

$\bar{\alpha}_j \stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot c_{k+j+1}^{(k+1)}$

$\bar{\beta}_{j+1} \stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot c_{k+j+2}^{(k+1)}$

$\bar{c}_{k+j+2}^{(k+1)} \stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot \beta_{j+1}$

$\bar{c}_{k+j+1}^{(k+1)} \stackrel{+}{=} \bar{c}_{k+j}^{(k)} \cdot (\alpha_j - x_k)$

$\bar{c}_{k+j}^{(k+1)} = \bar{c}_{k+j}^{(k)}$

$c_k^{(k+1)} = c_k^{(k)} - (\alpha_0 - x_k) c_{k+1}^{(k+1)} - \beta_1 c_{k+2}^{(k+1)}$

$\bar{\alpha}_0 \stackrel{+}{=} \bar{c}_k^{(k)} \cdot c_{k+1}^{(k+1)}$

$\bar{\beta}_1 \stackrel{+}{=} \bar{c}_k^{(k)} \cdot c_{k+2}^{(k+1)}$

$\bar{c}_k^{(k+1)} = \bar{c}_k^{(k)}$

$\bar{c}_{k+1}^{(k+1)} \stackrel{+}{=} \bar{c}_k^{(k)} \cdot (\alpha_0 - x_k)$

$\bar{c}_{k+2}^{(k+1)} \stackrel{+}{=} \bar{c}_k^{(k)} \cdot \beta_1$

$c_n^{(n)} = c_n^{(n-1)}$

$c_{n-1}^{(n)} = c_{n-1}^{(n-1)} - (\alpha_0 - x_{n-1}) c_n^{(n)}$

$\bar{\alpha}_0 \stackrel{+}{=} \bar{c}_{n-1}^{(n-1)} \cdot c_n^{(n)}$

$\bar{c}_n^{(n)} = \bar{c}_n^{(n-1)}$

$\bar{c}_n^{(n)} = \bar{c}_{n-1}^{(n-1)} \cdot (\alpha_0 - x_{n-1})$

$\bar{c}_{n-1}^{(n)} = \bar{c}_{n-1}^{(n-1)}$

return $\bar{\alpha}, \bar{\beta}$

NEvaluate

In the notation of Higham [1990], the three-term recurrence is:

$$p_{-1}(x) = 0; p_0(x) = 1; \quad p_{j+1}(x) = \theta_j(x - \beta_j)p_j(x) - \gamma_j p_{j-1}(x)$$

Their θ_j, β_j, and γ_j correspond to our $1/\gamma_{j+1}, \alpha_j/\gamma_{j+1}$, and γ_j/γ_{j+1}, respectively. Under their normalization $p_0(x) = 1$, suppose the solution of their algorithm is $\tilde{c}$. Then the solution under $p_0(x) = \gamma_0^{-1}$, as in the orthonormal polynomials, is $\gamma_0 \cdot \tilde{c}$. Following the template of the previous VJP derivations, here is NEvaluate written in indexed notation.

procedure NEvaluate(x, α, γ, c)

$u^{(n)} = [c_n, \ldots, c_n]$

for $k \in [n-1, \ldots, 0]$ **do**

$u^{(k)} = ((x - \alpha_k)/\gamma_{k+1}) \cdot u^{(k+1)} - \frac{\gamma_{k+1}}{\gamma_{k+2}} u^{(k+2)} + c_k$

$\mu = u^{(0)}/\gamma_0$

return μ

The adjoints and u are derived as before.

$$\begin{aligned}
\bar{u}^{(0)} &= \bar{\mu}/\gamma_0 \\
u^{(0)} &= \mu \cdot \gamma_0 \\
\bar{\gamma}^{(0)} &= -\bar{\mu} \cdot u^{(0)}/\gamma_0^2 \\
\bar{u}^{(k+1)} &\stackrel{+}{=} \bar{u}^{(k)} \cdot (x - \alpha_k)/\gamma_{k+1} \\
\bar{u}^{(k+2)} &= \bar{u}^{(k)} \cdot (-\frac{\gamma_{k+1}}{\gamma_{k+2}}) \\
\bar{\alpha}_k &= \bar{u}^{(k)} \cdot (-u^{(k+1)}/\gamma_{k+1}) \\
\bar{\gamma}_{k+1} &= \bar{u}^{(k)} \cdot (-(x - \alpha_k)u^{(k+1)}/\gamma_{k+1}^2 - u^{(k+2)}/\gamma_{k+2}) \\
\bar{\gamma}_{k+2} &= \bar{u}^{(k)} \cdot (\gamma_{k+1} u^{(k+2)}/\gamma_{k+2}^2) \\
u^{(k+2)} &= \frac{\gamma_{k+2}}{\gamma_{k+1}}(-u^{(k)} + ((x - \alpha_k)/\gamma_{k+1}) \cdot u^{(k+1)} + c_k)
\end{aligned}$$

Computing these quantities in reverse order yields the VJP algorithm, as follows on the left. The final algorithm, on the right, elides no-ops and indexing.

procedure $\overline{\text{NEvaluate}}(x, \alpha, \gamma, \mu, u^{(1)}, \bar{\mu}, \bar{u}^{(1)})$

$\bar{\alpha} = \bar{\gamma} = [0, \ldots, 0]$

$u^{(0)} = \mu \cdot \gamma_0$

$\bar{u}^{(0)} = \bar{\mu}/\gamma_0$

$\bar{\gamma}_0 = -\bar{\mu} \cdot u^{(0)}/\gamma_0^2$

for $k \in [0, \ldots, n-1]$ **do**

$u^{(k+2)} = \frac{\gamma_{k+2}}{\gamma_{k+1}}(-u^{(k)} + ((x - \alpha_k)/\gamma_{k+1}) \cdot u^{(k+1)} + c_k)$

$\bar{\alpha}_k = -\bar{u}^{(k)} \cdot (u^{(k+1)}/\gamma_{k+1})$

$\bar{\gamma}_{k+1} \stackrel{+}{=} -\bar{u}^{(k)} \cdot ((x-\alpha_k)u^{(k+1)}/\gamma_{k+1}^2 + u^{(k+2)}/\gamma_{k+2})$

$\bar{\gamma}_{k+2} = \bar{u}^{(k)} \cdot (\gamma_{k+1}u^{(k+2)}/\gamma_{k+2}^2)$

$\bar{u}^{(k+1)} \stackrel{+}{=} \bar{u}^{(k)} \cdot (x - \alpha_k)/\gamma_{k+1}$

$\bar{u}^{(k+2)} = \bar{u}^{(k)} \cdot (-\frac{\gamma_{k+1}}{\gamma_{k+2}})$

return $\bar{\alpha}, \bar{\gamma}$

procedure $\overline{\text{NEvaluate}}(x, \alpha, \gamma, \mu, v, \bar{\mu})$

$\bar{v} = 0$

$\bar{\alpha} = \bar{\gamma} = [0, \ldots, 0]$

$u = \mu \cdot \gamma_0$

$\bar{u} = \bar{\mu}/\gamma_0$

$\bar{\gamma}_0 = -\bar{\mu}^T u/\gamma_0^2$

for $k \in [0, \ldots, n-1]$ **do**

$\bar{\alpha}_k = -\bar{u}^T v/\gamma_{k+1}$

$\bar{\gamma}_{k+1} \stackrel{+}{=} -\bar{u}^T(x - \alpha_k) \cdot v/\gamma_{k+1}^2$

if $k < n-1$ **then**

$w = \frac{\gamma_{k+2}}{\gamma_{k+1}}(-u + \frac{x-\alpha_k}{\gamma_{k+1}} \cdot v + c_k)$

$\bar{\gamma}_{k+1} \stackrel{+}{=} -\bar{u}^T w/\gamma_{k+2}$

$\bar{\gamma}_{k+2} = \bar{u}^T w \cdot \gamma_{k+1}/\gamma_{k+2}^2$

$u, v = v, w$

$\tau = \bar{u}$

$\bar{u} = \bar{v} + \bar{u} \cdot (x - \alpha_k)/\gamma_{k+1}$

$\bar{v} = -\tau \cdot \frac{\gamma_{k+1}}{\gamma_{k+2}}$

return $\bar{\alpha}, \bar{\gamma}$

3.8 Discussion

This chapter explores a synthesis between modern gradient-based optimization and classical sequences of orthogonal polynomials. The key observation is that the most computationally convenient representation of a sequence of orthogonal polynomials consists of the coefficients of its three-term recurrence. By enabling backpropagation for polynomial evaluation and interpolation — that is, by deriving the vector-Jacobian products of these algorithms — we can use

orthogonal polynomials to parameterize (or reparameterize) a variety of contemporary learning and optimization problems. Based on the algorithm of Bella et al. [2009], our core techniques might extend from polynomial Vandermonde matrices to quasiseparable matrices. This chapter establishes basic technical underpinnings in the hope that they may be used in future, larger-scale applications.

This chapter focused on expanding the expressivity of fixed polynomial transforms, or exactly matching the feasible region of some optimization problems. The next chapter is different: it approximates (or slightly reduces) the expressivity of a neural network layer, while considerably improving its speed and tractability.

Chapter 4

Linear Dynamical Systems for Sequence Modeling

Abstract

Running nonlinear RNNs for T steps takes $\Omega(T)$ time, even with parallel execution. This chapter's construction, called LDStack, approximately runs them in $O(\log T)$ parallel time, and obtains arbitrarily low error via repetition. Though this specific construction is not a state-of-the-art sequence-to-sequence model, many of the architectural techniques can be useful for designing future models. The most interesting technique is replacing nonlinearity across time with nonlinearity along depth, through a provably-consistent scheme of local corrections. This allows nonlinear RNNs to be approximated by a stack of multiple-input, multiple-output (MIMO) linear dynamical systems (LDS). Next, this chapter shows that MIMO LDS can be approximated by an average or a concatenation of single-input, multiple-output (SIMO) LDS. Finally, this chapter presents an algorithm for running (and differentiating) SIMO LDS in $O(\log T)$ parallel time. On long sequences, LDStack is much faster than traditional RNNs, yet it achieves similar accuracy in initial experiments. Furthermore, LDStack is amenable to linear systems theory. Therefore, it improves not only speed, but also mathematical tractability. This chapter is based on the published work of Kaul [2020].

4.1 Introduction

Nonlinear RNNs have two crucial shortcomings. The first is computational: running an RNN for T steps is a sequential operation which takes $\Omega(T)$ time. The second is analytical: it is challenging to gain intuition about the behavior of a nonlinear RNN, and even harder to prove this behavior is desirable. These shortcomings have motivated practitioners to abandon RNNs altogether and to model time series by other means. These include hierarchies of (dilated) convolutions [Gehring et al., 2017, Oord et al., 2016] and attention mechanisms which are differentiable analogues of key-value lookups [Bahdanau et al., 2014, Vaswani et al., 2017]. In these models, the underlying parallel primitives are convolution and matrix multiplication, respectively.

This chapter addresses both of these shortcomings. It presents a method to approximately run and differentiate nonlinear RNNs in $O(\log T)$ parallel time, by rebuilding them from linear dynamical systems (LDS). In these, the next state $s_{t+1} = As_t + Bx_t$ is a linear function of the current state s_t and input x_t. LDS are a mainstay of control theory and many engineering applications because their behavior can be understood and regulated [Zhou et al., 1996]. Single-input, multiple-output (SIMO) LDS, which map a sequence of input numbers to a sequence of output vectors, are our core primitive. Using parallel scans, these can be run and differentiated in $O(\log T)$ parallel time [Blelloch, 1990].

Summary of Main Ideas. This chapter's approach is to (1) approximate the RNN by a stack of multiple-input, multiple output (MIMO) LDS, then (2) approximate the MIMO LDS by an aggregation of single-input, multiple-output (SIMO) LDS, and finally (3) run the SIMO LDS in $O(\log T)$ parallel time using scans and reductions. In step (1), we take the LDS, measure the deviations of its linear steps from desired nonlinear ones, and add those as corrections to the LDS in the subsequent layer. This scheme is naturally parallel, since the corrections are based on only local information; surprisingly, it is provably consistent. A multiplicative variant has already been extensively used to analyze nonlinear, continuous-time dynamical systems [Tomás-Rodríguez and Banks, 2010].

For step (2), we consider two kinds of aggregation: averaging and concatenation. The averaging approach uses a standard technique in randomized numerical linear algebra: the d-dimensional inputs x_t are repeatedly, randomly projected to a single dimension. The concate-

nation approach pre-applies a decoupling $d \times d$ transformation to the inputs. Then, the inputs are given to d coupled SIMO LDS, each of size n/d. This approach builds upon the canonical form of Luenberger [1967], which decomposes the MIMO LDS into smaller SIMO LDS, whose sizes are called the *controllability indices* of the MIMO system. Unfortunately, these quantities are onerous to estimate or to even compute. Using a perturbed Luenberger form, we show that a uniform size n/d may be used with essentially no loss in generality.

Finally, step (3) exploits the linear-algebraic structure of SIMO LDS. It is known that linear recurrences $s'_{t+1} = \lambda \circ s'_t + b_t$, which involve entrywise multiplication $\circ$, can be run in $O(n \log T)$ parallel time via scans and reductions. A SIMO LDS can be taken to this form via diagonalization, i.e. by running the LDS in the basis of its eigenvectors. When the SIMO LDS is in a canonical form, its eigenvectors have closed-form expressions in terms of its eigenvalues. Accordingly, the set of SIMO LDS is exactly parameterized by just n numbers, which are provided to the recurrence solver.

Outline. This chapter presents its key contribution — the LDStack layer, which can replace nonlinear RNNs — in a bottom-up fashion. Section 4.2 reviews background material on linear dynamical systems. Section 4.3 presents parameterizations of SIMO LDS, and an algorithm for running them in $O(\log T)$ parallel time. Section 4.4 presents two ways that SIMO LDS can be combined to effectively replace MIMO LDS in machine learning applications. Section 4.5 shows that nonlinear RNNs can be approximated by stacks of MIMO LDS with nonlinearities interspersed solely along depth. LDS and LDStack are empirically evaluated on artificial and real datasets. LDS achieve state-of-the-art performance on the copy memory problem. LDStack can be substantially faster than traditional RNNs, while achieving competitive accuracy. Ways to improve these constructions, and incorporate them into future work, are discussed in Section 4.8.

4.2 Linear Dynamical Systems

Linear dynamical systems have enjoyed a renaissance in machine learning theory. There have been many recent advances in algorithms for learning LDS from input-output data [Hardt et al., 2016a, Oymak and Ozay, 2019, Sarkar and Rakhlin, 2019, Simchowitz et al., 2019]. The sample

complexity of this task is well-studied [Jedra and Proutiere, 2019, Simchowitz et al., 2018]. As analytical testbeds, they capture the behavior of optimization algorithms [Lessard et al., 2016] and establish baseline performance for reinforcement learning [Matni et al., 2019, Recht] and online learning [Ghai et al., 2020, Hazan et al., 2017, Kozdoba et al., 2019]. Efficient and robust algorithms have recently been developed for controlling LDS [Dean et al., 2019, Hazan et al., 2020].

This section reviews some basic material about LDS. At time $t \in [T]$, let the input be $x_t \in \mathbb{R}^d$. Starting from an initial state $s_0 \in \mathbb{R}^n$, an LDS produces subsequent states s_{t+1}:

$$s_{t+1} = As_t + Bx_t = A^{t+1}s_0 + \sum_{\tau=0}^{t-1} A^{\tau+1}Bx_{t-\tau} \qquad y_t = Cs_t + Dx_t + D_0 \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times d}$. By recursively unrolling the first equality, we see the states are a convolution of the inputs (with an infinite kernel size and only one stride dimension). Outputs $y_t \in \mathbb{R}^m$ may be optionally produced, using $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times d}$, and $D_0 \in \mathbb{R}^m$.

4.2.1 SIMO Canonical Form

An LDS is *reachable*, roughly speaking, if we can take it to any state by supplying the right input.

Definition 3 (Reachability)**.** *A state $s \in \mathbb{R}^n$ is reachable if there is a sequence of inputs $x_1, \ldots, x_T$ which leads to $s_T = s$. An LDS is reachable if every state $s \in \mathbb{R}^n$ is reachable* [1].

Lemma 6 (Hautus)**.** *An LDS is reachable iff A is nonsingular and, for all $\gamma \in \mathbb{C}$, the $n \times (n+d)$ matrix $[\gamma I - A; B]$ has full rank n.*

A reachable SIMO LDS $(\tilde{A}, \tilde{B}, \tilde{C}, D)$ is placed in canonical form (A, B, C, D) by $\mathcal{T} \in$

[1] In continuous time, reachability and controllability are equivalent. In discrete time, they are equivalent when A is nonsingular.

$\mathbb{R}^{n\times n}$:

$$A = \mathcal{T}\tilde{A}\mathcal{T}^{-1} = \begin{pmatrix} 0 & 0 & 0 & -a_0 \\ \ddots & 0 & 0 & \vdots \\ 0 & 1 & 0 & -a_{n-2} \\ 0 & 0 & 1 & -a_{n-1} \end{pmatrix} \qquad B = \mathcal{T}\tilde{B} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad C = \tilde{C}\mathcal{T}^{-1} \tag{4.2}$$

$\mathcal{T}^{-1}$ is the controllability matrix of $(\tilde{A}, \tilde{B})$ [Ding, 2010], which will be defined in (4.4). $a_0, \ldots, a_{n-1}$ are the coefficients of A's characteristic polynomial $t \mapsto t^n + \sum_{i=0}^{n-1} a_i t^i$. Equation (4.2) is called the Frobenius companion form, and is one of many similar companion forms [Eastman et al., 2014, Fiedler, 2003]. We also consider the transpose form, which replaces (A, B) by $(A^T, [0, \ldots, 0, 1]^T)$. A key property of these forms is that A is determined entirely by its eigenvalues λ. Specifically, $a_i = (-1)^{n-i} e_{n-i}(\lambda)$, where e_i is the ith elementary symmetric polynomial. Thus, by using these forms, the number of parameters needed to define (A, B) drops from $n^2 + n$ to just n, since B is fixed and A is determined by λ.

4.2.2 Diagonalization

$A = V^{-1}\Lambda V$ where Λ is a diagonal matrix of the eigenvalues λ. V is the Vandermonde matrix in λ with entries $V_{i,j} = \lambda_i^{j-1}$. Its rows are the (row) eigenvectors of A. Since A is not symmetric, the eigenvectors are neither real nor orthonormal. However, since A is real, any complex eigenvalues come in conjugate pairs: if $\lambda_j = \alpha_j - \beta_j i$ is an eigenvalue, then so too is $\overline{\lambda}_j = \alpha_j + \beta_j i$. Defining $s'_t = Vs_t$, $B' = VB$ and $C' = CV^{-1}$, we diagonalize the system to a *modal* form:

$$s'_{t+1} = VAs_t + VBx_t = \lambda \circ s'_t + B'x_t \qquad y_t = C's'_t + Dx_t + D_0 \tag{4.3}$$

The transpose form is often factored in a slightly different way.

Lemma 7. *$A^T = U\Lambda U^{-1}$ where the jth column of U is $u_j = \left[\frac{1}{\lambda_j{}^{n-i}}\right]_{1\le i\le n}$. (Brand [1964], Leslie [1945]; see the appendix for a self-contained proof.)*

As discussed in the previous chapter, multiplication by V and V^{-1} are equivalent to polynomial evaluation and interpolation, respectively. That is, Vc evaluates a univariate polynomial,

with coefficients c in the monomial basis, at points $\lambda_1, \ldots, \lambda_n$; $V^{-1}y$ recovers the coefficients. Naively performing these operations may be numerically unstable, due to high-degree powers of λ. These operations may be more accurately performed in $O(n^2)$ time by Horner's method and the algorithm of Björck and Pereyra [1970], respectively. These algorithms are (numerically stable) special cases of those described in the previous chapter.

4.2.3 MIMO Luenberger Form

Let b_i be the ith column of B. The controllability matrix of a MIMO LDS has dimensions $n \times (n \cdot d)$:

$$\mathcal{C} = [b_1, \ldots, b_d, Ab_1, \ldots, Ab_d, \ldots, A^{n-1}b_1, \ldots, A^{n-1}b_d] \tag{4.4}$$

From left to right, take n columns, but skip a column if it is linearly dependent on the columns taken so far. If this procedure skips $A^u b_i$, it will also skip the higher powers $A^{u+1}b_i$. For $i \in [d]$, the *controllability index* μ_i is the first power of A skipped for b_i. For reachable LDS, $\sum_i \mu_i = n$.

The Luenberger form $(A^{*d}, B^{*d}E, C, D)$ expresses any reachable, multiple-input LDS as the concatenation of d coupled, reachable, single-input LDS, whose sizes equal the controllability indices [Luenberger, 1967]. Visual examples of A^{*d} and B^{*d} are given in Figure 4.3. A^{*d} has, along the block diagonal, d transpose-form SIMO LDS transition matrices of sizes μ_i. It has off-diagonal entries which couple the SIMO LDS at their inputs. Similarly, B^{*d} is the block diagonal matrix of d transpose-form B vectors, each of dimension $\mu_i \times 1$. E is an invertible, upper triangular matrix which depends on the original system parameters. It is pre-applied to the inputs.

4.3 SIMO LDS in $O(n \log T)$ Parallel Time and n Parameters

The following result makes reachable SIMO LDS our key computational primitive.

Proposition 6. *Reachable, SIMO, n-state LDS are exactly represented by their distinct, nonzero, complex eigenvalues $\lambda \in \mathbb{C}^n$, without further constraints. These eigenvalues can be concretely*

1. Initialize real variables and use them to define eigenvalues λ. In the standard parameterization (left), the variables are α and β, whose total length is n. In the unit parameterization (right), the variables are θ, whose length is $n/2$.

$$a \sim \text{Normal}(0, 1/n)^n \qquad \theta \sim \text{Uniform}(-2\pi, 2\pi)^{n/2}$$
$$\lambda = \text{roots}\Big(t \mapsto t^n + \sum_{i=0}^{n-1} a_i t^i\Big) \qquad \lambda = [\exp(\theta i), \exp(-\theta i)]$$
$$\alpha, \beta \text{ satisfy } \lambda = [\alpha + \beta i, \alpha - \beta i]$$

2. Given a sequence of inputs $x \in \mathbb{R}^T$, compute the sequence of states s'_{t+1}, and their gradients $\nabla s'_{t+1}$ with respect to the underlying real parameters. Use the algorithm of Proposition 7 on the recurrence $s'_{t+1} = \lambda \circ s'_t + B'x_t$ given in (4.3), where B' is the all-ones vector.

3. (Optional). Convert $s_t = V^{-1}s'_t$ using the algorithm of Björck and Pereyra [1970]. Finally, compute the outputs y_t using an additional dense layer, as in Equation (4.1). Alternatively, compute $y_t = \text{Re}(C's'_t) + Dx_t + D_0$ using a relaxation $C' \in \mathbb{C}^{m \times n}$.

Figure 4.1: Summary of how reachable SIMO LDS, with spectral parameterizations, can be used as a fast layer in a neural network. Also consider the "hinge" parameterization in the appendix. Martin and Cundy [2018] implemented the PLR algorithm (per Proposition 6) in CUDA; we extend it for complex inputs.

parameterized by n real numbers. Given the parameters and a length-T sequence of inputs x, it is possible to compute the LDS outputs, and their gradients with respect to the parameters, in $O(n \log T + n^2)$ time on $O(T)$ parallel processors.

It is underpinned by the following algorithm for parallel linear recurrences (PLR).

Proposition 7. *Let $\lambda_1, \ldots, \lambda_T$ and $b_1, \ldots, b_T$ be sequences of n-dimensional vectors. Let $\circ$ denote entrywise product between vectors. For $t \in [T]$, the recurrence $s'_{t+1} = \lambda_t \circ s'_t + b_t$, and its gradients, can be computed in $O\left(n\left(\frac{T}{p} + \log p\right)\right)$ depth (aka parallel time) on p parallel processors. This is $O(n \log T)$ parallel time when $p = O(T)$. [Martin and Cundy, 2018]*

Proposition 6 involves three steps: (1) the complex LDS eigenvalues λ must be concretely parameterized by real numbers, (2) those real parameters must be reasonably initialized, and (3) the LDS must be diagonalized according to (4.3). At first glance, it seems more straightforward to directly parameterize λ and B' in the diagonal form (4.3). Unfortunately, this does not exactly

capture the set of reachable SIMO LDS, unless additional constraints are imposed. If λ and B' are taken to be real, then only a subset is expressed; if they are complex, then a superset is expressed, and the number of parameters doubles. For analytical and practical reasons, it is desirable to exactly use reachable LDS. (For example, if LDS are stacked in a neural network, then reachability would ensure each layer can supply a full spectrum of input to the subsequent layer.)

Parameterization. The standard approach is to separately parameterize the real and imaginary (if present) parts of λ. Since the complex eigenvalues present in conjugate pairs, this requires only n real parameters (α, β) in total. More specifically, the complex pairs are $\lambda_j = \alpha_j - \beta_j i$ and $\bar{\lambda}_j = \alpha_j + \beta_j i$. The real eigenvalues just have α_j. For long-term dependencies, it is useful to constrain $|\lambda_j| = 1$, as in orthogonal or unitary A [Arjovsky et al., 2016]. This constraint is trivial in our framework. Suppose λ_j has polar representation (r_j, θ_j). Then a zero real part of $\ln \lambda_j = \ln r_j + \theta_j i$ corresponds to magnitude $r_j = 1$. Parameterize $\ln \lambda$ with 0 real part and $\pm\theta$ imaginary part, then exponentiate.

Initialization. For the previously defined real variables, typical random initializations, such as sampling from a truncated normal, lead to numerical instability. In the standard parameterization, we found it useful to initialize near unit eigenvalues. It is known that a monic polynomial with random coefficients has roots λ of magnitude close to 1 [Hughes and Nikeghbali, 2008]. These may be obtained by randomly initializing the coefficients a in (4.2), and then computing the eigenvalues of A [Aurentz et al., 2015]. For the unit parameterization, the coordinates θ_j must be kept numerically distinct. For moderate n, uniform random initialization is suitable. For large n, a low-discrepancy sequence, such as the van der Corput sequence, may be preferable.

Diagonalization. The two computational tasks are computing B' (for use in PLR) and converting between s_t and s'_t. For the standard form, $B' = V[1, 0, \ldots, 0]^T = [1, \ldots, 1]$ since that is the first column of V. As reviewed in Section 4.2, conversion between s_t and s'_t may be accomplished by polynomial evaluation and interpolation algorithms. For the transpose form expressed in terms of U, B' is the last column of U^{-1}. For completeness, this is derived in the appendix.

Lemma 8. *Given the (unnormalized) definition of U in Lemma 7, the complex conjugate of the last column of U^{-1} is $B' = \left[\lambda_i^{n-1} / \prod_{j \neq i} (\lambda_i - \lambda_j)\right]_{1 \leq i \leq n}$*

Related Work. LDS are often reparameterized for computational benefit [Shalit and Chechik, 2014], sometimes in terms of induced subspaces De Cock and De Moor [2002], Huang et al. [2017]. Chang et al. [2018] also study complex eigenvalue parameterizations with zero real part. Hsu et al. [2020] analyze LDS clustering using the Vandermonde decomposition. Previous algorithms attempt to run LDS in constant time with respect to T [Kozdoba et al., 2019, Martens, 2010]. However, these works rely on stability assumptions and approximations: they do not exactly compute forward and backward passes of LDS. Furthermore, they require the inputs to be partially and completely noise, respectively. Surprisingly, Lemma 8 does not plainly appear in the literature, even in recent work on generalizations of Vandermonde matrices [Rawashdeh, 2018]. Its proof uses the same technique as the "eigenvectors from eigenvalues" theorems that have gained recent attention in disparate areas of applied mathematics [Denton et al., 2019]. These results are more general, but do not yield closed-form expressions, and do not directly apply to the inverse matrix U^{-1}.

4.4 Approximating MIMO LDS by SIMO LDS

4.4.1 Improper Learning: Random Projection

MIMO LDS can be approximated by the average of r SIMO LDS, each produced by randomly projecting the input vectors to a single dimension. These LDS share the same weights λ.

