Reliable Validation: New Perspectives on Adaptive Data Analysis and Cross-Validation

by

Samuel Scott Elder

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Signature redacted

Author ..........

Department of Mathematics
August 7, 2018

Certified by ......

Signature redacted

Jonathan Kelner
Professor
Thesis Supervisor

Certified by ...........

Signature redacted

Tamara Broderick
Assistant Professor
Thesis Supervisor

Accepted by ........

Signature redacted

Jonathan Kelner
Chairman, Department Committee on Graduate Theses
DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.
Reliable Validation: New Perspectives on Adaptive Data Analysis and Cross-Validation

by

Samuel Scott Elder

Submitted to the Department of Mathematics on August 7, 2018, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

Validation refers to the challenge of assessing how well a learning algorithm performs after it has been trained on a given data set. It forms an important step in machine learning, as such assessments are then used to compare and choose between algorithms and provide reasonable approximations of their accuracy.

In this thesis, we provide new approaches for addressing two common problems with validation. In the first half, we assume a simple validation framework, the holdout set, and address an important question of how many algorithms can be accurately assessed using the same holdout set, in the particular case where these algorithms are chosen adaptively. We do so by first critiquing the initial approaches to building a theory of adaptivity, then offering an alternative approach and preliminary results within this approach, all geared towards characterizing the inherent challenge of adaptivity.

In the second half, we address the validation framework itself. Most common practice does not just use a single holdout set, but averages results from several, a family of techniques known as cross-validation. In this work, we offer several new cross-validation techniques with the common theme of utilizing training sets of varying sizes. This culminates in hierarchical cross-validation, a meta-technique for using cross-validation to choose the best cross-validation method.

Thesis Supervisor: Jonathan Kelner
Title: Professor

Thesis Supervisor: Tamara Broderick
Title: Assistant Professor
Acknowledgments

I would like to thank my co-advisors, Jon Kelner and Tamara Broderick, for numerous helpful discussions about this work. I would also like to thank Jerry Li, Adam Sealfon, Thomas Steinke, Will Stevenson, the MIT machine learning theory group, and the Broderick group for numerous helpful conversations for the entire duration of my PhD. Simulations were done on the the MIT Lincoln Labs Supercloud, and I would like to thank Lauren Milechin for getting me started on that system. I also received financial support from the United States Department of Defense (DoD), through the National Defense Science and Engineering Graduate Fellowship (NDSEG), and from the National Science Foundation.
Contents

1 Introduction 9

2 Adaptive Data Analysis 11

2.1 Introduction ........................................... 11
     2.1.1 The Original Adaptive Problem .................... 14
     2.1.2 Static Data Analysis .......................... 16
     2.1.3 Previous Adaptive Lower Bounds ................. 17
     2.1.4 Summary of Results ............................ 19

2.2 Bayesian Adaptive Data Analysis .......................... 20

2.3 New Adaptive Challenges .................................. 22
     2.3.1 First Challenge: Posterior Uncertainty .......... 22
     2.3.2 Obfuscation Techniques ......................... 29
     2.3.3 Second Challenge: Slightly Correlated Queries ... 31
     2.3.4 Discussion .................................. 36

2.4 New Adaptive Guarantees .................................. 42
     2.4.1 Subgaussian Concentration and Accuracy ......... 42
     2.4.2 Beta Distribution Subgaussianity ................. 45
     2.4.3 Dirichlet Prior ................................ 49
     2.4.4 Discussion .................................. 50

2.5 Conclusions ........................................... 52

3 Cross-Validation 55

3.1 Introduction ........................................... 55
Chapter 1

Introduction

With the widespread use of machine learning algorithms comes the need to properly evaluate them. Validation refers to this process of estimating algorithm performance. Validation addresses a key question we should ask of any algorithm: How do we know that the answers it gives are accurate? It also forms a building block of a wide variety of subroutines, as validation estimates can be used to choose between different algorithms or hyperparameter settings within those algorithms.

Arlot and Celisse [1], in a survey on cross-validation, distinguish between the validation goals of estimation and identification. That is, validation can be used to estimate the performance of algorithms, and also identify which of a number of algorithms is best. While both use cases are common, in this thesis, we will focus almost exclusively on the former.

For an algorithm trained on a data set, there are a variety of options for how to estimate its performance. One can look at its performance on the same data it was trained on, an estimate known as the resubstitution estimate. However, this will generally be overoptimistic, since the algorithm will likely fit that data better than fresh data. This leads to a number of correction or regularization approaches, which seek to adjust this estimate upwards depending on some algorithm-specific features like the dimensionality of a model.

However, as algorithms grow more and more complex and opaque, appropriate adjustment factors are harder to come by. Therefore, it is attractive to have validation
methods that treat the algorithm as a black box. Fortunately, we can correct for the bias of the resubstitution estimate by simply evaluating the algorithm on a fresh data set it has not seen, known as a **holdout set**.

This solution presents new challenges. First, is that estimate based on the holdout set robust to any changes we might make to our algorithm? How many times can we access it before we have to worry about the same overoptimism we run into with the resubstitution estimate? This line of questions forms the basis for the new theoretical field of *adaptive data analysis*. Sections 2.1 through 2.4 describe my contributions to this field: We first critique the original problem formulation, showing that the worst-case assumptions lead to unrealistic information asymmetries, then offer an alternative based on a Bayesian prior and show the first positive and negative results for the new problem.

Second, how do we pick this holdout set? In practice, researchers often obtain multiple holdout estimates with different disjoint training and holdout data sets and average the results together, a practice known as *cross-validation*. Sections 3.1 through 3.5 describe my contributions to this field. All cross-validation methods are biased, and we show how that bias can be eliminated by extrapolation. At a higher level, there are many different options for cross-validation, and we offer a data-driven approach to choosing the best technique which we call hierarchical cross-validation.

While these two halves of this thesis are fairly disjoint, they complement each other well, each addressing important concerns related to reliable validation of algorithms from different perspectives.
Chapter 2

Adaptive Data Analysis

2.1 Introduction

The growing field of adaptive data analysis seeks to understand the problems that arise from the way that many analysts study data: by an adaptive process of iterative measurements. Researchers often make those measurements on the same data set, inadvertently breaking an important assumption of classical statistical guarantees: the data and the reported measurement process are no longer independent. Rather than requiring this independence by fiat (e.g. mandating pre-registration of experiments or only using a holdout database once an "exploratory" phase of the analysis is complete), the aim of the nascent field of adaptive data analysis is to understand how accuracy decays under adaptive measurement, and build algorithms for extending that accuracy.

There is hope that such an understanding would solve one component of the current replicability crisis in experimental science. In practice, researchers often pick their experimental or data analysis techniques after observing the data, meaning any such results may no longer hold in a replication study with fresh, independent data. New statistical techniques will not be able to prevent dishonest researchers from cutting corners, but they could at least help honest researchers find true effects more reliably than current practice and more cheaply than replication studies.

Dwork, Feldman, Hardt, Pitassi, Reingold, and Roth (hereafter DFHPRR) formu-
lated this problem in 2014 in a seminal paper with different components published in *NIPS* [9], *Science* [10], and *STOC* [11]. After introducing the problem, they proposed several approaches based on algorithms developed for the purpose of achieving differential privacy, a strong stability guarantee. Subsequent analysis by Bassily, Nissim, Smith, Steinke, Stemmer, and Ullman (hereafter BNSSSU) [2] improved the achieved bounds by first streamlining and generalizing this approach.

Simultaneous work on lower bounds offers to explain the difficulties inherent in adaptive data analysis. Hardt and Ullman [15] and Steinke and Ullman [27] used a construction from privacy known as interactive fingerprinting codes to argue a nearly matching lower bound to the stronger guarantees of BNSSSU [2]. The results are quite strong, but as we will argue in Section 2.1.3, they take advantage of an unnatural information asymmetry in the original problem that does not arise in typical applications. This information asymmetry is actually a fundamental component of not only the lower bounds featured in [15] and [27], but also every example cited in the literature of a serious problem resulting from adaptivity.

To better understand other challenges to adaptivity that do not require this unnatural information asymmetry, we translate the original problem to a Bayesian context with a public prior. This naturally obviates the previous lower bound techniques, and also allows us to include other information about the population that might be known from the experimental design or prior publicly released data.

All of the previous techniques defined for the original problem have natural analogues in the Bayesian context, with the role of the empirical mean now played by the posterior mean. Having obviated the previous lower bound techniques, the natural first question is what other obstacles to efficient adaptive data analysis exist, and whether the analogues of state-of-the-art techniques are successful in eliminating those difficulties.

In Theorem 5, we answer both of these questions by introducing a new problem on which all analogues of previously proposed techniques fail to achieve the optimal sample complexity. Rather than taking advantage of information asymmetry, this difficulty relies upon two important components: A difficult learning problem based
on error-correcting codes with unusually high uncertainty in one direction, and a
technique for using nearly orthogonal measurements to extract information about
tiny effects from a general family of obfuscation techniques, including those using
noise, rounding and proxy mechanisms.

This result illustrates a new type of challenge faced by the usual approaches to
adaptive data analysis. At the same time, this difficulty is not quite as constraining as
those that exploit information asymmetry, which suggests that adaptive data analysis
as commonly practiced might actually be easier than those previous lower bounds
suggested.

In support of this possibility, we also prove new positive results in this Bayesian
context in the specific case that the public prior is a Dirichlet prior. Unlike previous
positive results, we show that in this scenario, no obfuscation is even necessary: the
posterior mean can accurately answer as many adaptive queries as static queries.

To prove this, we develop a new connection between guarantees in Bayesian adap-
tive data analysis and subgaussian concentration of posteriors. In our main technical
result, Theorem 10, we prove that the Beta($\alpha, \beta$) distribution is $O(1/(\alpha + \beta + 1))$-
subgaussian, a new concentration bound with implications outside of adaptive data
analysis and which has already been improved by Marchal and Arbel [20]. We also
examine other conjugate priors beyond the Dirichlet prior and conjecture based on
empirical data that they also share this property.

While we study the full Bayesian context with a completely specified and accurate
prior for clarity, we also think of this formulation as a heuristic analysis tool for
understanding the original frequentist problem in situations where there is no risk
of information asymmetry. Therefore, the message of this work is not to encourage
researchers to attempt to write down accurate priors and translate every problem into
a Bayesian context, but instead to use analogous techniques such as regularization to
appropriately take into account prior beliefs.
2.1.1 The Original Adaptive Problem

As formulated by DFHPRR [11],1 adaptive data analysis is a game between two players, the curator and the analyst. The analyst is given a distribution \( \bar{p} \) on a universe \( \mathcal{X} \), while the curator only receives \( n \) samples from \( \bar{p} \).

The analyst then asks him2 \( q \) statistical queries: For some function \( f : \mathcal{X} \to [0, 1] \), what is \( \mathbb{E}_{\bar{p}}(f) \)? In the case that \( f \) only takes on values 0 or 1, these are known as counting queries and correspond to asking the probability of some event \( f^{-1}(1) \subset \mathcal{X} \). Either way, all true answers are within \([0, 1]\), which provides an appropriate normalization within which to discuss error.

His goal is to answer every query to within an additive error of \( \epsilon \) on the true population, or \( \epsilon \)-accurately, with probability at least \( 1 - \delta \) (over both the sample randomness and any randomness that he introduces). The problem is summarized in the following table. The central question is: How many samples \( n \) does he need to do this, as a function of the parameters \( q, \epsilon, \) and \( \delta \)? Equivalently, given \( n \) samples, how many queries \( q \) can he answer successfully, as a function of \( n, \epsilon, \) and \( \delta \)?

---

1Notation has been slightly changed, because their results use variables \( \epsilon \) and \( \delta \) as they are used in differential privacy, while we use those variables in the more standard randomized algorithm notation (e.g. as used in PAC learning).

2For clarity, throughout this paper, we will refer to the curator using male pronouns and the analyst using female pronouns.
### Adaptive Data Analysis: Original Frequentist Version

Two players: Curator and Analyst

Both receive problem parameters: $\epsilon, \delta \in (0, 1)$, $n \in \mathbb{N}$ and universe $\mathcal{X}$.

Analyst receives distribution $\tilde{p}$ over $\mathcal{X}$, unknown to the curator.

Curator receives $n$ independent samples from $\tilde{p}$.

- Analyst asks query $f : \mathcal{X} \to [0, 1]$.
- Curator replies with answer $a \in \mathbb{R}$.
- Answer is $\epsilon$-accurate if $|\mathbb{E}_{x \sim \tilde{p}} f(x) - a| < \epsilon$.

Repeat the indented interaction for $q$ total queries.

Curator wins if all answers are $\epsilon$-accurate with probability $\geq 1 - \delta$.

How does this game correspond to data analysis in the real world? The usual story goes like this: The analyst represents a powerful machine learning algorithm, aiming to find a very good fit to the true distribution. She will generally try to come to the strongest conclusion possible, which will probably involve difficult queries which are likely to reveal the most information, or those on which the curator is likely to be far off. To prevent this overfitting in the worst case, we model the analyst adversarially, which means that she could even know the distribution $\tilde{p}$. The curator algorithm provides a layer of protection to the data set and answers these queries in order to make sure that all of the information the analyst gets is $\epsilon$-accurate.

This might not perfectly describe how data analysis is done in every application, but there is a domain where it is a decent fit: machine learning competitions. In a typical competition, the administrators often randomly split a data set into three similarly-sized components: a training set given to competitors, a test set reserved for official scoring at the end, and a holdout set to allow the competitors to try out their learning algorithms throughout the competition. In some circumstances, such as classification learning, the score of a submitted algorithm is actually exactly equal to a counting query corresponding to the region of the universe that the algorithm correctly classifies.
In this context, the official scoring using the unseen test set is an instance of static data analysis, the traditional domain where all of the queries (submissions) are specified before results are announced. However, the submissions measured throughout the competition repeatedly using the holdout set are necessarily and intentionally adaptive; competitors are supposed to use their scores to improve their algorithms. Adaptivity can even take place between competitors as they collaborate to produce the best blended approach. Therefore, the traditional guarantees fly out the window and the public (unofficial) leaderboard throughout the competition is often inaccurate (see, e.g. [33], originally cited in [11]).

Of course, as stated, it probably seems quite strange that the analyst also receives the true distribution $p$ if she is supposed to be seeking to learn it. If competitors already know the entire distribution, there is no need to hold the competition! We’ll save this important criticism for later, though, first focusing on the static case, which will serve for us as a benchmark for the best adaptive algorithms could achieve.

2.1.2 Static Data Analysis

If the analyst chooses all query functions $f_1, f_2, \ldots, f_q$ before hearing the curator’s answers to any of them, her strategy is said to be static. Under static data analysis, the queries and data are independent, and a very simple curator strategy achieves remarkable accuracy: the empirical mean. This curator strategy simply answers query function $f$ with $\frac{1}{n} \sum_{i=1}^{n} f(x_i)$, where the $\{x_i\}_{i=1}^{n}$ are the data points.

By a standard Hoeffding and union bound argument, the empirical mean is $\epsilon$-accurate against $q$ static queries with probability $1 - \delta$ if

$$n \geq n_{\epsilon}(q, \epsilon, \delta) := O \left( \frac{1}{\epsilon^2 \log \frac{q}{\delta}} \right).$$ (2.1)

This bound is tight for static data analysis, with a fairly simple matching example: $q$ independent $\epsilon$-biased coins. In short, the queries could require the analyst to estimate whether some biased coins are $\epsilon$ more likely to land on heads or tails, and each such estimation requires $\Omega \left( \frac{1}{\epsilon^2 \log \frac{q}{\delta}} \right)$ samples to keep the total error probability
less than $\delta$.

This isn’t an unusual challenge, either; estimating any $q$ independent probabilities (not very close to 0 or 1) to within an $\epsilon$ additive error will require a sample complexity within a constant factor of $n_x$. In the final private leaderboard of a machine learning competition, therefore, the administrators can accurately score an exponential number of submissions in the number of data points, an excellent dependence.

But when we move to the adaptive setting, such as when updating the public leaderboard of a competition, these upper bound guarantees no longer hold. Of course, adaptivity always gives the analyst more options, so at least $n_x$ samples are necessary. But is that number still sufficient? Can the best curator algorithms still answer exponentially many adaptive queries?

### 2.1.3 Previous Adaptive Lower Bounds

Initially, the answer appeared to be a strong no. The existing literature has produced an array of very strong lower bounds, arguing for a wide gap between the adaptive and static problems. However, we will argue these all rely on that strange consequence of the worst-case nature of this problem, that the analyst receives the true distribution, opening the possibility of a narrower gap without that extra knowledge.

First, let us describe the lower bounds. The simplest is found in an appendix of DFHPRR [11], viewable in the arXiv preprint. The authors describe a model over $\mathbb{R}^d$ where the empirical mean algorithm fails, motivating their proposed variations on it. The distribution is $N(0, I_d)$, a zero mean $d$-dimensional Gaussian, and the analyst first queries the dot product with each of the standard basis vectors (rescaled and truncated to fall within $[0, 1]$). After discovering where the sample is biased, she then queries along another diagonal direction, chosen to compile the errors of the first $d$ queries to produce an error that is typically $\sqrt{d}$ times larger. Since the empirical mean error is proportional to $\frac{1}{\sqrt{n}}$ and this requires $q = d + 1$ queries, this means that $n = \Omega(q)$ is necessary for the empirical mean to be constant-accurate. Linear query dependence is awful: It can also be achieved trivially by looking at a fresh batch of data for every query, showing that the empirical mean massively underperforms once
adaptivity is allowed.

Similarly, Blum and Hardt [4] describe another adaptive attack on the empirical mean in a slightly different but a bit more general setting. Instead of querying a Gaussian along coordinate axes, they produce a series of random queries before again aggregating the results to produce a query on which the empirical mean will be $\sqrt{q/n}$-inaccurate. This again shows that the empirical mean strategy is deficient; it can only answer a linear number of queries.

Moving beyond attacks specific to the empirical mean, the most frequently cited general lower bound constructions due to Hardt and Ullman [15] and Steinke and Ullman [27] build on a long literature of privacy-preserving algorithms, and in particular on an attack known as interactive fingerprinting codes. At a high level, the analyst asks queries that can only be answered successfully if the curator has seen particular data points, and in so doing is able to reconstruct the data that the curator has seen and query the remainder that he hasn't seen. This is harder than it sounds at first, because the analyst must use queries that force the curator to reveal knowledge of a particular point or answer $\epsilon$-inaccurately, not just $1/n$-inaccurately as the empirical mean frequently does. This attack either requires a common cryptographic assumption (one-way functions) and only applies to computationally bounded adversaries, or high ($d \sim n^2$) dimension, but in either case, it can be achieved with only $O(n^2)$ queries, which is remarkable given its generality.

These results imply the improved estimates of BNSSSU [2] are nearly tight for this problem. This would seem to be the end of the story: In adaptive data analysis, the curator can answer only quadratically many queries, far fewer than the exponential number of queries that can be answered in static data analysis, at least in high dimension.

However, these strong lower bound examples rely on a key information asymmetry between the analyst and curator that we will now argue is unrealistic. In both the Gaussian and fingerprinting cases, the analyst in fact knows the exact true distribution, and that the curator is left guessing from the data. This feature is critical to those constructions: In the Gaussian example, the analyst must know the true answer
(or in other words, where the origin is) to be able to determine in which direction
the analyst's answers are wrong. In the interactive fingerprinting attack, the ana-
lyst must know the possible samples that the curator could see so she can construct
queries designed to test for them. Even the boosting attack takes advantage of this
information asymmetry by limiting the curator to the empirical mean; see Section
2.3.1 and Appendix B.1 for the full details.

In the picture of how this would be used in machine learning, though, this asym-
metry is unrealistic. The analyst is trying to learn something about the distribution
from the data, but there's no learning to be done when she already knows it. In a
competition, if some competitor already knows the full distribution of the data, there
is no need to keep any data in a holdout set or for that matter, even to have them
enter the competition in the first place.

Finally, Nissim and Stemmer [26] have recently attempted to make the interactive
fingerprinting attack information-symmetric by encrypting it with public key encryp-
tion. From one perspective, this gives a computational lower bound, arguing that com-
putationally bounded curators can not answer more than $O(n^2)$ queries even in the
information-symmetric setting. At the same time, the problem description in terms of
public keys and private keys is exponentially long, and the key step is that the compu-
tationally bounded curator does not have enough time to read this entire description
and do proper inference to the private keys. This result sheds some light on the
problem mathematically, but ultimately just hides the same information-asymmetric
attack behind a computational barrier. We are interested in new categories of diffi-
culties to adaptivity that don't simply hinge on being unable to do inference, so we
will generally consider both players to be computationally unbounded.

2.1.4 Summary of Results

The rest of this paper consists of three major contributions: We define the problem of
Bayesian adaptive data analysis, demonstrate a new type of difficulty, and show that
in some natural Bayesian scenarios, adaptivity poses no additional overfitting threat.

In more detail, we begin in Section 2.2 by giving a new Bayesian formulation of the
problem that incorporates information symmetry via a public prior. We also justify and clarify the importance of studying Bayesian adaptive data analysis primarily as a means to understanding adaptive attacks that do not simply take advantage of information asymmetry.

In Section 2.3, we introduce a new type of adaptive attack with two components. First, in Section 2.3.1, we describe a series of problems based on error-correcting codes on which posterior inference leaves the curator uncertain about particular queries. This possibility motivates the need for obfuscation techniques for answering queries, and in Section 2.3.2, we describe the previously proposed techniques, including a new improvement of a classic suggestion, query-dependent rounding. However, in Section 2.3.3, we describe an augmented problem on which an adaptive analyst can extract information even from the best of the obfuscation techniques using only $\tilde{O}(n^4)$ queries.

To complement the picture, we also provide and demonstrate tools for proving positive results in Section 2.4. The first tool, in Section 2.4.1, is a simple condition in terms of the probabilistic notion of subgaussianity, for the posterior mean itself to be accurate without any obfuscation. To use this tool, we first show in Section 2.4.2 that the $\text{Beta}(\alpha, \beta)$ distribution is $1/(4(\alpha + \beta) + 2)$-subgaussian, a new probabilistic result of independent interest, before using this in Section 2.4.3 to prove that under the Dirichlet prior, the posterior mean is always an accurate estimate.

After understanding the definition in Section 2, the negative and positive results, in Sections 3 and 4 respectively, can be read independently.

### 2.2 Bayesian Adaptive Data Analysis

In this section, we introduce the new problem of Bayesian adaptive data analysis. Just two lines are changed (those referring to the prior$^3 \mathcal{P}$):

---

$^3$To be clear, this is a prior directly over distributions, without the common intermediate layer of a parameter.
In this way, we prevent the analyst from employing strategies that rely on side knowledge of the distribution, by giving all such knowledge to the curator as well. This is the main point; we are not claiming that an accurate public prior can be written down for every problem in practice. Instead, by examining the situations where it can, we hope to explore the possible difficulties that do not arise simply from exploiting side information. The hope is that observations in this domain will naturally translate to heuristics in real-life scenarios where information symmetry in an informal sense is appropriate to assume.\(^4\)

Future work could consider the case where the three priors in the problem (those that the curator and analyst receive and the true prior the distribution is drawn from) are close in some suitable sense, in order to measure the difficulties with a small amount of side information, but we focus on the case where all three are equal for simplicity.

The prior \(\mathcal{P}\) may be discretely or continuously supported (within the simplex of probability distributions \(\Delta(\mathcal{X})\)), and the examples we construct will feature both.

\(^4\)All of that said, this problem as stated is also of interest. It is analogous to the area of Bayesian analysis focused on finding frequentist properties (like posterior convergence) of Bayesian models (see, e.g. the survey paper [24] or Chapter 10 of [29]).
We call a particular universe $\mathcal{X}$ and prior $\mathcal{P}$ a model $\mathcal{M} = (\mathcal{X}, \mathcal{P})$.

As designed, this assumption of information symmetry obviates all of the previous potential lower bound models. If the analyst knows the distribution exactly, so does the curator, and then the curator can simply give the exact answer without referencing any samples. And if the analyst is in the dark, she can’t determine how the curator’s answers are inaccurate to aggregate the errors. To illustrate this, in Appendix A, we thoroughly analyze how the Gaussian example from DFHPRR becomes completely unconcerning with a reasonable prior.

We are left with only the basic lower bound examples we considered in the static case, which do translate nicely into this setting, since that analyst’s queries don’t require knowledge of the distribution. For the $\epsilon$-biased coin, we can consider the prior to be uniform on the two cases, and similarly for $q$ copies of it. In these cases, it is still necessary to see enough data to reliably distinguish probabilities of $\frac{1}{2} \pm \epsilon$ from each other, so Equation (2.1) still holds.

Is that really enough, though? This is the key question: Under information symmetry, can the static bound (2.1) be achieved for adaptive queries? If not, what new attacks can the analyst employ, and what bounds do those place on query complexity?

### 2.3 New Adaptive Challenges

As a starting point for the adaptive problem, suppose that the curator simply answers with the posterior mean. That is, the curator uses the data to update the prior by Bayes’ rule to a posterior and answers the query with the average answer according to those updated weights. What could go wrong?

#### 2.3.1 First Challenge: Posterior Uncertainty

By reading his posterior, the curator actually knows the probability that any answer he gives will be $\epsilon$-accurate. Therefore, the first question we can ask is whether the curator knows of any queries that he cannot answer so that the probability that his answer is $\epsilon$-accurate is less than $1 - \delta$. 
We will eventually answer this question in the affirmative, but it is worth dwelling on the difficulty of finding such an example. By the static data analysis result, any query specified in advance will have the desired posterior concentration around the empirical mean with all but exponentially small probability. Therefore, if we want the curator to have some query he will answer inaccurately with constant probability, there need to be exponentially many such potential queries.

The example we will construct is motivated by the famously difficult robust learning problem of learning parities, but we will eventually need to generalize it, so we introduce the general family of classification models.

**Classification Models in General**

In a classification model, the universe is a product $\mathcal{X} = \mathcal{Y} \times \mathcal{Z}$, where we think of $\mathcal{Y}$ as an underlying known population, and $\mathcal{Z}$ as a set of labels generated by some unknown function $\ell: \mathcal{Y} \to \mathcal{Z}$ that we are trying to learn.

That is, each hypothesis $\tilde{p}_j$ in the support of the prior corresponds to some possible function $\ell_j: \mathcal{Y} \to \mathcal{Z}$. All hypotheses have the same marginal $\tilde{r}$ on $\mathcal{Y}$, and after drawing a sample $y \sim \tilde{r}$, output the point $(y, \ell_j(y))$. In other words, the hypothesis $\tilde{p}_j$ has the graph of function $\ell_j: \mathcal{Y} \to \mathcal{Z}$ as its support, with weights determined by $\tilde{r}$.

In the examples we consider in this paper, $\tilde{r}$ will be uniform on a finite $\mathcal{Y}$ and $\mathcal{Z} = \mathbb{F}_2$. In equations, hypothesis

$$\tilde{p}_j(y, z) = \begin{cases} 1/|\mathcal{Y}| & z = \ell_j(y) \\ 0 & \text{otherwise.} \end{cases}$$

One important feature of classification models to us is that they have very easy-to-understand posteriors. Each potential sample point $(y, z)$ occurs with probability either $r_y$ or 0 under each hypothesis. Therefore, if the prior is uniform over a set of

---

5 This example was actually the main exception to problems that can be studied with statistical queries; see [17]. However, we will take this example further with a generalization based on error-correcting codes.

6 To write succinct equations that relate $\mathbb{F}_2$, the field on two elements, to $[0, 1]$ we will frequently abuse notation and conflate the two elements of $\mathbb{F}_2$ with the real numbers 0 and 1.
functions, the posterior remains uniform over all such functions which are consistent with all of the observed samples, and puts a zero weight on any that are inconsistent with even one. Call the hypotheses that agree with all of the samples eligible.

As an aside, we now have the notation to discuss the appropriate analog to the boosting attack of Blum and Hardt [4] in this setting, and demonstrate how it also uses information asymmetry, albeit more subtly than the other algorithms. However, this is tangential to our main purpose, so we defer this discussion to Appendix B.1.

**Linear Classification**

For our first example, also known as the problem of learning parities, we consider the uniform prior over linear classification functions $\ell_j$ on $\mathbb{F}_2^m$, which we call $LC_m$. In other words, uniformly random coefficients $a, b_1, b_2, \ldots, b_m \in \mathbb{F}_2$ are chosen, and the distribution is uniform on the graph of the $m$-dimensional $\mathbb{F}_2$ function $\ell(y_1, \ldots, y_m) = a + \sum_k b_k y_k. \footnote{This model is a slight variant on the classical problem of learning parities. Technically, we are looking at degree at most 1 polynomials, only half of which are linear in the linear algebraic sense (those with $a = 0$). We do this to introduce a symmetry: Every point in $Y \times Z$ is equally likely, rather than making $\binom{6}{1}$ impossible.}$

This model also has an easy-to-understand posterior:

- From a series of samples $\{(y_i, z_i)\}_{i=1}^n$, we can construct the function restricted to the affine span of $\{y_i\}$. At any point outside of the affine span, the function is equally likely to be 0 or 1.

- Considering the samples in order, call $(y_i, z_i)$ novel if $y_i$ is not in the affine span of $y_1, \ldots, y_{i-1}$. Each novel sample cuts the number of eligible hypotheses in half, by symmetry.

- If $i$ points are affinely independent, their affine span has size $2^{i-1}$. Therefore, each novel sample also doubles the size of the affine span of the samples.

- The probability that the first $m$ samples are all novel is hence

$$1 - \frac{1}{2^m} - \frac{2}{2^m} - \frac{2^2}{2^m} - \cdots - \frac{2^{m-2}}{2^m} = \frac{1}{2} + 2^{-m}.$$
With these observations, we can prove that the analyst has a winning strategy if
the curator gets a precise number of samples:

**Theorem 1.** Under model $LC_n$, there is an adaptive analyst strategy which causes
the posterior mean curator strategy to answer $\frac{1}{4}$-inaccurately with probability $> \frac{1}{2}$,
using only $n + 2$ queries.

*Proof.* To be clear, we take $n = m$, i.e. the number of samples is equal to the dimen-
sion. As we've computed, this means that with probability $\frac{1}{2} + 2^{-m} > \frac{1}{2}$, all of the
samples are novel. After $m$ novel samples, the number of eligible hypotheses is down
to two, which agree on half of $\mathbb{F}_2^m$ and disagree on the other half. The posterior puts
weights of $1/2$ on each of these, so the posterior mean puts a weight of $2^{-m}$ on the
$2^{m-1}$ known points and $2^{-m-1}$ on each of the $2^m$ unknown possible points.

Therefore, by querying the indicator function on individual points $\{(y, z)\}$, the
analyst has access to an oracle for whether the curator knows the value of the function
there. By sampling an affine basis for the entire space (such as $0, e_1, \ldots, e_m$), she can
compute that affine span with only $m + 1 = n + 1$ queries, and determine which two
hypotheses are still eligible.

Then she can exploit the remaining ignorance of the curator by querying the entire
graph of one of those hypotheses. The correct answer will be $\frac{1}{2}$ or 1, equally likely,
so the posterior mean answer of $\frac{3}{4}$ is $\frac{1}{4}$-inaccurate, as desired. $\square$

Note that this argument shows that this particular curator cannot even accurately
answer more than linearly many queries, which we've already mentioned is trivial. In
fact, the analyst technically doesn't even need all $n + 1$ exploratory queries:

**Theorem 2.** Under model $LC_n$, there is an adaptive analyst strategy which causes the
posterior mean curator strategy to answer $\frac{1}{4}$-inaccurately with probability $> \frac{1}{2}$ using
only two queries.

*Proof.* Consider some enumeration $y_1, y_2, \ldots, y_{2^m}$ of $\mathbb{F}_2^m$ and the query function de-
finite by $f(y_i, 1) = 2 \cdot 3^{-i}$ and $f(y_i, 0) = 0$. The posterior mean on this query is
therefore $2^{-m} \sum_{i=1}^{2^m} g(y_i)3^{-i}$, where $g(y) = 0$ if the curator knows that $f(y) = 0,$
\( g(y) = 1 \) if the curator doesn’t know \( f(y) \), and \( g(y) = 2 \) if the curator knows that \( f(y) = 1 \). Therefore, by reading off the digits of the ternary expansion of \( 2^m \) times the posterior mean’s answer to this query, the analyst can determine everything the curator knows. She can then query one of the two eligible hypotheses, and the curator will be 1/4-inaccurate as before. \( \square \)

This isn’t a matter of posterior inference being the wrong thing to do here; it’s easy to see that a variant of the attack in Theorem 2 would be able to find the data points if the curator decided to ignore the prior and use the empirical mean on this problem instead. The important feature is that there is a query that the curator does not have the information to answer reliably accurately, whether he uses all of the information he does have or not.

For intuition’s sake, it might help to look at the way the uncertainty of the curator evolves as he gets more data points. Initially, he is slightly uncertain in nearly every direction, but as he eliminates hypotheses, that uncertainty is reduced in most directions while increasing in a smaller number. When he reaches the last two hypotheses, all of the remaining uncertainty is concentrated along the direction of their disagreement. This particular direction is one out of exponentially many equally likely possibilities originally, so the analyst can’t simply guess it. But if she can learn what the curator knows, she can determine it and query it.

**Polynomial Classification**

The example of linear classification naturally generalizes. For the prior of polynomial classification \( PC_{m,k} \), uniformly randomly choose \( M = 1 + m + \binom{m}{2} + \cdots + \binom{m}{k} \) coefficients \( c_i, S \in \mathbb{F}_2 \) for \( i = 0, \ldots, k \) and \( S \subset [m] \) of size \( k \), and let the distribution be uniform on the graph of the polynomial

\[
\ell(x_1, \ldots, x_m) = \sum_{i=0}^{k} \sum_{S \in \binom{[m]}{i}} c_{i,S} \prod_{j \in S} x_j.
\]
Note that these are all possible polynomials of degree at most $k$, since $x_i^2 = x_i$ over $\mathbb{F}_2$. We will typically think of $k < m$ or constant so $M = \Theta(m^k)$, but this definition is valid for any $1 \leq k \leq m$.$^8$

This model has some of the same properties as linear classification.

- By counting coefficients, there are initially $2^M$ eligible hypotheses.

- Each sample introduces a linear constraint on the coefficients. If this constraint is not already known, it cuts the number of available hypotheses in half. As before, we call such samples novel.

- By the theory of Reed-Muller codes, any two degree $\leq k$ polynomials differ on at least a $\frac{1}{2^k}$ fraction of $\mathbb{F}_2^m$. Therefore, each data point is novel with probability at least $\frac{1}{2^k}$, until there is only one eligible hypothesis left.

By Markov’s inequality, this implies that with probability at least $1/2$, less than $2^{k+1}M$ data points are necessary to eliminate all but one eligible hypothesis. Right before the last novel sample, there must be exactly two eligible hypotheses remaining. Therefore, there exists some $M \leq n < 2^{k+1}M$ such that with probability $\frac{1}{2^{k+2}M} \geq \frac{1}{2^{k+2n}}$, there are exactly two eligible hypotheses remaining after the curator receives $n$ data points.

In this case, despite the $n = \Omega(m^k)$ data points, the curator still can’t distinguish two hypotheses that disagree on at least $1/2^k$ of the space. Therefore, if $\epsilon < \frac{1}{2^k+1}$, the curator will know a query which he cannot answer $\epsilon$-accurately with a probability greater than $1/2$.

**Error-Correcting Code Classification**

This construction actually works for any binary linear error-correcting code, not just Reed-Muller codes:

**Definition.** Let $\mathcal{C} \subset \mathbb{F}_2^m$ be a linear error-correcting code of length $m$ over $\mathbb{F}_2$. Define the model $\mathcal{M}_\mathcal{C}$ over universe $[m] \times \mathbb{F}_2$ to have the following prior: Each codeword

---

$^8$For $k = m$, this is actually independent classification again.
$C \in \mathcal{C}$ corresponds to a hypothesis $h_C$ which is a distribution with weight $\frac{1}{m}$ on $(i, C_i)$ for each $i \in [m]$, and all such hypotheses are equally likely.

In the theory of error-correcting codes, $C \subset \mathbb{F}_2^m$ is said to have dimension $k$ if $|C| = 2^k$ and distance $d$ if for any two distinct codewords $C, C' \in C$, $C_i \neq C'_i$ for $d$ values of $i \in [m]$. Finally, recall that $C$ is linear if $C$ is a linear subspace of the vector space $\mathbb{F}_2^m$.

**Lemma 3.** Suppose $C \subset \mathbb{F}_2^n$ is a linear code with length $m$, dimension $k$ and distance $d$. Then there exists some $k - 1 \leq n \leq \frac{2mk}{d}$ such that if a curator receives $n$ samples from $\mathcal{M}_C$, with probability at least $\frac{d}{4mk}$, the curator’s posterior will place equal weight on exactly 2 hypotheses who differ on a subset of weight at least $\frac{d}{m}$.

**Proof.** The curator begins with $2^k$ eligible hypotheses, and since $C$ is linear, each new observation $(i, C_i)$ is consistent with either all or half of the eligible hypotheses. Since the false hypotheses all have distance at least $d$ from the true hypothesis, there is a probability of at least $\frac{d}{m}$ that each sample is not consistent with all hypotheses, until there is only one eligible hypothesis left. If the curator receives and updates on samples one at a time, there will be some number of samples $n$ after which exactly 2 eligible hypotheses remain. By Markov’s inequality, with probability at least $\frac{1}{2}$, this occurs before $n \leq \frac{2mk}{d}$. Therefore, there is some $k - 1 \leq n \leq \frac{2mk}{d}$ such that this occurs at precisely $n$ data points with probability at least $\frac{d}{4mk}$. $\square$

Reed-Muller codes, which correspond in this fashion to polynomial classification, are linear binary codes with length $2^m$, rate $M = (\frac{m}{\leq k})$, and distance $2^{m-k}$. As we will see, though, the best attacks will take a number of queries polynomial in the length of the code, so we would like to reduce this. We will instead use the Justesen code (see [16]), a linear code over $\mathbb{F}_2$ with rate and distance proportional to $m$; we can take for instance $k = m/4$ and $d = m/10$ (for large enough $m$). Then by Lemma 3 with $C$ a Justesen code, there is some $\frac{m}{4} \leq n \leq 5m$ such that with probability $\geq \frac{1}{10m}$, after receiving $n$ samples from $\mathcal{M}_C$, the curator’s posterior has two eligible hypotheses left which differ on a subset with weight at least $1/10$. Call such a model $\mathcal{M}_C = J_m$ for simplicity.
2.3.2 Obfuscation Techniques

The standard response is that both the posterior mean and empirical mean are too precise: They give away too much information in unnecessary bits of precision. The literature has several methods for hiding what the curator knows while still maintaining query accuracy:

**Generic rounding.** This is fairly standard practice. For instance, submission scores on machine learning competitions run by Kaggle are frequently reported to only five or six decimal places. This would easily mitigate the queries used in the attack in Theorem 2, but will still be vulnerable to well-calibrated versions of the queries in the attack in Theorem 1.

In particular, in classification problems, the analyst can get information by straddling a rounding boundary $B$ with queries like this:

$$f(y, z) = \begin{cases} B & y \neq y' \\ z & y = y'. \end{cases} \quad (2.2)$$

If the curator knows that $\ell(y) = 0$, the posterior mean will lie in the interval below $B$, and otherwise, it will lie in the interval above $B$, so this query will tell the analyst whether the curator knows that $\ell(y) = 0$. Randomly shifting the boundary points would reduce the chance of this, but there will still be a nontrivial probability of information leakage.

**Independent noise.** To avoid this sort of certainty, the curator can instead add a small amount of noise independently to every answer, such as Laplacian or Gaussian noise. While Laplacian noise is calibrated to match the definitions of differential privacy, it is easier to verify that Gaussian noise with variance $O(\frac{1}{n})$ will adjust the answer by less than $\epsilon/2$ with probability $1 - \delta/2q$ if $n = n_s(q, \epsilon/2, \delta/2)$. Therefore, if the original answer was $\epsilon/2$-accurate with probability $1 - \delta/2q$, the noised answer will be $\epsilon$-accurate with probability $1 - \delta/q$. Since $n_s(q, \epsilon/2, \delta/2) = O(n_s(q, \epsilon, \delta))$, this only requires a constant factor increase in the sample complexity to maintain accuracy.

Adding noise effectively deals with the problem of linear classification (as we prove
in Appendix B.2), but it is not enough to achieve the static sample complexity on polynomial classification $PC_{m,k}$ for $k > 1$. The brief explanation is that each answer yields a small but nonzero amount of information (on the order of $2^{-2m}$ bits), so repeating the same query $2^{O(m)}$ times will allow the analyst to learn what the curator knows at every point. For linear classification, this takes exponentially many queries in $n$, but for polynomial classification, $n = \Omega(m^k)$, and therefore, this only requires $2^{O(n^{1/k})}$ queries. For more details, see Appendix B.3. In any case, we will achieve a much stronger result using Justesen code classification in the next section.

**Proxy distributions.** The famous private multiplicative weights algorithm of Hardt and Rothblum [14] maintains a proxy distribution that can be tracked by the analyst, and answers according to it if that answer is close enough to the empirical mean. If it isn’t, the method falls back to a noisy version of the empirical mean and updates the proxy distribution in a predictable way based on the query, essentially to avoid making the same mistake again. This method comes with very strong adaptive guarantees in low dimension, by a relative entropy argument showing that the proxy distribution cannot be updated too many times from its arbitrary initialization, but yet it maintains privacy.

We can naturally translate this algorithm into the Bayesian setting. The prior mean can serve as the initial proxy distribution, incorporating any public information about the problem, and we can replace uses of the empirical mean with the posterior mean. This won’t lead to any sharper guarantees, but this Bayesian version will behave similarly and allows us to consider these algorithms in corresponding settings.