Proposition 8. *Let $x_1, \ldots, x_T \in \mathbb{R}^d$ and $y_1, \ldots, y_T \in \mathbb{R}^m$ be the inputs and outputs of a reachable MIMO LDS with parameters (A, B, C, D). For each $j \in [r]$, let g_j be a d-dimensional standard normal vector, $x_t^{[j]} = x_t^T g_j$ be projected scalar inputs, and (A, Bg_j, C, D) be the parameters of a SIMO LDS producing outputs $y_t^{[j]}$. Let $\hat{y}_t = \frac{1}{r}\sum_{j=1}^{r} y_t^{[j]}$ be the average output. For each $t \leq T$, $\mathbf{E}\,||y_t - \hat{y}_t||^2 = \sum_{j=1}^{m} 2\,||Z_{t,j}||_F^2 / r$, where $Z_{t,j} = \sum_{\tau=1}^{t-1} x_{t-\tau} C_{j,:} A^\tau B$. Furthermore, the SIMO LDS are almost surely reachable, and share the same canonical form matrix.*

The proof of this equality uses standard techniques. Here is some brief intuition for the result. Suppose $m = 1$ and each x_t has standard $N(0, 1)$ components, as is typical in dynamical systems literature. Also assume that A's spectral radius $\rho < 1$ (i.e. the LDS is *strictly stable*), $||B||_2 \leq 1$,

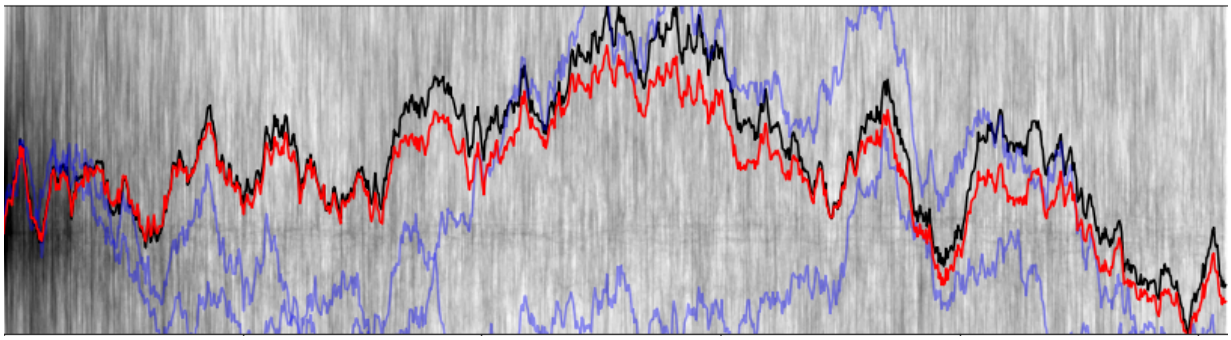


Figure 4.2: Illustration of a MISO LDS (black), of state size $n = 16$, operating on inputs of $d = 32$ dimensions over $T = 1024$ timesteps, approximated by SISO LDS. In light gray (nearly filling the background) are 512 SISO LDS, induced by random projections per Proposition 11. These have very high variance and do not approximate the MISO LDS. The two blue lines represent the average of two independent subsamples of 16 SISO LDS. These small averages still do not approximate the MISO LDS. The red line is the average of all 512 SISO LDS. This is fairly close to the MISO LDS.

and $||C|| \leq 1$. By the definition of the Frobenius norm and independence of each input:

$$\mathbf{E}\ \mathrm{tr}(Z_t^T Z_t) = \mathrm{tr} \sum_{\tau=1}^{t-1} B^T A^{\tau T} C^T \left(\mathbf{E}\ x_{t-\tau}^T x_{t-\tau} \right) C A^\tau B \leq\ d \sum_{\tau=1}^{t-1} \rho^{2\tau} \leq d \frac{\rho^2}{1-\rho^2} \tag{4.5}$$

Related Work. Gaussian projections are a key technique in randomized algorithms [Johnson and Lindenstrauss, 1984, Kannan and Vempala, 2017]. Model reduction is the approximation of large-size LDS by smaller-size LDS [Antoulas, 2005]. Proposition 8 does not reduce the size of the LDS, but rather the dimension of its inputs.

4.4.2 Proper Learning: Perturbed Luenberger Form

Proper learning of LDS, also known as system identification, is the task of recovering the parameters (A, B, C, D) from input-output data. The Luenberger form, reviewed in Section 4.2.3, exactly decomposes a MIMO LDS into a concatenation of smaller, SIMO LDS. It establishes a promising connection between proper learning of MIMO LDS and proper learning of SIMO LDS. However, as a parameterization used during learning, it has a crucial problem: the controllability indices, defining the sizes of the SIMO LDS, are not known. In practice, the SIMO LDS must be sized according to a loose upper bound, which then makes learning improper. For-

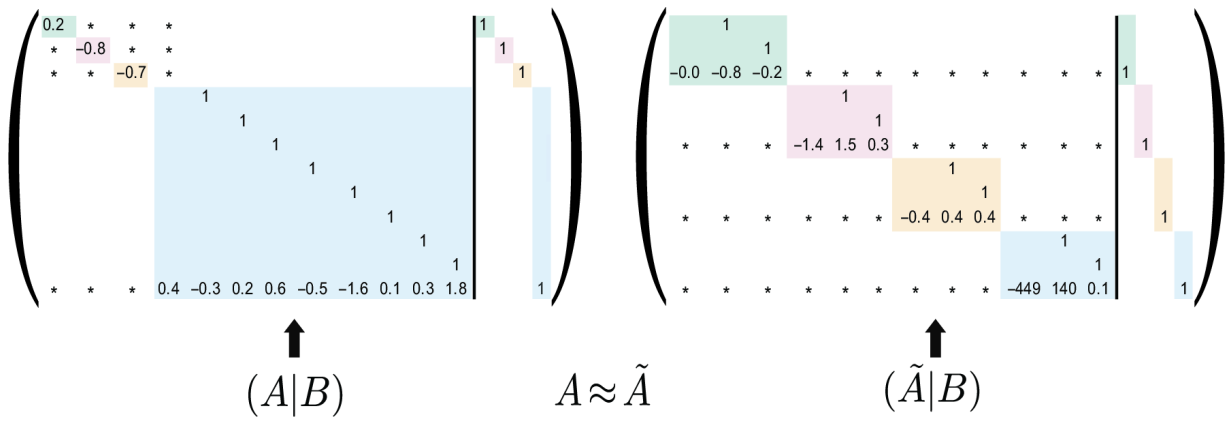


Figure 4.3: Left: is the Luenberger canonical form of a multiple-input LDS, with A to the left of the vertical line and B to the right. It decomposes into four single-input LDS of sizes $9, 1, 1$ and 1, which match the controllability indices. After the addition of a tiny amount of noise (in the form of a Gaussian matrix with variance 0.00000001), the canonical form decomposes into evenly-sized single-input LDS. The asterisks denote nonzero values which couple the single-input LDS.

tunately, the following result shows that any MIMO LDS is nearly equal to a concatenation of coupled SIMO LDS, each of known size.

Proposition 9. *Let n be divisible by d. Let (A, B) be the parameters of a reachable size-n LDS taking d-dimensional inputs. For any $\epsilon > 0$, there exists a perturbed system $(\tilde{A}, B)$ such that (1) $||A - \tilde{A}|| \leq \epsilon$, and (2) the controllability indices of $(\tilde{A}, B)$ are all n/d. Therefore, the Luenberger form of $(\tilde{A}, B)$ is a concatenation of d coupled SIMO LDS, each of size n/d.*

We may effectively treat any MIMO LDS data as if it originated from a system with equal controllability indices, i.e. equally-sized SIMO LDS. This result suggests that proper learning of LDS is largely equivalent to proper learning of SIMO LDS, which supports the latter's consideration as a key primitive. We present the perturbed Luenberger form as a conceptual reduction from MIMO to SIMO, rather than a practical algorithmic tool. The practical issue is that the SIMO LDS are coupled: the next state for each LDS depends on not just its own state, but also on the state of the other $(n/d) - 1$ LDS. This prevents the LDS from running independently, and thereby hinders parallelization.

Related Work There is a vast literature on system identification [Ljung, 1999]. Subspace identification (SSID) is the prevalent technique, utilized by the state-of-the-art work cited in

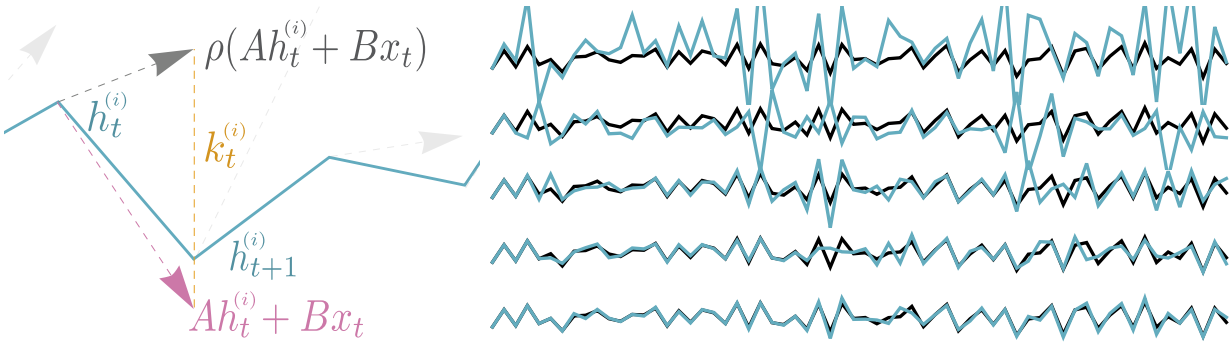


Figure 4.4: *Left*: A step of local correction within LDStack. Suppose the ith layer's states $h_t^{(i)}$ are all computed. We consider, at each t, two hypothetical steps from $h_t^{(i)}$: the linear step $Ah_t^{(i)} + Bx_t$ and the nonlinear step $\rho(Ah_t^{(i)} + Bx_t)$. Their difference is the correction $k_t^{(i)}$, which is added to $h_t^{(i+1)}$ in the next layer. Note that $h_{t+1}^{(i)} = Ah_t^{(i)} + Bx_t + k_t^{(i-1)}$ does not necessarily coincide with the hypothetical linear step, since it was corrected in this fashion. The faint gray arrows illustrate that the corrections are computed in parallel using only local information. *Right*: RNN (black) approximated using stacked LDS of increasing depth (from top to bottom). Observe the "correct from the start" behavior described in Proposition 10.

the introduction. SSID does not reduce MIMO to SIMO, as we do. It is well known that the controllability indices are numerically unstable [Jordan and Sridhar, 1973]. Our result shows this numerical instability is a blessing, since a small perturbation renders it useful. There are deterministic methods of modifying the original system to obtain (nearly) equal controllability indices, at the expense of increased state size [Cook, 1978]. The (mis)use of MIMO canonical forms as parameterizations for learning is discussed in [Glover and Willems, 1974]. They discuss a numerical advantage of Luenberger's (pseudocanonical) form over MIMO canonical forms, and base a system identification method upon it [Glover, 1973]. Subsequent works on 'overlapping' parameterizations also avoided the problem of unknown structural indices [Corrêa and Glover, 1984, Gevers and Ah-Chung, 1985].

4.5 Approximating Nonlinear RNNs by Stacked LDS

Let $h_{t+1} = \rho(Ah_t + Bx_t)$ be an RNN which takes inputs $x_t \in \mathbb{R}^d$ and an initial state $h_0 \in \mathbb{R}^n$, and produces subsequent states $h_t \in \mathbb{R}^n$. Its nonlinearity ρ has deviation from linearity $\delta(a) = \rho(a) - a$. This deviation is used to define local corrections to an LDS, as follows:

$$h_{t+1} = (Ah_t + Bx_t) + \delta(Ah_t + Bx_t) \quad \longrightarrow \quad h_{t+1}^{(i+1)} = Ah_t^{(i+1)} + Bx_t + \overbrace{\delta(Ah_t^{(i)} + Bx_t)}^{k_t^{(i)}} \tag{4.6}$$

On the left is a trivial equality involving δ. Its first term is a linear transition from h_t; its deviation from a correct (nonlinear) transition is measured by the second term. The approximation starts with a plain LDS $h_{t+1}^{(0)} = Ah_t^{(0)} + Bx_t$; then, its deviations are used as corrections $k_t^{(0)}$ to a subsequent LDS. Iterating this construction yields a stack of corrected LDS. As the previous layer's states $h_t^{(i)}$ become close to the next layer's $h_t^{(i+1)}$, the corrections become more accurate. With enough layers, the nonlinear RNN is exactly recovered. More generally, the layers are "correct from the start". Since the initial state $h_0^{(0)} = h_0$ is correct, the first layer gets the first state correct: $h_1^{(1)} = Ah_0 + \delta(Ah_0) = h_1$. The second layer gets the second state correct, and so forth, yielding a consistency guarantee. This holds in the worst-case setting; empirically, close approximation is observed with far fewer than T layers.

Proposition 10. *$h_t^{(\Delta)} = h_t$ for all $t \in [\Delta]$ and all $x \in \mathbb{R}^T$. Thus, $h^{(T)} = h$ for all $x \in \mathbb{R}^T$.*

Note that this convergence guarantee is uniform for all input sequences x. The local corrections made to the states are specific to each x. This strong guarantee states that the scheme of iterated local corrections matches the nonlinear RNN not just on an individual example, but on an entire dataset.

Since the stacked LDS have nonlinearity along depth, they may seem just as difficult to analyze as the original nonlinear RNN. Fortunately, our construction is a discrete, additive version of a continuous, multiplicative scheme developed in control theory [Tomás-Rodríguez and Banks, 2010]. It has been extensively used to analyze nonlinear dynamical systems via sequences of linear approximations. Controllers for aircraft, supertankers, and autopilots have been derived with this approach [Çimen and Banks, 2004]. It is possible to derive explicit solutions for the linear approximation in terms of an underlying Lie algebra [Banks, 2002]. The appendix describes this control-theoretic precursor of our construction. It is reasonable to expect that some of the same analytic techniques will carry over.

This is a simple but useful technique which can be used in both practice (to improve the expressive power of linear sequence models, without an increase in parameters) and in theory (to analyze and build correspondences between nonlinear sequence models and linear ones).

Related Work. Generalizing earlier works [Balduzzi and Ghifary, 2016, Bradbury et al., 2017], Martin and Cundy [2018] advocate the removal of nonlinearities across time, while introducing nonlinearity along depth. Given an RNN, they replace nonlinear dependencies across time with a "linear surrogate" amenable to PLR. These new RNNs can run in parallel, but it is not clear they can approximate the original nonlinear RNNs, and they are not as well-studied as LDS. Restricted subclasses of RNNs can be approximately differentiated in constant time [Liao et al., 2018]. There are substantial efforts to understand nonlinear RNNs [Karpathy et al., 2015] and develop provable learning algorithms for them [Allen-Zhu and Li, 2019, Allen-Zhu et al., 2019a, Foster et al., 2020].

The culmination of our results is the neural network layer $\mathbf{LDStack}(\boldsymbol{\rho}, \boldsymbol{n}, \boldsymbol{\Delta}, \boldsymbol{r})(\boldsymbol{x}, \boldsymbol{h_0})$. It takes (a batch x of) length-T sequences of d-dimensional vectors, and an n-dimensional initial state h_0. It returns (a batch $\hat{h}$ of) length-T sequences of n-dimensional states. It uses $O(\Delta n^2 \log T \log r)$ time on $O(rT)$ parallel processors.[2] Its settings are the nonlinearity ρ, state size $n > 1$, depth $\Delta \geq 1$, and number of projections $r \geq 1$. It has $O(n + n^2 d)$ trainable weights and rd fixed weights.

LDStack details. Suppose the (unknown) RNN has parameters $(\tilde{A}, \tilde{B})$ which define a reachable LDS. Let $\mathcal{C}$ be its $n \times (n \times d)$ controllability matrix. At initialization, random projections $g_j \in \mathbb{R}^d$ are drawn, for $j \in [r]$. The first layer is an average of plain SIMO LDS. Let $x_t^{[j]} = x_t^T g_j$ be the projected input of the jth SIMO LDS. $s_{j,t+1}^{(0)'} = \lambda \circ s_{j,t}^{(0)'} + B' x_t^{[j]}$ are computed in parallel, per Section 4.3. To compute the corrections, reverse the canonical and diagonal transformations $\mathcal{T}_j$ and V according to (4.2) and (4.3). Recall from (4.2) that $\mathcal{T}_j^{-1} = \mathcal{C}_j$, the $n \times n$ controllability matrix of the jth SIMO LDS $(\tilde{A}, \tilde{B} g_j)$. Then $\mathcal{C}_j = [Bg_j, ABg_j, \ldots, A^{n-1} Bg_j] = \mathcal{C} \cdot g_j$. Eliding superscripts:

[2]For simplicity, this time bound does not internally parallelize $O(n^2)$ matrix-vector multiplication and linear system solving. The analogous bound for nonlinear RNNs is $O(n^2 T)$.

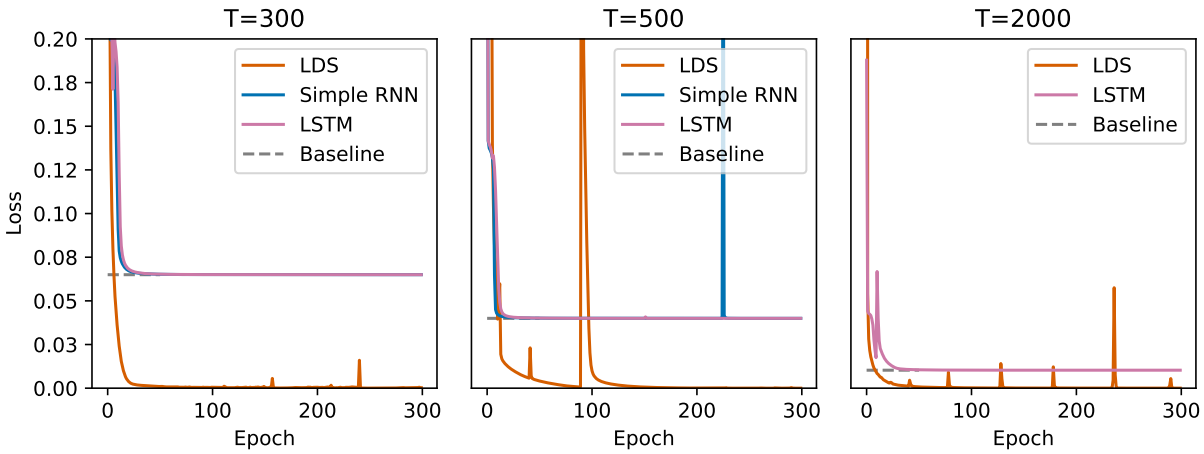


Figure 4.5: On the copying memory problem, standard RNNs do not outperform a trivial baseline. We solve it with the simplest model to date: a unitary SISO LDS, as described in Figure 4.1.

$$\tilde{A}\tilde{s}_{j,t} + \tilde{B}x_t = \mathcal{T}_j^{-1} A \underbrace{\mathcal{T}_j \tilde{s}_{j,t}}_{s_{j,t}} + \mathcal{T}_j^{-1} B x_t = \mathcal{T}_j^{-1}(As_{j,t} + Bx_t) = \mathcal{T}_j^{-1} V^{-1}(\Lambda \overbrace{Vs_{j,t}}^{s'_{j,t}} + VBx_t)$$
$$= \mathcal{T}_j^{-1} V^{-1}(\lambda \circ s'_{j,t} + B'x_t)$$

We introduce a free parameter $W \in \mathbb{C}^{n \times n \times d}$ which ideally satisfies $W \cdot r_j = (\mathcal{C} \cdot r_j)V^{-1}$, so it can directly perform the reverse transformations $\mathcal{T}_j^{-1}V^{-1}$. Averaging within (4.6), the corrections are $\tilde{k}_t^{(0)} = \delta(\frac{1}{r}\sum_{j=1}^{r} \tilde{A}\tilde{s}_{j,t}^{(0)} + \tilde{B}x_t^{[j]})$. Now we compute the next layer. Take the corrections back to the diagonalized basis as $k_{j,t}^{(0)'} = V\mathcal{T}_j\tilde{k}_t^{(0)}$. The corrected SIMO LDS are run in parallel using $s_{j,t+1}^{(1)'} = \lambda \circ s_{j,t}^{(1)'} + B'x_t^{[j]} + k_{j,t}^{(0)'}$. After Δ layers, $\hat{h}_t^{(\Delta-1)} = \frac{1}{r}\sum_{j=1}^{r} \tilde{s}_{j,t}^{(\Delta-1)}$ are returned.

4.6 Experiments

Copy memory problem [Arjovsky et al., 2016, Hochreiter and Schmidhuber, 1997]. The goal is to remember the first 10 entries r of an input sequence, withhold output for T steps (for which the inputs are just "blanks"), and, upon seeing a "go" input at time $T + 10$, to output r. There is a SISO LDS which achieves zero error [Henaff et al., 2016], so we do not consider LDStack of higher depth. Unitary RNNs are known to solve the problem, so we use the unit parameterization of Figure 4.1. Arjovsky et al. [2016] use LSTM, simple tanh RNN, and uRNN of respective sizes $n = 40, 80$, and 128 for parameter counts of roughly 6500. We use $n = 160$, which results in

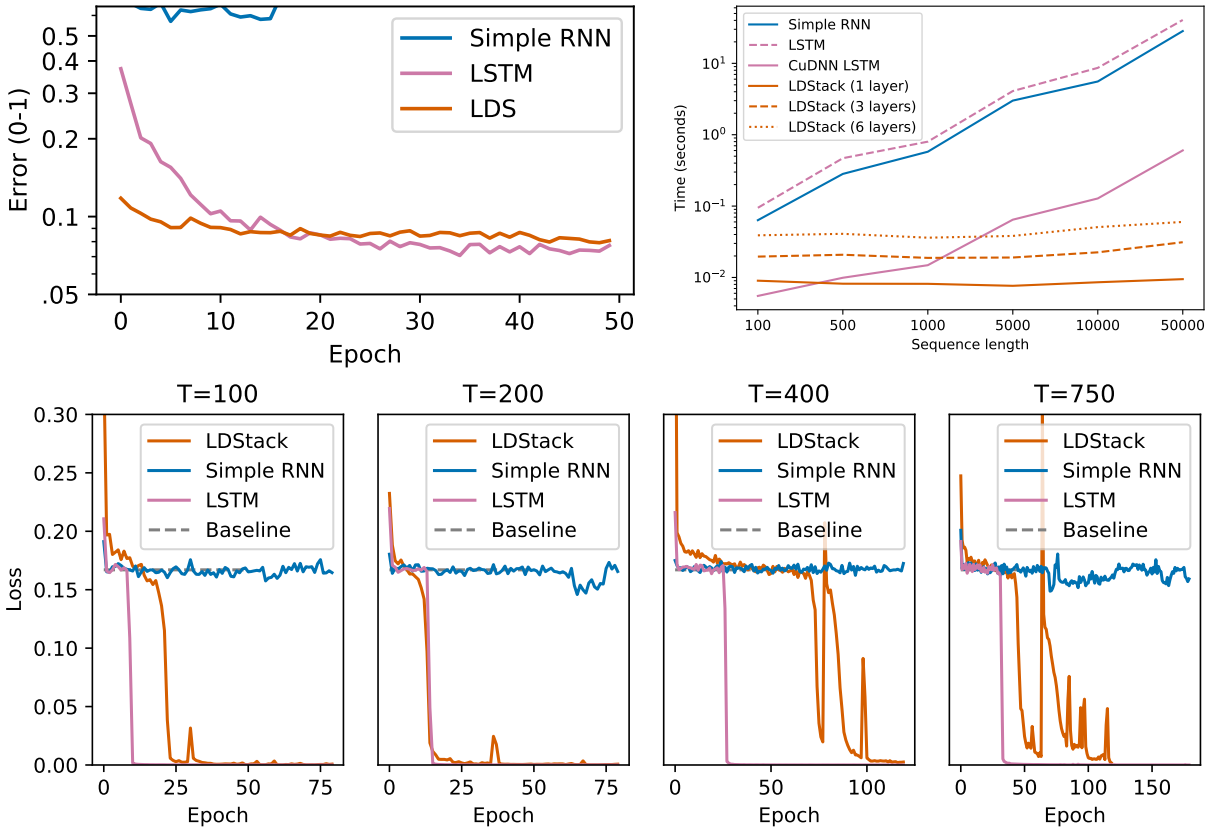


Figure 4.6: *Top left*: Sequential permuted MNIST. *Top right*: Runtimes for different sequence lengths. *Bottom*: the adding problem, with larger sequence lengths representing more challenging problems.

just 3380 parameters, including $C' \in \mathbb{C}^{n\times n}$. Our solution is the state of the art: it uses the simplest (linear) RNN with the fewest parameters to solve the $T = 2000$ instance. Previously, such performance demanded full-capacity uRNNs [Wisdom et al., 2016] or nonlinear RNNs [Lezcano-Casado and Martínez-Rubio, 2019].

Sequential permuted MNIST. The images are presented as length-784 sequences of pixels. Their order is arbitrary, but fixed across all images. We compare an $n = 384$ SIMO LDS having $\sim 16,500$ parameters to an $n = 128$ LSTM having $\sim 68,000$ parameters, as well as an $n = 128$ tanh RNN having $\sim 18,000$. The LDS and the LSTM achieve similar accuracies of 91.8% and 92.3%. This performance is not state of the art: for example, Chang et al. [2018] achieve 95.8% accuracy with 10,000 parameters. However, the LDS steps take 73ms, compared to 324ms for the RNN.

Runtime comparison. LDStack (prototype code in both Python and CUDA) is always faster than unfused RNNs (whose GPU kernels have not been manually coalesced). At longer sequence lengths, it is even faster than the highly-optimized, fused CuDNN LSTM. The $O(T)$ and $O(\log T)$ asymptotics manifest plainly.

Adding problem [Arjovsky et al., 2016, Hochreiter and Schmidhuber, 1997]. Each input has dimension $T \times 2$. The output is the sum of the two numbers (from the first dimension) which are marked by ones (in the second dimension); the rest of the outputs are zeros. Trivially returning 1 achieves mean-squared error 0.167. This problem cannot be solved by an LDS, so it exercises both random projection and nonlinear approximation by stacking. We use LDStack with state size $n = 32$, depth $\Delta = 2$, and $r = 6$ projections. This has 4,175 parameters, compared to $\sim 27,000$ and $\sim 17,000$ for an LSTM and tanh RNN, respectively, having $n = 80$. The simple RNN fails to beat the trivial baseline. The LSTM and LDStack both solve the problem up to $T = 750$, though the latter takes longer to converge, and is more unstable in later epochs. Overall, the LSTM and LDStack are roughly comparable in accuracy, but LSTM is slower.

4.7 Appendix

4.7.1 Proof of Lemma 7

Proof. We wish to show $Au_j = \lambda_j u_j$. If the theorem is true, then $\lambda_j u_{j,i} = \lambda_j \frac{1}{{\lambda_j}^{n-i}} = \frac{1}{\lambda_j^{n-(i+1)}} = u_{j,i+1}$. Recall the state update of the controllable LDS, which shifts $n-1$ entries and computes a dot product in the last entry:

$$Au_j = \begin{bmatrix} u_{j,2} \\ \vdots \\ u_{j,n-1} \\ -\sum_i a_{i-1} u_{j,i} \end{bmatrix} = \begin{bmatrix} \lambda_j u_{j,1} \\ \vdots \\ \lambda_j u_{j,n} \\ -\sum_i a_{i-1} / {\lambda_j}^{n-i} \end{bmatrix}$$

It suffices to show:

$$-\sum_i a_{i-1}/\lambda_j^{n-i} = \lambda_j u_{j,n} = \lambda_j \text{ i.e. } \sum_{1\le i\le n} \frac{a_{i-1}}{\lambda_j^{n-(i-1)}} = -1 \tag{4.7}$$

It is well known that the characteristic polynomial of A is $p(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_{n-1}t^{n-1} + t^n$. By definition, its roots (those t where $p(t) = 0$) are the eigenvalues of A.
So each λ_j satisfies:

$$0 = a_0 + a_1\lambda_j + a_2{\lambda_j}^2 + \ldots + a_{n-1}{\lambda_j}^{n-1} + {\lambda_j}^n = \qquad {\lambda_j}^n\left(1 + \sum_{1\le i\le n} \frac{a_{i-1}}{\lambda_j^{n-(i-1)}}\right)$$

Either we have a null eigenvalue $\lambda_j = 0$, or we have the desired equation (4.7). □

4.7.2 Proof of Lemma 8

Proof. Let v_i be the ith row of U^{-1}. The dual basis of U is $(U^{-1})^T$, i.e. $u_i^T v_i = 1$ and for all $j \neq i$, $u_i^T v_j = 0$. Since B' is the conjugate of the nth column of U^{-1}, it is determined by the nth coordinates of the v_i. We derive these by employing the adjugate technique of Denton et al. [2019]. Recall the determinant $\det(A) = \prod_i \lambda_i$ is the product of the eigenvalues. Also recall the following general definition of the adjugate matrix, when A is diagonalizable but not necessarily Hermitian:

$$\text{adj}(A)_{i,j} = \sum_{k=1}^{n}\left(\prod_{l\neq k}\lambda_l\right) u_{k,i}\bar{v}_{k,j}$$

For any k, replace A by $\lambda_k\lambda I_n - A$. This causes all but one of the summands to vanish, yielding the following simplication:

$$\text{adj}(\lambda_k I - A)_{i,j} = \left(\prod_{l\neq k}(\lambda_k - \lambda_l)\right) u_{k,i}\bar{v}_{k,j}$$

Setting $i = 1$ and $j = n$, and substituting the previously derived entries of u_k:

$$\text{adj}(\lambda_k I - A)_{1,n} = \left(\prod_{l \neq k} (\lambda_k - \lambda_l) \right) \frac{1}{\lambda_k^{n-1}} \bar{v}_{k,n} \tag{4.8}$$

By the Laplace expansion of the adjugate matrix of A, $\text{adj}(\lambda_k I - A)_{1,n}) = (-1)^{1+n}\det(M)$, where M is the minor of $\lambda_k I - A$ produced by removing its nth row and 1st column. It is straightforward to show that the only eigenvalue of M is -1 with multiplicity $n-1$, and therefore $\det(M) = (-1)^{n-1}$. Therefore $\text{adj}(\lambda_k I - A)_{1,n} = (-1)^{2n} = 1$. Combining this with (4.8) obtains an equality for each $\bar{v}_{k,n}$, which matches the desired result. □

4.7.3 Proof of Proposition 8

Proposition 8 is an easy corollary of the following proposition, which involves MISO LDS rather than MIMO LDS.