Recall that the investigatory queries in Theorem 1 were meant to only yield one bit of information each. Adding noise increases the entropy but still yields a small amount of information on every query. The value of the private multiplicative weights approach in low dimension is that nearly every query will be answered by the proxy distribution, which tells the analyst almost nothing about the exact answer of the posterior mean. In this way, private multiplicative weights functions as a type of consistent noise which is immune to repeated queries. However, we will soon construct an example in high dimension where this is no longer the case, and where the slight
information leakage from every query guarantees the private multiplicative weights algorithm will also fail.

**Query-dependent rounding.** Another way to tell the analyst almost nothing is to round in a way that depends on the query, but not on the data. This is a new approach that to our knowledge does not appear previously in the literature. In the Bayesian context, we can project the prior onto the query to get a distribution on $[0, 1]$ and avoid placing any boundaries near parts of that distribution with high density (or large point masses). This is indeed possible with only a constant factor loss:

**Lemma 4.** Let $D$ be a distribution on $[0, 1]$. Then there exists a partition of the interval $0 = x_0 < x_1 < x_2 < \ldots < x_m = 1$ with $\epsilon/3 < x_{i+1} - x_i < \epsilon$ such that $\forall \eta > 0$,

$$\mathbb{P}_D \left[ \bigcup_{i=1}^{m-1} (x_i - \eta, x_i) \right], \mathbb{P}_D \left[ \bigcup_{i=1}^{m-1} (x_i, x_i + \eta) \right] < \frac{6\eta}{\epsilon}.$$  

Moreover, if $D$ is discretely supported with support size $s$, there is an algorithm\(^9\) that can compute the partition in time $O(s^3/\epsilon)$.

For the proof, see Appendix B.4. This would easily handle queries like (2.2) by moving the boundaries away from $B$. It also takes care of linear classification with even fewer than $n_s$ queries (also shown in Appendix B.4), although its performance against the polynomial and Justesen code classification problems is less clear.

In any case, we will now construct a model on which none of these obfuscation methods can block information leakage.

2.3.3 Second Challenge: Slightly Correlated Queries

To extract small amounts of information out of strong obfuscation techniques like private multiplicative weights or smart rounding, we will produce a model that allows for queries like (2.2), except with $B$ replaced by an almost uniformly distributed

\(^9\)While randomized rounding can achieve bounds like this with constant probability for particular values of $\eta$, this result requires this to hold for the same partition for all $\eta$. 

31
random variable in $[0,1]$, independent of a polynomial classification problem. Each query will use a new random variable, so previous updates to the proxy by private multiplicative weights will not change this. (Of course, this increases the dimension of the problem, so this doesn’t contradict the results of [14].)

Here are the components of this problem:

- Given models $M_1 = (\mathcal{X}_1, \mathcal{P}_1)$ and $M_2 = (\mathcal{X}_2, \mathcal{P}_2)$, we say that a sample from the tensor product $M_1 \otimes M_2$ is an ordered pair of independent samples from each of the two models. In other words, the combined prior $\mathcal{P}_1 \otimes \mathcal{P}_2$ consists of independently sampling distributions $\tilde{\mathcal{P}}_i$ from each individual prior $\mathcal{P}_i$ and taking the product distribution $\tilde{\mathcal{P}}_1 \times \tilde{\mathcal{P}}_2$ on universe $\mathcal{X}_1 \times \mathcal{X}_2$. We call the tensor product of $r$ copies of the same model the $r$th tensor power $M^{\otimes r}$.

- Tensor products allow the analyst to amplify error probabilities. In particular, if the analyst can induce a probability of error of $\delta$ by answering $q$ queries on model $M$, then on some model $M^{O(1/\delta)}$, by asking the same questions on each component, she can induce a constant error probability with only $q/\delta$ queries.

- For a given model $M = (\mathcal{X}, \mathcal{P})$, we say that a sample from the $r$th power of $M$, which we will write as $M^r$, is an $r$-tuple of samples from the same hypothesis $\tilde{\mathcal{P}} \sim \mathcal{P}$. Effectively, this means that the curator simply gets $rn$ samples rather than $n$.

- The uniform model $U_2$, which has universe $\{1, 2\}$ and the uniform prior for $\mathcal{P}[2]$ on $[0,1]$.10 If the data has $n_i$ copies of option $i$, the posterior mean of this distribution will be at $(\frac{n_1 + 1}{n_1 + n_2 + 2}, \frac{n_2 + 1}{n_1 + n_2 + 2})$.

In this notation, the problematic model is the following:

$$J_m \otimes (U_2^{S})^{\otimes(q-1)}.$$

10This is also an example of a beta distribution, specifically Beta(1,1), a family which we will study in much more detail in Section 2.4.2.
That is, we take a tensor product of the quadratic classification problem and $q - 1$ instances, each repeated 8 times, of a uniformly randomly biased coin.

Before examining each of these components, let’s describe what the analyst does. Her attack will again consist of a series of $q - 1$ exploratory queries followed by one final exploitative query. Each of the exploratory queries will use a fresh uniform random variable to probe the value of the function at a point $y_i$. The query function, motivated by (2.2), is

$$f_i : [m] \times \mathbb{F}_2 \times (\{1, 2\}^8)^{q-1} \to [0, 1]$$

$$(y, z, (x_{1,1}, \ldots, x_{1,8}), \ldots, (x_{q-1,1}, \ldots, x_{q-1,8})) \mapsto \begin{cases} \frac{m - 1}{4n + 1} & \text{if } y = y_i \\ x_{i,1} & \text{if } y \neq y_i. \end{cases}$$

Conditional on $y \neq y_i$, the expectation of this function is simply the probability of coin $i$ landing on 1.

The big idea is that these slightly correlated queries will primarily concern the uniform random variables, but the relevant information to the analyst is in the slight adjustment depending on whether $\ell(y_i) = 1$. The 8th power and that fraction in the query are much more technical components: They’re designed to directly tune an observation of a single additional 2 on a uniform random variable to the change in information about whether $f(y_i) = 1$ for the posterior mean. Note that we take $n \geq m/4$ so that the fraction is less than 1.

To demonstrate this tuning, suppose that out of the $8n$ samples of the $i$th coin, $s_i$ of them were 1’s. If the curator has enough information to deduce that $\ell(y_i) = 0$, the posterior mean is

$$0 \cdot \frac{1}{m} + \frac{s_i + 1}{8n + 2} \left(1 - \frac{1}{m}\right) = \frac{s_i + 1}{8n + 2} \cdot \frac{m - 1}{m} = a_{s_i},$$

for clarity. On the other hand, if the curator deduces that $\ell(y_i) = 1$, the posterior mean is

$$\frac{m - 1}{4n + 1} \cdot \frac{1}{m} + \frac{s_i + 1}{8n + 2} \left(1 - \frac{1}{m}\right) = \frac{s_i + 3}{8n + 2} \cdot \frac{m - 1}{m} = a_{s_i + 2}.$$
Finally, if the curator is uncertain on the value of $\ell(y_i)$, the posterior mean is the average of these values, or 
\[
\frac{s_i + 2}{8n + 2} \cdot \frac{m - 1}{m} = a_{s_i+1}.
\]

Recall that for the uniform prior, the $s_i$ are uniformly distributed over the integers between 0 and $8n$. Therefore, nearly every possible posterior mean value has the same probability under each of the three cases. The only exceptions are on the ends: $a_0$ is only possible if $\ell(y_i) = 0$, $a_1$ if $\ell(y_i) = 0$ or is unknown, $a_{8n+1}$ if $\ell(y_i) = 1$ or is unknown, and $a_{8n+2}$ if $\ell(y_i) = 1$. This shifting of some of the probability mass from one end of the interval $[0, 1]$ to the other will be something we can detect by simple counts no matter how hard the mechanism tries to obfuscate the posterior mean.

**Theorem 5.** Suppose that the curator always outputs some possibly randomized function $g(f(\mathcal{P}), \mathbb{E}[f(\mathcal{P}')] )$ of the prior and posterior mean on a query function $f$. Then under model $(J_m \otimes (U_2^q)^{\otimes (q-1)} \otimes m)$, for some $m = \Theta(n)$, there is an adaptive analyst strategy which causes the curator to answer constant-inaccurately with constant probability, using only $q = O(n^4 \log n)$ queries.

The functional notation here indicates that the output is additionally allowed to depend on the prior distribution $f(\mathcal{P})$ on the query in addition to the posterior mean $\mathbb{E}[f(\mathcal{P}')]$, like the smart rounding algorithm does. As is, this Theorem does not cover the private multiplicative weights algorithm, but as we will see shortly, it will also fail to solve this.

**Proof.** First, on a single component of this model, the curator has at least a $\frac{1}{10m}$ probability of having a posterior with only two eligible hypotheses. The outer tensor product exists in order to amplify this probability to a constant, with the cost of a factor of $m = O(n)$ in the number of queries.

In this case, the analyst must simply find out what the curator knows and query one of the two remaining hypotheses. As we've previewed, the analyst determines this by asking $q - 1$ queries given in (2.4). These all have the same distribution on the prior $f_i(\mathcal{P})$ (which we hereafter omit from $g$ for clarity), so the curator's answers may only depend on the posterior mean $\mathbb{E}[f_i(\mathcal{P}')]$. Recall that this always takes on
the values of \( a_s \) for some value of \( s \in \{0, 1, \ldots, 8n + 2\} \) depending on the uniform random value and what the curator knows about \( \ell(y_i) \).

Clearly the curator's function must have \( g(a_0), g(a_1) < \frac{1}{2} < g(a_{8n+1}), g(a_{8n+2}) \) with near certainty or one of these answers will come into effect but be too far off. Therefore, the curator will give answers below \( \frac{1}{2} \) with a probability \( \Omega \left( \frac{1}{n} \right) \) higher if \( \ell(y_i) = 0 \) and at least \( \Omega \left( \frac{1}{n} \right) \) lower if \( \ell(y_i) = 1 \), as compared with the case where \( \ell(y_i) \) is unknown.

Therefore, by taking \( y_i \) to be the same point \( y \in \mathcal{Y} \) for \( O(n^2 \log n) \) values of \( i \), the analyst obtains estimates of this probability that are additively precise to within \( O \left( \frac{1}{n} \right) \) on each of the counts of the potential values with error probability less than \( \frac{1}{n} \). By comparing these results across all \( y \in \mathcal{Y} \), the analyst can determine what the curator knows about the function using only \( O(n^3 \log n) \) queries, and in the final query, exploit this knowledge.

**Corollary 6.** Under the same model, the same adaptive analyst strategy causes the Bayesian private multiplicative weights curator to answer constant-inaccurately with constant probability using only \( O(n^4 \log n) \) queries.

**Proof.** Recall that this curator first looks at the answer that a proxy distribution gives. The proxy distribution is initialized as the prior mean, a distribution which makes every uniform random variable equally likely to be 0 or 1. Since all updates to the posterior are functions of the previous queries, the proxy distribution will never reflect any knowledge of the random variables that have not previously played a role in queries. Therefore, the proxy distribution will answer the query (2.4) with an answer within \( \frac{1}{2} \pm \frac{1}{m} \).

The curator releases this proxy answer if it is close enough to the true posterior mean. This definition of close enough is probabilistic, but the probability of a deviation on the order of \( \frac{1}{2} \) is negligible. Therefore, if the posterior mean is any of \( a_0, a_1, a_{8n+1}, a_{8n+2} \), which are all far from \( \frac{1}{2} \), the curator will ignore the proxy answer and use the fallback method instead, which fell under the scope of Theorem 5.11

\[ ^{11}\text{We do make one slight adjustment to the analyst strategy: Instead of simply counting the} \]
spending $O(n^4 \log n)$ queries figuring out what the curator knows, the analyst exploits this knowledge in the usual way.

### 2.3.4 Discussion

These results are perhaps a bit surprising. The same analyst strategy of employing slightly correlated queries to leak information was able to defeat two different strong curator approaches, smart rounding and private multiplicative weights. Theorem 5 is also quite general, showing a limitation to any attempt to obfuscate.

Notably, this is not merely a result of the focus of the Bayesian context on the posterior mean. While the parameters in the queries that the analyst asks are directly tuned to match the posterior mean, it is easy to see that the same could be done for the empirical mean, posterior median, or any other such aggregation.

The general attack here consists of two steps: (1) use slightly correlated queries to learn what the curator knows, and (2) exploit that knowledge to find a query which he is unable to answer. Whatever method the curator used to represent the data originally, the analyst would be able to find it and ask the query the curator won’t be able to answer accurately.

We now survey some features and proposed fixes to this problem. If your natural inclination is to ask, “Well, what about X?” then this is the section for you; otherwise, it can be skipped.

### Stability

A key component of these polynomial classification problems is the moment when the curator is uncertain between two hypotheses which differ on a significant ($> 2\epsilon$) fraction of the space. In such a situation, with decent probability, the next data point will distinguish those hypotheses, moving the posterior mean by at least $\epsilon$ with respect to some query.

---

number of answers above or below $\frac{1}{2}$, we count the number above $\frac{3}{4}$ or below $\frac{1}{4}$. In this way, all of the answers we are counting are due to the fallback method and not the proxy distribution. Once again, we must still have $g(a_0), g(a_1) < \frac{1}{2}$ and $g(a_{rn+1}), g(a_{rn+2}) > \frac{3}{4}$, so the difference in the counts or lack thereof will be noticeable depending on what the curator knows about $\ell(y)$.  

36
In other words, the posterior mean algorithm is not stable on classification problems, since its answers to some queries change significantly when a single sample is added. This suggests that there might be promise in the perspective of another line of work, that of algorithmic stability. In 2002, Bousquet and Elisseeff [6] offered stability in a slightly different context as a condition guaranteeing generalization, and both BNSSSU [2] and Hardt, Recht and Singer [13] have adapted this notion to the field of adaptive data analysis.

However, mandating stability in this context is actually counterproductive. In classification, every sample point is either equally likely or impossible under each hypothesis, and therefore yields very strong information. As is, the posterior only puts nonzero weight on the eligible hypotheses, those that agree on every point. To achieve stability, we must soften this requirement, for instance by putting a weight proportional to $c^k$ on a hypothesis that makes $k$ errors, for some $0 < c < 1$.

Unfortunately, this is actually even more vulnerable to the same attack. Suppose the curator has enough data to give the true hypothesis between 1/4 and 3/4 of the total weight. Then the weight is high enough that the analyst’s investigatory queries can isolate that hypothesis, but low enough that querying it directly will surely be incorrect.

Moreover, attempting to introduce stability makes this intermediate state last even longer than it did previously! By only reducing incorrect hypotheses’ weights by a factor of $c > 0$, the mass on the true hypothesis can only spend more time in the range [1/4, 3/4]. Thus, stable versions of the posterior mean algorithm actually behave poorly on an even wider range of parameters than the posterior mean itself.

**Decomposition**

This problem is hard for the usual methods, but it’s actually quite easy to match the static bound if we allow ourselves to break it open. With a tensor product model like this one, we can distinguish between data from the two component models, and ignore the data from $J_m$ (until there’s enough to uniquely determine the function) while fully updating based on the data from $(U_2^q)^\otimes(q-1)$. Since each of those behave...
well on their own, this “decomposition” curator strategy works on this problem in particular.

Unfortunately, it’s difficult if not impossible to turn this specific strategy hack into a general algorithm. For instance, if we introduce tiny perturbations to the independence of these two components, it becomes hard to separate our knowledge of each, or even to define the two components in the first place.

Adaptivity Detection

Another class of algorithms that does well, but in a fragile way, against this particular problem aims to detect when the analyst is using adaptivity and utilizes previously unseen data whenever it does.

The first such algorithm in the literature, DFHPRR’s EffectiveRounds, reserves about 99% of its data for a “checker” and splits the remaining points into $r$ equal-sized “estimation samples.” The algorithm proceeds with answers from the first estimation sample until those answers deviate from the checker by more than a noisy threshold, after which the sample is discarded. In this way, if there are only $r$ rounds of adaptivity, the algorithm only needs $O(r)$ times as much data. (To make this better than simple sample splitting, this works even if the researcher doesn’t know when the rounds of adaptivity occur.)

The explore-exploit attack we’ve described only uses two rounds of adaptivity, so EffectiveRounds would defeat it with only a constant increase in the sample size. However, the tensor product construction allows us simply to repeat the same problem $n$ times, and so defeat this algorithm using only $\tilde{O}(n^5)$ queries.

Could we apply a similar philosophy to do better in a Bayesian context? Perhaps. One note of hope: When the Justesen code problem posterior is uncertain between two hypotheses, only a constant number of additional data points are needed to resolve this uncertainty, in expectation. So if we get stuck in a problem like this, we only need to look at a few more data points to break out. One could imagine a curator algorithm that keeps some number of data points in reserve and updates on them only if the posterior is not concentrated on some query.
Unfortunately, looking at even one extra data point is too many. By a similar trick, the subsequent copies of the Justesen classification problem could involve a slightly higher parameters $m$, causing the curator to look at at least one new data point every $\tilde{O}(n^4)$ queries. Repeat this $n$ times, and the curator runs out of points to add after only $\tilde{O}(n^5)$ queries. Again, this isn’t elegant, and there could potentially be a viable curator algorithm along these lines, but it won’t be trivial.

**Restricting Models**

Perhaps the general problem that we’ve posed is too difficult, and we should restrict the models somehow, in addition to ruling out use of asymmetric information. There seem to be two general approaches in the literature to doing this, bottom-up and top-down.

The bottom-up approaches are the most promising, but the most restrictive. In general, they start with a particular use case of adaptive data analysis and build algorithms to handle those in particular, hoping to again to achieve tight sample complexity results in a specific setting.

For instance, Russo and Zou [25] focus on the situation where researchers compute a variety of statistics and report only the “best” one or several, such as the smallest $p$-values. In this setting, they are able to control the bias by bounding the mutual information between the choice of statistic(s) and all of the statistics’ realized values.

Similarly, Blum and Hardt [4] examine the situation of releasing an approximately correct machine learning competition leaderboard, a common and obvious source of overfitting to a frequently used holdout set. With this objective, the leaderboard algorithm can avoid releasing scores unless a new submission is a significant amount better than the previous best, which effectively limits the information leakage.

Alternatively, a top-down approach hopes to solve the general problem under some restricted conditions that preclude examples like the ones introduced in this paper. The trickiest step here is formulating what the restriction should be.

One possibility is to restrict the dimension of the problem. Indeed, the universe of $J_m \otimes (U_2^8)^{\otimes q-1}$ has dimension $\log|\mathcal{X}| = \log m + 1 + 8(q - 1)$. While big data often
deals with situations where the dimension is greater than the number of data points \( n \), it might not be as high as \( n^4 \) (our number of queries).

This is essentially the technique that the guarantees for the original Private Multiplicative Weights [14] algorithm utilize, although their results are written with a multiplicative factor of the dimension (to some power) in the required sample complexity.

While this does yield effective algorithms, it isn’t a priori clear why the dimension should play a role in the first place. After all, every query projects the space of distributions \( \Delta(\mathcal{X}) \) down to a single dimension, fitting it within \([0,1]\), so the queries themselves are dimension-independent.

**Controlled Trials**

We have one interesting new restriction to offer that might make the problem easier. To understand the motivation for this restriction, consider a common setup in scientific analysis: Randomized controlled trials, also known as A/B testing in business.

In this framework, samples are drawn independently from a population and assigned to one of two groups. One of the groups receives a change of some kind while the other group stays the same or receives an ineffective version of the change (placebo) if appropriate. The study aims to compare some output variables on each group, possibly restricted to subpopulations of the original sample. This type of problem is ripe for high generalization error, because of the range of questions to ask and the opportunity to ask them adaptively.

To formulate this in the same sort of mathematical framework, we consider the universe to be a product \( \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \). Here, \( \mathcal{X} \) will capture the demographic data about the population, and we will assume the marginal over \( \mathcal{X} \) is known. \( \mathcal{Y} \) then captures the group data: Whether the sample was placed in the experimental group or the control group. For simplicity, we will assume there are just two groups, so \( \mathcal{Y} = \{0, 1\} \), and the samples are uniformly randomly assigned to each group, independent of their demographic data. Finally, \( \mathcal{Z} \) is the output variable in question. For even greater simplicity, we will also take \( \mathcal{Z} = \{0, 1\} \), corresponding to a single binary variable.
being studied, such as whether someone recovered from an illness or died.

Since the object of study is the difference between the two groups, we will require that all queries be of a specific form:

$$E(x,y,z) \sim P[z | x \in S] y = 1 - E(x,y,z) \sim P[z | x \in S] y = 0$$  

(2.5)

for some subset $S \subset \mathcal{X}$. In other words, this is measuring the difference in the probability of the outcome variable being 1 between the two groups on a specific subpopulation, scaled by the (publicly known) fractional size of the subpopulation.$^{12}$

Now let us transform this query somewhat to relate it to other problems we've seen. First, we notice that up to a factor of two and a shift, this query is the same as $P[y = z$ and $x \in S]$. In other words, for the purposes of queries of the form in (2.5), we can collapse $\mathcal{Y} \times \mathcal{Z} = \{0, 1\}^2$ down to two points, identifying the pairs $(0, 0)$ and $(1, 1)$, and $(0, 1)$ and $(1, 0)$. Let $w$ be the indicator function on the event $y = z$, so we are asked to estimate $E[w | x \in S]$.