Proposition 11. *Let $x_1, \ldots, x_T$ be any sequence of d-dimensional inputs, and let $y_1, \ldots, y_T$ be the corresponding outputs of a reachable MISO LDS with parameters (A, B, C, D). For each $j \in [r]$, let g_j be a d-dimensional standard normal vector, $x_t^{[j]} = g_j^T x_t$ be a projected sequence of scalar inputs, and (A, Bg_j, C, D) be the parameters of a SISO LDS producing outputs $y_t^{[j]}$. Let $\hat{y}_t = \frac{1}{r}\sum_{j=1}^r y_t^{[j]}$ be the average output. For each $t \leq T$, $\mathbf{E}(y_t - \hat{y}_t)^2 = 2||Z_t||_F^2/r$, where Z_t is defined below in (4.9). Furthermore, the SISO LDS are almost surely reachable, and share the same canonical form matrix.*

Proof. While proving this result, let us take $D = 0$ and $s_0 = 0$ for notational simplicity. (These are just constant terms which do not affect the result.) From the convolution representation (4.1) and the random construction of the SISO LDS, we find that the approximation is unbiased:

$$\mathbf{E}\, \hat{y}_t = \mathbf{E}\, \frac{1}{r} \sum_j \sum_{\tau=1}^{t-1} CA^\tau B g_j g_j^T x_\tau = \sum_{\tau=1}^{t-1} CA^\tau B \left(\frac{1}{r} \mathbf{E} g_j g_j^T \right) x_{t-\tau} = y_t$$

Therefore the mean squared error is just the variance:

$$\mathbf{E}\, (y_t - \hat{y}_t)^2 = \mathbf{E}\, ((\mathbf{E}\, \hat{y}_t) - \hat{y}_t)^2 = \mathbf{V}(\hat{y}_t)$$

By the independence of the g_j, and the cyclic property and linearity of trace, we reduce to the variance of a quadratic in normal variables:

$$\begin{aligned}
\mathbf{V}(\hat{y}_t) =& \mathbf{V}\left(\sum_{\tau=1}^{t-1} \operatorname{tr}(CA^\tau B\left(\frac{1}{r}\sum_{j=1}^{r} g_j g_j^T\right) x_{t-\tau})\right) \\
=& \frac{1}{r^2}\sum_{j=1}^{r} \mathbf{V}\left(\sum_{\tau=1}^{t-1} \operatorname{tr}(g_j^T x_{t-\tau} CA^\tau B g_j)\right) \\
=& \frac{1}{r^2}\sum_{j=1}^{r} \mathbf{V}\left(g_j^T g_j \sum_{\tau=1}^{t-1} CA^\tau B x_{t-\tau}\right) \\
=& \frac{1}{r^2}\sum_{j=1}^{r} \mathbf{V}\left(g_j^T \underbrace{\sum_{\tau=1}^{t-1} x_{t-\tau} CA^\tau B}_{Z_t} g_j\right)
\end{aligned} \tag{4.9}$$

The inner quadratic is not changed by replacing Z_t, which is asymmetric, with $\bar{Z}_t = \frac{1}{2}(Z_t + Z_t^T)$, which is symmetric, diagonalizable, and shares the same eigenvalues $\nu_1, \ldots, \nu_d$. g_j retains its distribution under the rotation U that diagonalizes $\bar{Z}_t$. We find the variance is just the squared Frobenius norm of Z_t:

$$\begin{aligned}
\mathbf{V}\left(g_j^T \bar{Z}_t g_j^T\right) =& \mathbf{V}\left(g_j^T U^T \operatorname{diag}(\nu) U g_j\right) \\
=& \mathbf{V}\left(\sum_{i=1}^{d} g_{j,i}^2 \nu_i\right) = 2\sum_{i=1}^{d} \nu_i^2 = 2\,||Z_t||_F^2
\end{aligned}$$

Now we verify that the SISO LDS are almost surely reachable, assuming the MISO LDS is reachable. By Lemma 6, we must show that if $[\gamma I - A; B]$ has full rank for all $\gamma \in \mathbb{C}$, then $[\gamma I - A; Bg_j]$ also does, almost surely. This holds because g_j has density with respect to Lebesgue measure.

To conclude the proof of Proposition 11, denote the MIMO LDS matrices above as $(\tilde{A}, \tilde{B})$. When projected to SIMO LDS $(\tilde{A}, \tilde{B}g_j)$, their canonical forms (A_j, B) are obtained via $\tilde{A}_j = \mathcal{T}_j^{-1} A \mathcal{T}_j$. Let v_i and λ_i be an eigenvector and corresponding eigenvalue of $\tilde{A}$: $\tilde{A}v_i = \lambda_i v_i$. Then $A_j \mathcal{T}_j v_i = \lambda_i \mathcal{T}_j v_i$, so the A_j share the same eigenvalues as $\tilde{A}$. Since A_j are companion matrices of the same form (4.2), this means they are actually the same matrix A.

□

4.7.4 Proof of Proposition 9

The following proposition implies Proposition 9.

Proposition 12. *Let n be divisible by d. Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times d}$ be full rank. Let (A, B) form a reachable MIMO LDS. Choose any $\epsilon > 0$ and any (Schatten) matrix norm $||\cdot||$. There is a $\delta > 0$ such that the following holds. Let G be an $n \times n$ matrix of normal variables of mean zero and variance δ, and $\tilde{A} = A + G$. Then, with nonzero probability, $||A - \tilde{A}|| \leq \epsilon$ and the controllability indices of $(\tilde{A}, B)$ are all equal to n/d.*

Proof. Clearly $||G|| \leq \epsilon$ with nonzero probability. The controllability indices are equal if the first n rows of the controllability matrix (4.4) are linearly independent. Thus, we must show that the following $n \times n$ matrix has full rank:

$$\mathcal{C}_{:,:n} = [B, (A+G)B, (A+G)^2B, \ldots, (A+G)^{n/d-1}B]$$

The first d columns are linearly independent by assumption. In the remaining columns, since G is normal — and therefore has density with respect to Lebesgue measure — linear independence follows from a standard argument. $\mathcal{C}_{:,:n}$ is full rank unless its determinant is zero. The determinant is a polynomial $p : \mathbb{R}^{n^2} \to \mathbb{R}$ in the (flattened) entries of $\mathcal{C}_{:,:n}$. For any such polynomial p, the set $p = 0$ has Lebesgue measure zero. □

4.7.5 Approximation of Nonlinear Systems by Time-Varying LDS

Tomás-Rodríguez and Banks [2010] describe a method of approximating continuous-time dynamical systems by linear, time-varying ones. We briefly review their method, showing how it gives rise to a multiplicative variant of LDStack. Consider the following nonlinear, discrete-time dynamical system: $h_{t+1} = \rho(Ah_t) + Bx_t$. Bx_t is usually inside the nonlinearity ρ, but we keep it separate for reasons that will be discussed below. ρ must be continuously differentiable. Furthermore, in order for the approximation scheme to be numerically stable, ρ must also be analytically "nice", as described below. We use the inverse square root activation $\rho(a) = a/\sqrt{1+a^2}$ as a running example.

We begin by viewing the RNN as an Euler discretization of a continuous-time dynamical system (e.g. Tallec and Ollivier [2018]). Using the Taylor expansion $h(t+\epsilon t) \approx h(t) + \epsilon t \cdot \dot{h}(t)$, and taking a step size of $\epsilon = 1$, we obtain the following nonlinear differential equation: $\dot{h} = \rho(Ah) - h + Bx$. (We elide the dependence on t to simplify notation). The first step is to convert the dynamical system to state-dependent coefficient (SDC) form: $\dot{h} = \mathcal{A}(h)h - h + Bx$. Here, the nonlinear update is factorized to resemble an LDS. SDC form does not allow $\mathcal{A}$ to depend on x, which is why Bx_t was kept outside of $\rho(\cdot)$. The SDC factorization can be derived in a straightforward manner.

Lemma 9. *The following is a valid SDC factorization when $\rho \in C^1$ and $\rho(0) = 0$. [Cimen, 2010]*

$$\mathcal{A}(h) = \int_0^1 \left. \frac{d\rho(Ah)}{dh} \right|_{h=\lambda h} d\lambda$$

We call ρ "nice" if the above factorization is numerically stable and can be analytically derived. For our example ρ, a brief calculation shows the SDC form is:

$$\dot{h} = \underbrace{\text{diag}(1/\sqrt{1 + (Ah)^2})A}_{\mathcal{A}(h)} h - h + Bx$$

Note that $\mathcal{A}(h)h$ is a multiplicative, entrywise correction of Ah based on its deviation from $\rho(Ah)$. Under weak conditions on $\mathcal{A}$, the SDC-form nonlinear system can be approximated by a sequence of linear, time-varying systems.

Theorem 6 (Informal)**.** *Let $\mathcal{A}$ be locally Lipschitz. Consider this sequence of time-varying LDS:*

$$\begin{aligned} \dot{h}^{(0)} &= \mathcal{A}(h_0)h^{(0)} - h^{(0)} + Bx & h_0^{(0)} &= h_0 \\ \dot{h}^{(i)} &= \mathcal{A}(h^{(i-1)})h^{(i)} - h^{(i)} + Bx & h_0^{(i)} &= h_0 \end{aligned}$$

As $i \to \infty$, the solution of $h^{(i)}$ converges to the solution of h. [Tomás-Rodríguez and Banks, 2010]

The nonlinear RNN approximation in Definition 4 is just a discretization of Theorem 6.

Definition 4 (Nonlinear RNN Approximation)**.** *Let ρ be a continuously differentiable activation function with $\rho(0) = 0$. For $t \in [T]$, let $h_{t+1} = \rho(Ah_t) + Bx_t$ be the n-dimensional states of an*

RNN with parameters (A, B). Let $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, as given by (9), be locally Lipschitz. This is a stack of time-varying LDS whose depth is indexed by i:

$$h_{t+1}^{(0)} = \mathcal{A}(h_0)h_t^{(0)} + Bx_t \qquad h_0^{(0)} = h_0$$
$$h_{t+1}^{(i)} = \mathcal{A}(h_t^{(i-1)})h_t^{(i)} + Bx_t \qquad h_0^{(i)} = h_0$$

Our additive variant is more algorithmically convenient, whereas the multiplicative variant is superior for approximation theory. Multiplicative corrections interfere with diagonalization, which is crucial for our algorithms. However, as illustrated in Figure 4.7, additive corrections can produce oscillations which lead to slower convergence. Note that this occurs when the LDS matrix A matches that of the nonlinear RNN - a choice made for analytic simplicity, when A is known. At relatively small depths Δ, it may be possible to achieve better approximation with a different LDS matrix A_Δ. In a practical learning setting, A_Δ is learned directly, without any reference to the unknown A.

Another Eigenvalue Parameterization

This is a simple trick to optimize over precisely the set of reachable linear dynamical systems, with the appropriate constraints on the eigenvalues.

A problem with the standard (α, β) parameterization of λ is that the number of real and complex eigenvalues is hardcoded. Two real eigenvalues cannot "cross over" to being complex conjugate pairs, and vice versa. To remedy this, we might consider independently parameterizing the real and imaginary parts of λ with $2n$ reals. Unfortunately, this does not constrain the complex numbers to be conjugate pairs, so then $\lambda_1, \ldots, \lambda_n$ are not necessarily the eigenvalues of a real matrix A. The following "hinge" parameterization, defined in terms of two real numbers (α, ω), avoids both of these issues. Let $h(a) = \max(0, a)$ be a ReLU. Consider these values:

$$\alpha + h(-\omega)i \qquad \text{and} \qquad \alpha + h(\omega) - h(-\omega)i$$

If $\omega > 0$, then the values simplify to α and $\alpha + \omega$, which are real. If $\omega < 0$, they simplify to

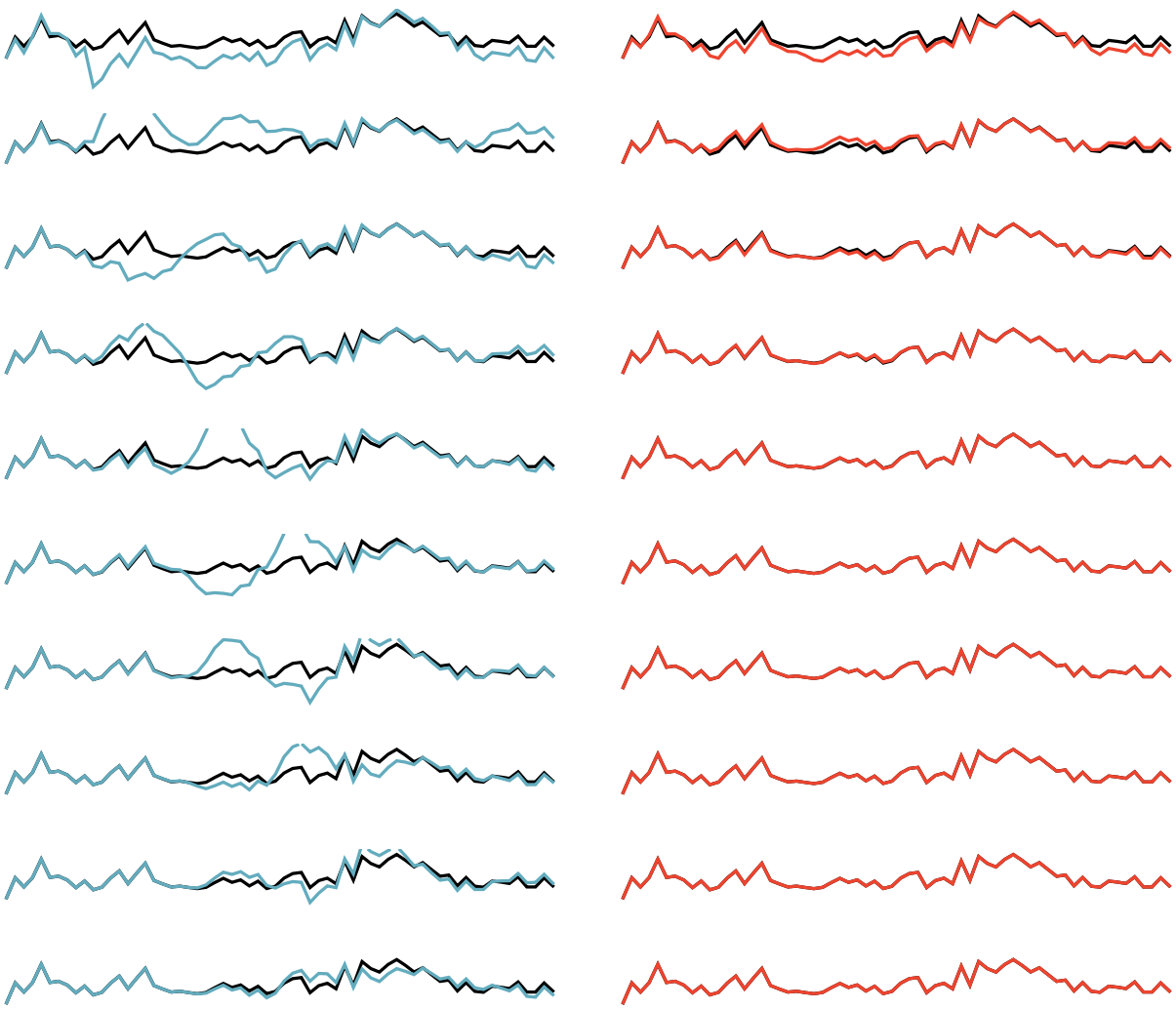


Figure 4.7: Additive and multiplicative approximations of a nonlinear RNN (black). The latter converge more quickly than the former, at least when the same matrix A is shared among the nonlinear RNN and the approximating LDS.

$\alpha \pm \omega i$, which are complex conjugate pairs. The values are distinct when $\omega \neq 0$.

4.7.6 Additional Experiment Details

In all the experiments, we used Adamax [Kingma and Ba, 2014] as the optimizer for LDS and LDStack. In some situations, we observed this choice substantially improved the rate of convergence. We used Adam as the optimizer for the LSTM and simple RNN. Abbreviate the learning rate and batch size as η and B, respectively. For the copy memory problem, $\eta = 0.01, B = 256$. For the runtime comparison, $n = 32$ and $B = 4$. For sequential permuted MNIST, $B = 128$. LDS used $\eta = 0.0003$, and the hinge parameterization described in Section 4.7.5. LSTM and simple RNN used $\eta = 0.01$. In the adding problem, $B = 32$ there were 100 steps per epoch. LDStack used $\eta = 0.003$ and the hinge parameterization. We observed faster convergence with a smaller $n = 32$ model LDStack than with a larger $n = 64$ one. LSTM and simple RNN used $\eta = 0.01$.

4.8 Discussion

This chapter explores a synthesis between nonlinear sequence-to-sequence models and linear dynamical systems, also known as state-space models. The results achieved in this chapter are among the most counterintuitive and interesting of the entire dissertation. For good reason, depth in neural networks is typically thought of as an architectural feature which inhibits analytical reasoning. However, an interesting approach in control theory indicates that, in some contexts, depth can actually facilitate analytical reasoning. This is because nonlinearity along time, which expresses complex dynamics, can be replaced by (approximations of) nonlinearity along depth, where deviations can be more easily bounded. Aside from analytical tractability, this replacement enables parallel computation along time, a crucial requirement of modern sequence-to-sequence models.

This chapter presents new architectural components for developing fast and trustworthy sequence models, based on the core primitive of SIMO LDS. Nonetheless, the specific constructions presented have significant technical limitations. Approximation guarantees for low-depth

stacks must be studied. This chapter does not closely examine algorithms for learning LDStack, even though RNNs suffer from the vanishing/exploding gradient problem. Finally, deep learning primitives are heavily optimized for GPUs [Chetlur et al., 2014]; our implementation requires similar treatment. Although LDStack scales well with T, its current implementation does not make efficient use of hardware: for example, memory use scales with depth. Reversibility and hardware-aware techniques could be exploited to address these issues.

Thus far, the dissertation has focused primarily on theoretical and methodological contributions. The next chapter, by comparison, is very applied and domain-specific. However, it is not disjoint from the theoretical work: it involves learning from long sequences of time-series data, which (chronologically) motivated the research of the present chapter. The application is in healthcare; this field's stringent requirements for rigorous and trustworthy algorithms inspired Chapter 2's foray into evidence-based medicine.

Chapter 5

Interpretable Deep Learning in Healthcare

Abstract

Brief, intense exercise can improve health due to its acute effect on the autonomic nervous system, particularly the sympathetic nervous system. Salivary amylase is a marker of sympathetic activity during exercise, but it requires specialized equipment to measure. This chapter investigates the feasibility of estimating the amylase response from heartbeat data recorded by commodity sensors. Heartbeat and amylase data are collected for $n = 71$ sessions of intense exercise performed in a commercial setting. A machine learning model exploits structure in the heartbeat signal: by identifying and removing the contribution of the parasympathetic nervous system, a residual with sympathetic information is obtained. Then, a convolutional neural network can be applied. This model has better accuracy than existing measures of exercise response, such as maximum heart rate, even though it doesn't use metadata such as age and gender. This suggests sympathetic activity may be (weakly) discerned from heartbeat data. With a larger dataset, a practical measure of sympathetic response to exercise could potentially be developed. This chapter's quantification of parasympathetic activity is more powerful than existing approaches and may have independent value. This chapter is based on the published work of Kaul et al. [2019].

5.1 Introduction

Intense exercise elicits a much different physiological response than moderate exercise. At low and moderate intensities, energy demands are satisfied by the oxidative (aerobic) pathway, and the initial increase in heart rate is accompanied by withdrawal of the parasympathetic nervous system. At high intensities, the glycolytic (anaerobic) energy pathway predominates; sweating, lipolysis, gluconeogenesis, and other characteristic responses are driven mostly by activation of the sympathetic nervous system rather than further parasympathetic withdrawal [Koistinen and Laitinen, 2004, Michael et al., 2017a, White and Raven, 2014]. Intense training has been shown to improve insulin sensitivity, blood pressure, aerobic capacity (VO_2max), and body composition [Batacan et al., 2017, Jelleyman et al., 2015, Jelleyman, 2018, Kessler et al., 2012, Milanović et al., 2015]. Even a single bout of intense exercise can have acute health benefits, such as enhancing glucose control [Jelleyman, 2018, Marliss and Vranic, 2002] or inhibiting the growth of colon cancer [Devin et al., 2019]. Intense exercise interventions are safe [Carl et al., 2016, de Jong et al., 2003, Wewege et al., 2018] and have the potential to improve outcomes for many patient populations [Elliott et al., 2015, Hannan et al., 2018, Jelleyman et al., 2015, Weston et al., 2014].

It is hard to measure if the desired response to intense exercise was actually achieved. Subjective measures, such as RPE (rate of perceived exertion) or RIR (repetitions in reserve), are common in athletic training. However, these measures are often unreliable in patient populations unaccustomed or indisposed to exercise [Aamot et al., 2014, Strzelczyk et al., 2001, Unick et al., 2014]. As previously mentioned, the activity of the sympathetic nervous system is of fundamental importance. There are many methods of measuring sympathetic activity, but none is considered a gold standard [Grassi and Esler, 1999]. The most common measure of sympathetic tone is the concentration of plasma epinephrine (a.k.a. adrenaline) or norepinephrine. Unfortunately, this requires invasive, confounding blood draws and complex laboratory analysis. The cardiac preejection period has been proposed as a valid, noninvasive measure of sympathetic activity [Michael et al., 2017a]. Unfortunately, it is recorded by bioimpedance cardiography, which is sensitive to the postural changes and heavy breathing that occur during exercise. Measurements can also be made at the periphery. Microneurography involves electrodes inserted directly into

muscle or skin nerves, which prohibits large movements during exercise [Vallbo et al., 2004]. Electrodermal activity (EDA) of sweat glands correlates with sympathetic activity during exercise [Boettger et al., 2010, Posada-Quintero et al., 2018]. However, peripheral measurements are not uniform across the body [Shoemaker et al., 2018], the most accurate measurement locations are not convenient [van Dooren et al., 2012], and are affected by local phenomena, such as vasoconstriction [Edelberg, 1964].

Salivary α-amylase (briefly, "amylase") has been identified as a marker of sympathetic tone, especially during exercise [Chicharro et al., 1998, Koibuchi and Suzuki, 2014, Nater and Rohleder, 2009]; (nor)epinephrine activates β_1-adrenergic receptors in the salivary glands, which causes granules of this enzyme to be released. Below a minimum threshold of intensity —– which seems to coincide with the accumulation of lactate in blood [Akizuki et al., 2014, Bocanegra et al., 2012, Calvo et al., 1997] —– the change in amylase is negligible. Above that threshold, it rises proportionally with intensity [De Oliveira et al., 2010, Li and Gleeson, 2004]. Amylase is typically measured by immunoassay of saliva sampled by passive drool. It can also be immediately measured by point-of-care devices [Shetty et al., 2011]. However, both methods incur nonneglible marginal cost due to nonreusable materials. Because salivary flow rate changes during exercise [Bosch et al., 2011, Rohleder and Nater, 2009], the point-of-care devices are prone to substantial error, if used without careful adherence to protocol [Peng et al., 2016].

Practitioners have resorted to metrics based on heart rate, because it is cheap and practical to measure with commodity sensors. These metrics include average and maximum heart rate, the decrease in heart rate 60 seconds after exercise (HRR60), and the rate at which heart rate returns to baseline (HRRτ). With sensors capable of recording the times between individual heartbeats, an assortment of heart rate variability (HRV) metrics, such as SDNN, RMSSD, PNN50, and LF, can be calculated during or after exercise [Shaffer and Ginsberg, 2017]. Though these metrics may be useful for monitoring recovery, assessing fitness, or related tasks, their utility as measures of the sympathetic response to intense exercise is limited. LF — the weight placed on low-frequency ($0.04 - 0.15$ Hz) components of the Fourier decomposition of the heartbeat signal — was previously thought to reflect slower-responding sympathetic tone, but is now understood to be parasympathetically driven [Moak et al., 2007, Reyes del Paso et al., 2013, Thomas et al.,

2019]. During exercise, HRV reaches a near-minimum at a relatively low intensity, analogous to parasympathetic tone [Boettger et al., 2010, Michael et al., 2017a]. Following exercise, HRV recovery, unlike sympathetic withdrawal, is substantially delayed by duration [Michael et al., 2017c] and moderate intensity [Michael et al., 2017b]. Similarly, HRR60 primarily, though not entirely, reflects parasympathetic reactivation [Kannankeril et al., 2004]. HRRτ seems to have a stronger dependence on sympathetic tone [Buchheit et al., 2007]; in skeletal muscle, accumulated metabolites stimulate metaboreceptors, which in turn maintain sympathetic tone [Fisher et al., 2013]. HRRτ, along with the other recovery metrics, require monitoring from 10 minutes to hours after the cessation of exercise, during which upright posture and further activity may confound results.

Since intense exercise is hard to monitor, it is hard to prescribe. Exercise intensity is typically specified as a percentage of some unknown, estimated quantity, such as maximum perceived exertion, maximum heart rate, or VO_2max. The error in this estimate is amplified when prescribing intense exercise at 95% VO_2max rather than moderate exercise at 60% VO_2max. Such imprecision stokes lingering concerns of overexertion during intense exercise. Mann et al. [2013] review the large variation of physiological responses to "poorly standardized" exercise protocols. Imprecise dosing of exercise raises concerns of overexertion and results in worse health outcomes. The SMARTEX heart study [Ellingsen et al., 2017a] found intense exercise was not better than moderate exercise for rehabilitating cardiac-failure patients, because some patients did not exercise at the correct intensity. In their view, "tight control of prescribed exercise intensity and intended load increase was somehow lost in the translation from a small proof-of-principle study to a larger multicenter trial of the efficacy under conditions closer to standard clinical practice" [Ellingsen et al., 2017b].

5.1.1 Novel Contribution

This chapter investigates the feasibility of measuring sympathetic response without specialized equipment. Using machine learning, it develops an algorithm which estimates the change in amylase enzyme activity (from before exercise to after, denoted as ΔAmylase) from heartbeat data readily collected during exercise. This is a supervised regression problem: the input is an

arbitrary-length sequence of interbeat intervals, and the label is ΔAmylase, a real value with units U/mL. We also consider the associated comparison problem: determining if one workout resulted in greater ΔAmylase than another workout.

To train and evaluate the algorithm, a realistic, albeit noisy, dataset is collected. Previous studies were conducted in laboratories or elite athletic settings, involved a narrow population of participants, exerted tight control over their diet and schedule, and specified a small number of exercises (primarily cycling). This new dataset is collected in commercial group training sessions, involves a diverse group of participants, observes them in their day-to-day exercise routine, and involves a wide variety of exercises. However, there may be substantial noise in both the heartbeat data (due to sensors slipping in full-body exercises) as well as the amylase measurements (due to changes in salivary flow and inadvertent misuse of equipment). Whereas laboratory settings have a tendency to produce optimistic results, this study is designed to produce pessimistic ones, reflecting practical realities of actual application.

This chapter's machine learning model is informed by the physiology of the autonomic nervous system. The parasympathetic system contributes a substantial, high-frequency component to the heartbeat signal. Heart rate variability (HRV) is a fairly reliable indicator of parasympathetic activity. By removing an HRV-derived component from the heartbeat signal, we obtain a residual with information about sympathetic activity. Multiple HRV metrics may be computed from second moments of the interbeat intervals. To quantify the parasympathetic contribution, rather than using an existing HRV metric, we allow more general, parametrized metrics of the same underlying moments. The generalization is mild enough to still consider them variability metrics rather than arbitrary statistics. The parameters of this metric are (pre)trained upon the data. Subsequently predicting ΔAmylase from the residual is a straightforward application of convolutional neural networks.

Clinical Relevance. A practical measure of sympathetic response could greatly improve the clinical practice of intense exercise. For doctors and researchers, a measure could ease the development and specification of intense exercise regimens. For patients and athletes, it could help ensure that improvements to health and fitness are actually being made. This chapter initiates the study of this problem in the hope of motivating further research. On the new dataset, heart

rate (the predominant metric) is no better than random guessing as a measure of sympathetic response. The bespoke model achieves a modest but discernible improvement. However, the dataset is too small ($n = 71$) to train a practically usable model.

Technical Significance. This chapter restores some optimism that heartbeat data contains information about sympathetic activity. Its novel approach of eliminating parasympathetic influence from the heartbeat signal proves useful; a naive application of CNNs is not as accurate. Its generalization of HRV metrics may have some independent utility for quantifying parasympathetic tone in the context of other applications.

5.1.2 Outline

Section 5.2 describes the cohort and the data collection process. Section 5.3 presents basic statistics of our dataset. These suggest the estimation problem is challenging, and motivate the use of machine learning. Section 5.4 presents the bespoke machine learning model, and compare it to previous work. Section 5.5 examines the results of fitting the model to the data. Section 5.6 reviews our findings and offer guidance for future research.

5.2 Study Design

The first goal of the study is to collect a dataset in which the participants and their activities are realistically observed rather than tightly controlled. This makes the data noisier, and in turn makes the estimation problem harder. However, a model trained upon this data has a better chance of transferring to practical use. Importantly, the goal of the study is just to estimate amylase, not to bolster its physiologic validity as a measure of sympathetic response.

5.2.1 Cohort Selection

The participants in this study are members of a commercial fitness facility in New Jersey. This membership is evenly split between genders and has an average age of 45, not including minors who are not eligible for study. Approximately 80% of the members are white and 20% of the members are of another race. No competitive or professional athletes are included.

Each exercise session is performed as part of an hour-long, instructor-led class. The workout is relatively brief, typically between 10 and 15 minutes. It is preceded by a warmup involving dynamic exercises, static stretching, and weightlifting practice. The workouts involve a wide variety of exercises, including bodyweight movements, rowing, cycling, running, gymnastics, dumbbell exercises, and barbell lifts. Most of the workouts are driven by one of two goals: to complete the workload as quickly as possible ("time priority"), or to complete as many rounds as possible within a given time ("task priority"). The workout prescriptions are generated semirandomly, to avoid repetition of exercises and to keep participants engaged. The workout prescriptions are just guidelines; participants make modifications, such as reducing the weight, so they can complete the workout.

Most of the workouts are intended to be intense, though longer workouts are necessarily less intense than the shorter ones. A small fraction of the workouts focus on "assistance" exercises which are performed more gingerly with large rest periods and lower overall difficulty. Relatively low ΔAmylase is expected for these workouts. They may be thought of as informal controls, with relatively low changes in amylase. Another portion of the workouts are performed competitively while being judged; relatively high ΔAmylase is expected for these.

5.2.2 Data Collection

The study began in February 2019 and lasted 4 weeks. Participants consented to the study after being informed of all aspects of the data collection process. During their first session, they were instructed how to use the necessary equipment. Participants arrived at class as they usually would; we did not control their food intake, time of day, or any other factors. However, they were instructed to arrive 10-15 minutes prior to class in order to initiate data collection. Participants performed their own measurements, having been individually instructed on proper protocol. (Since they exercised throughout the day, it was not feasible to supervise them at all times.) Roughly 5 minutes before the class started, they measured their amylase by inserting a saliva measurement stick (pictured in Figure 5.1) under their tongue for 30 seconds. They took special care to saturate the stick with saliva, to avoid problems with salivary flow rate. The stick was then read with a portable colorimetric meter [Shetty et al., 2011]. They then securely wore a

Figure 5.1: Data collection equipment. Left: pouches of saliva swabs. Center: point-of-care salivary amylase meter, with a single saliva swab. Right: Polar H10 cheststrap heart rate monitors.

Polar H10 (single-lead ECG cheststrap) heart rate monitor. The HRM connected to their personal smartphone via Bluetooth. To ease data collection and improve compliance, we implemented a custom iOS application, in which the participant could enter all their information, and the HRM could log its data. The interface of this application is presented in Figure 5.2. For the entire class, users remained within 15 meters of their phone, to prevent the HRMs from disconnecting. Roughly one minute after the main portion of the workout, prior to cooling down, the participants gauged their rate of perceived exertion, and again measured their amylase. Participants noted if any unusual circumstances befell the workout or their preparation for it; these include illness, injury, a high temperature in the gym, or an atypically large dose of caffeine. Finally, users submitted all of the data through the application. Using the start and end times of the data collection, we ensured that data was collected continuously.

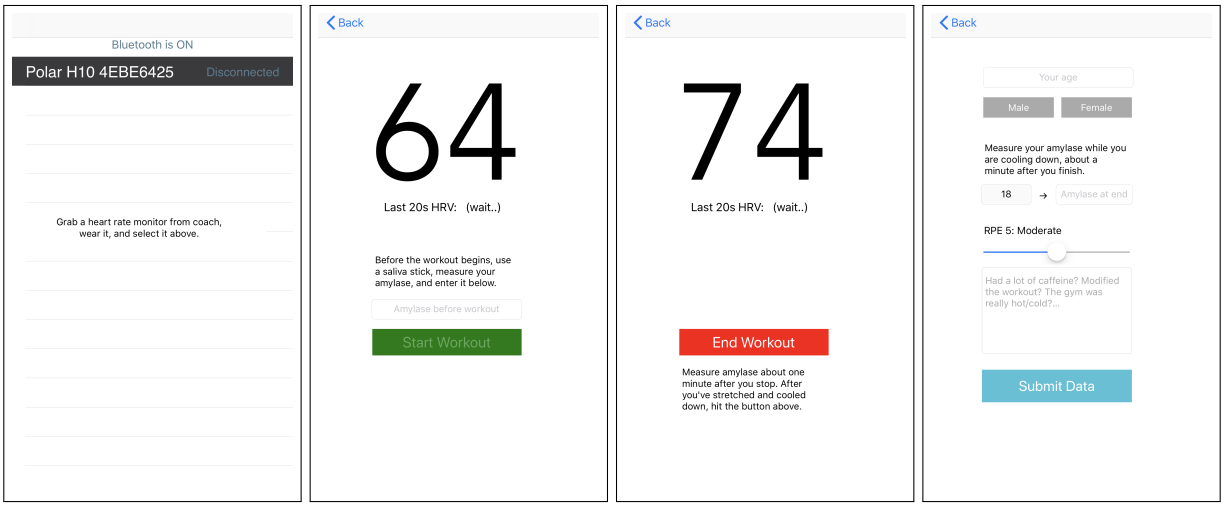


Figure 5.2: Screens of the iOS data collection application used in the study. **A**: The user connects to a Bluetooth heart rate monitor. **B**: Heart rate is prominently displayed. Starting amylase is recorded, and the green "start workout" button is pressed. **C**: Heart rate is continuously updated during the workout, after which the red "end workout" button is pressed. **D**: The remainder of the data (age, gender, ending amylase, and RPE) are collected.

Total workouts	85
User errors	3
HRM errors	2
App. errors	6
Too much noise	3
Dataset size	71

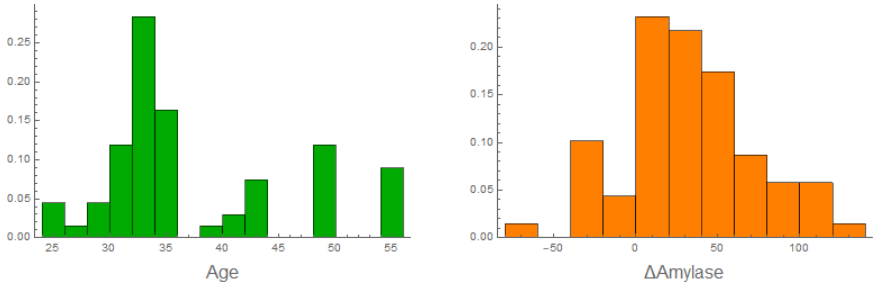


Figure 5.3: Basic statistics of the dataset. As the table shows, a substantial number of errors were encountered during data collection. The mean age is 37. The mean absolute magnitude of ΔAmylase is 40. The mean absolute deviation of ΔAmylase is 29.4.

Samples were excluded on the following grounds. (1) User error: the user did not follow proper protocol (e.g. forgot to measure their amylase at the cessation of exercise). (2) HRM error: the heart rate monitor prematurely disconnected from the phone. (3) Application error: an unknown technical error was encountered while using the iOS application. (4) Too much noise: the heart rate monitor slipped or malfunctioned, resulting in excessively noisy measurement.

5.3 Basic Data Preprocessing and Analysis

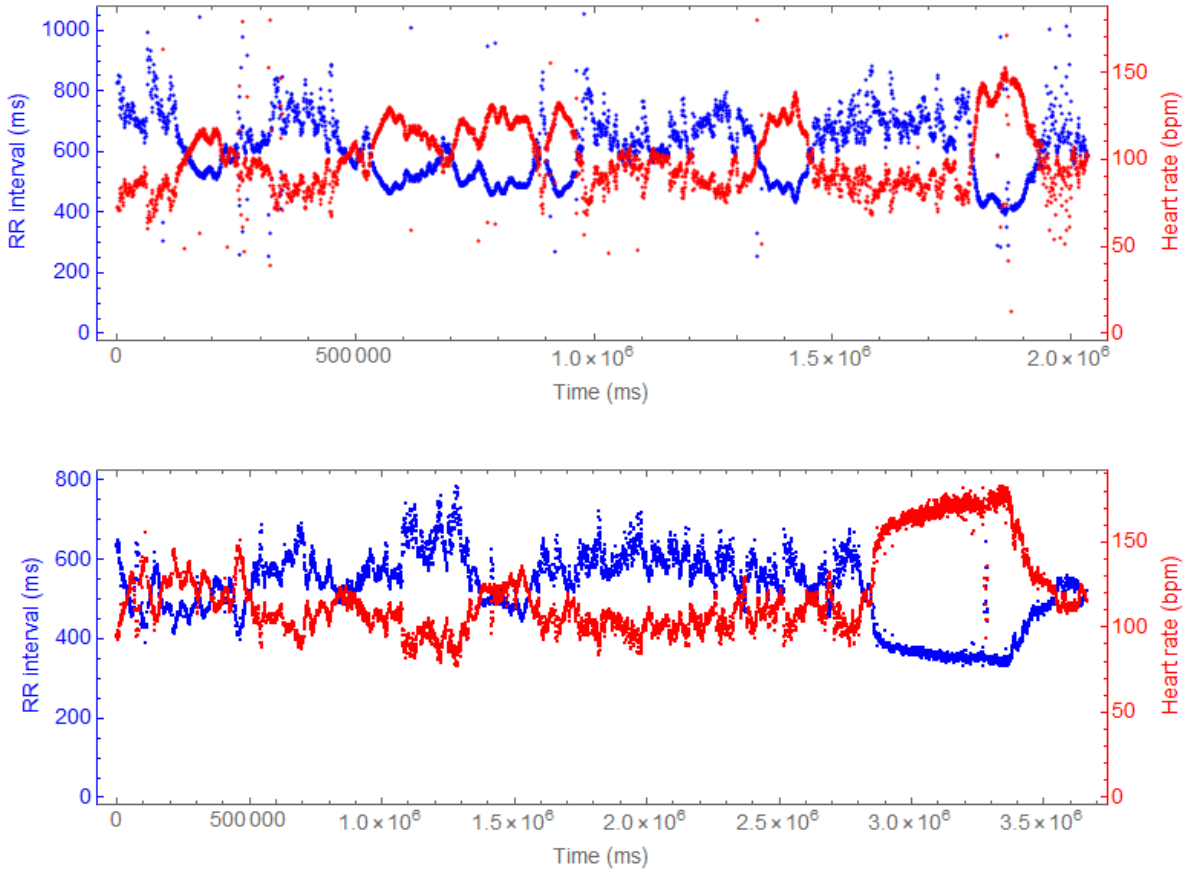


Figure 5.4: *Top*: a tachogram of the raw heartbeat data of a single workout. RR intervals are recorded, and converted to instantaneous heart rate via the equation HR $= 60000/\text{RR}$. The workout is 30 jump-rope double-unders and 15 dumbbell snatches, for as many rounds as possible within 10 minutes. The main working portion is near the end of the session, and is preceded by a substantial warmup. The data are presented without any smoothing or noise filtering. The large amount of noise may be partially attributed to jumping. *Bottom*: the previous tachogram processed by the noise filtering algorithm.

In total, 71 workouts were successfully recorded from 19 participants. Though small by the standards of modern machine learning, this is relatively large compared to previous studies. It is enough to cover a wide variety of exercise plans performed at varying degrees of intensity. In this section, we review the salient features of the dataset, and thereby gain some intuition for the design of the machine learning model.

5.3.1 Noise

As seen in Figure 5.4, the RR interval data are noisy. The noise consists primarily of isolated, falsely-ectopic beats which spike below or above the otherwise-smooth curve. Ectopic beats are known to impede calculation of heart rate variability [Lippman et al., 1994]. Accordingly, we eliminated the ectopic beats with the following noise filtering algorithm. First, we calculate a median filter on the entire RR interval sequence using a window size of 24. Next, we measure the relative deviation of the original RR interval from the filtered value. If this is more than 20%, then we consider the RR interval ectopic and replace it with the filtered value. If more than 15% of the beats are ectopic, then we exclude the entire sample from subsequent analysis. Three samples were excluded in this manner.

The ΔAmylase measurements are similarly noisy. Immediately following intense exercise, participants were sometimes forgetful or simply too tired to completely adhere to the measurement protocol. On occasion, saliva measurements were performed in duplicate. Based on these measurements, we estimate that the inherent noise in the ΔAmylase values is roughly 10. For similar reasons, RPE was essentially ignored; participants rarely changed it from its default value of 5. No such omissions were made for age or gender.

5.3.2 Heart Rate Metrics and $\triangle$Amylase

The most basic metric derived from RR intervals is heart rate, via the equation $\text{HR} = 60000/\text{RR}$. Beyond heart rate, heart rate variability (HRV) measures the activity of the parasympathetic nervous system in terms of the variation of time between heartbeats. Higher variation corresponds to higher parasympathetic activity. The most commonly used HRV statistics are RMSSD (root mean square of successive differences) and SDNN (standard deviation of normal-to-normal intervals). The equations in Section 5.4.3 formally define these statistics; for further detail, see Shaffer et al. [2014] or Shaffer and Ginsberg [2017] for reviews of these statistics. In Figure 5.5, we display, for a variety of workouts, SDNN computed over sliding windows of 24 beats.

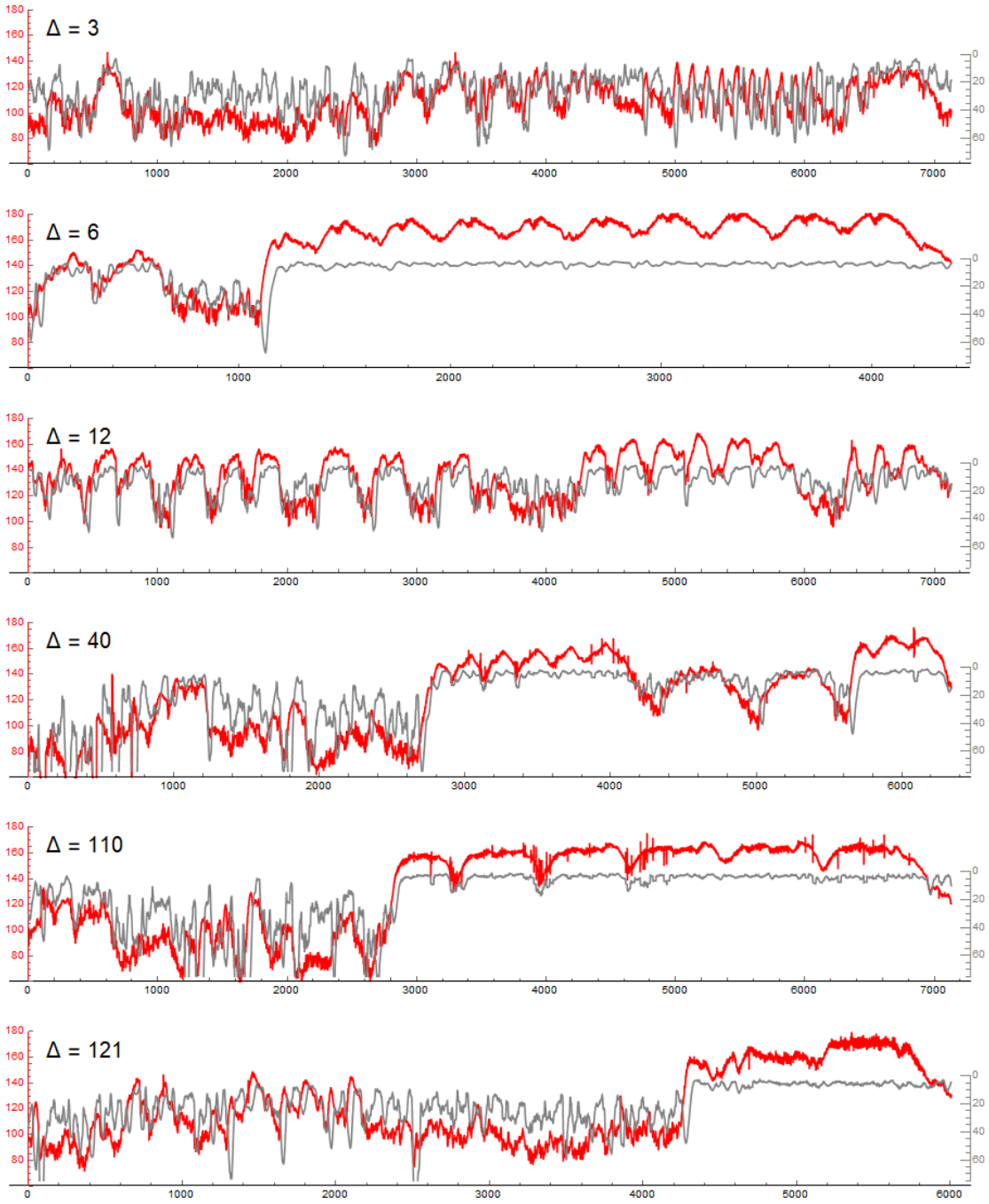


Figure 5.5: Heart rate (red) and SDNN (a measure of parasympathetic activity, gray) compared for workouts with different amylase changes (given by Δ in the top left). The vertical axis of SDNN is flipped for visual clarity. All of these workouts, except for the one with $\Delta = 6$, illustrate an interesting pattern. When Δ is low, changes in HR are mirrored by changes in SDNN. When Δ is high, HR and SDNN become decoupled: SDNN remains flat while HR may continue to increase. This manifests as the red line spiking above the gray line. HR changes that do not coincide with SDNN changes may be sympathetically driven.

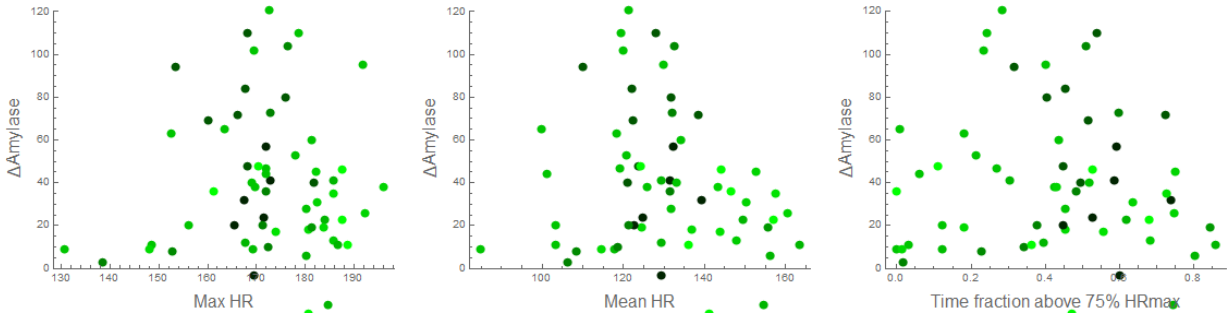


Figure 5.6: Various heart rate metrics, age and ΔAmylase do not seem to correlate. Bright green denotes young age, and black denotes old age. Five workouts resulted in negative ΔAmylase; these remain in the dataset, but are clipped off the plot for visual clarity. Maximum heart rate is estimated as HRmax $= 208 - 0.7 \cdot$ Age [Tanaka et al., 2001]. Heart rate metrics are not a good measures of sympathetic response, even taking age into account.

As expected, there does not seem to be a discernible correlation between basic heart rate metrics and ΔAmylase, even when controlling for age. Figure 5.6 illustrates the lack of correlation. This is consistent with the consensus that heart rate is not a reliable measure of the response to exercise, at least across individuals. While these metrics cannot be directly used to predict ΔAmylase, Figure 5.5 illustrates an interesting possibility of using SDNN to isolate sympathetic activity. We will examine this relationship more closely in Section 5.4.2, and ultimately use it to derive a custom machine learning model.

5.4 Machine Learning with a Structured Model

5.4.1 Problem Formulation

Let the length-T sequence of interbeat RR intervals be $x = [x_1, \ldots, x_T]$. Each $x_t \in \mathbb{R}^+$ has millisecond units. So, if R-peak t occurs half a second after R-peak $t-1$, then $x_t = 500$. Let ΔAmylase $\in \mathbb{R}$ (now abbreviated as just Δ) be the difference in amylase incurred during exercise. The goal is to find a function f which minimizes the following mean absolute error in predicting Δ (left). We also consider the induced pairwise comparison problem (right):

$$\mathbf{E}\,|f(x) - \Delta| \qquad\qquad \mathbf{P}\,((\Delta < \Delta') \iff (f(x) < f(x')))$$

In the present context, these problems are challenging for the following reasons:

- The sequences are long — typically longer than 7000 elements — and are highly nonstationary, with long-range dependencies. This precludes the use of many RNNs, which are typically trained on short segments, lest they become a serial computational bottleneck.
- The signal-to-noise ratio is low. As discussed in Section 5.3.1, there is noise in both the inputs and the outputs. There is only a single point of supervision Δ for each sequence x.
- Age, gender, and starting time are excluded from the predictive model. We do this to avoid overfitting, and to see if ΔAmylase can be estimated from heartbeat data alone.

5.4.2 Key Intuitions

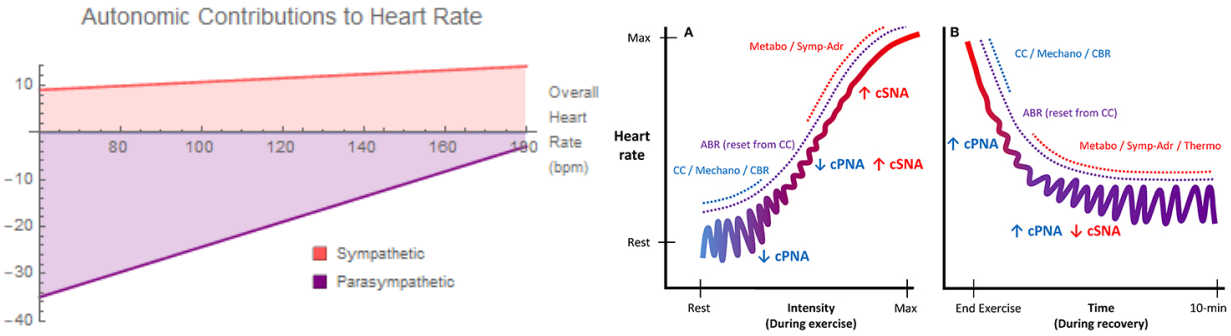


Figure 5.7: Schematic relationship of the parasympathetic (PS) and sympathetic (S) nervous systems during intense exercise. The left figure, following White and Raven [2014], shows that PS withdraws mostly at moderate intensities, and S activates mostly at high intensities. The right figure, from Michael et al. [2017a], shows that PS is faster-acting than S; it withdraws before S activates, and reactivates before S withdraws.

Let us motivate the design of the machine learning model by examining the relationship between HR and HRV in Figure 5.6. Large gaps between HR and (inverse) HRV seem to be a necessary condition for high ΔAmylase. That is, high ΔAmylase seems to be contingent upon sudden increases in HR despite no decrease in HRV. In workouts with low ΔAmylase, changes in HR seem to be mirrored by changes in HRV. It makes some visual sense to subtract (inverted) HRV to obtain the S activity that would explain the unaccounted HR changes. This idea makes some physiologic sense as well, due to the complementary relationship between the PS and S systems (Figure 5.7) . The physiology suggests another telltale clue: HRV increasing while HR

remains elevated.

The overall model takes the form $f(x) = s(h(x) - (h \circ p \circ v \circ z)(x))$, where z are squared differences of x, v is a generalized HRV metric, p is the pretrained parasympathetic layer, $h(r) = 60000/r$ converts from RR intervals to instantaneous HR, $h(x) - h(p)$ is the residual, and s is the sympathetic layer. The model architecture is described in more detail below.

5.4.3 Pretrained Parasympathetic Layer

To eliminate the parasympathetic influence on heart rate, it is tempting to simply subtract an existing HRV metric from heart rate. However, this is not quantitatively satisfactory for multiple reasons. First, it is not clear which HRV metric to use. Second, doing this in the manner of Figure 5.6 would involve translating and scaling the metric; these operations would have to be calibrated to the data. Finally, simply subtracting HRV may not be the best way of removing information from the signal. Ideally, the residual should be independent of parasympathetic activity, in that predicting the former from the latter should not be possible. Rather than attempting to ameliorate these issues by hand, we employ machine learning.

Squared differences. We observe that the most commonly-used HRV metrics can be expressed in terms of the squared differences $z_{i,j} = (x_i - x_j)^2$. This is obvious for RMSSD and PNN50. In the following equation for SDNN, $\mu = \frac{1}{m}\sum_{i \le m} x_i$ is the mean, and we invoke the usual decomposition of variance.

$$\begin{aligned}
\text{RMSSD}^2 =& \frac{1}{m}\sum_{i<m}(x_{i+1} - x_i)^2 = \frac{1}{m}\sum_{i<m} z_{i,i+1} \\
m^2 \cdot \text{SDNN}^2 =& m\sum_{i \le m}(x_i - \mu)^2 = m\sum_{i \le m} x_i^2 - m^2\mu^2 \\
=& \frac{1}{2}\sum_{i,j} x_i^2 + x_j^2 - 2x_i x_j = \frac{1}{2}\sum_{i,j} z_{i,j} \\
\text{PNN50} =& \frac{1}{m}\sum_{i \le m} 1\left(|x_{i+1} - x_i| \ge 50\right) = \frac{1}{m}\sum_{i \le m} 1\left(z_{i,i+1} \ge 50^2\right)
\end{aligned}$$

Put another way, the squared differences are sufficient statistics for HRV. However, these statistics do not capture all the information in the heartbeat signal. This restriction is beneficial because

if the (purportedly) parasympathetic model overlaps too much with remaining layers, it becomes less interpretable, and may fail its purpose of isolating a useful residual. We will soon see why squared differences, rather than the more typical mean and moments $x_i x_j$, are being used.

> *This generalization of heart-rate variability could have broader applications in electrocardiology and psychophysiology.*

Generalized HRV metric. The architecture of this part is as follows. For each window of size m, with stride 1, compute z. Optionally apply a nonlinearity for statistics like PNN50. Then, take a linear combination of the entries or z. The present model initializes this to the constant $1/(2m^2)$ matrix, to compute SDNN. It uses a window size of $m = 24$, which is considered an ultra-short-term measure of HRV.

Parasympathetic contribution. Let us momentarily ignore the unit conversion h. To make $x - p(v)$ difficult to predict from v, pretrain p to predict x from v, by minimizing mean squared error. In this way, $\tilde{x}$ becomes the unpredictable part of x. We expect this prediction will be difficult and that $x - p$ will be substantial. The HRV metrics could have also been written in terms of the mean and second moments $x_i x_j$, but this data would allow x to be easily recovered. The squared differences do not reveal much about the mean of x. This makes $p(v(z(\cdot)))$ a kind of autoencoder.

Now let us examine the use of h. Because x and p have units of RR intervals, there is an implicit bias for p to more closely fit low-intensity periods, which is when we expect more parasympathetic activity. This is because when HR is high, RR intervals are small, so the squared error is limited. For the sympathetic layer, we want the opposite numerical tendency, so we use units of heart rate. Computing $h(x - p)$ leads to near division-by-zero where x is close to p. Instead, we use $h(x) - h(p)$.

Given this functional overview, we can finally settle on the actual architecture of $p(v)$. Since the parasympathetic system is high-frequency and fast-acting, it uses 3 layers of convolutions of size 4 with linear activation.

5.4.4 Sympathetic Layer

The sympathetic layer is designed to be sensitive to the sharp increases and prolonged elevations coincident with sympathetic response. These sequential, spatially-local patterns can be recognized by one-dimensional convolutions. To increase the receptive field (i.e. to allow for some amount of sequential dependence), dilated convolutions are employed [Oord et al., 2016]. In particular, the model stacks a block of dilated convolutions, each followed by a standard convolution with a stride length of 2, which halves the output dimension. To mute the output in uninteresting regions, gated convolutions are used, which doubles the number of convolution parameters [Dauphin et al., 2017]. Together, the convolution layers output a single response value for each segment. The cumulative sympathetic response is just the sum of the responses for each segment. This locality assumption is plausible, but could be reexamined in future work.

Compared to the parasympathetic layer, the sympathetic layer has a generic architecture. This is appropriate because the theory of how the sympathetic nervous system affects heartbeat intervals is much less well-developed. Though the chosen architecture seems to make the sympathetic layer work well, it is possible that other choices may work better.

5.4.5 Implementation Details

The experiment splits the dataset into training and testing sets of equal size. To avoid imbalances, it rejects splits where the train and test means of ΔAmylase differ by more than 5. (Due to the small size of the dataset, this is a potential concern). Variable-length sequences are either zero-padded or left-truncated to a uniform length of 7000.

The model is implemented in TensorFlow 1.13. Both pretraining and training use the Adam optimizer with the default learning rate. For pretraining, the entire training set is used in each batch; training uses just a single example in each batch. Both pretraining and training run for 60,000 steps. During training, the parasympathetic layer's parameters are frozen. For both the regression and comparison problems, the ℓ_1 loss is used. To examine the potential of over-parametrization, the number of parameters in the sympathetic layer were informally varied.

5.4.6 Related Work

Most work on classifying cardiologic signals starts with nearly-continuous ECG waveforms sampled at approximately 200Hz [Hannun et al., 2019, Lehman et al., 2018]. The goal is often to detect arrythmia or determine risk of myocardial infarction. We use coarser RR interval data for the following reasons. It isn't possible to access raw waveforms from consumer-grade monitors; the Bluetooth standard provides only for heart rate and RR intervals. During intense exercise, the waveform is likely to be extremely noisy, since even the extracted RR intervals are noisy. Lastly, sympathetic response is relatively slow compared to parasympathetic response and other ECG dynamics, so the time scale of RR intervals is more appropriate.

Most algorithms for assessing autonomic function from RR intervals originate from the field of signal processing. The algorithms are handcrafted, not learned, for the sake of simplicity, interpretability, and computational efficiency. Even though RR interval data is relatively abundant, machine learning algorithms which process them are somewhat uncommon. In 2002, the PhysioNet challenge involved generating artificial RR interval sequences and discriminating them from real ones [Moody, 2002]. Tsipouras et al. [2005] classify heartbeats using a handcrafted classifier which sequentially operates on windows of RR intervals. Gjoreski et al. [2017] employ a 7-layer fully-connected ReLU network upon the raw RR intervals. [Faust et al., 2018] apply an LSTM to detect atrial fibrillation. Asl et al. [2012] extracts time-series features before using a neural network.

The extraction of a sympathetic residual should not be confused with (and in fact is diametrically opposed to) residual networks in deep learning [He et al., 2016b]. The layers of these networks have skip connections which add the original input to their transformation of the input. With a skip connections, the parasympathetic layer would output $x + p$ rather than just p. Skip connections seem to ease optimization because, if many such layers are stacked, then each individual layer can be very close to an identity transformation, with each p modeling a slight change (or "residual") of the input. By contrast, the point of our approach is to eliminate the extraneous parasympathetic component from the original input.