Moreover, we can consider this as an average value of the function $f : \mathcal{X} \rightarrow [0, 1], x \mapsto E[w | x]$ corresponding to the difference in the effect of treatment between the groups, rescaled to fall within $[0, 1]$ with $f(x) = 1/2$ meaning no effect. Therefore, the queries amount to asking the agreement between the true probability function $g : \mathcal{X} \rightarrow [0, 1]$ and indicator functions $f(x) = 1_{x \in S}$. In other words, these are merely a slight relaxation of classification problems! There is one small change: Instead of measuring agreement with functions to the set $\{0, 1\}$, we are measuring agreement with functions to the interval $[0, 1]$, where we say that the value $t \in [0, 1]$ corresponds to agreement $t$ with $1$ and $1 - t$ with $0$. This is just a convex relaxation of the original classification problem framework we introduced Section 2.3.1.

Restricting to (soft) classification problems might not seem promising since most of our problematic examples come from that framework. However, the tensor product, an integral construction of the final challenging model that created slightly correlated

$^{12}$This scaling matches that of statistical queries. It also seems appropriate, as opposed to say $E(x,y,z) \sim P[z | y = 1, x \in S]$, because when $S$ is small, we would need more data to estimate this probability accurately.
queries, cannot be expressed in a classification model. This gives hope that some version of the posterior mean might solve all classification problems, and therefore provide a framework for avoiding overfitting from multiple comparisons in parallel group randomized controlled trials.

2.4 New Adaptive Guarantees

In addition to the negative result of Theorem 5, we offer a new category of Bayesian adaptive data analysis problems on which the Bayesian curator can indeed achieve the static query complexity \( n_\epsilon = O \left( \frac{1}{\epsilon^2 \log \frac{1}{\epsilon}} \right) \). In the framework of Section 2.3.4, this is an instance of a family of bottom-up examples.

We will first characterize when the posterior mean itself, without any obfuscation, serves as an accurate answer. This characterization begins with a tool from probability theory.

2.4.1 Subgaussian Concentration and Accuracy

A random variable \( X \) with zero mean is said to be \( \sigma^2 \)-subgaussian if for all \( \lambda \), 

\[
E[\exp(\lambda X)] \leq \exp(\lambda^2 \sigma^2 / 2).
\]

Here, \( \sigma^2 \) is also known as the variance proxy. Following the seminal work on subgaussian random variables [8], define \( \tau(X) = \min\{\sigma \geq 0 : X \text{ is } \sigma^2\text{-subgaussian}\} \). Recall the following:

**Proposition 7.** Some basic facts about subgaussian random variables.

1. As the name suggests, the variance gives a lower bound on the subgaussian variance proxy: \( \text{Var}[X] \leq \tau^2(X) \). If the two are equal, \( X \) is said to be strictly subgaussian.

2. (See [8] Theorem 1.2) The space of subgaussian random variables is a Banach space with respect to the norm \( \tau(X) \). That is, it has the right scaling, \( \tau(aX) = a\tau(X) \) for all \( a > 0 \), satisfies the triangle inequality \( \tau(X + Y) \leq \tau(X) + \tau(Y) \) and is complete on the space of subgaussian random variables.
3. (See [8] Theorem 1.3) $X$ is $\sigma^2$-subgaussian if for all integers $k \geq 2$,

$$
\mathbb{E}[X^{2k}] \leq \left( \frac{\sigma^2}{\sqrt{3.1}} \right)^k (2k-1)!!, \text{ where } (2k-1)!! = (2k-1)(2k-3) \cdots (1) = \frac{(2k)!}{2^k k!}.
$$

If $X$ is symmetric (i.e. $X$ and $-X$ have the same distributions, so all odd moments are zero), the factor of $\sqrt{3.1}$ can be dropped.

If $\mathbb{E}X \neq 0$, we will abuse notation slightly and write $\tau(X) = \tau(X - \mathbb{E}X)$. That is, we will consider random variables $X$ with nonzero mean to be $\sigma^2$-subgaussian if their centered versions $X - \mathbb{E}X$ are. Note that in this context, conditions like in Proposition 7.3 apply to the centered moments $\mathbb{E}[(X - \mathbb{E}X)^{2k}]$.

Now, let us return to the Bayesian adaptive data analysis problem and consider a single query for a moment, setting $q = 1$. In this case, the static bound states that $n = n_s = O \left( \frac{1}{\epsilon^2 \log \frac{1}{\delta}} \right)$ samples are sufficient for estimation of $\mathbb{E}_{x \sim \rho}[f(x)]$ to additive error $\epsilon$, with probability $1 - \delta$. We will show that for this particular relationship between $\epsilon$ and $\delta$, this follows from a subgaussianity property on the posterior.

**Proposition 8.** If the curator’s posterior distribution is $O(1/n)$-subgaussian with respect to every query, then the posterior mean-answering curator answers correctly and achieves the static sample complexity of $n = O \left( \frac{1}{\epsilon^2 \log \frac{9}{\delta}} \right)$.

Before proving the proposition, let us explicate this condition. Each query $f : \mathcal{X} \to [0,1]$ projects every possible population $\bar{\rho}$ to a value $\mathbb{E}_{x \sim \bar{\rho}} f(x) \in [0,1]$. Therefore, $f$ projects every possible posterior distribution on populations to a distribution on $[0,1]$. The assumption is that this projected distribution of the curator’s posterior is $O(1/n)$-subgaussian, for every possible query $f$.

The claim is that as long as this condition holds, the curator doesn’t care what queries the analyst asks; the posterior mean will be accurate on all of them. Indeed, the curator could actually release the entire posterior mean (not just its value on every query), giving the analyst everything she will ever learn from his answers. If we can show that the probability of error on any query is less than $\frac{\delta}{q}$, then a union bound will give a total error probability at most $\delta$, no matter which queries are asked.
Proof. Suppose the curator’s posterior distribution with respect to a given query is $c/n$-subgaussian. Equivalently stated, suppose the error $X = \mathbb{E}_{x \sim p}[f_i(x)] - a_i$ of the posterior mean is a (centered) $c/n$-subgaussian random variable. By Markov’s inequality, setting $\lambda = ne/c$,

$$
\mathbb{P}[X > \epsilon] \leq \exp(-\lambda\epsilon) \mathbb{E}[^\lambda X] \leq \exp\left(\frac{\lambda^2c}{2n} - \lambda\epsilon\right) = \exp\left(-\frac{nc^2}{2c}\right).
$$

We can also prove an identical bound for $\mathbb{P}[X < -\epsilon]$, and therefore, the error probability $\delta_1$ satisfies

$$
\delta_1 \leq 2 \exp\left(-\frac{nc^2}{2c}\right) \iff n \leq \frac{2c}{\epsilon^2} \log \frac{2}{\delta_1} = O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta_1}\right).
$$

Taking $\delta_1 = \delta/q$, we have shown the required bound.

This is rather remarkable; it says that the curator can do just as well against adaptive queries as against static queries if his posterior has this concentration property. Of course, such a concentration property will not hold in every case; for instance, it is far from true for the posteriors in the series of examples considered in the previous section. But in cases where it does hold, like the Dirichlet prior and posterior we will investigate shortly, the curator doesn’t have to do any obfuscation.

As an added bonus, the subgaussian framework also simplifies the set of queries we must consider:

**Proposition 9.** If the curator’s posterior distribution is $O(1/n)$-subgaussian with respect to every counting query, then the posterior mean curator answers correctly and achieves the static sample complexity of $n = O\left(\frac{1}{\epsilon^2} \log \frac{q}{\delta}\right)$.

**Proof.** The key here is Proposition 7.2. Since $\tau(X)$ is a norm, convex combinations of $\sigma^2$-subgaussian random variables will also be $\sigma^2$-subgaussian:

$$
\tau(\lambda X + (1 - \lambda)Y) \leq \lambda \tau(X) + (1 - \lambda)\tau(Y) \leq \lambda \sigma + (1 - \lambda)\sigma = \sigma.
$$

Recall that counting queries only have values 0 or 1, and therefore form the vertices
in the hypercube of possible queries \([0,1]^X\). All other queries are convex combinations of these\(^\text{13}\), and therefore will be \(O(1/n)\)-subgaussian as well (with no loss in the constant). Therefore, by Proposition 8, the posterior mean curator wins on all statistical queries if he wins on counting queries.

\[\square\]

### 2.4.2 Beta Distribution Subgaussianity

The first posterior distribution we will investigate is the ubiquitous Beta distribution. We will see in the next section that this distribution is also the projection of the Dirichlet distribution on any counting query.

**Definition.** The **beta distribution** \(\text{Beta}(\alpha, \beta)\) is a continuous distribution on \([0,1]\) with density

\[
\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1},
\]

where \(\Gamma(n)\) is the gamma function, satisfying \(\Gamma(n) = (n-1)!\) for \(n \in \mathbb{N}\).

The fraction in the density formula above is simply a normalization constant. Notice that \(\text{Beta}(1,1)\) is the uniform distribution on \([0,1]\). For \(\alpha = \beta \to 0\), \(\text{Beta}(\alpha, \beta)\) converges to the Rademacher 0-1 random variable, \(\frac{1}{2}(\delta_{x,0} + \delta_{x,1})\). Finally, the (raw) moments of the beta distribution are given by

\[
\mathbb{E}[X^k] = \prod_{r=0}^{k-1} \frac{\alpha + r}{\alpha + \beta + r} = \frac{\alpha_k}{(\alpha + \beta)_k}, \tag{2.6}
\]

where \((x)_k = x(x+1)\ldots(x+k-1)\) is a **rising factorial**. In particular, the mean and variance are given by

\[
\mathbb{E}[X] = \frac{\alpha}{\alpha + \beta} \quad \text{Var}[X] = \frac{\alpha \beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.
\]

To see the beta distribution as a posterior, consider a Bernoulli trial, a single

\[\text{13}\text{If } X \text{ is infinite, then some queries might not be convex combinations of finitely many counting queries, but we can find a sequence of finite convex combinations converging to them in } \infty\text{-norm simply by picking more possible values for the function in } [0,1] \text{ at each step. This will converge in } \infty\text{-norm, so the completeness of } \mathcal{r} \text{ from Proposition 7.2 shows that they must also have the same bound on their subgaussian norm.}\]
event with two possible outcomes: success or failure. If our prior over the probability of success is given by $\text{Beta}(\alpha, \beta)$, then upon receiving $n_1$ successes and $n_2$ failures, a Bayesian update yields a posterior proportional to

$$x^{\alpha-1}(1-x)^{\beta-1}x^{n_2}(1-x)^{n_1} = x^{\alpha+n_2-1}(1-x)^{\beta+n_1-1},$$

so after renormalizing, it must be the $\text{Beta}(\alpha+n_2, \beta+n_1)$ distribution. This is what is meant by calling the family of beta distributions a **conjugate prior**: If the prior is a beta distribution, the posterior will be as well.

Surprisingly, despite substantial focus including the entire *Handbook of Beta Distribution and its Applications* [12], the following concentration result does not appear to be known:

**Theorem 10.** The beta distribution $\text{Beta}(\alpha, \beta)$ is $\frac{1}{4(\alpha + \beta) + 2}$-subgaussian.

**Remark.** Numerical data suggests that

$$\tau^2(\text{Beta}(\alpha, \beta)) \leq \tau^2(\text{Beta}((\alpha + \beta)/2, (\alpha + \beta)/2)) = \text{Var}(\text{Beta}((\alpha + \beta)/2, (\alpha + \beta)/2)) = \frac{1}{4(\alpha + \beta + 1)}.$$  

That is, $\text{Beta}(\alpha, \beta)$ appears to be maximized for fixed $\alpha + \beta$ when $\alpha = \beta$, where it appears to be strictly subgaussian. Since the variance is a lower bound for $\tau^2$ (Proposition 7.1), we conclude that Theorem 10 is tight up to a factor of $1 + o(1)$, seen as a function of $\alpha + \beta$. However, numerical data is very clear that the lower bound is tight:

**Conjecture 1.** The beta distribution $\text{Beta}(\alpha, \beta)$ is $\frac{1}{4(\alpha + \beta + 1)}$-subgaussian.

Before proving Theorem 10, let’s see what this concentration result means for a nearly trivial case of Bayesian adaptive data analysis.

---

14Subsequent to this work appearing on the arXiv, Marchal and Arbel [20] were able to improve this result, characterizing the subgaussian variance proxy for $\text{Beta}(\alpha, \beta)$ for all $\alpha, \beta$, and in the process proved Conjecture 1. However, we retain my original weaker theorem for the novelty of my proof and to keep the appropriate chronology.
Corollary 11. If the prior on the parameter of a Bernoulli distribution is given by Beta($\alpha, \beta$) for any $\alpha, \beta$, the posterior mean curator strategy wins.

Proof. If the prior is Beta($\alpha, \beta$), the posterior is Beta($\alpha', \beta'$) with $\alpha' + \beta' = \alpha + \beta + n > n$.

With a Bernoulli trial, there are two nontrivial counting queries, the probabilities of success and failure, or $X$ and $1 - X$. The latter is an affine function of $X$, so by Propositions 7.2 and 9 it suffices to show that $X \sim$ Beta($\alpha', \beta'$) is $O(1/n)$-subgaussian.

Indeed, Theorem 10 shows that the posterior is $\frac{1}{4(\alpha' + \beta') + 2} < \frac{1}{4n}$-subgaussian, so by Proposition 8, the posterior mean curator strategy wins.

Thus, we see that as a conjugate family, the beta distributions have an interesting property: Only some beta distributions can be posteriors after observing and updating on $n$ data points. All such posteriors have $\alpha + \beta > n$, and Theorem 10 says that they must concentrate to the degree we require.

We have actually found two very different and new styles of proofs of subgaussianity results like Theorem 10. The first is perhaps more expected: We bound all of the raw moments of the beta distribution and combine these into a bound on the moment generating function $E[e^{\lambda X}]$. This proof is still nonstandard because it doesn’t use the centered moment generating function $E[e^{\lambda(X - EX)}]$ or the centered moments, and therefore is more algebraically involved. For the full details, see Appendix C.1.

Even more surprisingly, we can get a simpler and stronger result by considering the beta distribution as a Bayesian posterior.

Proof of Theorem 10. The key to this proof is Azuma’s Inequality:

Lemma 12 (Azuma’s Inequality (c.f. [30] Lemma 3.7)). Let \{X_k\}_{k \in \mathbb{N}} be a martingale adapted to the filtration \{F_k\}_{k \in \mathbb{N}} such that for all k, $$(X_k - X_{k-1})|F_{k-1}$$ is $\sigma_k^2$-subgaussian. Then $(X_n - X_0)|F_0$ is $\sum_{k=1}^{n} \sigma_k^2$-subgaussian.

The martingale we will construct is surprisingly related to our problem: Let $F_0$ be the Beta($\alpha, \beta$) prior over the parameter of a Bernoulli random variable, let $F_k$ be the updated information upon receiving $k$ samples from the random variable, and let $X_k$
be the resulting posterior mean. Then as $n \to \infty$, $X_n$ approaches the true parameter of the random variable almost surely, so $X_n - X_0$ approaches the error of the original posterior mean.

Moreover, it is fairly elementary to check that $X_k - X_{k-1}$ is a martingale with the appropriate subgaussian variance proxy. Suppose that the posterior after the first $k - 1$ data points is Beta($\alpha', \beta'$), where of course $\alpha' + \beta' = \alpha + \beta + k - 1$. Then the posterior mean $X_{k-1} = \frac{\alpha'}{\alpha' + \beta'}$. Conditioning on the samples the curator has seen so far, the next sample will be a success with probability $X_{k-1}$ and a failure with probability $1 - X_{k-1}$. That is:

\[
\begin{align*}
\text{w.p. } \frac{\alpha'}{\alpha' + \beta'}, \quad & X_k = \frac{\alpha' + 1}{\alpha' + \beta' + 1} \\
\implies & X_k - X_{k-1} = \frac{\alpha' + 1}{\alpha' + \beta' + 1} - \frac{\alpha'}{\alpha' + \beta'} = \frac{\beta'}{(\alpha' + \beta')(\alpha' + \beta' + 1)};
\end{align*}
\]

\[
\begin{align*}
\text{w.p. } \frac{\beta'}{\alpha' + \beta'}, \quad & X_k = \frac{\alpha'}{\alpha' + \beta' + 1} \\
\implies & X_k - X_{k-1} = \frac{\alpha'}{\alpha' + \beta' + 1} - \frac{\alpha'}{\alpha' + \beta'} = \frac{-\alpha'}{(\alpha' + \beta')(\alpha' + \beta' + 1)}
\end{align*}
\]

This difference is clearly mean-zero, so $\{X_k\}$ is indeed a martingale, as expected. As a centered binary random variable, by Theorem 3.1 in [7], the subgaussian variance proxy of $X_k - X_{k-1}$ is given by

\[
\tau^2(X_k - X_{k-1}) = \frac{1}{(\alpha' + \beta' + 1)^2} K \left( \frac{\alpha'}{\alpha' + \beta'} \right), \quad K(p) = \begin{cases} 
0 & p \in \{0, 1\} \\
\frac{1}{4} & p = \frac{1}{2} \\
\frac{p^{-(1-p)}}{2(\ln p - \ln(1-p))} & \text{otherwise.}
\end{cases}
\]

In particular (Lemma 2.1 in [7]), $K(p) \leq \frac{1}{4}$ for all $p \in [0, 1]$, so $X_k - X_{k-1}$ is $\frac{1}{4(\alpha' + \beta' + 1)^2} = \frac{1}{4(\alpha + \beta + k)^2}$-subgaussian.
Therefore, by Azuma's inequality, $\text{Beta}(\alpha, \beta)$ is subgaussian with variance proxy

\[
\sum_{k=1}^{\infty} \frac{1}{4(\alpha + \beta + k)^2} \leq \sum_{k=1}^{\infty} \frac{1}{4(\alpha + \beta + k + 1/2)(\alpha + \beta + k - 1/2)}
\]

\[
= \sum_{k=1}^{\infty} \left( \frac{1}{4(\alpha + \beta + k - 1/2)} - \frac{1}{4(\alpha + \beta + k + 1/2)} \right)
\]

\[
= \frac{1}{4(\alpha + \beta) + 2'}
\]

as desired.\(^{15}\)

\[\square\]

### 2.4.3 Dirichlet Prior

Of course, a Bernoulli random variable is a nearly trivial example of Bayesian adaptive data analysis: There is (essentially) only one possible query, making adaptivity meaningless. However, the result fortunately generalizes to a much more useful framework: Dirichlet priors on categorical random variables.

Recall that a categorical random variable has support \{1, \ldots, k\} for some positive integer \(k\), and probabilities \(p_1, \ldots, p_k\) for each value. \(k = 2\) corresponds to a Bernoulli trial again, but if \(k > 2\), there are many possible queries, each corresponding to a vector \(\vec{v} = (f(1), \ldots, f(k)) \in [0, 1]^k\) and asking for the dot product \(\vec{p} \cdot \vec{v}\).

The conjugate prior for the categorical random variable is the Dirichlet distribution \(\text{Dir}(\alpha_1, \ldots, \alpha_k)\), the natural generalization of the beta distribution. Its probability density function over the \(k\)-dimensional simplex \(\{(x_1, \ldots, x_k) : x_1 + \cdots + x_k = 1, x_i \geq 0 \forall i\}\) is given by

\[
\frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} x_1^{\alpha_1} \cdots x_k^{\alpha_k}.
\]

Therefore, upon receiving data with counts \(c_i\) of category \(i\), the posterior is given by \(\text{Dir}(\alpha_1 + c_1, \ldots, \alpha_k + c_k)\), making the Dirichlet distribution a conjugate prior for the categorical distribution.

**Theorem 13.** *In the direction of any query vector \(\vec{v}\), the Dirichlet distribution is* \(^{15}\)The final steps of this analysis are also tight; \(\sum_{k=1}^{\infty} 1/(4(\alpha + \beta + k)^2) \geq 1/((4(\alpha + \beta) + 2 + 1/(3(\alpha + \beta))))\).

\[^{15}\text{The final steps of this analysis are also tight; } \sum_{k=1}^{\infty} 1/(4(\alpha + \beta + k)^2) \geq 1/((4(\alpha + \beta) + 2 + 1/(3(\alpha + \beta))))\]
In exactly the same way, this guarantees accuracy of the posterior mean:

**Corollary 14.** If the prior on the parameter of a categorical random variable is Dir(α₁, ..., αₖ) for any α₁, ..., αₖ, the posterior mean curator strategy wins.

**Proof Sketch of Theorem 13.** First, we use Proposition 9 to reduce to the case where the vector \( \vec{v} \in \{0, 1\}^k \). We then show that the Dirichlet prior with respect to such a query is Beta \( \left( \sum_{v_i=1} \alpha_i, \sum_{v_i=0} \alpha_i \right) \). By Theorem 10, this implies that the distribution is \( \frac{1}{2(\alpha_1 + ... + \alpha_k + 1)} \)-subgaussian, as desired.

The only difficulty in the full proof is parameterizing the simplex appropriately. For the full details, see Appendix C.2.