Various consumer devices calculate proprietary scores for workouts. Some of these scores purportedly measure the response to intense, anaerobic exercise. Since they do not correspond

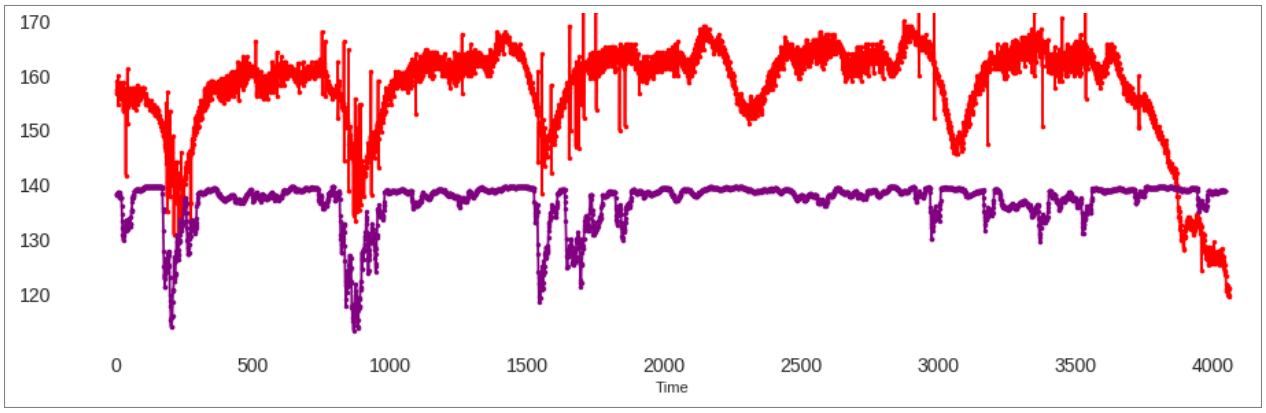


Figure 5.8: A single example of heart rate $60000/x$ in red, along with the learned parasympathetic contribution $60000/p$ in purple. As described in Section 5.4.2, this parasympathetic contribution is subtracted from the signal to isolate the sympathetic component. On this example, the parasympathetic contribution visually appears to be a refinement of simple thresholding at roughly 150 BPM, which is popularly thought of as a demarcation between low-intensity and high-intensity exercise. It may be compared with SDNN in Figure 5.5, but it is quantitatively superior to all such baseline HRV metrics.

to any real physiologic quantity, it is difficult to assess the validity of these scores relative to a gold standard. It is also difficult to assess the relevance of these scores to health outcomes. It is easier to reason about the outcomes associated with amylase response, since it is part of the neuroendocrine system.

5.5 Machine Learning Results

First, we examine the results of pretraining. The quantitative error incurred during pretraining is not pertinent, so we examine some of the qualitative aspects of the learned features. In Figure 5.8, we see that parasympathetic contribution roughly accords with SDNN in Figure 5.5, but is more steady at higher HR. In Figure 5.9, we see that the learned HRV metric is substantially different than the known ones.

Next, we examine the accuracy of the algorithm on the regression and comparison problems. As a baseline method, we consider a plain CNN, whose architecture is the same as the sympathetic layer. We also consider what happens if we didn't pretrain the parasympathetic layer, but merely subtracted RMSSD from HR. For the comparison problem, we consider using maximum

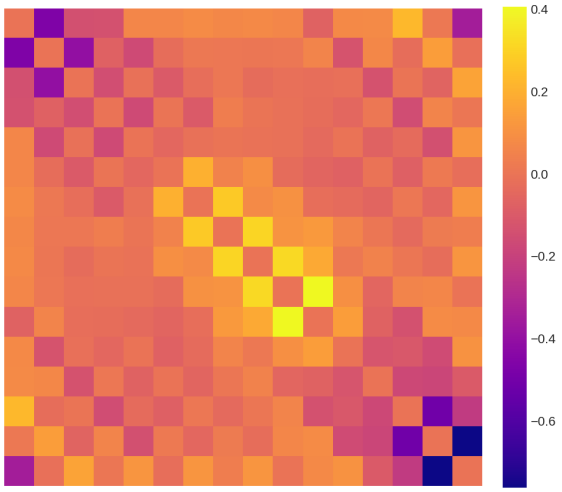


Figure 5.9: The matrix of weights on the squared differences $z_{i,j}$. Each entry was initialized to the same value, but after training, there is clearly nonuniformity. The algorithm seems to take advantage of the flexibility afforded by the additional parameters, with a nontrivial weight pattern, and both positive and negative weights.

heart rate as a ranking. We find that the novel algorithm is superior to all of these baseline approaches. This suggests that both our modeling effort and pretraining are worthwhile. However, the accuracy of all the methods is still relatively poor. This is likely due to the small amount of training data.

Algorithm	Training Error		Testing Error	
	Regression	Comparison	Regression	Comparison
Plain CNN	**8.02**	**0.11**	26.29	0.53
No Pretrained PS	12.28	0.37	19.21	0.41
Our Algorithm	10.04	0.28	**15.12**	**0.34**
Max HR	–	0.48	–	0.52

Figure 5.10: Quantitative evaluation of the algorithm. Our algorithm is superior in both the regression and comparison problems.

5.6 Discussion

This chapter explores a synthesis between traditional, handcrafted electrocardiological statistics and statistics that are learned from labeled data for a specific predictive purpose. It defines a family of statistics which generalize the handcrafted ones and embed parameters which can be learned by gradient descent. This chapter is satisfying because its physiologically-informed approach leads to better predictive performance than a naive one based on more flexible models. The design of the generalized statistics imparted an empirically-helpful inductive bias. Furthermore, the new statistics preserve some of the interpretation and intuitions surrounding the traditional ones.

Overall, this chapter made the following contributions. (1) It initiated the study of a novel supervised learning problem. (2) It describes the collection of a dataset large enough to conduct a pilot study. (3) It generalizes HRV metrics to allow them to express parasympathetic contribution to heart rate. (4) It trains and evaluates a physiologically-informed machine learning model which outperforms baseline methods used by practitioners. Our main result is that Δ*Amylase is (weakly) discernible solely from heartbeat data.* This is supported by the nontrivial regression and comparison accuracy of our model. A secondary result is that *it seems easier to predict* Δ*Amylase after attempting to subtract parasympathetic contribution*. It is possible that that this is simply due to the larger number of parameters in the pretrained model. However, this is unlikely, since the sympathetic layer already has a large number of parameters, and its performance was not substantially affected by adding more parameters.

The following limitation must be recognized: *the validity of* Δ*Amylase as a marker of exercise intensity is outside the scope of the study*. This question is examined by previous works mentioned in the introduction. Importantly, the validity of ΔAmylase may depend on the patient population. For example, cardiac rehabilitation patients are often prescribed β-receptor antagonists, which inhibit both heart rate and likely amylase response. We are presently focused on the purely quantitative prediction task. Furthermore, considering the small size of our dataset, this chapter should be considered a pilot study. It identifies, formalizes, and initiates the study of a machine learning problem, but does not adequately solve it. Accordingly, we offer a methodological suggestions for future studies: *Point-of-care devices are not recommended for large-scale*

use. Delicate use of the devices is necessary, and probably would not occur without careful instruction and/or supervision.

The entire dissertation has thus far focused on improving predictive models for preexisting problems. But machine learning also involves formalizing and understanding the problems that should be solved in the first place. This is especially true in modern machine learning, where desiderata besides speed and accuracy abound. The next chapter investigates how classical computational considerations interact with modern desiderata of fairness.

Chapter 6

Towards Computationally-Tractable Multi-Group Fairness

Abstract

This chapter uses the geometric quantity of margin — the distance between a decision boundary and a classified point, or the gap between two scores — to formalize the principle of equal opportunity: the chance to improve one's outcome, regardless of group status. This approach recognizes, for example, that a strongly rejected individual was offered less recourse than a weakly rejected one, despite the shared outcome. It also leads to simpler algorithms, since continuous margins are easier to analyze and optimize than discrete outcomes. This chapter formalizes two ways that a protected group may be guaranteed equal opportunity: (1) (social) mobility: acceptance should be within reach for the group (conversely, the general population shouldn't be cushioned from rejection), and (2) contrast: within the group, good candidates should get substantially higher scores than bad candidates, preventing the so-called 'token' effect. A simple linear classifier seems to offer roughly equal opportunity both experimentally and mathematically. This chapter is based on the published work of Kaul [2018].

In machine learning, the outcome of a candidate x is often determined by a real-valued score $s(x) \in [-1, 1]$. A deterministic classifier $c(x) = \mathrm{sgn}(s(x)) \in \{-1, 1\}$ uses the sign of the score to determine whether the individual is accepted or rejected. A randomized, confidence-based classifier returns $\mathrm{sgn}(s(x))$ with probability $|s(x)|$, and guesses randomly otherwise. An accurate classifier minimizes the probability of misclassification $\mathbf{P}\left(c(x) \neq y_x\right)$ relative to the correct outcomes $y_x \in \{-1, 1\}$. In ranking, the score is used to compare candidates. An accurate ranking maximizes the probability of ranking a good candidate x higher than a bad candidate x': $\mathbf{P}\left(s(x) > s(x')\right)$.

Since discrete optimization problems are typically harder than their continuous variants, underpinning outcomes by scores is computationally expedient. The continuous optimization problems are often based on a quantity called the margin: a distance in either the input space (of x) or the output space (of $s(x)$). In the input space, this is a distance between x and the decision boundary. (For a linear classifier $c(x) = \mathrm{sgn}(\langle w, x\rangle)$, this typically refers to $|\langle w, x\rangle|$.) In the output space, $s(x) - s(x')$ is the margin by which x is ranked higher than x'.

Besides being accurate, a score should be fair. Suppose candidates belong to either a protected group Π or the general population Π^c; for example, Π may be an underrepresented minority. In classification, the most well-known definition of group fairness is demographic parity, which equalizes the acceptance rate of Π and Π^c. Rather than enforcing equal outcomes, this chapter focuses on fair process. It formalizes two aspects of equal opportunity as 'mobility' and 'contrast'. Before the formal discussion, here is some high-level motivation for the definitions. Suppose a candidate in Π is declined a job offer and seeks to improve her chance the next time she applies. If she can devote just a few hours per week to prepare, the *magnitude* of her effort is limited. Mobility allows candidates to become accepted through a reasonable amount of effort. Also, the candidate *directs* her effort by becoming more like her successful peers than the unsuccessful ones. Contrast ensures that good candidates have much higher scores (i.e. acceptance probabilities) than bad ones, which makes it easier to discern the underlying differences between good and bad peers. Since these guarantees should have the same strength for Π and Π^c (on average), the groups have equal opportunity.

Mobility and contrast are closely related to margins in input and output space, respectively.

This chapter adapts these quantities to capture equal opportunity while retaining their analytic tractability. As a result, we can prove that mobility and contrast (or at least precursors thereof) are offered by a very simple linear classifier computed by averaging the data. These results are validated on adult income data.

Notation. Let $\langle w, x \rangle = \sum_i w_i x_i$ be the inner product in n-dimensional Euclidean space $\mathbb{R}^n$. Let $X \subset \mathbb{R}^n$ be the set of all candidates; to ease notation, assume it has finite size $|X|$. Each candidate has a correct outcome y_x equal to either -1 ('bad') or 1 ('good'). The protected group is a subset $\Pi \subset X$, and the general population is the complement Π^c. We partition the good and bad members of Π:

$$\Pi_+ = \{x \in \Pi : y_x > 0\} \quad \Pi_- = \Pi \setminus \Pi_+$$

We similarly partition Π^c into $\prod_+^c$ and $\prod_-^c$. Let $c : X \to \{-1, 1\}$ and $s : X \to [-1, 1]$ be a classifier and score.

6.1 (Social) Mobility

Take a candidate $x \in \mathbb{R}^n$ and change them by adding $o \in \mathbb{R}^n$. The direction of the change o represents an 'opportunity' if it causes a rejected x to be accepted, or an 'offense' otherwise. The size of the change $||o||$ represents 'effort' to be accepted, or 'slack' to be rejected. The margin of a candidate x is the smallest $||o||$ such that $c(x + o) \neq c(x)$. This standard margin definition allows arbitrary o, which may correspond to unnatural or unlikely changes, and would be incompatible with the principle of equal opportunity:

> "Even if all are eligible to apply for a superior position and applications are judged fairly on their merits, one might hold that genuine or substantive equality of opportunity requires that all have a *genuine* opportunity to become [accepted]." [Arneson, 2015]

For example, if a classifier is biased towards males, females may not have mobility, because the 'opportunity' to change their gender is hollow. Such o are more commonly referred to as 'adversarial perturbations' which cause the classifier to err after minimal change of the input

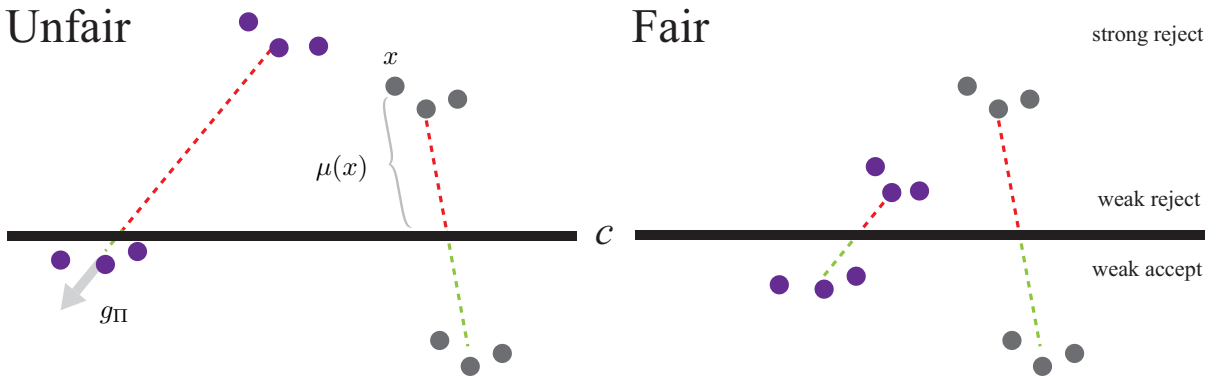


Figure 6.1: Suppose the horizontal line c classifies the protected group Π and general population Π^c perfectly; it is still unfair in the first scenario. Rejected members of Π are a far distance from acceptance, whereas those accepted are a close distance from rejection. By contrast, rejected members of Π^c aren't as far, and accepted ones are cushioned from rejection. This imbalance is rectified in the 'fair' scenario. μ_Π and μ_{Π^c} are, respectively for Π and Π^c, the average distance of accepted members minus the distance of rejected members. The corresponding directions g_Π and g_{Π^c} are thought of as 'genuine' opportunities, as explained below.

[Goodfellow et al., 2014, Hardt et al., 2016b].We restrict attention to the actual (i.e. present in the data) difference between good candidates and bad ones; this leads to the following definition of a 'genuine' opportunity vector.

Definition 5. *For Π, the genuine opportunity can be signified by the following vector:*

$$g_\Pi = \frac{1}{|\Pi_+|}\sum_{x\in\Pi_+} x - \frac{1}{|\Pi_-|}\sum_{x'\in\Pi_-} x' \tag{6.1}$$

The genuine opporunity vector for Π^c can be defined analogously as g_{Π^c}.

The genuine margin of x is its distance to the decision boundary along this vector. For rejected x, this is the effort, following the genuine opportunity, needed to become accepted.

Definition 6. *For any $x \in \Pi$, the genuine margin $\mu(x) \in \mathbb{R}$ is the smallest (in absolute value) ϵ such that*

$$c\left(x + \epsilon \cdot \frac{g_\Pi}{||g_\Pi||}\right) \neq c(x)$$

For any $x \in \Pi^c$, $\mu(x)$ is defined the same way, with g_{Π^c} replacing g_Π.

For linear classifiers, the genuine margin is easy to compute. For $x \in \Pi$:

$$c(x) = \text{sgn}(\langle w, x \rangle) \quad \Leftrightarrow \quad \mu(x) = \frac{\langle w, x \rangle}{|\langle w, g_\Pi / ||g_\Pi|| \rangle|} \tag{6.2}$$

For nonlinear c, it may be estimated by line search on ϵ. For each group, we consider the average genuine margin. This is positive if the group is cushioned from rejection, and negative if acceptance is beyond reach.

Definition 7. *The genuine margin of* Π *is the average of the genuine margins of its constituents:*

$$\mu_\Pi = \frac{1}{|\Pi|} \sum_{x \in \Pi} \mu(x)$$

Finally, mobility is defined as a group notion of fairness.

Definition 8. c *offers* Π *mobility if* $\mu_\Pi = \mu_{\Pi^c}$.

Mobility concerns input margins: how changes in x affect the discrete outcome $c(x)$. Dwork et al. [2012] instead bound the effect on the real-valued outcome, positing that similar individuals x and x' (with respect to the distance $||x - x'||$) should have similar outcomes: $|s(x) - s(x')| \leq ||x - x'||$. Fish et al. [2016] equalize acceptance rates between Π and Π^c by reclassifying candidates who were perhaps likely to be misclassified anyway: those having small margin. Zafar et al. [2017b] prevents indirect use of sensitive features used by limiting their correlation with the (signed) margin. Luong et al. [2011] impose this requirement on nearest-neighbor classifiers.

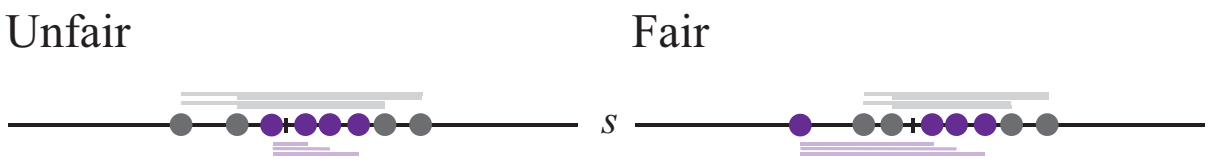


Figure 6.2: Scores (with zero marked in the middle) for the protected group and the general population. In both scenarios, the protected group has a higher acceptance rate, since more candidates have positive score. Nonetheless, the left scenario is unfair because good candidates receive nearly the same scores as bad ones. By contrast, good candidates in the general population are clearly distinguished by their higher scores.

6.2 Contrast

The following definition takes probability of correct comparison, as defined in the introduction, and relaxes the outcome indicator (either 0 or 1) to a continuous value.

Definition 9. *The average margin of comparison within* Π *is*

$$\kappa_\Pi = \frac{1}{|\Pi_+|}\frac{1}{|\Pi_-|}\sum_{x\in\Pi_+, x'\in\Pi_-} s(x) - s(x')$$

Similarly define κ_{Π^c} by replacing Π with Π^c.

Definition 10. s *offers* Π *contrast if* $\kappa_\Pi = \kappa_{\Pi^c}$.

This definition captures two key ideas. The first is that comparisons *within* groups should be accurate. Suppose a college accepts the best students from the general population, but guesses randomly within a protected group, or perhaps accepts based on an ancillary attribute such as athleticism. This so-called 'token' effect may distort incentives or otherwise misdirect students wishing to improve themselves. The second idea is that the scores, in either their calculation or their subsequent use, involve randomness or error. For example, recall randomized classifiers from the introduction. As another example, if outcomes in $\{-1, 1\}$ are sampled with mean $s(x)$ and $s(x')$ for good x and bad x', then the probability of a correct comparison is just $(1+s(x))(1-s(x'))/4$. In these scenarios, the magnitude of scores matters as well as their ordering. With these ideas in mind, let us compare this definition to ones previously proposed in the literature, and understand their respective benefits and drawbacks.

Accuracy of between-group comparisons. In the contextual bandit problem, an algorithm compares candidates $x_1, \ldots x_k$ from k known groups (or arms), each with true (but unknown) values $y_1, \ldots y_k$. It randomly samples candidate x_i with probability based on a score $s(x_i)$. It learns that candidate's value, and thereby estimates the values of future candidates. In this context, Joseph et al. [2016] disallow $s(x_i) > s(x_j)$ if $y_i < y_j$; a candidate's potentially high value must be considered, even if their group has low overall value. This enforces accurate comparison *between* groups; the algorithm must explore and estimate values for each group, not just the overall population. It crucially relies on random, possibly erroneous choices to learn about groups without explicitly preferring them. This randomness is presently considered a

nuisance; contrast mitigates its impact on the outcomes.

The probability of correct comparison is equal to the area under the ROC curve [Cortes and Mohri, 2004], which quantifies the tradeoff between false positive rate (FPR) and true positive rate (TPR). Contrast can be reinterpreted in terms of these quantities after some basic algebraic manipulation:

$$\kappa_\Pi = \frac{1}{|\Pi_+|} \sum_{x \in \Pi_+} s(x) - \frac{1}{|\Pi_-|} \sum_{x' \in \Pi_-} s(x')$$

For a randomized classifier, this quantity is the expectation of $\mathrm{TPR}_\Pi - \mathrm{FPR}_\Pi$. Let us think about how mobility affects these rates. Suppose $\mathrm{TPR}_{\Pi^c} = \mathrm{TPR}_\Pi$ but $\mathrm{FPR}_{\Pi^c} > \mathrm{FPR}_\Pi$; that is, the general population is accidentally accepted more often. To offer contrast, the classifier could reduce these accidents by decreasing FPR_{Π^c}. However, it could also increase TPR_Π and therefore increase the acceptance rate of Π^c, which was already higher. Perhaps worse, it could decrease TPR_Π and reduce accuracy. Contrast deems this scenario inopportune for the general population even though they enjoy better outcomes. This shows that contrast does not equalize acceptance rates between the groups, nor does it necessarily promote accuracy.

Inherent tradeoffs for discrete error rates. Equalized odds, as defined by Hardt et al. [2016c], requires the FPRs and TPRs to be the same between both groups. Hardt et al. [2016c] find this notion too strong because it penalizes classifiers which are more accurate on the general population. They identify equal opportunity with equal TPRs. For example, good students should have equal chances of being admitted to college, regardless of their group. However, bad students in Π may be scrutinized more than bad students in Π^c. This could allow bad students to be admitted due to wealth or influence. More generally, Zafar et al. [2017a] seek to equate the FPRs, TPRs, FNRs, etc. It is not always possible to equate such quantities, which makes various notions of fairness irreconcilable. Chouldechova [2017], Kleinberg et al. [2016] initiated the study of such tradeoffs, proving that TPRs and TNRs typically cannot be equated for calibrated scores. By formalizing contrast as an analytically tractable margin, we hope to avoid such impossibility

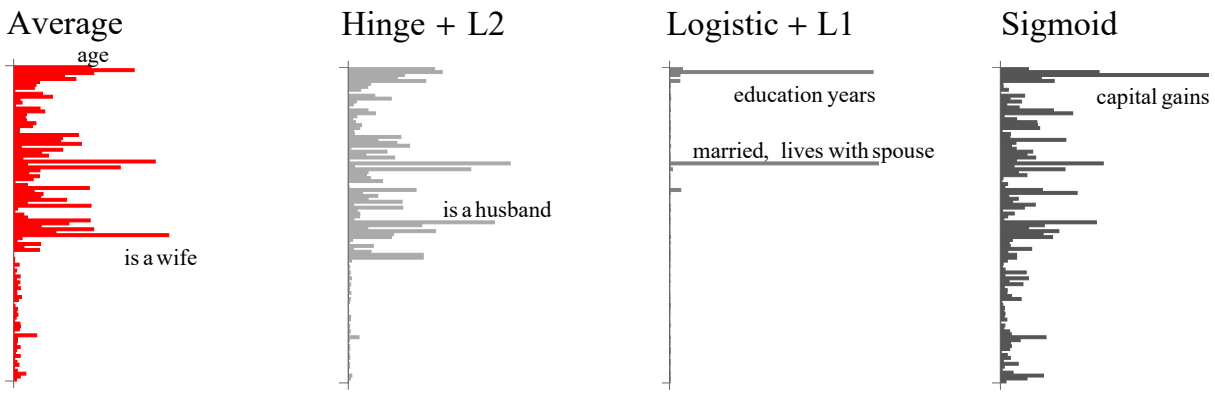


Figure 6.3: Absolute coordinate values (i.e. dependence on features) of different unit-norm discriminant vectors, each computed on the dataset of adult income. Let Π and Π^c be females and males respectively constituting roughly $1/4$ and $3/4$ of the candidates, whose income is classified as high or low. The average vector, as defined in eq. (6.3) is compared to standard, Π-unaware penalized loss minimizers: hinge loss with ℓ_2-norm penalty (aka SVM), logistic loss with ℓ_1 penalty, and nonconvex sigmoid loss with no penalty. As expected, the ℓ_1 penalty encourages sparsity; the other vectors are not sparse. The unpenalized vector uses capital gains, which is predictive but only relevant for a small fraction of the population. Average and SVM are similar, except the former heavily emphasizes "is a wife" rather than "is a husband". This is because the average adjusts for the minority Π.

results. If y_x were continuous rather than binary, their margins (from a decision threshold) relate to fairness. When they are very different for Π and Π^c, different TPRs (e.g. 'hits' in police searches) are not necessarily unfair [Simoiu et al., 2016].

6.3 The Average Vector

As we will see below, common ways of learning a linear classifier do not result in mobility and contrast. This section shows that a simpler classifier based on averaging does yield both, under appropriate assumptions. We focus on scores and classifiers induced by $w \in \mathbb{R}^n$:

$$s_w(x) = \psi(\langle w, x \rangle) \quad \rightarrow \quad c_w(x) = \operatorname{sgn}(\langle w, x \rangle)$$

The activation function $\psi : \mathbb{R} \rightarrow [-1, 1]$ approximates the sign function, but is differentiable with maximum slope β: $\psi(0) = 0$, $\psi'(0) = \beta$, and $|\psi'(a)| \leq \beta$ for all a. A common choice is

tanh. As $\beta \to \infty$, $\psi \to \text{sgn}$ and $s_w \to c_w$. The typical approach to choosing w is to minimize the expectation, over the data, of a loss function plus a penalty function. We analyze a simpler average of the data.

Definition 11. *The average of the genuine opportunities of* Π *and* Π^c*, as defined in eq.* (6.1)*, is:*

$$g = \frac{1}{2}\left(\frac{g_\Pi}{||g_\Pi||} + \frac{g_{\Pi^c}}{||g_{\Pi^c}||}\right) \tag{6.3}$$

The figure above compares the average to other vectors.

6.3.1 Theoretical Support

Preprocessing the data allows the average to offer mobility.

Proposition 13. *If the data are centered:*

$$\frac{1}{|\Pi|}\sum_{x\in\Pi} x = \frac{1}{|\Pi^c|}\sum_{x\in\Pi^c} x$$

then the average offers mobility to Π.

Proof. Since g is the average of two vectors, it has the same angle between both of them:

$$\langle g, g_\Pi/||g_\Pi||\rangle = \langle g, g_{\Pi^c}/||g_{\Pi^c}||\rangle.$$

By eq. (6.2):

$$\mu_\Pi = \frac{1}{|\langle g, g_\Pi/||g_\Pi||\rangle|}\langle g, \frac{1}{|\Pi|}\sum_{x\in\Pi} x\rangle = \frac{1}{|\langle g, g_{\Pi^c}/||g_{\Pi^c}||\rangle|}\langle g, \frac{1}{|\Pi^c|}\sum_{x\in\Pi^c} x\rangle = \mu_{\Pi^c}$$

□

Contrast is guaranteed if the score is very smooth (i.e. the slope of the sigmoid is small):

Proposition 14. *If* $||g_\Pi|| = ||g_{\Pi^c}||$*, as* $\beta \to 0$*,* s_g *offers contrast to* Π.

Proof. As $\beta \to 0$, $\frac{d}{d\beta} s_g(x) = \langle g, x \rangle$. By definition of κ_Π:

$$\begin{aligned} \frac{d}{d\beta} \kappa_\Pi \Big|_{\beta=0} &= \frac{1}{|\Pi_+|} \sum_{x \in \Pi_+} \langle g, x \rangle - \frac{1}{|\Pi_-|} \sum_{x' \in \Pi_-} \langle g, x' \rangle \\ &= \langle g, g_\Pi \rangle \end{aligned}$$

Similarly $\frac{d}{d\beta} \kappa_\Pi \Big|_{\beta=0} = \langle g, g_{\Pi^c} \rangle$. To equate these quantities, we must show:

$$||g_\Pi|| + \langle \frac{g_{\Pi^c}}{||g_{\Pi^c}||}, g_\Pi \rangle = ||g_{\Pi^c}|| + \langle \frac{g_\Pi}{||g_\Pi||}, g_{\Pi^c} \rangle$$

Dividing both sides by $||g_\Pi|| = ||g_{\Pi^c}||$ completes the proof. □

These propositions have strong, possibly unrealistic preconditions; the conclusion reflects upon their pertinence, and the next section validates the average on real data.

6.4 Experimental Validation

The well-known adult income dataset consists of 48,842 individuals, each described by 14 features, and whether or not they earn more than $50,000 per year [Kohavi, 1996]. Over 75% of the incomes are higher; eliminating this imbalance reduces the number of data to 15,682. Each categorical feature with k possible values is 'one-hot' encoded using k binary features, and the the auxiliary 'final weighting' attribute is removed. This results in 107 total features, each standardized to mean 0 and variance 1. Mobility and contrast do not directly involve the discrepancy between training and test distributions, so the entire dataset is used at once.

Two experiments compare the average with some standard linear classifiers which are unaware of Π. In the first experiment, Π is generated by selecting a single defining feature (for example, "is a husband"). This produces minority (or majority) groups in a relatively realistic fashion. In the second experiment, Π is just a random half of the population. This 'null' experiment decorrelates the features, outcomes, and group memberships. The results of the first experiment should substantially differ from the second.

Mobility and contrast are defined by exact equalities, but we will observe just approximate

equality. The absolute difference between the two sides of definition 8 or definition 10 is not as important as the relative difference. We measure differences by the absolute difference of logarithms, with values close to zero still being ideal:

$$d(a, b) = |\log(a/b)| \tag{6.4}$$

Results. These are depicted in Section 6.3.1. In the first experiment, the average offers roughly equal mobility and contrast, whereas the other classifiers do not. This difference is in some sense significant, since it disappears in the second experiment. As expected, the differences are much smaller in the second experiment, since they are between two random sums of the same mean.

6.5 Remarks

A key underlying idea of this chapter is that, even if decisions are binary, the margin by which they are established is morally important. They determine how much effort is needed to improve one's outcome, or how sensitive the outcome is to randomness and error. We accordingly formalize equal opportunity in terms of an input margin (mobility) and an output margin (contrast). We illustrate the virtues of a very simple averaging classifier with some basic mathematical analysis and an experiment on a moderately-sized dataset. Let us highlight the limitations of our contributions with a view to future research.

As previously discussed, mobility and contrast are not comprehensive definitions of fairness: they may further imbalance outcomes or increase error rates. We loosely compared them to other previously proposed definitions, but we could not meaningfully say one definition is better than another. In some scenarios, equal opportunity is just a means to a more quantitative end: better outcomes. If a rule supposedly ensures equal opportunity, then imposing it upon candidates eager to improve themselves should eventually lead to better outcomes. Perhaps definitions of equal opportunity could be quantitatively compared along these lines.

Proposition 13 and proposition 14 only support the average classifier when it is, respectively, very accurate or very close to random. They also assume the genuine opportunities are com-

parably sized (i.e. $||g_{\Pi}|| = ||g_{\Pi^c}||$). This may be ensured by rescaling or reweighting the data. However, the relative advantage of the average over other vectors, as illustrated in the experiment, may instead depend on whether the genuine opportunities coincide (i.e. $\langle g_{\Pi}, g_{\Pi^c} \rangle$ is large). Intuitively, if the way to become accepted differs considerably for Π and Π^c, then it is more difficult to accommodate both groups. A classifier unaware of Π is less likely to do so by accident; the average, or another Π-aware method, may then have a larger relative advantage. The average should be perceived as a simple, effective baseline rather than an optimal solution. It is likely to be outperformed by a more computationally involved algorithm which explicitly attempts to minimize error while maximizing mobility and contrast.

6.6 Discussion

This chapter explores a synthesis between newly-proposed discrete definitions of fairness, and continuous quantities preferred in optimization. Unlike the other chapters, this one proposes objectives rather than solving them. By its nature, this chapter is the most subjective and difficult to evaluate. It also pursed a rather different strategy for synthesis: rather than wrapping a modern algorithm, or swapping out some of its components, it attempted to find a compromise between computational and ethical concerns. As the concluding chapter will discuss, this is perhaps the least convincing synthesis of the dissertation, for reasons that can be analyzed and learned from in hindsight.

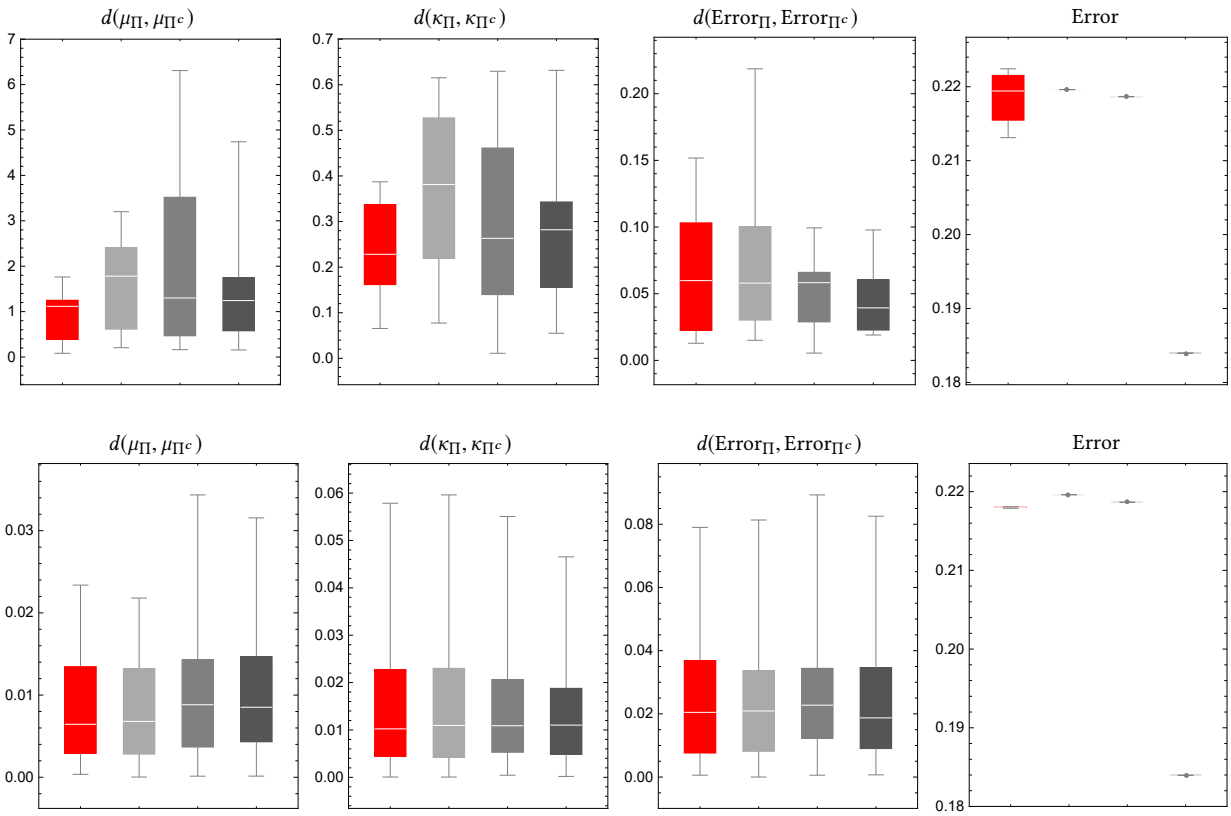


Figure 6.4: Two experiments, top and bottom, compare the average vector to standard, Π-unaware penalized loss minimizers: hinge loss with ℓ_2-norm penalty (aka SVM), logistic loss with ℓ_1 penalty, and nonconvex sigmoid loss with no penalty. As described in the main text, d is a measure of relative difference. The top experiment involves 10 realistic Π. The average roughly offers mobility ($\mu_\Pi \approx \mu_{\Pi^c}$) whereas the others do not. The average and nonconvex classifier roughly offer contrast ($\kappa_\Pi \approx \kappa_{\Pi^c}$), though the former has better interquartile range. However, the misclassification error of the average is often substantially higher. (The other classifiers have the same error rate for every Π since they are not aware of it.) These distinctions vanish in the bottom experiment, where Π is just a random half of the population.

Chapter 7

Conclusion and Future Work

The field of machine learning has recently witnessed remarkable empirical advances which have profoundly enhanced not just benchmark metrics, but the role of machine learning in society. This progress has not been uniform along all dimensions; the gap between classical and modern machine learning mirrors growing concerns about rigor, efficiency, interpretability, and fairness. This dissertation adopts a fundamentally optimistic view: that in some (though not all) situations, the seemingly-inviolable tradeoffs between classical and modern machine learning can be carefully sidestepped through a nontrivial synthesis of the two approaches. Stated concisely:

> **Thesis**: It is often possible to restore safety, efficiency, and tractability to modern machine learning by prudently incorporating classical techniques.

To conclude this dissertation, let us reflect upon this thesis — *when* was such synthesis possible? — and offer suggestions for future research.

7.1 Review and Subsequent Developments

Each chapter examines this thesis in a different area of machine learning. Let us review the contributions of these chapters, and examine how each area of machine learning evolved subsequent to each of these works.

Chapter 2 (Meta-Analysis with Untrusted Data)

This chapter proposes synthesis between modern regression models trained on large quantities of untrusted data, and rigorous estimates of causal effect based on small amounts of trusted data. To researchers in machine learning, it is somewhat unsurprising that prior beliefs or inductive bias can be safely incorporated into learning algorithms; as discussed earlier in the chapter, a variety of statistical techniques enable this combination. In evidence-based medicine, however, such a synthesis between observational data and rigorous causal inference is both counterintuitive and remarkable. To successfully apply conformal prediction to this field, this chapter fundamentally advanced some core methodology in (full) conformal prediction. In particular, it shows that full conformal prediction can be fast for a wide class of learning algorithms. Furthermore, this full conformal prediction algorithm is simple enough to analyze its behavior in the presence of noise. Though I feel this work is promising, it is too recent to compare it to new developments in the field.

Chapter 3 (Differentiating Through Orthogonal Polynomial Transforms)

This chapter implements a synthesis between modern gradient-based optimization and classical sequences of orthogonal polynomials. The key observation is that the most computationally convenient representation of a sequence of orthogonal polynomials consists of the coefficients of its three-term recurrence. By enabling backpropagation for polynomial evaluation and interpolation — that is, by deriving the vector-Jacobian products of these algorithms — we can use orthogonal polynomials to parameterize (or reparameterize) a variety of contemporary learning and optimization problems.

As a layer within neural networks, structured linear maps (subsuming not just orthogonal polynomial transforms, but also low rank and sparse matrices) have gained popularity [Dao et al., 2020, Fu et al., 2024]. The main advantage has been performance (for example, achieving subquadratic matrix-matrix multiplication) rather than interpretability or inductive bias. Since the DXT layer is a drop-in replacement for fixed polynomial transforms, the hope is that it can preserve some interpretability of the original signal processing pipeline; for example, intuitions about the DCT eliminating high-frequency stimuli irrelevant to the human visual system. How-

ever, it is unclear whether inspecting an orthogonal polynomial basis constitutes an essential form of interpretability for a wide audience [Lipton, 2016]. The second part of this chapter, pertaining to the connection between orthogonal polynomials and optimization theory, has remained (surprisingly) understudied. The paper of Lasserre [2020] explicating this connection has been cited less than 5 times by other researchers.

Chapter 4 (Linear Dynamical Systems for Sequence Modeling)

This chapter develops a synthesis between nonlinear sequence-to-sequence models and linear dynamical systems, also known as state-space models. The results achieved in this chapter are among the most counterintuitive and interesting of the entire dissertation. For good reason, depth in neural networks is typically thought of as an architectural feature which inhibits analytical reasoning. However, an interesting approach in control theory indicates that, in some contexts, depth can actually facilitate analytical reasoning. This is because nonlinearity along time, which expresses complex dynamics, can be replaced by (approximations of) nonlinearity along depth, where deviations can be more easily bounded. Aside from analytical tractability, this replacement enables parallel computation along time, a crucial requirement of modern sequence-to-sequence models.

Since the publication of this work, there has been intense interest in using linear systems (also called state-space models) as the basis for modern sequence modeling. Simultaneously with this work, Gu et al. [2020] showed that long-range memory problems could be, in a sense, optimally solved by choosing the matrices A, B, C and D according to the HiPPO framework. The subsequent S4 architecture implemented HiPPO models as convolutions, which allowed parallel computation over time [Gu et al., 2021a]. Subsequent works showed that plain diagonal recurrences could achieve similar performance as S4 [Gu et al., 2022, Gupta et al., 2022]. The convolution kernel of Gu et al. [2021a] can be computed through Vandermonde matrix multiplication [Gu et al., 2022]. Gu et al. [2021b] also provably replace nonlinearity across time with nonlinearity along depth, through a closely-related scheme of Picard iteration. (However, their result is only for continuous time, and does not allow the nonlinearity to apply to both the state and the projected input; it is meant to theoretically motivate existing gating schemes, rather than

derive a new one). Orvieto et al. [2023] also observed that RNNs with nonlinearity across time can be empirically replaced by stacks of diagonally-parameterized linear systems, each implemented as a parallel scan. Gu and Dao [2023] introduced time-varying parameters (as a selection mechanism) and also replaced convolutions with parallel scans. This architecture finally allows state-space models to achieve competitive predictive performance with transformers on tasks such as language modeling, while being more computationally efficient.

Chapter 5 (Interpretable Deep Learning in Healthcare)

This chapter carefully generalizes traditional, handcrafted electrocardiological statistics into statistics that are learned from labeled data for a specific predictive purpose. It defines a family of statistics which generalizes the handcrafted ones, and embeds parameters which can be learned by gradient descent. This chapter is satisfying because its physiologically-informed approach leads to better predictive performance than a naive one based on more flexible models. The design of the generalized statistics imparted an empirically-helpful inductive bias. Furthermore, the new statistics preserve some of the interpretation and intuitions surrounding the traditional ones. (Unlike in Chapter 3, this preservation of interpretation coincided with performance improvements).

The scientific conclusion of this chapter — that sympathetic activity seems weakly discernible from heart rate signals — was confirmed simultaneously and independently by Valenza et al. [2018]. The Sympathetic Activity Index decomposes the heart rate signal using Laguerre basis functions. Specifically, it expresses the RR interval sequence as a sum of orthogonal Laguerre functions, where lower and higher frequency functions correspond to the sympathetic and parasympathetic systems, respectively. Valenza et al. [2018] fit these functions using supervised learning, but in a very different manner than this dissertation. To fit the low-frequency sympathetic functions, Valenza et al. [2018] administered atropine to seven subjects, which blocks their parasympathetic systems, and had them perform supine-to-stand tests. Similarly, to fit the high-frequency parasympathetic functions, subjects performed the same tests with blocked sympathetic systems. By obtaining data on each system independently, they more cleanly fit their model. (However, they do not utilize a quantitative ground-truth measure, such as salivary amy-

lase excretion; they simply consider the sympathetic system as "active" while standing). Their laboratory data collection, though more invasive, allowed identification of the underlying sympathetic signal.

Chapter 6 (Towards Computationally-Tractable Multi-Group Fairness)

This chapter found a compromise between newly-proposed discrete definitions of fairness and continuous quantities preferred in optimization. Unlike the other chapters, this one proposes objectives rather than solving them. By its nature, this chapter is the most subjective and difficult to evaluate. It also pursued a rather different strategy for synthesis: rather than wrapping a modern algorithm, or swapping out some of its components, it attempted to balance computational and ethical concerns.

The motivation of this chapter is that combinatorial or discrete definitions of fairness, especially multi-group fairness, can be computationally challenging to satisfy. This was, in a sense, soon confirmed by other work: the celebrated notion of multicalibration was proposed as a comprehensive definition of fairness [Hebert-Johnson et al., 2018]. Multicalibration requires a predictor to be (approximately) calibrated not just overall, and not just for some subgroups specified in advance, but for any group that can be efficiently isolated by a boolean hypothesis class $\mathcal{C}$. Algorithmically, multicalibration can be thought of as a game against an auditor, which seeks to find subsets $\Pi \subset \mathcal{C}$ on which the predictor is poorly calibrated; the learner can then use this counterexample to improve the predictor. The foundational observation is that achieving multicalibration over $\mathcal{C}$ is equivalent to (weak) agnostic learning over $\mathcal{C}$. Thus, when $\mathcal{C}$ is comprised of linear classifiers, as in Chapter 6, it is computationally difficult to ensure multicalibration [Feldman et al., 2009, Kalai et al., 2008]. In this sense, relaxations (such as those in Chapter 6) are generally necessary. However, embracing, rather than avoiding, the computational challenges of fairness has led to strong research developments. Multicalibration is closely connected to (and in some restricted senses, equivalent to) outcome indistinguishability [Dwork et al., 2021] and omniprediction [Gopalan et al., 2023], two recently-proposed desiderata for predictive models.

7.2 Thesis Assessment

As discussed in the previous section, each chapter enjoyed a different degree of success. Organized in reverse chronological order, they tended to become more successful over time. While it is tempting to speculate on many different factors, I offer three high-level observations about where the most research progress was made.

My first observation is that success was largely correlated with an immediate, definite, quantitative goal for the synthesis. This seems like a trivial observation; surely it is obvious that machine learning research should be quantitatively driven. However, in the context of AI safety, it can be difficult to completely quantify all goals (e.g. tractability), or to pursue the appropriate balance between competing quantitative goals. Chapter 2 (meta-analysis) had the clearest quantitative goal of obtaining the tighest possible prediction intervals meeting the specified coverage guarantee; no attention was paid to other issues. This chapter is somewhat unusual in that it pursues a "wrapping" strategy but obtains rigorous guarantees. Other research in this vein — for example, methods of interpreting black-box predictive models — are sometimes criticized on account of their lack of well-specified guarantees [Rudin, 2019]. Chapter 6 (fairness) defined its own quantative goals, and found a compromise which did not align with subsequent research. Chapters 3, 4, and 5 balanced expressive power against considerations of tractability and speed.

My second observation is that the "swapping" research strategy tended to require relatively extensive engineering. Chapters 3 and 4 both involved implementation of custom CUDA operations, the latter with complex numbers. However, even these efforts were not truly sufficient to do justice to the theoretical developments. For example, the implementation of LDStack has an unoptimized, memory-inefficient backwards pass, which makes it difficult to achieve competitive performance with state-of-the-art sequence-to-sequence models. Subsequent work on state-space models has concentrated substantially on hardware-aware algorithms [Dao et al., 2022, Gu and Dao, 2023]. This is because, even if state-space models are theoretically more efficient than models such as transformers, they can fall short in practice due to lower hardware utilization. As noted by Fu et al. [2023], it can be very challenging for algorithmic researchers to improve upon vendor-optimized implementations and even custom hardware. This has consequences for research on primitives for machine learning: initial algorithmic work likely will

not deliver benchmark-beating performance due to these practical issues. Expectations for such research should be set accordingly.

My third observation is that settling on arbitrary compromises or balance points between classical and modern techniques should be avoided; there should ideally be major practical downsides for choosing a different design. Chapter 3 is a good example of this, since it embodies both a success and failure in this regard. The DXT is not as widespread as more expressive layers because there isn't always a compelling reason to limiting expressivity to just polynomial transforms as opposed to, for example, the slightly-larger class of quasiseparable matrices. However, in the context of applications such as Mop, there is a strong rationale for optimizing over exactly the set of orthonormal polynomial sequences, since any larger feasible region would constitute a relaxation that would not yield the solution to the original problem. In Chapter 2, it is not possible to introduce any (direct) nonlinearity across time without preventing the use of parallel scans and convolutions; this is likely why linear systems are a mainstay of fast sequence-to-sequence models. Chapter 6's fairness definitions are based on margin quantities, which are merely one device for achieving computational tractability. Other approaches could also achieve tractability without departing from well-recognized conceptions of fairness.

7.3 Future Work

7.3.1 Meta-Analysis

Revisiting Network Meta-Analysis

As presented, conformal meta-analysis is a practical algorithm which could be immediately applied to answer scientific questions. However, before doing so, it is important to understand the settings in which conformal meta-analysis would likely deliver meaningful, notable conclusions. It is ideally applied to questions where there is (1) a high amount of heterogeneity, and (2) a relatively large number of included trials ($n \geq 100$), originating from a broad question and/or an active research field. Network meta-analyses often satisfy both of these criteria. A network meta-analysis compares multiple interventions (e.g. "drug A", "drug B", and so on) against placebo.

In addition to including placebo-controlled trials (e.g. drug A versus placebo) it also includes active-comparison trials (e.g. drug A versus drug B) and uses them to indirectly reason about e.g. drug B versus placebo. This leads to a graph where the nodes are interventions and the edges are pairwise comparisons of interventions from trials. Inference about treatment effects involves the graph Laplacian, and an analogy to electrical networks: within-trial variances corresponds to resistances, observed effects correspond to voltages, and the inverse-variance weighted estimates of true effects corresponds to current flow [Rücker, 2012]. Network meta-analysis tends to include more trials because it expands the inclusion criteria. For example, recent meta-analyses of glaucoma treatments, antipsychotics and antidepressants included 114, 402 and 522 trials, respectively [Cipriani et al., 2018, Huhn et al., 2019, Li et al., 2016].

Because they involve a relatively large number of trials, are technically sophisticated, and answer broader questions with more clinical relevance, network meta-analyses tend to be highly cited and regarded. In a sense, if systematic review and meta-analysis is the highest form of evidence, then network meta-analysis is considered the true apex. However, network meta-analysis leans on homogeneity (also called "validity") much more strongly than pairwise meta-analysis. This is because network meta-analysis doesn't estimate one global average, but multiple averages derived from purportedly valid chains of comparisons among multiple trials. In pairwise meta-analysis, heterogeneity tends to be "swept under the rug" because it affects the interpretation of the analysis rather than the quantitative estimates themselves, which remain unbiased. If there is a high degree of heterogeneity in pairwise meta-analysis, then the average treatment effect is less informative about each individual trial's setting, but it is nonetheless a valid average. However, it is not clear whether network meta-analysis remains unbiased in the presence of heterogeneity. This seems to depend on the distribution of the pairwise comparisons (i.e. the edges in the graph), which may depart substantially from the uniform distribution. For example, if drug B is often compared to drug C, and trials of drug C tend to downplay its effectiveness, then drug B might be inappropriately considered more effective than drug A, which inherits less credit from the biased trials of drug C. Fortunately, these issues can be rigorously avoided, since network meta-analysis is just a special case of regression (i.e. conformal meta-analysis) where the different interventions are encoded in the features. It would be interesting to see how the

conclusions of conformal meta-analyses would differ from previous network meta-analyses.

Beyond Split Conformal and Score Functions

In practice, split conformal prediction is the most widely-used form of conformal prediction. This is because split conformal doesn't involve retraining the underlying predictor, and it is very easy to implement. The downside of split conformal is it is statistically inefficient, requiring the training data to be split into a proper training set and a calibration set. Methods such as the jackknife+, cross-validation+ [Barber et al., 2021], and cross-conformal prediction [Vovk, 2015] offer a middle ground between full and split conformal, retraining the underlying predictor roughly a constant number of times, while reducing the statistical overhead of data splitting. However, there are still many situations where training the model more than once is prohibitive. In these situations, fully-conformal kernel ridge regression (as presented in Chapter 2) could viably replace split-conformal prediction in most applications, without retraining the underlying predictor or imposing any other practical difficulties.

The key observation is that, in practice, good predictors are almost always based on good features. In deep learning, predictions are usually linear in some learned feature space: if w is a vector, and $\phi(x)$ is a learned feature map, then the prediction typically takes the form $\mu(x) = w^T\phi(x)$. So, in most practical situations, training results not just in μ, but a Gaussian process (μ, κ), where $\kappa(x, x') = \phi(x)^T\phi(x')$ describes the feature space. So, rather than running split-conformal prediction with scores based on μ, it is possible to run fully-conformal kernel ridge regression with (μ, κ), and actually learn from the calibration data. The potential benefits of this scheme are hinted at by Simulation 4 in Chapter 2, which examined the benefits of KRR learning a posterior rather than just treating the prior as fixed. Since the calibration set is usually not nearly as large as the proper training set, and since fast algorithms for KRR have been developed [Alaoui and Mahoney, 2015, Avron et al., 2017], there would be limited practical downside to replacing split conformal with fully-conformal KRR. At a higher level, the idea is to more routinely base conformal prediction not just on score functions derived from μ, but on features as well.

Replacing split conformal with fully-conformal KRR could be helpful when learning con-

formal predictors [Stutz et al., 2022]. In this application, a differentiable analogue of conformal prediction is used as a loss layer while training a predictor with gradient descent. The goal is to more specifically train the predictor to produce tight conformal prediction intervals. The methodology of Stutz et al. [2022] is based upon split conformal prediction, which reduces the effective batch size due to splitting. The linear algebraic operations in fully-conformal KRR are differentiable, so it could similarly be used as a loss layer. Aside from increasing the batch size, it would allow gradients to pass through not just μ, but also κ. That is, grading both the predictions and the features could plausibly lead to a more well-behaved loss layer.

Tighter Analysis, Weaker assumptions, and Stronger Guarantees

The conformal meta-analysis algorithms presented in Chapter 2 are intended to be simple, strong baselines; they are not optimal solutions to the predictive problems of meta-analysis. As discussed in Chapter 2, the analysis of the presented algorithms seems empirically loose. It should be noted, however, that a gap between empirical and provable coverage is not uncommon in the conformal prediction literature [Barber et al., 2021, Stutz et al., 2023]. Relaxing the assumptions needed to obtain similar guarantees seems both viable and desirable. Particular attention needs to be paid to the sequential dependence (i.e. nonexchangeability) of the features x. On a short time scale, it is not completely unreasonable to model the features as approximately independent, simply because, at the edge of scientific knowledge, trial design may seem random or even haphazard in hindsight. However, at longer timescales, trials are definitely designed in recognition of the past history of results, progressively targeting more promising or novel treatments. Furthermore, living systematic reviews [Elliott et al., 2017] would keep the same meta-analysis updated for decades. There are now a number of approaches which extend conformal prediction beyond exchangeability. The methods developed by Barber et al. [2023] are appropriate when the distribution departs from exchangeable in a mild or predictable manner. In an adversarial setting, performing gradient descent to control interval sizes can offer approximately $1 - \alpha - \frac{1}{n}$ coverage [Gibbs and Candes, 2021].

Conformal prediction is often criticized for offering only marginal guarantees; indeed, conditional guarantees are not generally possible without further distributional assumptions [Foygel Bar-

ber et al., 2021]. Meta-analysis is highly unusual among regression problems in that prevalent algorithms do not use features. Because existing guarantees are all marginal, this downside of conformal prediction is not as sharply felt in its application to meta-analysis. Nonetheless, in the future, it would be desirable to offer group or class-conditional guarantees [Ding et al., 2024, Gibbs et al., 2023].

7.3.2 Other Ideas

Minimal Values of Practical Optimization Problems

This scheme is conceptually interesting, but the crucial question remains: is there a broad class of applications where it helps find the minimum value faster than gradient descent, moment estimation, or random sampling? The most promising approach is to augment Mop with additional structure in the objective f and the prior distribution ρ. Black-box sampling access to these functions is unnecessarily limiting in many settings. Structured representations of probability distributions — such as normalizing flows, graphical models, diffusion models, and Bayesian neural networks — offer computational access beyond efficient sampling. (For example, normalizing flows allow the probability density function to be evaluated.) It might be possible to develop better estimates $\hat{\sigma}$ and $\hat{\Sigma}$ using this additional information. Aside from incorporating additional information, there is the question of identifying objectives where knowing the minimal value is as useful as knowing the minimizer. This happens in a variety of combinatorial problems; for example, in network flow problems, knowing the capacity of the network (i.e. the maximum flow) is just as important as identifying the flow which saturates it. In game theory, the value of a game is often just as important as the strategies that attain it. In machine learning, knowing the minimal attainable loss would be an interesting tool for studying nonconvex learning.

Replacing Positive-Definite Constraints Via Orthogonal Polynomial Reparameterization

Positive-definite Hankel matrices M arrange a vector m as $M_{i,j} = m_{i+j}$. They are the univariate analog of multivariate moment matrices appearing in semidefinite relaxations of polynomial opti-

mization problems [Blekherman et al., 2012]. Optimizing over positive-definite Hankel matrices is equivalent to optimizing over the coefficient vectors (α, β) defining sequences of orthogonal polynomials. The latter parameterization has a notable advantage: it replaces the positive-definite Hankel constraint, which is computationally burdensome, with trivial entrywise positivity constraints on β. This raises an interesting question: is it possible to more efficiently solve these optimization problems by reparameterizing them in terms of α and β? In general, the answer is probably no: the map between (α, β) and m is known to be poorly conditioned [Gautschi, 1985]. (A formula for α and β involving powers of the Jacobi matrix is given by Simon [1998].) However, there may be special cases where the objective can be cleanly rephrased directly in terms of α and β. It should be noted that this reparameterization may introduce numerical challenges of its own. However, it is often possible to design optimization algorithms that can handle poorly-behaved objectives, so long as difficult constraints are eliminated. The Burer-Monteiro approach to rank-constrained semidefinite programming is a good example of this, where convexity is sacrificed to avoid the challenging rank and semidefinite constraints [Burer and Monteiro, 2003].

Chapter 3's approach can be extended to positive-definite Toeplitz matrices, which are more common in statistics and optimization applications. For example, the covariance matrix of an ARMA time series model is positive-definite Toeplitz. The extension is possible due to Szegő's theorem, which shows that positive-definite Toeplitz matrices correspond to orthogonal polynomials on the complex unit circle [Szego, 1939]. Furthermore, such polynomials obey a recurrence relation in terms of coefficients known as the Schur parameters. Evaluation and interpolation algorithms have been adapted to this recurrence [Ammar et al., 1993, Bella et al., 2007].

Understanding the Power of Depth in Replacing Nonlinearity

Though state-space models are briskly gaining popularity for sequence modeling, there is still a technique from Chapter 4 which has been underutilized: replacing nonlinearity across time by nonlinearity along depth, using the simple, principled mechanism of iterated local corrections. It would be interesting to understand the rate of convergence of the corrections, i.e. the number of layers required to approximate a nonlinear RNN. It would also be interesting to determine if multiplicative corrections, which experimentally seemed to have a faster rate of convergence,

could be efficiently implemented. Finally, these corrections could easily handle the time-varying construction of Gu and Dao [2023]; indeed, they were initially proposed for time-varying systems [Tomás-Rodríguez and Banks, 2010].

At a higher level, it is possible that this line of work could inform recent research on the interplay between transformers and recurrent models [Katharopoulos et al., 2020], particularly the ability of transformers to perform complex tasks with low depth [Liu et al., 2023a, Sanford et al., 2024]. In particular, Liu et al. [2023a] show that automata (which are just a special case of recurrent neural networks of length T) can be expressed by $O(\log T)$-depth transformers. This is a "shortcut" in the sense that an RNN of length T requires depth T to compute. However, the results of Chapter 4 suggest that some RNNs could be plausibly approximated by stacks of linear systems, also of logarithmic depth.

Iterated Local Corrections in Practical Architectures

The scheme of iterated local corrections could potentially improve modern state-space model architectures. Here are two avenues for possible improvement.

(1) Democratizing training by reducing memory requirements. State-space models have greatly improved speeds of the forward pass, enabling inference of large neural networks on edge devices. However, computing gradients (the backward pass) remains out of reach for edge devices, because the amount of memory required to store activations scales linearly with the number of layers L. Recomputation and reversibility are two techniques which reduce the memory requirement to $O(1)$, but they cause severe tradeoffs in speed and expressiveness, respectively. Recomputation (also known as gradient checkpointing) is typically used to reduce the memory requirement to $O(\sqrt{L})$ [Chen et al., 2016]; reducing it to $O(1)$ impractically requires a factor $O(L)$ more computation. In reversible neural networks, the inputs can be recomputed from the outputs. Reversibility substantially constrains the expressive power of the network because it hinders the ability to forget irrelevant information [MacKay et al., 2018]. Diagonal state-space models such as Mamba are not (necessarily) reversible. However, their VJP can nevertheless be computed in $O(1)$ memory. (Due to linearity, the VJP is zero where the input sequence cannot be recomputed). However, this applies only to the SSM itself and not to the surrounding operations

in the Mamba block. Thus, major memory savings are achievable only if the Mamba block can be substantially simplified. This may indeed be possible, as described in the next suggestion.