### 2.4.4 Discussion

We have shown that any Dirichlet posterior concentrates, making the posterior mean a winning curator strategy if the prior is given by a Dirichlet. Notably, this does not mean that the empirical mean is at all accurate; if \( \alpha_1 + ... + \alpha_k \gg n \), then the prior will dominate the posterior mean estimate and the empirical mean could be quite inaccurate on some queries, such as those that a boosting attack might find.

The proof of Theorem 10 is quite interesting on several accounts. From one perspective, it shows that stability guarantees accuracy for the posterior mean. However, we should be careful in morally applying this result back to the original non-Bayesian problem. For instance, the empirical mean is also stable, but as several attacks show, it can be quite inaccurate. The key is that the posterior mean is a martingale, while the empirical mean is clearly not: If the empirical mean after a finite number of entries is 0, its expectation in the future should still be nonzero.

As we look to extend this result, the family of beta distributions have a key monotone property as the posteriors for Bernoulli trials that made this approach possible: When the posterior updates on an additional data point, the positive parameter \( \alpha + \beta \)
increases by 1. This monotonicity means that only some beta distributions in the family can be the posterior of a curator who has already received \( n \) data points, i.e. those with \( \alpha + \beta > n \).

Other conjugate priors also feature similar monotonicity features:

- For the \( m \)-binomial distribution with conjugate prior \( \text{Beta}(\alpha, \beta) \), \( \alpha + \beta \) increases by \( m \) with each binomial sample.

- For the \( m \)-multinomial distribution with conjugate prior \( \text{Dir}(\alpha_1, \ldots, \alpha_k) \), \( \alpha_1 + \ldots + \alpha_k \) increases with \( m \) with each multinomial sample.

- For the Poisson distribution with conjugate prior \( \Gamma(\alpha, \beta) \), the parameter \( \beta \) increases by 1 with each Poisson sample.

- For the geometric distribution with conjugate prior \( \text{Beta}(\alpha, \beta) \), \( \alpha \) increases by \( m \) with each geometric sample.

Therefore, by this general approach in Proposition 8, in any of these conjugate prior models for adaptive data analysis, the posterior mean curator is accurate when \( n = O \left( \frac{1}{\delta} \log \frac{1}{\delta} \right) \) if each of the projections of these distributions onto queries has the appropriate subgaussian concentration. We formulate the necessary conditions into a series of conjectures (using Proposition 9):

**Conjecture 2.** For any \( S \subset \{0, 1, \ldots, m\} \), the random variable

\[
\sum_{k \in S} \binom{m}{k} p^k (1 - p)^{m-k}, \quad p \sim \text{Beta}(\alpha, \beta)
\]

is \( O \left( \frac{m}{\alpha + \beta} \right) \)-subgaussian.

**Conjecture 3.** For any subset \( S \subset \{0, 1, \ldots\} \), the random variable

\[
\sum_{k \in S} p(1 - p)^k, \quad p \sim \text{Beta}(\alpha, \beta)
\]

is \( O \left( \frac{1}{\alpha} \right) \)-subgaussian.
Conjecture 4. For any subset $S \subseteq \{(x_1, \ldots, x_k) \in \mathbb{Z}^n : x_i \geq 0, x_1 + \cdots + x_k = m\}$, the random variable

$$\sum_{(x_1, \ldots, x_k) \in S} \frac{m!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, \quad (p_1, \ldots, p_k) \sim \text{Dir}(\alpha_1, \ldots, \alpha_k)$$

is $O\left(\frac{m}{\alpha_1 + \cdots + \alpha_k}\right)$-subgaussian.

Conjecture 5. For any subset $S \subseteq \{0, 1, \ldots\}$, the random variable

$$\sum_{k \in S} \frac{\lambda^k e^{-\lambda}}{k!}, \quad \lambda \sim \text{Gamma}(\alpha, \beta)$$

is $O\left(\frac{1}{\beta}\right)$-subgaussian.

Numerical data suggests that all of these conjectures are true, and the simplest special cases of each are relatively easy to check using Lemma 25. However, both the raw moments and the step sizes of most of these transformed random variables are not simple to compute, so new methods will be required to prove any of these conjectures.

### 2.5 Conclusions

What makes adaptive data analysis inherently hard? What stops the curator from accurately answering as many adaptive queries as he can static queries?

The picture from previous lower bounds was bleak: Powerful analysts with complete knowledge of the distributions they were pretending to study could compile errors and catch the curator making a mistake. Having already somehow gotten access to the distribution itself, these analysts just needed $O(n^2)$ queries to crack the curator’s $n$ samples and stump him with a query about the rest of the distribution he hadn’t seen. The proposed solutions therefore naturally sought strong differential privacy techniques to protect every data point from the gaze of these nearly omniscient analysts.
For practitioners, though, this worry about superintelligences disguising themselves as curious seekers of truth seems over the top. At the very least, surely this can't be the only difficulty with adaptivity! To try to understand what else makes adaptivity challenging, we translated the original formulation of adaptive data analysis over to the information-symmetric Bayesian setting, which we then set out to explore.

Our initial scouting report is mixed. On the one hand, we found a new type of difficulty from analysts using only $\tilde{O}(n^4)$ slightly correlated queries to subtly leak information about a problem with high posterior uncertainty, even past the strongest known obfuscation techniques. On the other hand, we also showed that for a natural family of conjugate prior models, the posterior is well-concentrated and the curator doesn't even need any obfuscation. Along the way, we proved a new result on the subgaussian concentration of beta distributions, which as a bonus has already found application outside of adaptive data analysis [23], and which has already been improved [20].

To be clear, we study Bayesian adaptive data analysis both for contexts where introducing a prior is appropriate and for the analogy to those in the original framework when we suspect that information asymmetric techniques are not playing a role. Not every problem can be easily written with an accurate prior, and our suggestion is not to attempt to write one anyways. The situation is actually somewhat analogous to the field of solar astronomy: When there is a solar eclipse, researchers can study the sun's corona with otherwise overpowering light from the sun itself blocked. In the same way, Bayesian adaptive data analysis blocks the otherwise dominant class of lower bound techniques, those exploiting information asymmetry. With those removed, we can better understand other sources of error in adaptive data analysis as commonly carried out in practice.
Chapter 3

Cross-Validation

3.1 Introduction

While the theory of adaptive data analysis is useful for aiming at the particular problem of characterizing the difficulties inherent in adaptivity, its validation model, a single reused holdout set, only matches some particular scenarios of common practice. This is perhaps most vividly illustrated by one of the citations that DFHPRR invoke to justify that overfitting is a real issue that needs to be addressed, a blog post about Kaggle strategy by David Wind [33]. Here are two of the Kaggle Masters Wind cites on the topic of overfitting to the public leaderboard:

"The public leaderboard is some help, [...] but one needs to be careful to not overfit to it especially on small datasets. Some masters I have talked to pick their final submission based on a weighted average of their leaderboard score and their CV score (weighted by data size). [...]’ (Steve Donoho) ‘Overfitting to the leaderboard is always a major problem. The best way to avoid it is to completely ignore the leaderboard score and trust only your cross-validation score. [...]’ (Josef Feigl)"

These are remarkable admissions. Kaggle is providing all competitors the opportunity to test their algorithm on a completely fresh holdout set, which will provide an unbiased estimate of how one’s algorithm will perform on the true data set. And yet, Feigl advocates ignoring that fresh data entirely in favor of the cross-validation scores obtained only using the publicly available training data, while the masters Donoho
spoke with support a mixture.

In addition to the obvious hope to improve holdout scores to prevent overfitting, we can make two additional conclusions: First, the validation model matters. Having the opportunity to average over multiple different validation sets gives us a lower-variance estimate than a single fresh holdout set, even if we have to introduce bias by removing this test data from the training data. Second, the use of cross-validation is much more common than the use of a single holdout set, even in situations where a fresh holdout set is provided due to the structure of the competition.

After all, most machine learning is not done in the context of a competition with this arbitrary access to a holdout set. The picture of the analyst and curator is helpful for the theory of adaptivity, but it is not very close to common practice. Therefore, there is much greater potential impact in designing new techniques for researchers hoping to do proper validation without a fixed partition between the training and test data.

Moreover, cross-validation is a ripe field for new algorithms. Classical methods, described in Section 3.1.2, give results that are unavoidably biased (some more than others, of course) by the fact that they measure performance of the algorithm given a smaller training set. They all also only use information from the performance of the algorithm at a single (wrong) training set size, which is problematic if the algorithm performs differently at different training set sizes. In Section 3.2, we will provide a new approach to handle problems of bias which we call Extrapolative Cross-Validation. Finally, even though there is a range of different cross-validation methods, little work has illuminated how to choose the best method for the data set and algorithm at hand. Ideally, this would be done algorithmically, to avoid accusations of cherry-picking. In Section 3.3, we introduce the first such algorithmic approach, a CV-for-CV method we naturally call Hierarchical Cross-Validation. Empirical results of both of these new families of methods, in Section 3.4, demonstrate how these methods compare to existing methods in mostly ways we expect, with some room for improvement particularly for hierarchical methods. Finally, both Extrapolative and Hierarchical Cross-Validation are most natural to define with an online learning framework, but
in Section 3.5, we describe how they can be applied in offline training contexts.

3.1.1 Notation

Let us define the problem of validation, generally speaking. We will focus on machine learning problems related to prediction as this seems to capture the majority of uses of cross-validation, with one major exception being density estimation.

There is a probability distribution $p$, unknown to the user, over ordered pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The second coordinate $y$ will capture the label or value our algorithm will be seeking to predict, while the first coordinate $x$ will capture covariates that we are hoping to use to predict it. In a regression framework, $\mathcal{Y}$ will be continuous and frequently real-valued, while in a classification framework, $\mathcal{Y}$ will be discrete. Either way, we will be given $n$ data points $(x_1, y_1), \ldots, (x_n, y_n)$.

To capture our accuracy, we will be given a loss function $\ell : \mathcal{Y} \times \mathcal{Y}$. Common loss functions include squared loss $\ell(y, y') = (y - y')^2$ in a regression context, or 0-1 loss $\ell(y, y') = \delta_{y,y'}$ in a classification context.

Finally, we will have an algorithm $A$, which we will treat for our purposes as (almost) a black box. The algorithm will take as input some number of data points $S_I = \{(x_i, y_i)\}_{i \in I}$ for $I \subseteq [n]$, sometimes with a lower bound on the training set size $|I| \geq n_{\min}$ (this is often a minor consideration). Its output will be a prediction function $A_S : \mathcal{X} \to \mathcal{Y}$ meant to capture the algorithm’s best guess for the label of a new data point with given values of its covariates.

The problem of validation is simple: predict the average error of the algorithm trained on the data. That is, estimate

$$E_{(x,y) \sim p}[\ell(A_{S_{[n]}(x)}, y)]$$

(3.1)

To make this estimation, validation methods will have access to estimates of the following form: $\ell_{I,j} := \ell(A_{S_I}(x_j), y_j)$ for some $I \subseteq [n]$ and $j \in [n]$. Which and how many of these $\ell_{I,j}$ the validation method is allowed to access will depend on the computation time and learning model under assumption, as we will describe next.
3.1.2 Classical Cross-Validation Methods

In their exhaustive 2010 survey of cross-validation procedures for model selection [1], Arlot and Celisse describe how cross-validation arose historically. The first observation, dating to the 1930s, was that estimates of risk obtained by testing the algorithm on the same data it is trained on, known as resubstitution, is inherently overoptimistic. This led to the introduction of the simple validation estimate:

\[ \ell_I := \frac{1}{n - |I|} \sum_{j \not\in I} \ell_{I,j}, \]

for some nonempty proper subset \( I \subset [n] \). That is, we train our algorithm on the training set \( S_I \) and test it on the disjoint validation set \( S_{[n] \setminus I} \), averaging the resulting losses.

This is almost the context examined in adaptive data analysis, where the training set is essentially ignored. The only difference is that we are using it for the purpose of estimating the loss of the algorithm trained on \( S_{[n]} \), rather than on \( S_I \).\(^1\)

This technique is known as "simple validation" because it only consists of measuring the performance of the algorithm trained on a single training set \( S_I \). Cross-validation refers to estimates that involve multiple training sets.

We will define a classical cross-validation technique as a simple average over \( \ell_I \) for \( I \in \mathcal{I} \). Here are the examples cited in Arlot and Celisse [1]:

- Leave-One-Out (LOO): \( \mathcal{I} = \binom{[n]}{n-1} = \{ [n] \setminus \{i \} : i \in [n] \} \). That is, our training sets are all possible subsets of size \( n - 1 \), and the validation sets are the corresponding singletons.

- Leave-\( p \)-out (LPO): \( \mathcal{I} = \binom{[n]}{n-p} \). In this generalization of LOO, we instead leave \( p \) data points out of each training set, so the corresponding validation sets each have size \( p \).

\(^1\)While some results in the literature compare this validation estimate to the algorithm trained on \( S_I \), this is merely all we can manage to prove about this method, not the ultimate goal. After all, the training subset \( I \) is defined by the method, not part of the original problem structure.
- V-fold CV (VFCV): For some partition $A_1, \ldots, A_V$ of $[n]$ into (nearly) equally-sized subsets, $\mathcal{I} = \{[n] \setminus A_i\}_{i=1}^V$.

- Balanced Incomplete CV (BICV): $\mathcal{I}$ is a balanced incomplete block design, i.e. $|\{I \in \mathcal{I} : k \in I\}|$ is independent of $k$ and $|\{I \in \mathcal{I} : k, l \in I\}|$ is independent of $k \neq l \in [n]$.

- Repeated Learning Testing (RLT): Choose $B$ random subsets of $[n]$ with some fixed size $n_t$ without replacement (that is, all subsets are distinct).

- Monte Carlo Cross-Validation (MCCV): Choose $B$ random subsets of $[n]$ with some fixed size $n_t$ with replacement (some subsets may be repeated).

Of course, not all of these are in widespread use. Following a recommendation of empirical work by Kohavi [18] in 1995, the most commonly used methods today are 5-fold and 10-fold cross-validation.

Besides the fact that all of these consist of simple averages, another remarkable feature of all classical cross-validation algorithms is that in every case, elements of $\mathcal{I}$ are all the same (or nearly the same) size. That is, the training algorithms are utilizing training sets of identical sizes, despite the fact that they are attempting to approximate the performance of the algorithm on an entirely different training set size.

There is one exception to this rule in the literature, which we will discuss next.

### 3.1.3 Online Cross-Validation

In 1999, Blum, Kalai and Langford [5] introduced a little-heralded alternative to the classical CV paradigm based on holdout sets which they called progressive cross-validation. In the notation above, they first randomly permute the data points by

\[2\] This presentation follows Arlot and Celisse [11], but we note that at least some of the literature also seems to treat RLT as sampling the subsets with replacement. The difference is not important to our work.
some permutation $\sigma \in S_n$, and then average

$$\ell_{\{\sigma(1), \ldots, \sigma(k-1)\}, \sigma(k)}$$

over $t \leq k < n$ for some minimum training set size $t$. That is, they train on a sequence of prefixes of the data, and test on the next data point.

As the authors prove, progressive cross-validation enjoys the same Hoeffding-style estimation guarantees as a single holdout. Moreover, although this is not the focus of their work, a single run of this progressive validation can then be repeated $B$ times using permutations $\sigma_1, \ldots, \sigma_B$ to form a progressive cross-validation estimate. There is then a corresponding panoply of options regarding the choices of permutations $\sigma_i$ and minimum training set size $t$. For instance, one possible analog to $V$-fold cross-validation would be to choose the $\sigma_i$ as $n/V$-cyclic shifts of a single randomly chosen permutation $\sigma_1$.

Of course, for progressive cross-validation to be relevant, a single progressive validation estimate needs to be as easy to compute as a single holdout estimate. Google ad researchers McMahan, Holt and Scully et al [21] offer a compelling practical example in their work for Google in ad click prediction: “We generally use progressive validation (sometimes called online loss) [5] rather than cross-validation or evaluation on a held out dataset. Because computing a gradient for learning requires computing a prediction anyway, we can cheaply stream those predictions out for subsequent analysis, aggregated hourly.”

This observation holds for any algorithm which is trained in an online fashion, i.e. for which the training update of adding a single point to the training set costs a constant amount of computation. Many algorithms in big data contexts are trained in this way in order to limit their computational time to linear in the data set size. Thus, it is helpful to distinguish the online from the offline validation problem. Both problems are summarized in the boxes below.
The Problem of Validation

Unknown probability distribution $\bar{\rho}$ over $\mathcal{X} \times \mathcal{Y}$.

$n$ samples are drawn independently from $\bar{\rho}$: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Algorithm $A$ trained on data set $S$ outputs predictions $A_S: \mathcal{X} \rightarrow \mathcal{Y}$.

Known loss function $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$.

**Offline setting:** Choose a series of $B$ pairs of training and validation sets $I_1, \ldots, I_B, J_1, \ldots, J_B \subset [n]$, $I_k \cap J_k = \emptyset$. Receive losses

$$\frac{1}{|I_k|} \sum_{j \in J_k} \ell(A_{S_{I_k}}(x_j), y_j),$$

where $S_I := \{(x_i, y_i) : i \in I\}$ for $1 \leq k \leq B$.

**Online setting:** Choose a series of $B$ permutations $\sigma_1, \ldots, \sigma_B$ on $[n]$ and a minimum training set size $t$. Receive losses

$$\ell(A_{S_{\sigma_j^{(1)}, \ldots, \sigma_j^{(k-1)}}}(x_{\sigma_j^{(k)}}, y_{\sigma_j^{(k)}}))$$

for $t < k \leq n$ and $1 \leq j \leq B$.

Goal in both settings is to estimate $E_{(x, y) \sim \bar{\rho}}[\ell(A_{S_{[n]}(x)}, y)]$.

### 3.2 Extrapolative Cross-Validation (XCV)

All of the proposed techniques we've seen so far propose simple averages of the losses provided. However, these will all be inherently biased. After all, if we expect the algorithm is going to improve with more training data, then (3.1) should be a decreasing function of $n$. Yet all of the training sets that we provide the algorithm in our cross-validation estimate are smaller, which means that we will be inherently pessimistic.

The literature to date is primarily focused on this bias-variance tradeoff: We accept that all estimates are biased, and recognize that the larger the training set, the less bias there will be. The tradeoff is that for a given computational budget $B$, the larger the validation sets, the lower the variance of their estimates. Therefore, there is a widely-cited bias-variance tradeoff in classical cross-validation techniques, at least once one limits the computational budget (which might forbid certain techniques like LOO).

However, there is no reason why we must be limited to simple averaging. We can think of an analogy with the algorithm's learning curve, i.e. the expectation of (3.1) over the choice of data sets $S$, seen as a function of $n$. Each (offline) loss estimate is
an unbiased estimator of a particular point on the curve corresponding to the training set size. We are interested in the curve’s value at \( n \), but only have unbiased estimates of its value at various points to the left.

Classical cross-validation techniques choose to estimate the position of one such point on the learning curve as well as they can, and use that as an estimate for the point in question to its right. But if we had estimates of the value of the function at multiple values, we could perform an extrapolation to obtain what we hope would be a nearly unbiased estimator of the value we care about, depending on our assumptions about the shape of the learning curve.

Given the rich continuum of training set sizes such extrapolation requires, this approach is most natural in the context of the online validation problem, so we will first formulate these methods in that context before working to bring it to the original problem. As motivation, we will work out what this looks like in the case of linear extrapolation.

### 3.2.1 Linear Extrapolative Cross-Validation (LXCV)

Suppose we are given function values \( f(0), f(1), \ldots, f(n-1) \) and are asked to estimate \( f(n) \) using linear extrapolation. We first fit a linear curve \( f(x) = mx + b \) to the known values using a least-squares error estimate. Minimizing this least squares estimate corresponds to setting the gradient in the two parameters equal to 0, yielding the following linear equations in \( m \) and \( b \):

\[
0 = \sum_{i=0}^{n-1} 2i(mi + b - f(i))
\]

\[
0 = \sum_{i=0}^{n-1} 2(mi + b - f(i))
\]
Simplifying and using the formula for the sum of consecutive squares, these become

\[
\sum_{i=0}^{n-1} if(i) = \frac{n(n-1)(2n-1)}{6} m + \frac{n(n-1)}{2} b
\]

\[
\sum_{i=0}^{n-1} f(i) = \frac{n(n-1)}{2} m + nb
\]

Solving the system for \(m\) and \(b\), or computing a cleverly-chosen linear combination of these two equations, our estimate for \(f(n)\) is then given by

\[
mn + b = \sum_{i=0}^{n-1} \frac{6i - 2n + 2}{n(n-1)} f(i).
\]

Notice that \(\sum_{i=0}^{n-1} \frac{6i - 2n + 2}{n(n-1)} = \frac{3n(n-1)-2n(n-1)}{n(n-1)} = 1\), so this is an affine combination of the function values \(f(i)\). This is expected, of course, because a vertical shift in the function by a constant should also shift the extrapolation estimate by that same constant. Moreover, the "average" function value lies at \(n\):

\[
\sum_{i=0}^{n-1} \frac{6i - 2n + 2}{n(n-1)} = \frac{n(n-1)(2n-1)}{n(n-1)} - \frac{n(n-1)}{n} = (2n-1) - (n-1) = n.
\]

Note also that these coefficients are linearly varying themselves, from a minimum of \(-2/n\) on \(f(0)\) to a maximum of \(+4/n\) on \(f(n-1)\). Thus, the first third of the function estimates actually contribute negatively to the extrapolation, since a greater function value at those points primarily indicates the function is decreasing more rapidly (or increasing less rapidly).

This then is **linear extrapolative cross-validation**: Generate the table of online losses

\[
\ell_{j,k} := \ell(A_{S_{(x_{j(1)},\ldots,x_{j(k-1)})}}(x_{\sigma_j(k)}), y_{\sigma_j(k)}),
\]

for \(t < k \leq n\) and \(1 \leq j \leq B\), then combine them using these linear extrapolation coefficients:

\[
\frac{1}{B} \sum_{j=1}^{B} \sum_{k=t+1}^{n} \frac{6(k-t-1) - 2(n-t) + 2}{(n-t)(n-t-1)} \ell_{j,k}.
\]
As these weights are both positive and negative, this affine combination allows us to extend beyond averages, weighted or unweighted, which will always have some bias as the weighted average training set size will be less than \( n \). In this case, we can show that the “average” training set size is indeed \( n \), and this is the best way to achieve this condition in a certain sense:

**Proposition 15.** Among vectors of coefficients \((c_0, \ldots, c_{n-1})\) with sum 1 satisfying \( \sum_{i=0}^{n-1} ic_i = n \), the vector corresponding to the linear extrapolation coefficients has the minimum 2-norm.

**Proof.** Let \( p_0, p_1, \ldots, p_{n-1} \) be a series of orthonormal polynomials on \( \{0, 1, \ldots, n-1\} \), i.e. \( \deg p_k \leq k \) and \( \sum_{i=0}^{n-1} p_j(i)p_k(i) = \delta_{j,k} \). Any vector \( c \) of coefficients can then be expanded in this basis as \( c_i = \sum_{k=0}^{n-1} a_k p_k(i) \), so \( \|c\|_2^2 = \sum a_k^2 \). Moreover, the conditions \( \sum c_i = 1 \) and \( \sum ic_i = n \) can be written as dot products with \( p_0 \) and \( p_1 \), so they dictate the values of \( a_0 \) and \( a_1 \) but allow the remaining coefficients to be anything. Therefore, the minimum value of \( \|c\|_2^2 \) occurs when all higher coefficients are 0, hence when \( c \) is the unique linear polynomial satisfying the two constraints, i.e. the linear extrapolation coefficients.

We can compute this 2-norm to be

\[
\sum_{i=0}^{n-1} \left( \frac{6i - 2n + 2}{n(n-1)} \right)^2 = \frac{4n + 2}{n(n-1)} = \frac{4}{n} + \frac{6}{n(n-1)}.
\]

Moreover, a bound on the 2-norm of the coefficients allows us to bound the variance of this estimate around its mean, a “sanity-check lower bound” a la Blum, Kalai and Langford (Theorem 4 in [5]).

**Proposition 16.** Hoeffding-type bounds hold as in progressive cross-validation. In particular if \( L \) is an extrapolation estimate with coefficients \((c_{t+1}, \ldots, c_n)\) and \( EL \) is its mean,

\[
P[L > EL + a] \leq \exp(-2a^2/\|c\|_2^2) \text{ and } P[L < EL - a] \leq \exp(-2a^2/\|c\|_2^2)
\]

64
Proof. As shown in the Proof Sketch of Theorem 4 in [5],

\[ \mathbb{E}[e^{\lambda c_k (\ell_{j,k} - E\ell_{j,k}) |\ell_{j,t+1}, \ldots, \ell_{j,k-1}}] \leq \exp(\lambda^2 c_k^2 / 8). \]

Therefore, in the same manner as their proof,

\[ \mathbb{E}[\exp(\lambda (\hat{L} - L))] = \mathbb{E} \left[ \exp \left( \lambda \sum_{k=t+1}^{n} c_k (\ell_{j,k} - E\ell_{j,k}) \right) \right] \]
\[ \leq \prod \exp(\lambda^2 c_k^2 / 8) = \exp(\lambda^2 \|c\|_2^2 / 8), \]

from which the conclusion follows as in Theorem 4 of [5].

For a simple average, \( \|c\|_2^2 = \frac{1}{n-t} \), so linear extrapolation is at most a constant factor more variable.

We note that this proposition holds for even a single progressive validation estimate, giving no insight into the benefit of averaging results for multiple permutations. In general, any such reasoning for cross-validation procedures (online or offline) is nearly impossible because the correlations between \( \ell_{j,k} \) for different permutations \( \sigma_j \) will greatly depend on the algorithm. For instance, in their attempt to estimate the variance of cross-validation, Nadeau and Bengio [22] require a very strong assumption about the algorithm’s behavior: Its predictions on a single data point outside the training set do not depend on the identity of the training set, only on its size.

### 3.2.2 Other Forms of Extrapolative Cross-Validation

Of course, we don’t necessarily expect that a linear extrapolation will perfectly match the shape of the learning curve. We could use higher order polynomials, but none will capture the typical asymptotic behavior of learning curves, diverging to \( \pm \infty \) rather than converging to a constant as \( n \to \infty \). In our experience, hyperbolic extrapolation also suffers from uncertainty about where to place the vertical asymptote of the hyperbola, a choice which significantly affects the fit.

If the learning curve is convex, i.e. there are diminishing returns for new data,
we would expect that linear extrapolation will tend to underestimate the mean and therefore be overoptimistic. We could then also consider a weighted average of the linear extrapolation and the average, or simply start the linear extrapolation further along the learning curve. This leads to a need to formalize all such linear combinations into what we call weight function notation.

Let \( w : [0, 1] \to \mathbb{R} \) be a real-valued function with \( w(0) = 0 \) and \( w(1) = 1 \). We will define the weighted extrapolation of \( f(0), f(1), \ldots, f(k - 1) \) according to weight function \( w \) as the following estimate:

\[
\sum_{i=0}^{k-1} \left( w \left( \frac{i + 1}{k} \right) - w \left( \frac{i}{k} \right) \right) f(i)
\]

Since \( w(1) = 1 \) and \( w(0) = 0 \), the sum of these weights telescopes to 1, so we are guaranteeing that this is an affine combination. For instance, this encapsulates the following examples we've already cited:

- Leave One Out corresponds to the step function \( w_{\text{loc}}(x) = \delta_{x,1} \), i.e. only the last coefficient is nonzero.

- Progressive Cross-Validation corresponds to a weight function of the form

  \[
  w_{\text{pcv},t}(x) = \begin{cases} 
  0 & \text{if } x < t \\
  \frac{x-t}{1-t} & \text{if } x \geq t,
  \end{cases}
  \]

  with free parameter \( t \) depending on the choice of minimum training set size. Of course, as \( t \to 1 \), this approaches Leave One Out. For convenience, we will call the other extreme of \( t \to 0 \) "Averaging Cross-Validation" or ACV for short. In this limit, the weight function is simply \( w_{\text{acv}}(x) = x \).

- Linear Extrapolative Cross-Validation almost corresponds neatly to a particular function here, with only small \( O(1/n^2) \) errors in the coefficients. Since \( w \) is essentially an integral/sum over the weights, the corresponding weight function is now the unique quadratic with \( w(0) = 0 \), \( w(1) = 1 \), and \( \int_0^1 w(x) \, dx = 1 \).
Solving the system, we find that quadratic is \( w_{\text{xev}}(x) = 3x^2 - 2x \).

- Quadratic Extrapolative Cross-Validation is the corresponding method to Linear Extrapolation using a quadratic fit to the data. It similarly corresponds to the cubic weight function \( w_{\text{qev}}(x) = 10x^3 - 12x^2 + 3x \).

- Any convex combination of these estimates corresponds to the same convex combination of the weight functions.

The advantage of using weight functions is that they remove the dependence on the exact number of data points under consideration. This will be most relevant when we want to evaluate the performance of a method on a different training set size, as we will see in the next section.

### 3.3 Hierarchical Cross-Validation

With such a wide array of cross-validation algorithms at our disposal, the natural question is which one we should use. In the conclusion of their survey, Arlot and Celisse [1] offer this guide: “When the goal is estimation and the signal-to-noise ratio (SNR) is large, the smallest bias usually is the best, which is obtained by taking \( n_t \sim n \). Otherwise, CV can be asymptotically suboptimal. Nevertheless, when the goal is estimation and the SNR is small, keeping a small upward bias for the estimation error often improves the performance, which is obtained by taking \( n_t \sim \kappa n \) with \( \kappa \in (0, 1) \).”

This is not a very surprising observation from the perspective of learning curves. When the SNR is large, the algorithm is learning quickly and therefore the potential for bias by estimating points far from the endpoint is high. Conversely, when the SNR is small, the estimates are noisy and therefore it can be best to average more of the data.

The next question is natural: How can we tell if the SNR is high or low? Or is that even the most parsimonious feature we can extract to tell us which method to use? More generally, if we are interested in extrapolating the losses we are given,
can we assess whether they are varying in a "nearly" linear, quadratic, or some other fashion before choosing the extrapolation method?

Zhang and Yang [34] asked a very similar question in 2015. From their abstract: “While there are various model selection methods, an unanswered but important question is how to select for one of them for data at hand.” Their problem is slightly different in that the model selection criteria they consider are based on regularization according to two information criteria, which are asymptotically identical to cross-validation methods. Their solution is striking, though: Just use cross-validation!

After all, our problem is exactly the type ripe for cross-validation methods: We don’t a priori know which extrapolation method is going to be most accurate, and want to pick the one that does the best job based on the data we have.

But that then raises the same question: Which cross-validation technique should we use to decide which cross-validation technique to use? For their particular choice of model selection criteria, Zhang and Yang found repeated 2-fold cross-validation accurately distinguished between the two possible model selection criteria they were considering. But what about in general?

This line of reasoning leads to a new family of meta-techniques which we call hierarchical cross-validation. If we wanted to be trendy, we’d call it deep cross-validation.

### 3.3.1 Aggregated Hierarchical Cross-Validation

The trouble with defining hierarchical cross-validation is that from this quick description, the computation time would quickly balloon out of proportion. Even if there are just two cross-validation techniques under consideration at each level, the number of problems grows exponentially: Which technique best captures the learning curve of the first? Which best captures the learning curve of the second evaluating the first? And so on. To avoid this exponential complexity, the formulation of hierarchical cross-validation that we will present here involves an aggregation step at each layer.

First, we will present the online version of hierarchical cross-validation, which is already naturally set up for this purpose. In Section 3.5, we will discuss how to use
these ideas in an offline context.

The input to the algorithm is the table of losses $\ell_{j,k}$ for $t < k \leq n$ and $1 \leq j \leq B$. The first step is to average over the permutations $\sigma_j$ to combine all of the predictions into a single vector which we'll call $\ell_k := E_j \ell_{j,k}$. To properly aggregate over weight functions, we will also require a measure $\mu$ on weight functions $w$. We will consider the support of $\mu$ to be the set of possible weight functions we are considering. For instance, for a discrete set of weight function options, $\mu$ could be the uniform measure, or weighted (ideally in some easily computable sense) towards options which we suspect will do better.

In some sense, estimating (3.1) amounts to predicting the expected value of $\ell_{n+1}$. Therefore, the best cross-validation-type analogy we can draw is how effectively a given weight function approximates $\ell_k$ given access to previous estimates $\ell_{t+1}, \ldots, \ell_{k-1}$. We compute this estimate for each weight function and compare it to the actual weight, scoring it with, say, squared loss. Then we can aggregate over weight functions to produce a single table with entries which we'll call $\ell'_k$.

We repeat this process $D$ times, where $D$ refers to the depth (height) of the hierarchy we are building. Then we descend, picking the best weight function for the next layer in the hierarchy by using the weight function from the previous layer. We get this second chain started with an arbitrary choice of weight function; its effect should be diminished the more layers we have.

That's the basic gist of the algorithm; here is the full-fledged recursion of Aggregated Hierarchical Cross-Validation (AHCV):

- The base case of the recursion is the loss table $\ell_{k}^{0} = (E_j \ell_{j,k})_{k=t+1}^{n}$.
- First, we define the estimates

  $$\hat{\ell}_k^{(d)}(w) := \sum_{i=t+d+1}^{k-1} \left( w \left( \frac{i - t - d}{k - t - d} \right) - w \left( \frac{i - t - d - 1}{k - t - d} \right) \right) \ell_i^{(t)}$$

  These estimates are based on the $\ell_i^{(d)}$ with $i < k$.
- Next, we define the error of such estimates, which we will take to be squared
error:
\[ \tilde{E}_k^{(t)}(w) := (\tilde{E}_k^{(t)} - \hat{E}_k^{(t)})^2. \]

- Finally, we will aggregate these errors according to the measure \( \mu \):
\[ E_k^{(t+1)} := E_{w \sim \mu} \tilde{E}_k^{(t)}(w). \]

Note that since we lack any prediction for the \( \ell_{t+1} \), this table is slightly smaller than the previous layer.

- This recursive process is then repeated \( D \) times.

- As the base case of the descending recursion, we first pick an arbitrary weight function \( w_{D+1} \) to evaluate the \( D \)th layer. By default, this is the average of all weight functions according to measure \( \mu \):
\[ w_{D+1}(x) = E_{w \sim \mu} w(x). \]

- To descend, we pick weight function \( w_d \) at layer \( d \) as graded by \( w_{d+1} \):
\[ w_d := \arg \min_{w \in \text{supp } \mu} \sum_{i=t+d+1}^n \left( w_{d+1} \left( \frac{i - t - d}{n - t - d} \right) - w_{d+1} \left( \frac{i - t - d - 1}{n - t - d} \right) \right) \ell_i(w). \]

- Finally, we output the estimate provided by \( w_0 \):
\[ \sum_{i=t+1}^n \left( w_0 \left( \frac{i - t}{n - t} \right) - w_0 \left( \frac{i - t - 1}{n - t} \right) \right) \ell_i(w). \]

3.3.2 An Analogy

If it is helpful, we can think of this in analogy to an overly bureaucratized education system. First, imagine the goal is to evaluate a single student’s performance, but in a way that will generalize to many types of students, those who learn quickly or slowly, and so on. All we have to work on is the student’s test scores.

How do we combine those scores to evaluate the student? What we want to predict is how they’ll do on the next test. Do we just take the last one (leave one out)? Maybe
we try to extrapolate? Or perhaps we just average them all? Let’s imagine that each of these options corresponds to a different grader’s opinion.

Well, we can evaluate which of these graders is correct by seeing how each of them predict the student’s later scores from their earlier ones. Again, the goal is to predict the grader’s error in predicting the student’s next test score. A teacher can then do this, grading each of the graders on how well they predict the student’s scores. The corresponding question arises: How do we aggregate the scores of a grader? How do we weigh accuracy for the student’s earlier scores versus accuracy of the student’s score on the final test?

When it comes to evaluating a teacher, though, the challenge is a little different. They each can be asked to predict the graders’ errors, but then the best teacher might depend on which grader they’re asked to evaluate. To prevent this exponential expansion of the question, we instead seek to find the best teacher for the average grader as according to the provided measure $\mu$.

Evaluating the average grader is the same sort of problem as evaluating the student in the first place, so we still have the problem that different teachers have different opinions on how to weigh graders’ performances at the beginning of the year versus the end, so we need a department head to sort out which teacher is best. And again, different department heads might have different opinions on how to grade the teachers, so we need an assorted list of administrators all tasked with evaluating the level below them.

In the end, the choice at the top of the hierarchy must be arbitrary, and we hope that its effect will be minimal on the lowest levels. We expect that at some level, some teacher or department head or administrator should be everyone’s favorite, which means that the higher levels won’t matter. But if we’re uncertain when that’ll happen, we can always add another layer to the bureaucracy...

In any case, once we’ve picked the top level administrators, they hire the next level administrator of their choosing, down to the department head, who picks the best teacher in their estimation, the teacher can pick the best grader, and we output that grader’s estimate of the student’s ability.
3.3.3 Other Layered Cross-Validation Methods in the Literature

As computation time has increased, the cross-validation literature has warmed to the possibility of running multiple layers of cross-validation. For instance, in 2006, Varma and Simon [31] noticed that when cross-validation is used to select the best of a number of algorithms, the cross-validation estimate of that algorithm’s performance tends to be optimistic, a version of the “winner’s curse.” They propose a two-layer method they call “nested CV”: First use a subset of the data to do the selection problem, then evaluate the result on a holdout set, and repeat.

This is a very different approach to this form of cross-validation. Later authors would in fact debate whether this technique is too computationally expensive, as Bernau, Augustin, and Boulesteix [3] would argue, or whether the advent of cloud computing would make it worth it, as Krstajic, Buturovic, Leahy, and Thomas [19] would contend.

By contrast, the hierarchical technique we have presented here does not actually require any more accesses to the algorithm than progressive cross-validation. It merely decides how to extrapolate from those accesses in a hierarchical fashion, repeating a series of basic computations, not algorithm training sessions. This allows us to propose such a method even when we treat the algorithm as a black box, an important criterion for reliable validation.

3.4 Empirical Online Cross-Validation Results

To illustrate the usefulness of these new methods, we first developed a new way to visualize their effectiveness called a bias-variance plot. As the name suggests, the goal is to plot the bias against the variance (or, technically, the standard deviation, so it is in the same units) in order to compare the possible tradeoffs and visualize where the components of the mean squared error are coming from.

Mathematically, what this means is that we first generate a large number of in-
stances, i.e. data sets drawn from a particular distribution. For each data set, we measure the true error of the algorithm trained on the data by testing against a much larger holdout set drawn from the same distribution. Then we compute the estimates from whatever validation methods we want to compare. The error of each estimate is then given by the estimate minus the true value.

This gives us an error (positive if it is over-pessimistic and negative if it is over-optimistic) for each method on each instance. The average of these errors is the empirical estimate of the bias of that method, and the variance is, well, the variance. We plot the bias against the standard deviation of each method in a bias-variance plot.

The accuracy goal of any of these methods is to minimize mean squared error, which means minimizing the Euclidean distance to the origin of the plot. To help visualize this, we also plot the origin (corresponding to the truth), and require that the two axes are on the same scale.

We additionally compute uncertainty estimates of each of these parameters as described in Appendix D. Without any further ado, here are some examples of bias-variance plots to illustrate the performance of our algorithms.

As an easy introduction, we illustrate that different methods are best for different shapes of learning curves. In Figure 3-1, we see the bias-variance plot for two different instances illustrating steep and nearly flat learning curves. Details of the models, algorithms and parameters are given in Appendix D.

In both instances, averaging CV is the most pessimistic, while (linear) extrapolative CV is the most optimistic. However, we can see that the effect of the variance is very different. In the case of a nearly flat learning curve, which also has the smallest bias, ACV has the lowest variance and nearly almost the lowest mean squared error. This reflects the fact that the learning curve has already nearly converged to its final value, and we want to average as many of the losses as we can. Meanwhile, in the steep and shallow learning curves, extrapolative cross-validation has the lowest variance, but leave-one-out is both nearly unbiased and fairly low variance.

When we reduce the budget, though, we see that looking at only the last data point
Figure 3-1: Comparing Leave One Out (LOO), Linear Extrapolative Cross-Validation (XCV), and Averaging Cross-Validation (ACV) and interpolations between them, assuming a large computational budget, on examples with steep (left) and nearly flat (right) learning curves.

Figure 3-2: Comparing Leave One Out (LOO), Linear Extrapolative Cross-Validation (XCV), and Averaging Cross-Validation (ACV) and interpolations between them, assuming a small computational budget, on examples with steep (left) and nearly flat (right) learning curves.
(our low-budget version of Leave One Out) becomes much less reliable, as illustrated in Figure 3-2. Meanwhile, the extrapolative and averaging methods are not much worse with a smaller budget.

Given that linear extrapolative cross-validation (XCV) is overoptimistic in every example we've seen, we might wonder whether quadratic extrapolation would work better. Figure 3-3 illustrates the performance of QXCV along with ACV and XCV and weighted averages of pairs on the same problems as in Figures 3-1 and 3-2. Indeed, QXCV occupies a similar space to LOO, generally unbiased but with a much more gracious decay in variance with small computational budgets.

Among the pure new algorithms introduced, QXCV generally performs best. In all of these instances, though, the best performer is actually not an endpoint, but is some interpolation or weighted average of multiple endpoints. This is somewhat surprising; even in these hypothetical examples we've rigged up to favor one or the other endpoint, hedging slightly against it seems to work even better.

Finally, we illustrate in Figure 3-4 how aggregated hierarchical cross-validation treats each of these four scenarios. For simplicity, we use the same interpolations of linear and quadratic extrapolation and averaging CV as in Figure 3-3. We can see that while AHCV tends to lie in the interior of the methods it is aggregating, it is generally unbiased. The variance also seems to improve with a larger computational budget, similar to QXCV.