(2) An alternative approach to hardware-efficient algorithms. Parallel scans, which are used to compute LDStack and Mamba [Gu and Dao, 2023], are very fast, with optimized implementations running at "memcpy speeds". Somewhat paradoxically, this extreme efficiency poses a problem for utilization of GPU hardware. GPUs are generally designed for computations which involve very little data transfer (i.e. use of limited memory bandwidth) compared to computation. Furthermore, modern GPUs include dedicated hardware for performing matrix multiplication, separate from the general-purpose cores used for parallel scans. These concerns motivated the design of Mamba-2, which replaces parallel scans with matrix multiplication by imposing a severe "scalar times identity" constraint on the transition matrices A of the state-space model [Dao and Gu, 2024]. The scheme of iterated local corrections replaces the computation of a single linear system with the computation of a whole stack of linear systems upon the same input data. This greatly increases the arithmetic intensity of the parallel scan. Furthermore, it directly achieves nonlinearity, which requires separate operations and parameters in Mamba. Thus, it is plausible (though not certain) that replacing the Mamba block by a stack of linear systems could fully utilize hardware while achieving nonlinearity. This leaves the issue of utilizing the dedicated matrix multiplication hardware. This hardware could be used asynchronously to compute Bx_t; the size of B could be increased to saturate this hardware.

Bibliography

Inger-Lise Aamot, Siv Hege Forbord, Trine Karlsen, and Asbjørn Støylen. Does rating of perceived exertion result in target exercise intensity during interval training in cardiac rehabilitation? a study of the borg scale versus a heart rate monitor. *Journal of science and medicine in sport*, 17(5):541–545, 2014.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://github.com/tensorflow/tensorflow/blob/e61bc26e00f48db9abaf165f343f1f44a10227a9/tensorflow/python/ops/linalg_grad.py#L539`. Gradient for MatrixSolve.

Kazunori Akizuki, Syouichirou Yazaki, Yuki Echizenya, and Yukari Ohashi. Anaerobic threshold and salivary α-amylase during incremental exercise. *Journal of physical therapy science*, 26(7):1059–1063, 2014.

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in neural information processing systems*, 28, 2015.

Zeyuan Allen-Zhu and Yuanzhi Li. Can sgd learn recurrent neural networks with provable generalization? In *Advances in Neural Information Processing Systems*, pages 10331–10341, 2019.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 6673–6685, 2019a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252. PMLR, 09–15 Jun 2019b. URL `https://proceedings.mlr.press/v97/allen-zhu19a.html`.

Gregory S Ammar, William B Gragg, and Lothar Reichel. An analogue for szegő polynomials of

the clenshaw algorithm. *Journal of computational and applied mathematics*, 46(1-2):211–216, 1993.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023.

Athanasios C Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Siam, 2005.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128, 2016.

David Armstrong. Professionalism, indeterminacy and the ebm project. *BioSocieties*, 2(1):73–84, 2007.

Richard Arneson. Equality of opportunity. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2015 edition, 2015.

Babak Mohammadzadeh Asl, Ahmad R Sharafat, and S Kamaledin Setarehdan. An adaptive backpropagation neural network for arrhythmia classification using rr interval signal. *Neural Network World*, 22(6):535, 2012.

Jared L Aurentz, Thomas Mach, Raf Vandebril, and David S Watkins. Fast and backward stable computation of roots of polynomials. *SIAM Journal on Matrix Analysis and Applications*, 36 (3):942–973, 2015.

Haim Avron, Kenneth L Clarkson, and David P Woodruff. Faster kernel ridge regression using sketching and preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 38(4): 1116–1138, 2017.

Christina Baek, J Zico Kolter, and Aditi Raghunathan. Why is SAM robust to label noise? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3aZCPl3ZvR.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

W. L. Baker, C. Michael White, J. C. Cappelleri, J. Kluger, C. I. Coleman, and From the Health Outcomes, Policy, and Economics (HOPE) Collaborative Group . Understanding heterogeneity in meta-analysis: the role of meta-regression. *International Journal of Clinical Practice*, 63(10):1426–1434, 2009. doi: https://doi.org/10.1111/j.1742-1241.2009.02168.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1742-1241.2009.02168.x.

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *Journal of the ACM (JACM)*, 57(3):1–46, 2010.

David Balduzzi and Muhammad Ghifary. Strongly-typed recurrent neural networks. In *International Conference on Machine Learning*, pages 1292–1300, 2016.

Idriz Balla, Elizana Petrela, and Anesti Kondili. Pharmacological conversion of recent atrial fib-

rillation: a randomized, placebo-controlled study of three antiarrhythmic drugs/yeni baslayan atriyal fibrilasyonun ilaçla sinüs ritmine döndürülmesi: Üç antiaritmik ilaçla gerçeklestirilen randomize, plasebo-kontrollü çalisma. *The Anatolian Journal of Cardiology*, 11(7):600, 2011.

SP Banks. Nonlinear delay systems, lie algebras and lyapunov transformations. *IMA Journal of Mathematical Control and Information*, 19(1_and_2):59–72, 2002.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486 – 507, 2021. doi: 10.1214/20-AOS1965. URL https://doi.org/10.1214/20-AOS1965.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Stephen Barnett. Some applications of the comrade matrix. *International Journal of Control*, 21 (5):849–855, 1975.

Roberto Barrio. Parallel algorithms to evaluate orthogonal polynomial series. *SIAM Journal on Scientific Computing*, 21(6):2225–2239, 2000.

Romeo B Batacan, Mitch J Duncan, Vincent J Dalbo, Patrick S Tucker, and Andrew S Fenning. Effects of high-intensity interval training on cardiometabolic health: a systematic review and meta-analysis of intervention studies. *Br J Sports Med*, 51(6):494–503, 2017.

Atılım Günes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018.

Tom Bella, Yuli Eidelman, Israel Gohberg, Israel Koltracht, and Vadim Olshevsky. A björck–pereyra-type algorithm for szegö–vandermonde matrices based on properties of unitary hessenberg matrices. *Linear algebra and its applications*, 420(2-3):634–647, 2007.

Tom Bella, Yuli Eidelman, Israel Gohberg, and Vadim Olshevsky. Computations with quasiseparable polynomials and matrices. *Theoretical Computer Science*, 409(2):158–179, 2008.

Tom Bella, Yuli Eidelman, Israel Gohberg, Israel Koltracht, and Vadim Olshevsky. A fast björck–pereyra-type algorithm for solving hessenberg-quasiseparable-vandermonde systems. *SIAM Journal on Matrix Analysis and Applications*, 31(2):790–815, 2009.

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Kjell Benson and Arthur J Hartz. A comparison of observational studies and randomized, controlled trials. *New England Journal of Medicine*, 342(25):1878–1886, 2000.

Jean-Paul Berrut and Lloyd N Trefethen. Barycentric lagrange interpolation. *SIAM review*, 46 (3):501–517, 2004.

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106:1039–1082, 2017.

Ioana Bica, Ahmed M Alaa, Craig Lambert, and Mihaela Van Der Schaar. From real-world patient data to individualized treatment effects using machine learning: current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics*, 109(1): 87–100, 2021.

Dean Billheimer. Predictive inference and scientific reproducibility. *The American Statistician*, 73(sup1):291–295, 2019.

Ake Björck and Victor Pereyra. Solution of vandermonde systems of equations. *Mathematics of computation*, 24(112):893–903, 1970.

Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

Guy E Blelloch. Prefix sums and their applications. In *Synthesis of parallel algorithms*, pages 35—60. Morgan Kaufmann Publishers Inc., 1990. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.6430.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003a. ISSN 0004-5411. doi: 10.1145/792538.792543. URL http://doi.acm.org/10.1145/792538.792543.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003b.

Olga L Bocanegra, Miguel M Diaz, Renata R Teixeira, Silvio S Soares, and Foued S Espindola. Determination of the lactate threshold by means of salivary biomarkers: chromogranin a as novel marker of exercise intensity. *European journal of applied physiology*, 112(9):3195–3203, 2012.

Silke Boettger, Christian Puta, Vikram K Yeragani, Lars Donath, Hans-Josef Mueller, Holger HW Gabriel, and Karl-Juergen Baer. Heart rate variability, qt variability, and electrodermal activity during exercise. *Medicine & science in sports & exercise*, 42(3):443–448, 2010.

Ignace Bogaert. Iteration-free computation of gauss–legendre quadrature nodes and weights. *SIAM Journal on Scientific Computing*, 36(3):A1008–A1026, 2014.

Michael Borenstein. Avoiding common mistakes in meta-analysis: Understanding the distinct roles of q, i-squared, tau-squared, and the prediction interval in reporting heterogeneity. *Research Synthesis Methods*, 15(2):354–368, 2024. doi: https://doi.org/10.1002/jrsm.1678. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jrsm.1678.

Jos A. Bosch, Enno C.I. Veerman, Eco J. de Geus, and Gordon B. Proctor. Alpha-amylase as a reliable and convenient measure of sympathetic activity: don't start salivating just yet! *Psychoneuroendocrinology*, 36(4):449 – 453, 2011. ISSN 0306-4530. doi: https://doi.org/10.1016/j.psyneuen.2010.12.019. URL http://www.sciencedirect.com/science/article/pii/S0306453011000072.

Alin Bostan, Bruno Salvy, and Éric Schost. Fast conversion algorithms for orthogonal polynomials. *Linear Algebra and its Applications*, 432(1):249–258, 2010.

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-Recurrent Neural Networks. *International Conference on Learning Representations (ICLR 2017)*, 2017.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Louis Brand. The companion matrix and its properties. *The American Mathematical Monthly*, 71(6):629–634, 1964.

Christopher J Bryan, Elizabeth Tipton, and David S Yeager. Behavioural science is unlikely to change the world without a heterogeneity revolution. *Nature human behaviour*, 5(8):980–989, 2021.

Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34:28811–28822, 2021.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Martin Buchheit, Paul B Laursen, and Saïd Ahmaidi. Parasympathetic reactivation after repeated sprint exercise. *American journal of physiology-heart and circulatory physiology*, 293(1): H133–H141, 2007.

Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, pages 453–510, 1989.

Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2):329–357, 2003.

Evgeny Burnaev and Ivan Nazarov. Conformalized kernel ridge regression. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pages 45–52. IEEE, 2016.

D Calvetti and L Reichel. Fast inversion of vandermonde-like matrices involving orthogonal polynomials. *BIT Numerical Mathematics*, 33(3):473–484, 1993.

Felipe Calvo, José L Chicharro, Fernando Bandrés, Alejandro Lucía, Margarita Pérez, Julián Álvarez, Luis L Mojares, Almudena F Vaquero, and Julio C Legido. Anaerobic threshold determination with analysis of salivary amylase. *Canadian Journal of Applied Physiology*, 22 (6):553–561, 1997.

Daniel L Carl, Pierce Boyne, Bradley Rockwell, Myron Gerson, Jane Khoury, Brett Kissela, and Kari Dunning. Preliminary safety analysis of high-intensity interval training (hiit) in persons with chronic stroke. *Applied physiology, nutrition, and metabolism*, 42(3):311–318, 2016.

Bo Chang, Minmin Chen, Eldad Haber, and Ed H Chi. Antisymmetricrnn: A dynamical system view on recurrent neural networks. In *International Conference on Learning Representations*,

2018.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

José L Chicharro, Alejandro Lucía, Margarita Pérez, Almudena F Vaquero, and Rosario Ureña. Saliva composition and exercise. *Sports medicine*, 26(1):17–27, 1998.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Tayfun Cimen. Systematic and effective design of nonlinear feedback controllers via the state-dependent riccati equation (sdre) method. *Annual Reviews in Control*, 34(1):32 – 51, 2010. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2010.03.001. URL `http://www.sciencedirect.com/science/article/pii/S1367578810000052`.

Tayfun Çimen and Stephen P Banks. Nonlinear optimal tracking control with application to super-tankers for autopilot design. *Automatica*, 40(11):1845–1863, 2004.

Andrea Cipriani, Julian PT Higgins, John R Geddes, and Georgia Salanti. Conceptual and technical challenges in network meta-analysis. *Annals of internal medicine*, 159(2):130–137, 2013.

Andrea Cipriani, Toshi A Furukawa, Georgia Salanti, Anna Chaimani, Lauren Z Atkinson, Yusuke Ogawa, Stefan Leucht, Henricus G Ruhe, Erick H Turner, Julian PT Higgins, et al. Comparative efficacy and acceptability of 21 antidepressant drugs for the acute treatment of adults with major depressive disorder: a systematic review and network meta-analysis. *The Lancet*, 391(10128):1357–1366, 2018.

Ed S Coakley and Vladimir Rokhlin. A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. *Applied and Computational Harmonic Analysis*, 34(3):379–414, 2013.

Bénédicte Colnet, Imke Mayer, Guanhua Chen, Awa Dieng, Ruohong Li, Gaël Varoquaux, Jean-Philippe Vert, Julie Josse, and Shu Yang. Causal inference methods for combining randomized trials and observational studies: a review. *Statistical science*, 39(1):165–191, 2024.

John Concato, Nirav Shah, and Ralph I Horwitz. Randomized, controlled trials, observational studies, and the hierarchy of research designs. *New England journal of medicine*, 342(25): 1887–1892, 2000.

PA Cook. On some questions concerning controllability and observability indices. *IFAC Proceedings Volumes*, 11(1):1699–1705, 1978.

Gilberto Oliveira Corrêa and Keith Glover. Pseudo-canonical forms, identifiable parametrizations and simple parameter estimation for linear multivariable systems: Input-output models. *Automatica*, 20(4):429–442, 1984.

Corinna Cortes and Mehryar Mohri. *AUC optimization vs. Error rate minimization*. Neural information processing systems foundation, 2004. ISBN 0262201526.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28, 2015.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Re. Learning fast algorithms for linear transforms using butterfly factorizations. volume 97 of *Proceedings of Machine Learning Research*, pages 1517–1527, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/dao19a.html.

Tri Dao, Nimit Sohoni, Albert Gu, Matthew Eichhorn, Amit Blonder, Megan Leszczynski, Atri Rudra, and Christopher Ré. Kaleidoscope: An efficient, learnable representation for all structured linear maps. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkgrBgSYDS.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017.

Katrien De Cock and Bart De Moor. Subspace angles between arma models. *Systems & Control Letters*, 46(4):265–270, 2002.

Zuzana de Jong, Marten Munneke, Aeilko H. Zwinderman, Herman M. Kroon, Annemarie Jansen, Karel H. Ronday, Dirkjan van Schaardenburg, Ben A. C. Dijkmans, Cornelia H. M. Van den Ende, Ferdinand C. Breedveld, Theodora P. M. Vliet Vlieland, and Johanna M. W. Hazes. Is a long-term high-intensity exercise program effective and safe in patients with rheumatoid arthritis?: Results of a randomized controlled trial. *Arthritis & Rheumatism*, 48(9): 2415–2424, 2003. doi: 10.1002/art.11216. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/art.11216.

VN De Oliveira, A Bessa, RPMS Lamounier, MG De Santana, MT De Mello, and FS Espindola. Changes in the salivary biomarkers induced by an effort test. *International journal of sports medicine*, 31(06):377–381, 2010.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47, 2019.

Jonathan J Deeks and Julian PT Higgins. Statistical algorithms in review manager 5. *Statistical methods group of the Cochrane Collaboration*, 1(11), 2010.

Ilker Demirel, Ahmed Alaa, Anthony Philippakis, and David Sontag. Prediction-powered generalization of causal inferences. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=QKnWXX3aVm.

Peter B Denton, Stephen J Parke, Terence Tao, and Xining Zhang. Eigenvectors from eigenvalues. *arXiv preprint arXiv:1908.03795*, 2019.

Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7(3):177–188, 1986. ISSN 0197-2456. doi: https://doi.org/10.1016/0197-2456(86)90046-2. URL https://www.sciencedirect.com/science/article/pii/0197245686900462.

James L Devin, Michelle M Hill, Marina Mourtzakis, Joe Quadrilatero, David G Jenkins, and Tina L Skinner. Acute high intensity interval exercise reduces colon cancer cell growth. *The Journal of physiology*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Inderjit S Dhillon, Beresford N Parlett, and Christof Vömel. The design and implementation of the mrrr algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):533–560, 2006.

Feng Ding. Transformations between some special matrices. *Computers & Mathematics with Applications*, 59(8):2676–2695, 2010.

Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. Class-conditional conformal prediction with many classes. *Advances in Neural Information Processing Systems*, 36, 2024.

James R Driscoll, Dennis M Healy Jr, and Daniel N Rockmore. Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs. *SIAM Journal on Computing*, 26(4):1066–1099, 1997.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1115-1. doi: 10.1145/2090236.2090255. URL http://doi.acm.org/10.1145/2090236.2090255.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108, 2021.

B Eastman, I-J Kim, BL Shader, and KN Vander Meulen. Companion matrix patterns. *Linear Algebra and its Applications*, 463:255–272, 2014.

Robert Edelberg. Effect of vasoconstriction on galvanic skin response amplitude. *Journal of applied physiology*, 19(3):427–430, 1964.

Øyvind Ellingsen, Martin Halle, Viviane Conraads, Asbjørn Støylen, Håvard Dalen, Charles Delagardelle, Alf-Inge Larsen, Torstein Hole, Alessandro Mezzani, Emeline M Van Crae-

nenbroeck, et al. High-intensity interval training in patients with heart failure with reduced ejection fractionclinical perspective. *Circulation*, 135(9):839–849, 2017a.

Oyvind Ellingsen, Martin Halle, Eva Prescott, and Axel Linke. Response by ellingsen et al to letters regarding article, "high-intensity interval training in patients with heart failure with reduced ejection fraction". *Circulation*, 136(6):611–612, 2017b. doi: 10.1161/CIRCULATIONAHA.117.029145. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.117.029145.

Adrian D Elliott, Kanchani Rajopadhyaya, David J Bentley, John F Beltrame, and Edoardo C Aromataris. Interval training versus continuous exercise in patients with coronary artery disease: a meta-analysis. *Heart, Lung and Circulation*, 24(2):149–157, 2015.

Julian H Elliott, Anneliese Synnot, Tari Turner, Mark Simmonds, Elie A Akl, Steve McDonald, Georgia Salanti, Joerg Meerpohl, Harriet MacLehose, John Hilton, et al. Living systematic review: 1. introduction—the why, what, when, and how. *Journal of clinical epidemiology*, 91: 23–30, 2017.

William Faulkner. *Requiem for a Nun*. Random House, New York, 1951.

Oliver Faust, Alex Shenfield, Murtadha Kareem, Tan Ru San, Hamido Fujita, and U Rajendra Acharya. Automated detection of atrial fibrillation using long short-term memory network with rr interval signals. *Computers in biology and medicine*, 102:327–335, 2018.

Alvan R Feinstein. Meta-analysis: statistical alchemy for the 21st century. *Journal of clinical epidemiology*, 48(1):71–79, 1995.

Shai Feldman, Bat-Sheva Einbinder, Stephen Bates, Anastasios N Angelopoulos, Asaf Gendler, and Yaniv Romano. Conformal prediction is robust to dispersive label noise. In *Conformal and Probabilistic Prediction with Applications*, pages 624–626. PMLR, 2023.

Vitaly Feldman, Subhash Khot, and Parikshit Gopalan. New results for learning noisy parities and halfspaces. In *In Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS*, pages 563–574, 2006.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

Christian Fiedler, Carsten W Scherer, and Sebastian Trimpe. Practical and rigorous uncertainty bounds for gaussian process regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7439–7447. AAAI, 2021.

Miroslav Fiedler. A note on companion matrices. *Linear Algebra and its Applications*, 372: 325–331, 2003.

Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 144–152. SIAM, 2016.

James P Fisher, Ahmed M Adlan, Alena Shantsila, J Frederik Secher, Henrik Sørensen, and Niels H Secher. Muscle metaboreflex and autonomic regulation of heart rate in humans. *The Journal of physiology*, 591(15):3777–3788, 2013.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Dylan Foster, Tuhin Sarkar, and Alexander Rakhlin. Learning nonlinear dynamical systems from a single trajectory. volume 120 of *Proceedings of Machine Learning Research*, pages 851–861, The Cloud, 10–11 Jun 2020. PMLR.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Dan Fu, Simran Arora, Jessica Grogan, Isys Johnson, Evan Sabri Eyuboglu, Armin Thomas, Benjamin Spector, Michael Poli, Atri Rudra, and Christopher Ré. Monarch mixer: A simple sub-quadratic gemm-based architecture. *Advances in Neural Information Processing Systems*, 36, 2024.

Daniel Y Fu, Tri Dao, Khaled Kamal Saab, Armin W Thomas, Atri Rudra, and Christopher Re. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=COZDy0WYGg.

Hei Tao Fung and Kevin J Parker. Design of image-adaptive quantization tables for jpeg. *Journal of Electronic Imaging*, 4(2):144–151, 1995.

Walter Gautschi. Computational aspects of three-term recurrence relations. *SIAM review*, 9(1): 24–82, 1967.

Walter Gautschi. Orthogonal polynomials—constructive theory and applications. *Journal of Computational and Applied Mathematics*, 12:61–76, 1985.

Walter Gautschi. How (un) stable are vandermonde systems. *Asymptotic and computational analysis*, 124:193–210, 1990.

Walter Gautschi. *Orthogonal polynomials*. Oxford university press Oxford, 2004.

Walter Gautschi. Orthogonal polynomials (in matlab). *Journal of computational and applied mathematics*, 178(1-2):215–234, 2005.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/gehring17a.html.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

MR Gevers and Tsoi Ah-Chung. A new and wider class of overlapping forms for the presentation of multivariable systems. *IFAC Proceedings Volumes*, 18(5):743–747, 1985.

Udaya Ghai, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. No-regret prediction in marginally stable systems. volume 125 of *Proceedings of Machine Learning Research*, pages 1714–1757. PMLR, 09–12 Jul 2020. URL http://proceedings.mlr.press/v125/

ghai20a.html.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *arXiv preprint arXiv:2305.12616*, 2023.

Martin Gjoreski, Hristijan Gjoreski, Mitja Luštrek, and Matjaž Gams. Deep affect recognition from rr intervals. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 754–762. ACM, 2017.

Keith Glover. *Structural aspects of system identification.* PhD thesis, Massachusetts Institute of Technology, 1973.

Keith Glover and Jan Willems. Parametrizations of linear dynamical systems: Canonical forms and identifiability. *IEEE Transactions on Automatic Control*, 19(6):640–646, 1974.

Israel Gohberg and Vadim Olshevsky. Fast algorithms with preprocessing for matrix-vector multiplication problems. *Journal of Complexity*, 10(4):411–427, 1994.

Israel Gohberg and Vadim Olshevsky. The fast generalized parker–traub algorithm for inversion of vandermonde and related matrices. *Journal of Complexity*, 13(2):208–234, 1997.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pages 2214–2224, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. URL http://arxiv.org/abs/1412.6572.

Parikshit Gopalan, Michael Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 39936–39956. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/7d693203215325902ff9dbdd067a50ac-Paper-Conference.pdf.

Stephen Jay Gould. The median isn't the message. *Ceylon Medical Journal*, 49(4), 2010.

Guido Grassi and Murray Esler. How to assess sympathetic activity in humans. *Journal of hypertension*, 17(6):719–734, 1999.

Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, 26(1):19–45, 2000.

Peter D Grünwald. The e-posterior. *Philosophical Transactions of the Royal Society A*, 381 (2247):20220146, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re. Hippo: Recurrent memory with optimal polynomial projections. *arXiv preprint arXiv:2008.07669*, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022.

Gordon Guyatt, David Sackett, D Wayne Taylor, John Chong, Robin Roberts, and Stewart Pugsley. Determining optimal therapy–randomized trials in individual patients. *The New England journal of medicine*, 314(14):889–892, 1986.

Gordon H. Guyatt, David L. Sackett, John C. Sinclair, Robert Hayward, Deborah J. Cook, Richard J. Cook, Eric Bass, Hertzel Gerstein, Brian Haynes, Anne Holbrook, Roman Jaeschke, Andreas Laupacls, Virginia Moyer, and Mark Wilson. Users' Guides to the Medical Literature: IX. A Method for Grading Health Care Recommendations. *JAMA*, 274(22):1800–1804, 12 1995. ISSN 0098-7484. doi: 10.1001/jama.1995.03530220066035. URL https://doi.org/10.1001/jama.1995.03530220066035.

Yuta Hamaguchi, Hisashi Noma, Kengo Nagashima, Tomohide Yamada, and Toshi A Furukawa. Frequentist performances of bayesian prediction intervals for random-effects meta-analysis. *Biometrical Journal*, 63(2):394–405, 2021.

Amanda L Hannan, Wayne Hing, Vini Simas, Mike Climstein, Jeff S Coombes, Rohan Jayasinghe, Joshua Byrnes, and James Furness. High-intensity interval training versus moderate-intensity continuous training within cardiac rehabilitation: a systematic review and meta-analysis. *Open access journal of sports medicine*, 9:1, 2018.

Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1): 65, 2019.

Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *arXiv preprint arXiv:1609.05191*, 2016a.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 111–122, New York, NY, USA, 2016b. ACM. ISBN 978-1-4503-4057-1. doi: 10.1145/2840728.2840730. URL http://doi.acm.org/10.1145/2840728.2840730.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In

Proceedings of Advances in Neural Information Processing Systems (NIPS), 2016c.

Joachim Hartung and Guido Knapp. On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in medicine*, 20(12):1771–1782, 2001.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 6702–6712, 2017.

Elad Hazan, Sham Kakade, and Karan Singh. The nonstochastic control problem. In *Algorithmic Learning Theory*, pages 408–421. PMLR, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016b.

Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1939–1948. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/hebert-johnson18a.html`.

Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent orthogonal networks and long-memory tasks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 2034–2042, 2016.

Rachel Heyard, Leonhard Held, Sebastian Schneeweiss, and Shirley V Wang. Design differences and variation in results between randomised trials and non-randomised emulations: meta-analysis of rct-duplicate data. *BMJ Medicine*, 3(1), 2024. doi: 10.1136/bmjmed-2023-000709. URL `https://bmjmedicine.bmj.com/content/3/1/e000709`.

Julian PT Higgins, Simon G Thompson, and David J Spiegelhalter. A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 172(1):137–159, 2009.

Julian PT Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J Page, and Vivian A Welch. *Cochrane Handbook for Systematic Reviews of Interventions*. John Wiley & Sons, Chichester, UK, 2nd edition, 2019.

Nicholas J Higham. Error analysis of the björck-pereyra algorithms for solving vandermonde systems. *Numerische Mathematik*, 50(5):613–632, 1987.

Nicholas J Higham. Fast solution of vandermonde-like systems involving orthogonal polynomials. *IMA Journal of Numerical Analysis*, 8(4):473–486, 1988.

Nicholas J Higham. Stability analysis of algorithms for solving confluent vandermonde-like systems. *SIAM Journal on Matrix Analysis and Applications*, 11(1):23–41, 1990.

Nicholas J Higham. Iterative refinement enhances the stability ofqr factorization methods for

solving linear equations. *BIT Numerical Mathematics*, 31(3):447–468, 1991.

Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

Geoffrey E. Hinton. To recognize shapes, first learn to generate images. In Paul Cisek, Trevor Drew, and John F. Kalaska, editors, *Computational Neuroscience: Theoretical Insights into Brain Function*, volume 165 of *Progress in Brain Research*, pages 535–547. Elsevier, 2007. doi: https://doi.org/10.1016/S0079-6123(06)65034-6. URL https://www.sciencedirect.com/science/article/pii/S0079612306650346.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Falk Hoffmann, Katharina Allers, Tanja Rombey, Jasmin Helbach, Amrei Hoffmann, Tim Mathes, and Dawid Pieper. Nearly 80 systematic reviews were published each day: observational study on trends in epidemiology and reporting over the years 2000-2019. *Journal of Clinical Epidemiology*, 138:1–11, 2021.

Max Hopkins, Michael Mitzenmacher, and Sebastian Wagner-Carena. Simulated annealing for jpeg quantization. In *2018 Data Compression Conference*, pages 412–412. IEEE, 2018.

Chloe Hsu, Michaela Hardt, and Moritz Hardt. Linear dynamics: Clustering without identification. In *International Conference on Artificial Intelligence and Statistics*, pages 918–929. PMLR, 2020.

Wenbing Huang, Mehrtash Harandi, Tong Zhang, Lijie Fan, Fuchun Sun, and Junzhou Huang. Efficient optimization for linear dynamical systems with applications to clustering and sparse coding. In *Advances in Neural Information Processing Systems*, pages 3444–3454, 2017.

Christopher P Hughes and Ashkan Nikeghbali. The zeros of random polynomials cluster uniformly near the unit circle. *Compositio Mathematica*, 144(3):734–746, 2008.

Maximilian Huhn, Adriani Nikolakopoulou, Johannes Schneider-Thoma, Marc Krause, Myrto Samara, Natalie Peter, Thomas Arndt, Lio Bäckers, Philipp Rothe, Andrea Cipriani, et al. Comparative efficacy and tolerability of 32 oral antipsychotics for the acute treatment of adults with multi-episode schizophrenia: a systematic review and network meta-analysis. *The Lancet*, 394(10202):939–951, 2019.

Hilde M Huizenga, Ingmar Visser, and Conor V Dolan. Testing overall and moderator effects in random effects meta-regression. *British Journal of Mathematical and Statistical Psychology*, 64(1):1–19, 2011.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

Joanna IntHout, John PA Ioannidis, and George F Borm. The hartung-knapp-sidik-jonkman method for random effects meta-analysis is straightforward and considerably outperforms the standard dersimonian-laird method. *BMC medical research methodology*, 14:1–12, 2014.

Joanna IntHout, John PA Ioannidis, Maroeska M Rovers, and Jelle J Goeman. Plea for routinely presenting prediction intervals in meta-analysis. *BMJ open*, 6(7):e010247, 2016.

John PA Ioannidis. Contradicted and initially stronger effects in highly cited clinical research.

Jama, 294(2):218–228, 2005.

Mourad Ismail, Mourad EH Ismail, and Walter van Assche. *Classical and quantum orthogonal polynomials in one variable*, volume 13. Cambridge university press, 2005.