However, these results are not the most promising endorsement of AHCV. Ideally, the chosen point would be near the optimum in the set of options every time, or at least would tend to vary in that direction. Looking at the upper left and lower right bias-variance plots, we see some evidence of this trend, but the other plots show results closer to the middle of all of the results, as if the hierarchy is choosing more or less randomly. This indicates that there is a bit of a disconnect between the assumed goal of extrapolating to the “next point” and the core problem of validation. Whatever that disconnect is, future work should illuminate the best way to set up this hierarchy.
Figure 3-3: Comparing Quadratic Extrapolative Cross-Validation (QXCV), Linear Extrapolative Cross-Validation (XCV), and Averaging Cross-Validation (ACV) and interpolations between them, assuming a large (top) and small (bottom) computational budget, on examples with steep (left) and nearly flat (right) learning curves.
Figure 3-4: Comparing Aggregated Hierarchical (AHCV) on Quadratic Extrapolative Cross-Validation (QXCV), Linear Extrapolative Cross-Validation (XCV), and Averaging Cross-Validation (ACV) and interpolations between them. All hierarchies are to depth 5, as the choice seems to stabilize by that point. In all instances, AHCV is the red dot that doesn’t match the color of the surrounding points.
3.5 Using Extrapolative and Hierarchical CV for Offline Validation

While both extrapolative and hierarchical cross-validation can be better formulated in an online framework, both families of techniques can also be used in the offline context. In this section, we will describe how that might be done.

In the online context, we had access to the algorithm trained on every data point before the one in question, the entire prefix of that particular permutation. When training is not online, we have to instead only train on certain prefixes to achieve the same style of data. In other words, we have to partition our data into folds.

In $V$-fold extrapolative CV, the data is first partitioned into $V$ folds of nearly even size $n/V$ as in VFCV. Instead of training on all but one of the folds, though, we train the algorithm on the first $k$ folds and then test on the $k + 1$th fold, for $k = 1, \ldots, V - 1$. Averaging the losses on the data points in each fold creates a series of loss averages that we can then extrapolate in any of the ways previously described, including hierarchical methods. If the computational budget allows, we can then repeat this procedure for multiple permutations of the folds.

3.6 Conclusions

We have presented two novel families of techniques for solving the problem of validation: extrapolative methods, and hierarchical methods. The former aims to avoid the inherent bias-variance tradeoff in classical cross-validation methods by extrapolating the performance of the algorithm on training sets of different sizes. The latter aims to solve the meta-problem of deciding which cross-validation method to use by applying the same data-driven philosophy that leads us to choose cross-validation in the first place.

In our simulations, linear extrapolation proved to be too optimistic, which is unsurprising given the frequent presence of diminishing returns on additional training data. Quadratic extrapolation, meanwhile, was nearly unbiased, behaving somewhat
similarly to Leave One Out, but only requires a smaller computational budget. Hierarchical methods also tended to be nearly unbiased but suffered from somewhat higher variance, especially when the computational budget is small.

Both extrapolative and hierarchical methods are more naturally presented in the online learning context, and all of our empirical work is done there. Future empirical work can characterize how those techniques do in the offline context.
Appendix A

The Gaussian Example with a Prior

To illustrate the improvement of this method, we reexamine the Gaussian example from DFHPRR. Recall that the setup of this problem involves a spherical $d$-dimensional Gaussian with known variance but unknown mean, and the analyst is seeking to find a direction in which the curator will give inaccurate answers.

To translate it into a Bayesian framework, we need a prior over the mean, since we can no longer make it known to the analyst but not the curator. For simplicity in computation, suppose the true center $c$ of the distribution is distributed as $N(0, \sigma I_d)$ for some $\sigma > 0$, and each data point is known to be generated from adding $N(0, I_d)$ to $c$. This normalization is appropriate, since dot products with unit vectors will likely deviate from the true mean by a constant, so only truncation of a constant fraction of the space is necessary to make such queries fall within $[0, 1]$. Therefore, up to constants, we can assume that querying the dot product with any unit vector is allowed.

In this context, $\sigma = 0$ corresponds to the mean being completely known (at the origin), and large $\sigma^2 \gg 1/n$ corresponds to a widely diffused prior for which the data will be needed to clarify the position. As a sum of Gaussians, it is easy to compute that the empirical mean $\hat{c}$ will be distributed as $N(0, (\sigma^2 + \frac{1}{n})I_d)$. For notational convenience, choose a new basis for $\mathbb{R}^d$ so that $\hat{c} = te_1$.

Updating to the posterior, another easy computation shows that the probability
density function for the center \(c\) is proportional to

\[
\exp \left( -\frac{\|c\|^2}{2\sigma^2} - \frac{n\|c - \hat{c}\|^2}{2} \right)
\]

\[
\propto \exp \left( -\frac{n + 1/\sigma^2}{2} \left( c_1 - \frac{nt}{n + 1/\sigma^2} \right)^2 - \frac{n + 1/\sigma^2}{2} \sum_{i>1} c_i^2 \right)
\]

\[
= \exp \left( -\frac{n + 1/\sigma^2}{2} \left\| c - \frac{n}{n + 1/\sigma^2} \hat{c} \right\|^2 \right).
\]

The posterior mean will then answer all queries for the mean with the center of this distribution, \(\frac{n}{n + 1/\sigma^2} \hat{c}\). Because this is now a Gaussian with variance \(\frac{1}{n+1/\sigma^2}\), the probability of \(\epsilon\) error in any query direction is \(\sim \exp(-\epsilon^2(n+1/\sigma^2)/2) < \exp(-n\epsilon^2/2)\), matching the static bound (2.1) up to constants.

In other words, all of the difficulty in the Gaussian example was due to the information asymmetry. This is most clear when we vary \(\sigma\). As \(\sigma \to \infty\), the posterior mean approaches the empirical mean, but the variance of the posterior is still bounded by \(1/n\), so it remains accurate. As \(\sigma \to 0\), the posterior mean takes less account of the data and approaches the origin. The original example corresponds to \(\sigma \to \infty\) for the curator’s prior while \(\sigma \to 0\) for the analyst’s prior, maximally asymmetric.

The same conclusion holds for the interactive fingerprinting attack, because the analyst no longer knows which fingerprints of data points to look for. The essential query in that problem relied on the data points that the analyst knew were possible but the curator didn’t, and with information symmetry, there are no such data points.

We are left with only the basic lower bound examples we considered in the static case, which do translate nicely into this setting, since that analyst’s queries don’t require knowledge of the distribution. For the \(\epsilon\)-biased coin, we can consider the prior to be uniform on the two cases, and similarly for \(q\) copies of it. In these cases, it is still necessary to see enough data to reliably distinguish probabilities of \(\frac{1}{2} \pm \epsilon\) from each other, so (2.1) still holds.

Is that really enough, though? This is the key question: Under information symmetry, can the static bound (2.1) be achieved for adaptive queries? If not, what new attacks can the analyst employ, and what bounds do those place on query complexity?
Appendix B

Additional Results Related to Adaptive Challenges

B.1 Boosting Attack on Independent Classification

The boosting attack of Blum and Hardt [4] takes place in a classification setting, so we will be working with the notation of Section 2.3.1, with the marginal \( r \) uniform over \( \mathcal{Y} \) and \( Z = \mathbb{F}_2 \).

Rather than considering linear or polynomial label functions, we will suppose that all possible label functions \( f_i : \mathcal{Y} \to \{0, 1\} \) are equally likely. This is a somewhat trivial "learning" scenario, since the curator only learns the value of the function at each point he sees, and nothing more. Still, it illustrates an important difference between the posterior mean and empirical mean when it comes to boosting and reconstruction-style attacks.

The attack as described in [4] works as follows: For the first \( q - 1 \) queries, the analyst picks random functions \( f'_i : \mathcal{Y} \to \{0, 1\} \) and asks for their correlation with the true labeling function. To write this in the notation of statistical queries, her \( i \)th query function is the indicator function on the graph of \( f'_i : f(y, z) = \delta_{z,f'_i(y)} \).

The expected value of this function is therefore the probability that \( f'_i \) and the true labeling function \( f \) agree on a uniformly random chosen \( y \in \mathcal{Y} \).

The analyst then collects all queries with agreement greater than \( 1/2 \), according
to the curator. She defines a new query by taking the majority label among all of
these queries, which will be biased to agree with $\ell$ more frequently on the points that
the curator has seen.

Suppose that $k$ of the $n$ data points have labels that match $\ell'$. The empirical mean
will simply answer the query with that fraction, $k/n$. The posterior mean is more
subtle, though: It averages the answers over all functions consistent with the data.
On the $n$ data points that he’s seen, the average is the same $k/n$, but on all $|\mathcal{Y}| - n$
data points he hasn’t, the average agreement is just 1/2. Therefore, the posterior
mean regularizes the empirical mean’s answer back towards the prior, answering with

$$\frac{1}{2} \frac{|\mathcal{Y}| - n}{|\mathcal{Y}|} + \frac{k}{n} \frac{n}{|\mathcal{Y}|} = \frac{1}{2} + \frac{2k - n}{2|\mathcal{Y}|}.$$

We can actually easily verify that the posterior mean makes the appropriate in-
fERENCE against random queries. The true agreement between the random $\ell'$ and the
true label function on unseen points is simply a rescaled binomial random variable
with mean equal to the posterior mean and variance $\frac{|\mathcal{Y}| - n}{|\mathcal{Y}|^2} < \frac{1}{n}$. By Hoeffding’s in-
equality, this differs by $\epsilon$ from its mean with probability $\sim \exp(-\Omega(ne^2))$, satisfying
the desired bound.

The empirical mean’s answer is also accurate against the random queries, but the
boosting attack distinguishes them. Both answers to the random queries fall on the
same side of 1/2, so the attack constructs the same biased query for each of them.
But when the empirical mean answers with a correlation that is $\sqrt{\frac{2}{n}}$ too high, the
posterior mean only goes up by $\sqrt{\frac{2}{n}} \frac{n}{|\mathcal{Y}|} = \frac{\sqrt{n}}{|\mathcal{Y}|}$, much less.

In fact, this is the correct increase. The above concentration argument for the
posterior mean’s accuracy actually applies to any query that is uncorrelated with the
true labels off of the data known to the curator. It is easy to see that this holds for
the biased query, and in fact, any query that the analyst or curator could construct,
since neither knows anything about the true labels off of the known data.

This is a subtler use of information asymmetry than the usual case, since the
boosting attack makes no reference to the true labels. Instead, the success of the
attack depends on the true support $|X|$ being significantly larger than $n$, a publicly known fact which the empirical mean curator ignores. Once the curator is allowed to take the support size into account and regularize towards $1/2$, he no longer makes this type of mistake.

As a further example, we could consider what would happen if the support size is instead drawn from some nontrivial prior. This is somewhat similar to the classical problem of estimating support size of a distribution (see e.g. [28]). However, the curator does not need to accurately estimate $|Y|$, but rather, $1/|Y|$ to an accuracy of roughly $1/n$.

In particular, if all of the data points are unique, then the curator probably (depending on his prior) learns that $|Y|$ is likely much larger than $n$. This makes his estimate for $1/|Y|$ negligible, so he will answer (very close to) the prior mean of $1/2$ for every query, even the boosted ones. This illustrates how important information present in the data can be ignored by the empirical mean, a deficiency that the boosting attack exploits.

While a Bayesian curator takes this into account perfectly, this also indicates morally that other regularization techniques would also prevent this sort of "learning." Indeed, in this circumstance, it is clear that any sort of cross-validation would properly evaluate this boosting algorithm if it attempted to learn in this fashion on the training data. Fortunately, many researchers already apply such techniques to keep themselves from overfitting.

B.2 Adding Gaussian Noise Beats Linear Classification

In this section, we prove the tangential claim that the noisy posterior mean curator strategy can successfully answer exponentially queries under the linear classification model. For clarity, we fix the variance of the noise at $\frac{1}{4n}$.

**Theorem 17.** Under model $LC_n$, the noisy posterior mean curator strategy can an-
swer \( q \) queries \( \epsilon \)-accurately using \( n = O \left( \frac{1}{\epsilon^2} \log \frac{2}{\delta} \right) \) samples.

The main idea of the proof is that the noisy posterior actually answers sufficiently similarly to the prior, and the prior answers sufficiently similarly to the true answer on almost every query. We first show the second claim via this lemma:

**Lemma 18.** Under model \( LC_m \), the Prior Mean curator strategy will answer any query \( \epsilon \)-accurately with probability at most \( 1 - \frac{2^{-m}}{4\epsilon^2} \).

**Proof.** The Prior Mean strategy is simple: by symmetry, it puts equal weights of \( 2^{-m-1} \) on every point in \( \mathbb{F}_2^m \times \mathbb{F}_2 \), and answers the average value of the function according to those weights. Abusing notation, we call the prior mean value \( \mathbb{E}_p(f) \).

To bound the difference of the real answer and the prior mean, define a new function \( f' : \mathbb{F}_2^m \rightarrow [-1, 1] \) by \( f'(y) = f(y, 0) - f(y, 1) \). Then we can rewrite

\[
E_{\mathbb{P}_j}(f) - E_p(f) = \frac{1}{2m+1} \sum_{y \in \mathbb{F}_2^m} \left( f(y, \ell_j(y)) - \frac{1}{2}(f(y, 0) + f(y, 1)) \right)
= \frac{1}{2m+1} \sum_{y \in \mathbb{F}_2^m} f'(y)(-1)^{\ell_j(y)}.
\]

The terms in this sum are independent, because for any two points \( y, y' \in \mathbb{F}_2^m \), the ordered pair \((\ell_j(y), \ell_j(y'))\) is equidistributed among \( \mathbb{F}_2^2 \) as \( \ell_j \) over all linear functions. So the variance of this deviation is the sum of the variances of the individual terms, or

\[
\text{Var}(E_{\mathbb{P}_j}(f) - E_p(f)) = \frac{1}{2^{2m+2}} \sum_{y \in \mathbb{F}_2^m} f'(y)^2 \leq \frac{2^m}{2^{2m+2}} = \frac{2^{-m}}{4}.
\]

The lemma follows immediately from Chebyshev’s inequality.

The rest of the proof involves the somewhat counterintuitive step of analyzing how the posterior of the analyst evolves, knowing the curator is using the noisy posterior mean algorithm.

**Proof of Theorem 17.** First, if \( n > 2m \), with probability \( 1 - 2^{-\Omega(n)} \), the curator will have enough data to completely determine the hypothesis (it just takes \( m + 1 \) novel points, and each point is novel with probability \( > 1/2 \)). In that case, the error of the
noisy posterior will simply be the Gaussian noise added, which is designed to be of size at most $\epsilon$ with probability at least $1 - \frac{\delta}{q}$, as desired.

So we may suppose $n \leq 2m$. The statement then amounts to showing that the analyst answers correctly on every query, except with probability $q \exp(-\Omega(m^2)) > q \cdot 2^{-\Omega(m)}$.

We wish to understand the analyst's knowledge of the distribution via her posterior. Just like the curator, the analyst's prior in linear classification is uniform on all $2^{m+1}$ possible linear functions. We will inductively prove the following carefully-calibrated claims:

**Lemma 19.** Let $\delta' = 2qm2^{-m/2}$. After $k$ queries for $k \leq q \leq 1/3\delta'$, with probability $1 - O(k^2\delta')$, (i) the curator's answers on all queries are correct, and (ii) the analyst's posterior puts weights within $\exp(\pm k/q)/2^{m+1}$ on each hypothesis, except for a $k\delta'^2$ fraction of the hypotheses, which themselves have at most a total weight of $k\delta'$.

**Proof.** We show this by induction on $k$. From the analyst's perspective, to answer a query, a random hypothesis according to the posterior is chosen, the curator receives $n$ data points from that posterior, answers the query according to the data, and then all of the hypothesis weights are adjusted accordingly. For simplicity, we will actually assume that the curator answers according to a single hypothesis, and argue that the curator's answers are only more true if he answers with the average of several hypotheses.

First, we label as "k-bad" all hypotheses that give answers more than $\epsilon' = \frac{1}{4qm} \ll \epsilon$ away from the mean on the $k$th query, and as "k-good" any hypothesis that is not $l$-bad for any $l \leq k$. We already know there won't be many $k$-bad hypotheses: By Lemma 18, a $\frac{2^{-m}}{4q^2} = 4q^2m^22^{-m} = \delta'^2$-fraction of the hypotheses are $k$-bad, so all but a $k\delta'^2$ fraction of the hypotheses will be $k$-good, as desired. We now must show that the $k$-good hypotheses maintain approximately similar weights, while bad hypotheses have small total weight, with high probability.

We then condition on the randomly chosen hypothesis (in the analyst's model of how the curator works) being $(k - 1)$-good, which adds $(k - 1)\delta'$ to the error
probability by the inductive hypothesis. Then we condition on it not being $k$-bad, which adds at most $\exp((k - 1)/q)\delta^2 < 4\delta^2 < \delta'$ to the error probability, again by the induction hypothesis's upper bound on all of the $(k - 1)$-good hypotheses' weights. In all, this adds only $k\delta'$ to the error probability, which is fine since we have $O(k^2 - (k - 1)^2)\delta' = O(k)\delta'$ extra error probability to work with. So we may assume that the true hypothesis's answer $a$ is within $\epsilon'$ of the mean $\mu$ on the query. Moreover, with the same error probability, the curator's estimate is also within $\epsilon'$ of the mean $\mu$ on the query,¹ so the curator's unnoised answer is $2\epsilon' < 1/qn < 1/2\sqrt{n} < \epsilon/2$-accurate. Since the noise is calibrated to be less than $\epsilon/2$ with probability $1 - \delta/2q$, this means that the curator's answers are accurate with the desired probability, satisfying (i).

The analyst then updates her beliefs according to this noised answer. Since the curator adds Gaussian noise with variance $1/4n > 1/2m$, the likelihood density of answering with $a'$ is proportional to $\exp(-m(a - a')^2)$, where $a$ is the answer of the true hypothesis.² By Bayes' rule, this is the factor we multiply the weights on hypotheses with answer $a$, before renormalizing. If $a'$ is outside of the range $[\mu - \epsilon', \mu + \epsilon']$, the largest and smallest weights to good hypotheses will occur on the ends of the interval, so the likelihood ratio of any pair of good hypotheses is bounded by

$$\exp|m(\mu + \epsilon' - a')^2 - m(\mu - \epsilon' - a')^2| = \exp(4me'|\mu - a|) \leq \exp(2me') = \exp(1/2q),$$

where we have simply bounded $|mu - a'| < \epsilon < 1/2$ with very high probability.³ If $a' \in [\mu - \epsilon', \mu + \epsilon']$, the smallest weights, near 1, will fall on any hypotheses with answers close to $a'$, so the likelihood ratio will be bounded by the weight at an endpoint: $\exp(m(2\epsilon')^2) < \exp(2me') = \exp(1/2q)$ as well.

¹In the real scenario where the curator answers according to multiple hypotheses, we have a little more work to do. Digging into the variance-based proof of Lemma 18 would show that averages of hypotheses are even more likely to fall close to the mean than individual hypotheses, and therefore, the unnoised answers of the curator will be $k$-good.

²The coefficient of $m$ in this expression is possibly lower, but this will only make the weights more similar to each other.

³For simplicity, we are analyzing a version of the noisy posterior mean algorithm that does not truncate all answers to fall within $[0, 1]$. However, it is clear that such truncation only reduces the information that the analyst receives.
Therefore, all of the good hypotheses maintain approximately the same weights as each other: No pair can have a likelihood ratio greater than \( \exp(1/2q) \). To show that this means all good hypotheses indeed have weights near their original values, we must bound the total weight on bad hypotheses.

The argument here is a little different. When the new \( l \)-bad (i.e. also \( l-1 \)-good) hypotheses are determined by the choice of the \( l \)th query, they collectively have weight at most \( \leq \exp(1/q)\delta'^2 < 4\delta'^2 \). Now, since this is a correct posterior, those hypothesis weights are also equal to the expectation of their weights after any future data. So by Markov’s inequality, after the \( k \)th query, the new \( l \)-bad hypotheses have weight at most \( \delta' \) with probability \( 1 - 4\delta' \), and collectively, by a union bound, all \( l \)-bad hypotheses for \( l \leq k \) have weight at most \( k\delta' \) with probability \( 1 - 4k\delta' \). Since we again have error probability \( O(k\delta') \) to spare, we have proved the last claim of the lemma.

Finally, since the bad hypotheses have total weight at most \( k\delta' \), the average weight on the \( k \)-good hypotheses is between \( \frac{1}{1-k\delta'^2} \) and \( 1-k\delta' \) times their original weight \((2^{-m-1})\). But \( 1-k\delta'^2 > 1-k\delta' \geq 1-k/3q > \exp(-k/2q) \), using the assumed bound on \( q \), so the average weight of the \( k \)-good hypotheses is within \( \exp(\pm k/2q) \) of their original weight. Since no pair of \( k \)-good hypotheses has weight ratio exceeding \( \exp(k/2q) \), this means that every \( k \)-good hypothesis has a weight within \( \exp(\pm k/q) \), as desired.