Dan Jackson and Ian R White. When should meta-analysis avoid making hidden normality assumptions? *Biometrical Journal*, 60(6):1040–1058, 2018.

Paras Jain, Ajay Jain, Aniruddha Nrusimha, Amir Gholami, Pieter Abbeel, Joseph Gonzalez, Kurt Keutzer, and Ion Stoica. Checkmate: Breaking the memory wall with optimal tensor rematerialization. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 497–511. 2020. URL https://proceedings.mlsys.org/paper/2020/file/084b6fbb10729ed4da8c3d3f5a3ae7c9-Paper.pdf.

Yassir Jedra and Alexandre Proutiere. Sample complexity lower bounds for linear system identification. *arXiv preprint arXiv:1903.10343*, 2019.

Charlotte Jelleyman, Thomas Yates, Gary O'Donovan, Laura J Gray, James A King, Kamlesh Khunti, and Melanie J Davies. The effects of high-intensity interval training on glucose regulation and insulin resistance: a meta-analysis. *Obesity reviews*, 16(11):942–961, 2015.

Charlotte Lauren Jelleyman. *High-intensity physical activity for improving glucose regulation: can science justify IT?* PhD thesis, Department of Cardiovascular Sciences, 2018.

J Jiang. Image compression with neural networks–a survey. *Signal processing: image Communication*, 14(9):737–760, 1999.

Ying Jin, Zhimei Ren, and Emmanuel J Candès. Sensitivity analysis of individual treatment effects: A robust conformal inference approach. *Proceedings of the National Academy of Sciences*, 120(6):e2214889120, 2023.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

D. Jordan and B. Sridhar. An efficient algorithm for calculation of the luenberger canonical form. *IEEE Transactions on Automatic Control*, 18(3):292–295, 1973.

Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Rawlsian fairness for machine learning. *CoRR*, abs/1610.09559, 2016. URL http://arxiv.org/abs/1610.09559.

Thomas Kailath and Vadim Olshevsky. Displacement-structure approach to polynomial vandermonde and related matrices. *Linear Algebra and Its Applications*, 261(1-3):49–90, 1997.

Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

Ravindran Kannan and Santosh Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135, 2017.

Prince J Kannankeril, Francis K Le, Alan H Kadish, and Jeffrey J Goldberger. Parasympathetic effects on heart rate recovery after exercise. *Journal of investigative medicine*, 52(6):394–401, 2004.

Emir Karaçağlar, İlyas Atar, Süleyman Özbiçer, Atilla Sezgin, Salih Özçobanoğlu, Ayse Canan Yazici, Bülent Özin, and Haldun Müderrisoğlu. Amiodarone versus direct current cardioversion in treatment of atrial fibrillation after cardiac surgery. *Turkish Journal of Clinics and Laboratory*, 10(1):26–32, 2019.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Shiva Kaul. Margins and opportunity. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 191–196, 2018.

Shiva Kaul. Linear dynamical systems as a core computational primitive. *Advances in Neural Information Processing Systems*, 33:16808–16820, 2020.

Shiva Kaul, Anthony Falco, and Karianne Anthes. Measuring the sympathetic response to intense exercise in a practical setting. In *Machine Learning for Healthcare Conference*, pages 680–703. PMLR, 2019.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Holly S Kessler, Susan B Sisson, and Kevin R Short. The potential for high-intensity interval training to reduce cardiometabolic disease risk. *Sports medicine*, 42(6):489–509, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

George E Kochiadakis, Nikos E Igoumenidis, Michail E Hamilos, Maria E Marketou, Gregory I Chlouverakis, and Panos E Vardas. A comparative study of the efficacy and safety of procainamide versus propafenone versus amiodarone for the conversion of recent-onset atrial fibrillation. *The American journal of cardiology*, 99(12):1721–1725, 2007.

Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page to appear, 1996.

Eri Koibuchi and Yoshio Suzuki. Exercise upregulates salivary amylase in humans. *Experimental and therapeutic medicine*, 7(4):773–777, 2014.

Docent Juhani Koistinen and Docent Tomi Laitinen. Effect of physical exercise on autonomic regulation of heart rate. 2004.

Michel Komajda, John JV McMurray, Henning Beck-Nielsen, Ramon Gomis, Markolf Hanefeld,

Stuart J Pocock, Paula S Curtis, Nigel P Jones, and Philip D Home. Heart failure events with rosiglitazone in type 2 diabetes: data from the record clinical trial. *European heart journal*, 31(7):824–831, 2010.

Mark Kozdoba, Jakub Marecek, Tigran Tchrakian, and Shie Mannor. On-line learning of linear dynamical systems: Exponential forgetting in kalman filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4098–4105, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Jean B Lasserre. Connecting optimization with spectral analysis of tri-diagonal matrices. *Mathematical Programming*, pages 1–15, 2020.

Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009.

Monique Laurent and Lucas Slot. Near-optimal analysis of univariate moment bounds for polynomial optimization. *arXiv preprint arXiv:2001.11289*, 2020.

Yann LeCun. Who is afraid of nonconvex loss functions? NIPS Workshop on Efficient Machine Learning, Whistler, 2007. URL `https://www.youtube.com/watch?v=8zdo6cnCW2w`.

Eric P. Lehman, Rahul G. Krishnan, Xiaopeng Zhao, Roger G. Mark, and Li-wei H. Lehman. Representation learning approaches to detect false arrhythmia alarms from ecg dynamics. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 3rd Machine Learning for Healthcare Conference*, volume 85 of *Proceedings of Machine Learning Research*, pages 571–586, Palo Alto, California, 17–18 Aug 2018. PMLR. URL `http://proceedings.mlr.press/v85/lehman18a.html`.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2014.

Lihua Lei and Emmanuel J Candès. Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5): 911–938, 2021.

Gregory Leibon, Daniel N Rockmore, Wooram Park, Robert Taintor, and Gregory S Chirikjian. A fast hermite transform. *Theoretical computer science*, 409(2):211–228, 2008.

Patrick H Leslie. On the use of matrices in certain population mathematics. *Biometrika*, 33(3): 183–212, 1945.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

Luz M Letelier, Kamol Udol, Javier Ena, Bruce Weaver, and Gordon H Guyatt. Effectiveness of

amiodarone for conversion of atrial fibrillation to sinus rhythm: a meta-analysis. *Archives of Internal Medicine*, 163(7):777–785, 2003.

Mario Lezcano-Casado and David Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. *arXiv preprint arXiv:1901.08428*, 2019.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020.

Tianjing Li, Kristina Lindsley, Benjamin Rouse, Hwanhee Hong, Qiyuan Shi, David S Friedman, Richard Wormald, and Kay Dickersin. Comparative effectiveness of first-line medications for primary open-angle glaucoma: a systematic review and network meta-analysis. *Ophthalmology*, 123(1):129–140, 2016.

TL Li and Michael Gleeson. The effect of single and repeated bouts of prolonged cycling and circadian variation on saliva flow rate, immunoglobulin a and-amylase responses. *J sports Sci*, 22(11-12):1015–1024, 2004.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "Ridgeless" regression can generalize. *The Annals of Statistics*, 48(3):1329 – 1347, 2020. doi: 10.1214/19-AOS1849. URL https://doi.org/10.1214/19-AOS1849.

Tengyuan Liang and Benjamin Recht. Randomization inference when n equals one. *arXiv preprint arXiv:2310.16989*, 2023.

Renjie Liao, Yuwen Xiong, Ethan Fetaya, Lisa Zhang, KiJung Yoon, Xaq Pitkow, Raquel Urtasun, and Richard Zemel. Reviving and improving recurrent back-propagation. *arXiv preprint arXiv:1803.06396*, 2018.

Jona Lilienthal, Sibylle Sturtz, Christoph Schürmann, Matthias Maiworm, Christian Röver, Tim Friede, and Ralf Bender. Bayesian random-effects meta-analysis with empirical heterogeneity priors for application in health technology assessment with very few studies. *Research Synthesis Methods*, 15(2):275–287, 2024.

NEAL Lippman, KENNETH M Stein, and BRUCE B Lerman. Comparison of methods for removal of ectopy in measurement of heart rate variability. *American Journal of Physiology-Heart and Circulatory Physiology*, 267(1):H411–H418, 1994.

Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

ZC Lipton. The mythos of model interpretability. icml 2016 workshop on human interpretability in machine learning (whi 2016), 2016.

Janusz Lisiecki, Michal Szolucha, Joaquin Anton Guirao, and Maitreyi Roy. Loading data fast with dali and the new hardware jpeg decoder in nvidia a100 gpus, 2020. URL https://developer.nvidia.com/blog/loading-data-fast-with-dali-and-new-jpeg-decoder-in-a100/.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=De4FYqjFueZ.

Jin-Guo Liu and Taine Zhao. Differentiate everything with a reversible programming language. *arXiv preprint arXiv:2003.04617*, 2020.

Ziyu Liu, Fahad M Al Amer, Mengli Xiao, Chang Xu, Luis Furuya-Kanamori, Hwanhee Hong, Lianne Siegel, and Lifeng Lin. The normality assumption on between-study random effects was questionable in a considerable number of cochrane meta-analyses. *BMC medicine*, 21(1): 112, 2023b.

Lennart Ljung. System identification. *Wiley encyclopedia of electrical and electronics engineering*, pages 1–19, 1999.

David Luenberger. Canonical forms for linear multivariable systems. *IEEE Transactions on Automatic Control*, 12(3):290–293, 1967.

Andreas Lundh, Joel Lexchin, Barbara Mintzes, Jeppe B Schroll, and Lisa Bero. Industry sponsorship and research outcome. *Cochrane database of systematic reviews*, (2), 2017.

Xiyang Luo, Hossein Talebi, Feng Yang, Michael Elad, and Peyman Milanfar. The rate-distortion-accuracy tradeoff: Jpeg case study. *arXiv preprint arXiv:2008.00605*, 2020.

Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 502–510. ACM, 2011.

Matthew MacKay, Paul Vicol, Jimmy Ba, and Roger B Grosse. Reversible recurrent neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Theresa Mann, Robert Patrick Lamberts, and Michael Ian Lambert. Methods of prescribing relative exercise intensity: physiological and practical considerations. *Sports medicine*, 43(7): 613–625, 2013.

JoAnn E Manson, Nancy R Cook, I-Min Lee, William Christen, Shari S Bassuk, Samia Mora, Heike Gibson, David Gordon, Trisha Copeland, Denise D'Agostino, et al. Vitamin d supplements and prevention of cancer and cardiovascular disease. *New England Journal of Medicine*, 380(1):33–44, 2019.

Errol B Marliss and Mladen Vranic. Intense exercise has unique effects on both insulin release and its roles in glucoregulation: implications for diabetes. *Diabetes*, 51(suppl 1):S271–S283, 2002.

James Martens. Learning the linear dynamical system with asos. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 743–750. Omnipress, 2010.

Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyUNwulC-.

Luca Masserano, Tommaso Dorigo, Rafael Izbicki, Mikael Kuusela, and Ann B Lee. Simulator-

based inference with waldo: Confidence regions by leveraging prediction algorithms and posterior estimators for inverse problems. *Proceedings of Machine Learning Research*, 206, 2023.

Nikolai Matni, Alexandre Proutiere, Anders Rantzer, and Stephen Tu. From self-tuning regulators to reinforcement learning and back again. *arXiv preprint arXiv:1906.11392*, 2019.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

Scott Michael, Kenneth S Graham, and Glen M Davis. Cardiac autonomic responses during exercise and post-exercise recovery using heart rate variability and systolic time intervals—a review. *Frontiers in physiology*, 8:301, 2017a.

Scott Michael, Ollie Jay, Kenneth S Graham, and Glen M Davis. Higher exercise intensity delays postexercise recovery of impedance-derived cardiac sympathetic activity. *Applied Physiology, Nutrition, and Metabolism*, 42(8):834–840, 2017b.

Scott Michael, Ollie Jay, Kenneth S Graham, and Glen M Davis. Longer exercise duration delays post-exercise recovery of cardiac parasympathetic but not sympathetic indices. *European journal of applied physiology*, 117(9):1897–1906, 2017c.

Zoran Milanović, Goran Sporiš, and Matthew Weston. Effectiveness of high-intensity interval training (hit) and continuous endurance training for vo 2max improvements: a systematic review and meta-analysis of controlled trials. *Sports medicine*, 45(10):1469–1481, 2015.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022.

Jeffrey P Moak, David S Goldstein, Basil A Eldadah, Ahmed Saleem, Courtney Holmes, Sandra Pechnik, and Yehonatan Sharabi. Supine low-frequency power of heart rate variability reflects baroreflex function, not cardiac sympathetic innervation. *Heart Rhythm*, 4(12):1523–1529, 2007.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

GB Moody. Rr interval time series modeling: the physionet/computers in cardiology challenge 2002. In *Computers in Cardiology*, pages 125–128. IEEE, 2002.

Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

Marilyn C Morris and Robert M Nelson. Randomized, controlled trials as minimal risk: an ethical analysis. *Critical care medicine*, 35(3):940–944, 2007.

Marcus R Munafò, Brian A Nosek, Dorothy VM Bishop, Katherine S Button, Christopher D Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J Ware, and John Ioannidis. A manifesto for reproducible science. *Nature human behaviour*, 1(1):1–9, 2017.

M Hassan Murad, Noor Asi, Mouaz Alsawas, and Fares Alahdab. New evidence pyramid. *BMJ Evidence-Based Medicine*, 21(4):125–127, 2016.

Kengo Nagashima, Hisashi Noma, and Toshi A. Furukawa. pimeta: an r package of prediction intervals for random-effects meta-analysis. arXiv, 2021.

Urs M Nater and N Rohleder. Salivary alpha-amylase as a non-invasive biomarker for the sympathetic nervous system: current state of research. *Psychoneuroendocrinology*, 34(4):486–496, 2009.

Willie Neiswanger and Aaditya Ramdas. Uncertainty quantification using martingales for misspecified gaussian processes. In *Algorithmic learning theory*, pages 963–982. PMLR, 2021.

Jersey Neyman. Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10(1):1–51, 1923.

Ilia Nouretdinov, Thomas Melluish, and Volodya Vovk. Ridge regression confidence machine. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 385–392, 2001.

Randal S Olson, William La Cava, Patryk Orzechowski, Ryan J Urbanowicz, and Jason H Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10:1–13, 2017.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.

Samet Oymak and Necmiye Ozay. Non-asymptotic identification of lti systems from a single trajectory. In *2019 American Control Conference (ACC)*, pages 5655–5661. IEEE, 2019.

Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *International journal of surgery*, 88:105906, 2021.

Christopher Partlett and Richard D Riley. Random effects meta-analysis: coverage performance of 95% confidence and prediction intervals following reml estimation. *Statistics in medicine*, 36(2):301–317, 2017.

J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2009. ISBN 9780521895606. URL `https://books.google.com/books?id=f4nuexsNVZIC`.

Henry T Peng, Erin Savage, Oshin Vartanian, Shane Smith, Shawn G Rhind, Catherine Tenn, and Stephen Bjamason. Performance evaluation of a salivary amylase biosensor for stress assessment in military field research. *Journal of clinical laboratory analysis*, 30(3):223–230, 2016.

Coby Penso and Jacob Goldberger. A conformal prediction score that is robust to label noise. *arXiv preprint arXiv:2405.02648*, 2024.

Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for clas-

sification under label shift. In *Uncertainty in artificial intelligence*, pages 844–853. PMLR, 2021.

Hugo F Posada-Quintero, Natasa Reljin, Craig Mills, Ian Mills, John P Florian, Jaci L VanHeest, and Ki H Chon. Time-varying analysis of electrodermal activity during exercise. *PloS one*, 13 (6):e0198328, 2018.

Daniel Potts. Fast algorithms for discrete polynomial transforms on arbitrary grids. *Linear algebra and its applications*, 366:353–370, 2003.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. Cambridge University Press, USA, 1992. ISBN 0521431085.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

EA Rawashdeh. A simple method for finding the inverse matrix of vandermonde matrix. *Matematički Vesnik*, 2018.

Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*.

Thomas W Reps and Louis B Rall. Computational divided differencing and divided-difference arithmetics. *Higher-order and symbolic computation*, 16(1-2):93–149, 2003.

Gustavo A Reyes del Paso, Wolf Langewitz, Lambertus JM Mulder, Arie Van Roon, and Stefan Duschek. The utility of low frequency heart rate variability as an index of sympathetic cardiac tone: a review with emphasis on a reanalysis of previous studies. *Psychophysiology*, 50(5): 477–487, 2013.

Lev Reyzin. Statistical queries and statistical algorithms: Foundations and applications. *arXiv preprint arXiv:2004.00557*, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

Kirsty M. Rhodes, Rebecca M. Turner, Ian R. White, Dan Jackson, David J. Spiegelhalter, and Julian P. T. Higgins. Implementing informative priors for heterogeneity in meta-analysis using meta-regression and pseudo data. *Statistics in Medicine*, 35(29):5495–5511, 2016. doi: https://doi.org/10.1002/sim.7090. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.7090`.

W Scott Richardson, Mark C Wilson, Jim Nishikawa, and Robert S Hayward. The well-built clinical question: a key to evidence-based decisions. *ACP journal club*, 123(3):A12–A13, 1995.

Richard D Riley, Julian P T Higgins, and Jonathan J Deeks. Interpretation of random effects meta-analyses. *BMJ*, 342, 2011. ISSN 0959-8138. doi: 10.1136/bmj.d549. URL `https://www.bmj.com/content/342/bmj.d549`.

Nicolas Rohleder and Urs M Nater. Determinants of salivary α-amylase in humans and methodological considerations. *Psychoneuroendocrinology*, 34(4):469–485, 2009.

Christian Röver. Bayesian random-effects meta-analysis using the bayesmeta r package. *arXiv preprint arXiv:1711.08683*, 2017.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Gerta Rücker. Network meta-analysis, electrical networks and graph theory. *Research synthesis methods*, 3(4):312–324, 2012.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

David L Sackett. Bias in analytic research. In *The case-control study consensus and controversy*, pages 51–63. Elsevier, 1979.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*, 2024.

Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5610–5618, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

H Schünemann, J Brożek, G Guyatt, and A Oxman, editors. *GRADE handbook for grading quality of evidence and strength of recommendations*. The GRADE Working Group, 2013.

Matthias Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269, 2002.

Matteo Sesia, YX Wang, and Xin Tong. Adaptive conformal classification with noisy labels. *arXiv preprint arXiv:2309.05092*, 2023.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Fred Shaffer and JP Ginsberg. An overview of heart rate variability metrics and norms. *Frontiers in public health*, 5:258, 2017.

Fred Shaffer, Rollin McCraty, and Christopher L Zerr. A healthy heart is not a metronome: an integrative review of the heart's anatomy and heart rate variability. *Frontiers in psychology*, 5: 1040, 2014.

Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

Uri Shalit and Gal Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 548–556, Bejing, China, 22–24 Jun 2014. PMLR. URL `http://proceedings.mlr.press/v32/shalit14.html`.

John Shawe-Taylor and Robert C Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9, 1997.

Vivek Shetty, Corwin Zigler, Theodore F Robles, David Elashoff, and Masaki Yamaguchi. Developmental validation of a point-of-care, salivary α-amylase biosensor. *Psychoneuroendocrinology*, 36(2):193–199, 2011.

Richard Shin and Dawn Song. Jpeg-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, volume 1, 2017.

J. Kevin Shoemaker, Stephen A. Klassen, Mark B. Badrov, and Paul J. Fadel. Fifty years of microneurography: learning the language of the peripheral sympathetic nervous system in humans. *Journal of Neurophysiology*, 119(5):1731–1744, 2018. doi: 10.1152/jn.00841.2017. URL `https://doi.org/10.1152/jn.00841.2017`. PMID: 29412776.

Kurex Sidik and Jeffrey N Jonkman. On constructing confidence intervals for a standardized mean difference in meta-analysis. *Communications in Statistics-Simulation and Computation*, 32(4):1191–1203, 2003.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. *arXiv preprint arXiv:1802.08334*, 2018.

Max Simchowitz, Ross Boczar, and Benjamin Recht. Learning linear dynamical systems with semi-parametric least squares. *arXiv preprint arXiv:1902.00768*, 2019.

C. Simoiu, S. Corbett-Davies, and S. Goel. The Problem of Infra-marginality in Outcome Tests for Discrimination. *ArXiv e-prints*, July 2016.

Barry Simon. The classical moment problem as a self-adjoint finite difference operator. *Advances in Mathematics*, 137(1):82–203, 1998.

James B Simon, Dhruva Karkada, Nikhil Ghosh, and Mikhail Belkin. More is better: when infinite overparameterization is optimal and overfitting is obligatory. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=OdpIjS0vkO`.

Uri Simonsohn, Joseph Simmons, and Leif D Nelson. Above averaging in literature reviews. *Nature Reviews Psychology*, 1(10):551–552, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023.

Richard S Slavik and Peter J Zed. Intravenous amiodarone for conversion of atrial fibrillation: Misled by meta-analysis? *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy*, 24(6):792–798, 2004.

Francis J Smith. An algorithm for summing orthogonal polynomial series and their derivatives with applications to curve-fitting and interpolation. *Mathematics of Computation*, 19(89): 33–36, 1965.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Teresa C Smith, David J Spiegelhalter, and Andrew Thomas. Bayesian approaches to random-effects meta-analysis: a comparative study. *Statistics in medicine*, 14(24):2685–2699, 1995.

Alicja Smoktunowicz. Backward stability of clenshaw's algorithm. *BIT Numerical Mathematics*, 42(3):600–610, 2002.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

Nathan Srebro and Ambuj Tewari. Stochastic optimization for machine learning. *ICML Tutorial*, 2010.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

T. D. Stanley and Stephen B. Jarrell. Meta-regression analysis: A quantitative method of literature surveys. *Journal of Economic Surveys*, 3(2):161–170, 1989. doi: https://doi.org/10.1111/j.1467-6419.1989.tb00064.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6419.1989.tb00064.x.

Theresa A Strzelczyk, Rebecca J Quigg, Pamela B Pfeifer, Michele A Parker, and Philip Greenland. Accuracy of estimating exercise prescription intensity in patients with left ventricular systolic dysfunction. *Journal of Cardiopulmonary Rehabilitation and Prevention*, 21(3):158–163, 2001.

David Stutz, Krishnamurthy Dj Dvijotham, Ali Taylan Cemgil, and Arnaud Doucet. Learning optimal conformal classifiers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=t8O-4LKFVx.

David Stutz, Abhijit Guha Roy, Tatiana Matejovicova, Patricia Strachan, Ali Taylan Cemgil, and Arnaud Doucet. Conformal prediction under ambiguous ground truth. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=CAd6V2qXxc.

SV Subramanian, Rockli Kim, and Nicholas A Christakis. The "average" treatment effect: A construct ripe for retirement. a commentary on deaton and cartwright. *Social science & medicine*, 210:77–82, 2018.

Gabor Szego. *Orthogonal polynomials*, volume 23. American Mathematical Soc., 1939.

Corentin Tallec and Yann Ollivier. Can recurrent neural networks warp time? In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJcKhk-Ab.

Hirofumi Tanaka, Kevin D Monahan, and Douglas R Seals. Age-predicted maximal heart rate revisited. *Journal of the american college of cardiology*, 37(1):153–156, 2001.

Anna Thomas, Albert Gu, Tri Dao, Atri Rudra, and Christopher Ré. Learning compressed transforms with low displacement rank. In *Advances in neural information processing systems*, pages 9052–9060, 2018.

Bianca Lee Thomas, Nicolaas Claassen, Piet Becker, and Margaretha Viljoen. Validity of commonly used heart rate variability markers of autonomic nervous system function. *Neuropsy-*

chobiology, 78(1):14–26, 2019.

Stuart P Thomas, Duncan Guy, Elisabeth Wallace, Roselyn Crampton, Pat Kijvanit, Vicki Eipper, David L Ross, and Mark J Cooper. Rapid loading of sotalol or amiodarone for management of recent onset symptomatic atrial fibrillation: a randomized, digoxin-controlled trial. *American heart journal*, 147(1):E3, 2004.

Simon G. Thompson and Julian P. T. Higgins. How should meta-regression analyses be undertaken and interpreted? *Statistics in Medicine*, 21(11):1559–1573, 2002. doi: https://doi.org/10.1002/sim.1187. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1187.

George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020), 2020. URL http://www.compression.cc.

Ingrid Toews, Andrew Anglemyer, John LZ Nyirenda, Dima Alsaid, Sara Balduzzi, Kathrin Grummich, Lukas Schwingshackl, and Lisa Bero. Healthcare outcomes assessed with observational study designs compared with those assessed in randomized trials: a meta-epidemiological study. *Cochrane Database of Systematic Reviews*, (1), 2024.

María Tomás-Rodríguez and Stephen P Banks. *Linear, time-varying approximations to nonlinear dynamical systems: with applications in control and optimization*, volume 400. Springer Science & Business Media, 2010.

Markos G Tsipouras, Dimitrios I Fotiadis, and D Sideris. An arrhythmia classification system based on the rr-interval signal. *Artificial intelligence in medicine*, 33(3):237–250, 2005.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, Anil Palepu, Basil Mustafa, Aakanksha Chowdhery, Yun Liu, Simon Kornblith, David Fleet, Philip Mansfield, Sushant Prakash, Renee Wong, Sunny Virmani, Christopher Semturs, S. Sara Mahdavi, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Karan Singhal, Pete Florence, Alan Karthikesalingam, and Vivek Natarajan. Towards generalist biomedical ai. *NEJM AI*, 1(3):AIoa2300138, 2024. doi: 10.1056/AIoa2300138. URL https://ai.nejm.org/doi/abs/10.1056/AIoa2300138.

Jessica L Unick, Sarah Gaussoin, Judy Bahnson, Richard Crow, Jeff Curtis, Tina Killean, Judith G Regensteiner, Kerry J Stewart, Rena R Wing, John M Jakicic, et al. Validity of ratings of perceived exertion in patients with type 2 diabetes. *Journal of novel physiotherapy and physical rehabilitation*, 1(1), 2014.

Gaetano Valenza, Luca Citi, J. Philip Saul, and Riccardo Barbieri. Measures of sympathetic and parasympathetic autonomic outflow from heartbeat dynamics. *Journal of Applied Physiology*, 125(1):19–39, 2018. doi: 10.1152/japplphysiol.00842.2017. URL https://doi.org/10.1152/japplphysiol.00842.2017. PMID: 29446712.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Paul Valiant. Three perspectives on orthogonal polynomials. FOCS 2016 Workshop: (Some)

Orthogonal Polynomials and their Applications to TCS, 2016. URL http://www.cs.columbia.edu/~ccanonne/workshop-focs2016/.

Ake B. Vallbo, Karl-Erik Hagbarth, and B. Gunnar Wallin. Microneurography: how the technique developed and its role in the investigation of the sympathetic nervous system. *Journal of Applied Physiology*, 96(4):1262–1269, 2004. doi: 10.1152/japplphysiol.00470.2003. URL https://doi.org/10.1152/japplphysiol.00470.2003. PMID: 15016790.

Marieke van Dooren, Joris H Janssen, et al. Emotional sweating across the body: Comparing 16 different skin conductance measurement locations. *Physiology & behavior*, 106(2):298–304, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Stephen Vavasis. Some notes on applying computational divided differencing in optimization. *arXiv preprint arXiv:1307.4097*, 2013.

Areti Angeliki Veroniki. Random-effects meta-analysis methods in revman (cochrane statistical editor training 2022). YouTube video, 2022. URL https://www.youtube.com/watch?v=4gsaUl5uh70.

Areti Angeliki Veroniki, Dan Jackson, Ralf Bender, Oliver Kuss, Dean Langan, Julian P.T. Higgins, Guido Knapp, and Georgia Salanti. Methods to calculate uncertainty in the estimated overall effect size from a random-effects meta-analysis. *Research Synthesis Methods*, 10(1): 23–43, 2019. doi: https://doi.org/10.1002/jrsm.1319. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jrsm.1319.

Wolfgang Viechtbauer. Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics*, 30(3):261–293, 2005.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.

Vladimir Vovk. Testing randomness online. *Statistical Science*, 36(4):595–611, 2021.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

Shirley V Wang, Sebastian Schneeweiss, Jessica M Franklin, Rishi J Desai, William Feldman, Elizabeth M Garry, Robert J Glynn, Kueiyu Joshua Lin, Julie Paik, Elisabetta Patorno, et al. Emulation of randomized clinical trials with nonrandomized database analyses: results of 32

clinical trials. *Jama*, 329(16):1376–1385, 2023.

Kassia S Weston, Ulrik Wisløff, and Jeff S Coombes. High-intensity interval training in patients with lifestyle-induced cardiometabolic disease: a systematic review and meta-analysis. *Br J Sports Med*, 48(16):1227–1234, 2014.

Michael A Wewege, Dohee Ahn, Jennifer Yu, Kevin Liou, and Andrew Keech. High-intensity interval training for patients with cardiovascular disease—is it safe? a systematic review. *Journal of the American Heart Association*, 7(21):e009305, 2018.

Daniel W White and Peter B Raven. Autonomic neural control of heart rate during dynamic exercise: revisited. *The Journal of physiology*, 592(12):2491–2500, 2014.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in neural information processing systems*, pages 4880–4888, 2016.

Mingzhang Yin, Claudia Shi, Yixin Wang, and David M Blei. Conformal sensitivity analysis for individual treatment effects. *Journal of the American Statistical Association*, 119(545): 122–135, 2024.

Hye Sun Yun, David Pogrebitskiy, Iain J Marshall, and Byron C Wallace. Automatically extracting numerical results from randomized controlled trials with large language models. *arXiv preprint arXiv:2405.01686*, 2024.

Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *26th International World Wide Web Conference (WWW'17)*, 2017a.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 962–970, Fort Lauderdale, FL, USA, 20–22 Apr 2017b. PMLR. URL `http://proceedings.mlr.press/v54/zafar17a.html`.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Sy8gdB9xx`.

Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

Tijana Zrnic and Emmanuel J Candès. Cross-prediction-powered inference. *Proceedings of the National Academy of Sciences*, 121(15):e2322083121, 2024.