Taking \( k = q \), then, the curator is \( \epsilon \)-accurate with probability at least \( 1 - O(q^2\delta') = 1 - O(q^3m2^{-m/2}) \). (Clearly this is meaningless unless \( q < 1/3\delta' \).) Since \( m \leq O(2^{m/6}) \), we can also write this as \( 1 - O(q^32^{-m/3}) \). Therefore, when \( q \leq O(2^{m/9}\delta^{1/3}) \), the error probability is at most \( \delta \). Translating this in terms of \( n \leq 2m \), we have shown that the curator wins if \( n > 18\log q + 6\log \Omega(1/\delta) = \Omega(\log q/\delta) \). Therefore, if \( n \geq \Omega \left( \frac{1}{2^q} \log \frac{3}{\delta} \right) \), the noise is small and the curator is accurate, as desired.

One can see this result as a Bayesian version of the classical result that statistical queries are unable to learn parities in [17].
B.3 Noise Against Polynomial Classification

In this section, we describe an adaptive analyst attack against the noisy posterior mean curator on the polynomial classification problem $PC_{m,k}$ of Section 2.3.1. This attack illustrates a simple problem on which the noisy posterior mean fails to achieve the static bound (2.1), but the parameters and class of curator algorithms defeated by the augmented Justesen code attack of Section 2.3.3 are both much stronger.

In the context of polynomial classification, $n = \binom{m}{\leq k} = \Omega(m^k)$. We will describe an analyst attack that only uses $2^{O(m)}$ queries. Since this is far less than $\exp(n) = \exp(\Omega(m^k))$, this shows that the noisy posterior mean curator algorithm falls short of the static bound (2.1) here.

The analyst first repeatedly queries the indicator function on one individual point $(y, z)$. Recall that the posterior mean puts a weight $0, 2^{-m-1},$ or $2^{-m}$ on that point according to whether the curator knows that $\ell(y) = z$ or $\ell(y) \neq z$. Noise with variance $\frac{1}{n}$ ordinarily drowns out this signal, but if the analyst asks $2^{2m}$ times, the average will have noise variance $\frac{1}{2^{2m}n}$, making these gaps about $\sqrt{n}$ standard deviations apart. Therefore, the average of these repeated queries will tell the analyst what the curator knows about $\ell(y)$ with probability $1 - \exp(-n)$. Repeating this for all $y \in \mathcal{Y}$ only takes $2^{3m}$ queries in all, from which the analyst can recreate everything the curator knows, and can find that query on which he will be confused.

Taking $k$ as small as possible, we see that the analyst can answer at most $q = \exp(O(n^{1/\log_{1/2}(2e)}))$ queries, or equivalently, $n = \log^{\log_{1/2}(2e)} q$ data points are necessary to answer $q$ queries. To wrap all of this up, we have just shown:

**Theorem 20.** Under $PC_{m,k}$, where $n \sim \binom{m}{k}$, there is an adaptive analyst strategy which causes the noisy posterior mean curator strategy to answer $\epsilon$-inaccurately with probability at least $\frac{\epsilon}{2n}$, using only $2^{O\left(n^{1/\log_{1/2}(2e)}\right)}$ queries.
B.4 Smart Rounding

In this appendix, we prove Lemma 4, and then we show that this smart rounding defeats linear classification.

Proof of Lemma 4. First, we show it suffices to prove a single point version of this, which is simple enough that it may be of independent interest:

**Lemma 21.** For any distribution $D$ on $[0, 1]$, there exists a point $x \in (0, 1)$ such that $\forall \eta > 0,$

$$\mathbb{P}_D[[x - \eta, x]], \mathbb{P}_D[[x, x + \eta]] \leq 2\eta.$$

**Remark.** This lemma is tight: If $D$ is identically equal to $1/2$, then taking $\eta = |\frac{1}{2} - x|$ makes one of the terms on the left equal to 1 and $2\eta \leq 1$.

For $i = 1, \ldots, m-1$, we will pick $x_i \in I_i (\frac{2i}{3} \epsilon, \frac{2i+1}{3} \epsilon)$, so $x_{i+1} - x_i > (\frac{2i+2}{3} - \frac{2i+1}{3}) \epsilon = \frac{\epsilon}{3}$ and $x_{i+1} - x_i < (\frac{2i+3}{3} - \frac{2i}{3}) \epsilon = \epsilon$. Define another distribution $D'$ as follows: For each $i$, move all of the weight from $[\frac{4i-1}{6} \epsilon, \frac{4i}{3} \epsilon]$ to the point $\frac{2i}{3} \epsilon$, and all of the weight from $[\frac{2i+1}{3}, \frac{4i+3}{6} \epsilon]$ to $\frac{2i+1}{3} \epsilon$. Since the statement of Lemma 4 is trivial for $\eta \geq \frac{5}{6}$, weight in those intervals is too far from $x_{i-1}$ or $x_{i+1}$ to be included in either term, so only its distance from $x_i$ matters. Moreover, we've decreased that distance, so it suffices to prove the result on $D'$ instead of $D$, under the constraint that each $x_i \in I_i$.

Assuming that $D'$ puts positive weight in $I_i$ (otherwise we can ignore that interval), consider $D'$ restricted to $I_i$, and rescaled by a factor of $3/\epsilon$ to become an interval on $[0, 1]$; call this new distribution $D'_i$. By Lemma 21, there exists some $x' \in [0, 1]$ such that $\forall \eta > 0,$

$$\mathbb{P}_{D'_i}[(x' - \eta, x')], \mathbb{P}_{D'_i}[(x', x' + \eta)] < 2\eta$$

$$\mathbb{P}_{D'_i}[(x - \eta \epsilon/3, x)], \mathbb{P}_{D'_i}[(x, x + \eta \epsilon/3)] < 2\eta \mathbb{P}_{D'_i}[x \in I_i]$$

$$\mathbb{P}_{D'_i}[(x - \eta, x)], \mathbb{P}_{D'_i}[(x, x + \eta)] < \frac{6\eta}{\epsilon} \mathbb{P}_{D}[x \in I_i]$$

$$\mathbb{P}_{D'_i} \left[ \bigcup_{i=1}^{m-1} (x_i - \eta, x_i) \right], \mathbb{P}_{D'_i} \left[ \bigcup_{i=1}^{m-1} (x_i, x_i + \eta) \right] < \frac{6\eta}{\epsilon}.$$
Figure B-1: Constructing the sequence in the proof of Lemma 21 from the cumulative distribution function.

where $x'$ is the image of $x$ under the rescaling, and in the final line we have used the fact that the intervals $I_i$ are disjoint. Having proved the result for $D'$, we have shown that Lemma 21 implies Lemma 4, apart from the algorithmic statement.

Proof of Lemma 21. By the density of continuous functions, we may assume that $D$ has a continuous cumulative density function (no point masses). Then we can write $P_D[(a, b)] = P_D[[a, b]] =: P[a, b]$ for short.

Suppose the statement is false, so for all $x \in [0, 1]$ there exists $\eta$ such that either $P[x - \eta, x] > 2\eta$ or $P[x, x + \eta] > 2\eta$. We will construct a series of points which will show a violation of this claim.

We start at $x = 0$, where we can rule out one of the possibilities, since $P[-\eta, 0] = 0$. We will first construct a series of points $0 = x_0 < y_0 < x_1 < y_1 < \cdots$ inductively such that (i) $P[0, x_k] \geq x_k$, (ii) $P[0, x_k] - 2x_k$ is strictly decreasing in $k$, and for all $\eta > 0$, (iii) $P[x_k - \eta, x_k] < 2\eta$ and (iv) $P[y_k, y_k + \eta] < 2\eta$. To visualize one step of this, see Figure B-1.

By assumption, there exists some $y > x_k$ such that $P[x_k, y] \geq 2(y - x_k)$. Define $y_k = \sup\{y : P[x_k, y] \geq 2(y - x_k)\}$, the largest such value. By continuity, we must have $P[x_k, y_k] = 2(y_k - x_k)$, which implies that this is where the cdf crosses the
solid line with slope 2 out of $x_k$ in Figure B-1. Then for any $\eta > 0$, we must have
\[ \Pr[x_k, y_k + \eta] < 2(y_k + \eta - x_k) = \Pr[x_k, y_k] + 2\eta, \] so $\Pr[y_k, a_k + \eta] < 2\eta$, satisfying (iv).
In Figure B-1, this shows that the function does not enter the upper red region.

Now define $x_{k+1} = \inf\{b : \Pr[x_k, b] \leq 2(b - y_k)\}$. This is depicted in Figure B-1 as the first intersection with a second solid line parallel to the first. Indeed, this must exist because 1 is a member of the set. This takes a few steps to verify: First, $\Pr[x_k, 1] \leq \Pr[x_k, y_k] = 2(y_k - x_k)$. Second, $\Pr[x_k, 1] = 1 - \Pr[0, x_k] \leq 1 - x_k$ by the inductive step property (i). Doubling the second inequality and subtracting the first (which aligns the inequalities), $\Pr[x_k, 1] \leq 2(1 - x_k) - 2(y_k - x_k) = 2(1 - y_k)$, as claimed. In Figure B-1, this shows that the upper right corner is still below the second solid line, so the function must cross it somewhere.

Again, continuity implies that $\Pr[x_k, x_{k+1}] = 2(x_{k+1} - y_k)$, so $x_{k+1} > y_k$ as claimed. Therefore, for $0 < \eta \leq x_{k+1} - x_k$, $\Pr[x_k, x_{k+1} - \eta] > 2(x_{k+1} - \eta - y_k) = \Pr[x_k, x_{k+1}] - 2\eta$, so $\Pr[x_{k+1} - \eta, x_{k+1}] < 2\eta$, satisfying (iii) for small $\eta$. This also implies that $\Pr[x_k, x_{k+1}] < 2(x_{k+1} - x_k)$, so $\Pr[0, x_{k+1}] = 2x_{k+1} - 2x_k < \Pr[0, x_k] - 2x_k$, satisfying (ii). If $\eta > x_{k+1} - x_k$, $\Pr[x_{k+1} - \eta, x_{k+1}] = \Pr[x_{k+1} - \eta, x_k] + \Pr[x_k, x_{k+1}] \leq 2(x_k - x_{k+1} + \eta) + 2(x_{k+1} - x_k) = 2\eta$ by the induction hypothesis, satisfying (iii) for large $\eta$ as well. This shows that the function does not enter the lower red region in Figure B-1.

Finally, since $x_{k+1} \geq y_k$, we have $\Pr[x_k, x_{k+1}] \geq \Pr[x_k, y_k] = 2(y_k - x_k)$. Averaging this with $\Pr[x_k, x_{k+1}] = 2(x_{k+1} - y_k)$, we have $\Pr[x_k, x_{k+1}] \geq x_{k+1} - x_k$. This shows that the function is above the dotted line with slope 1 when it reaches $x_{k+1}$. Therefore, $\Pr[0, x_{k+1}] = \Pr[0, x_k] + \Pr[x_k, x_{k+1}] \geq x_k + (x_{k+1} - x_k) = x_{k+1}$, satisfying (i).

Therefore, our recursive sequence construction goes through. Now, as an increasing sequence in the compact interval $[0, 1]$, this sequence must have a limit $L$. We claim that for all $\eta > 0$, $\Pr[L - \eta, L], \Pr[L, L + \eta] \leq 2\eta$, i.e. that $L$ proves the lemma. Let us first prove the first claim. Suppose for some $\eta > 0$, $\Pr[L - \eta, L] > 2\eta$. Because $x_0 = 0$ and $x_k \to L$, there exists some $k$ such that $x_k < L - \eta \leq x_{k+1}$. By condition (iii) on $x_{k+1}$, we must have $\Pr[L - \eta, x_{k+1}] \leq 2(x_{k+1} - (L - \eta))$, so $\Pr[x_{k+1}, L] \geq 2\eta - 2(x_{k+1} - (L - \eta)) = 2(L - x_{k+1}).$

On the other hand, by continuity, we must have $\lim_{k \to \infty} (\Pr[0, x_k] - 2x_k) = \Pr[0, L] -
2L. Since we proved (property (ii)) that this sequence is decreasing, we must have \( P[0, x_{k+1}] - 2x_{k+1} > P[0, L] - 2L \). Therefore, \( P[x_{k+1}, L] < 2(L - x_{k+1}) \). This is a contradiction, so no such \( \eta \) exists.

Finally, suppose that for some \( \eta > 0 \), \( P[L, L + \eta] > 2\eta \). Since \( P[0, x_k] - 2x_k \) converges to \( P[0, L] - 2L \), there exists some \( k \) such that \( (P[0, x_k] - 2x_k) - (P[0, L] - 2L) < P[L, L + \eta] - 2\eta \). Rearranging, this means that \( P[x_k, L + \eta] > 2(L + \eta - x_k) \). But since \( y_k = \text{sup}\{y : P[x_k, y] \geq 2(y - x_k)\} \), this implies that \( y_k \geq L + \eta \), which is impossible as \( L > y_k \) as the limit of an increasing sequence. So no such \( \eta \) exists, and we conclude that \( L \) satisfies what we need.

Finally, for the algorithmic statement, since Lemma 4 calls Lemma 21 \( O(1/\epsilon) \) times, it suffices to bound how long each call will take:

**Corollary 22.** If \( D \) is a discretely supported distribution with support size \( s \), then there exists an \( O(s^3) \)-time algorithm to find some \( x \in [0, 1] \) such that \( \forall \eta > 0 \), \( P[[x - \eta, x]], P[[x, x + \eta]] \leq 2\eta.\)

**Proof.** Consider the set of such \( x \), which is nonempty by Lemma 21. We claim this set is closed. Indeed, consider a sequence \( x_1, x_2, \ldots \) that each satisfy the condition and converge to some \( x \in [0, 1] \). Suppose without loss of generality that \( P[[x, x + \eta]] > 2\eta + \epsilon \) for some \( \epsilon > 0 \). Then there exists some \( x_k \) such that \( |x_k - x| < \epsilon/2 \). If \( x_k > x \), then by assumption, \( P[[x, x_k]] \leq 2(x_k - x) \) and \( P[[x_k, x + \eta]] \leq 2(x + \eta - x_k) \) so \( P[[x, x + \eta]] \leq 2\eta \), a contradiction. If \( x_k < x \), then since \( x_k > x - \epsilon/2 \), \( P[[x, x + \eta]] \leq P[[x, x_k + \eta + \epsilon/2]] \leq 2(\eta + \epsilon/2) \), a contradiction. So indeed, \( x \) must satisfy the condition and the set of such \( x \) is closed.

Therefore, this set has a maximum. At this maximum, we must have at least one equality; suppose first that \( P[[x, x + \eta]] = 2\eta \) for some \( \eta > 0 \). Since \( D \) is discretely supported and for any \( \epsilon > 0 \), \( P[[x, x + \eta - \epsilon]] \leq 2(\eta - \epsilon) < 2\eta = P[[x, x + \eta]], x + \eta \) must be in the support. Similarly, if \( P[[x - \eta, x]] = 2\eta, x - \eta \) must be in the support.

Let the support of \( D \) be \( d_1 < d_2 < \cdots < d_s \). The tight interval \([x - \eta, x]\) or \([x, x + \eta]\) contains some consecutive subset of these \( d_i, d_{i+1}, \ldots, d_j \), and either \( d_i = x - \eta \) or \( d_j = x + \eta \). Moreover, the sum of the weights at these points is exactly \( 2\eta \), so the two
possible values for $x$ can be determined from the subset. There are $O(s^2)$ possible subsets of consecutive points, so $O(s^2)$ possible $x$’s to check. Moreover, for each candidate $x$, we only have to check $s$ possible values of $\eta (|x - d_i|$ for $i = 1, \ldots, s$). Therefore, the brute force algorithm of checking all such $x$ takes $O(s^3)$ time to find one that works.

Now, let us show that this smart rounding, like the noisy posterior, defeats linear classification:

**Theorem 23.** Under model $LC_n$, the smart rounded posterior mean curator algorithm answers $q$ queries $\epsilon$-accurately with probability $1 - \delta$, with $n = O \left( \log \frac{2}{\delta} \right)$ samples.

**Proof.** First consider a single query. By Lemma 18, at least a $1/2$ probability mass of the prior lies within $2^{-m/2} = 2^{-n/2}$ of the prior mean along that query direction.

We will first use this fact to show that the prior mean is not near a boundary of the resulting smart partition.

By Lemma 4 with $\eta = \epsilon/24$, less than a total of $\frac{1}{4}$ of the prior probability weight lies within $\epsilon/24$ on either side of all boundaries. Therefore, those regions cannot hold all of the probability mass within $2^{-n/2}$ of the prior mean, which implies that the prior mean is at least $\epsilon/24 - 2^{-n/2}$ away from the nearest boundary.

By Lemma 18 again, the probability of the posterior mean falling into a different interval than the prior mean is less than $2^{-n}/(\epsilon/24 - 2^{-n/2})^2 = (\epsilon/(24 \cdot 2^{-n/2}) - 1)^{-2}$. If $2^{-n/2} \leq \frac{\epsilon \sqrt{q}}{48\sqrt{q}}$, then, the probability of a different answer than the prior mean is at most $\frac{1}{(2\sqrt{q/\delta - 1})^2} < \frac{\delta}{2q}$.

Therefore, the smart rounded posterior mean answers the same as the Smart Rounded Prior Mean with probability at least $1 - \delta/2q$. Since the Smart Rounded Prior Mean doesn’t depend on the data, it can be simulated by the analyst, and its answers provide no additional information. Therefore, if the answers match, the analyst only learns that an event with probability $\delta/2q$ did not take place. Repeating for all $q$ queries, all answers match with probability at most $\delta/2$.

---

4 Technically, Lemma 18 only applied to the true answer’s deviation from the prior mean. But the posterior mean consists of an average of several true answers, which will only concentrate tighter towards the prior mean. This is immediately clear when it comes to variance-based arguments like the proof of Lemma 18, since $\text{Var}(\frac{a+b}{2}) \leq \frac{1}{2}(\text{Var}(a) + \text{Var}(b))$. 95
The Smart Rounded Prior Mean is also $\epsilon$-accurate: By Lemma 18 one last time, the prior mean is within $\epsilon/2$ of the true answer with probability $1 - 2^{-n}/\epsilon^2 \geq 1 - \frac{\delta}{48q} > 1 - \frac{\delta}{2q}$. Moreover, the smart rounding answers with the midpoint of an interval of width less than $\epsilon$ containing the prior mean, so it is only at most $\epsilon/2$ off. Therefore, the smart rounded posterior mean is within $\epsilon$ of the true answer with probability $1 - \delta$ over all $q$ queries, as desired.

Like with noise (Appendix B.2), we’ve demonstrated accuracy of rounding by comparing with the prior mean, which doesn’t depend on the data. But unlike the prior mean, these algorithms also behave well on $LC_n$ for $n > m$, which is when there is enough data for the posterior to nail down the true hypothesis. In this case, it would still take $\exp(\Theta(m))$ queries for the analyst to learn this hypothesis, but this can be far less than $\exp(\Theta(n))$.

The next natural question would be whether smart rounding can succeed against polynomial classification. Unfortunately, that answer isn’t immediately clear. We need an even smaller fraction of the hypotheses to lie outside of the same region around the posterior mean, and the corresponding concentration bounds from the $(2^k - 1)$-wise independence of degree $\leq k$ polynomials aren’t strong enough. I’d conjecture that it does work, but this also isn’t the end of the story; there is yet another difficult model.
Appendix C

Additional Results Related to Adaptive Guarantees

C.1 Subgaussianity from Raw Moments

In this appendix, we prove a claim that is strictly weaker than Theorem 10 using a different technique that has other applications:

**Proposition 24.** The beta distribution $\text{Beta}(\alpha, \beta)$ is $\frac{1}{2(\alpha + \beta + 1)}$-subgaussian.

**Proof.** We will first prove a more general lemma based on ratios of raw moments:

**Lemma 25.** If $X$ is a random variable with positive raw moments satisfying

$$
\frac{\mathbb{E}[X^{j+2}]}{\mathbb{E}[X^j]} \leq \mathbb{E}[X^2] + (j + 1)^2,
$$

(C.1)

for every nonnegative integer $j$, then for all $\lambda > 0$, $\mathbb{E}[e^{\lambda(X-E[X])}] \leq e^{\lambda^2 \sigma^2/2}$.

**Remark.** Note that for $j = 0$, this condition says that $\text{Var}[X] \leq \sigma^2$.

**Proof.** Notice that technically, subgaussianity is a claim about the centered distribution, or $X - EX$. However, our condition does not involve centered moments, but raw moments. That is, rather than proving

$$
\mathbb{E}[e^{\lambda(X-E[X])}] \leq e^{\lambda^2 \sigma^2/2}
$$
by expanding termwise in $\lambda$, we move the mean term to the other side and show

$$E[e^{\lambda X}] \leq e^{\lambda E[X] + \lambda^2 \sigma^2/2}$$

by expanding termwise in $\lambda$. First, we consider the even terms, which ask for bounds on the even moments. Since all moments and powers of $\lambda$ are positive, we can use the condition as a telescoping product:

$$[\lambda^{2k}]E[\exp(\lambda X)] = \frac{1}{(2k)!}E[X^{2k}] = \frac{1}{(2k)!} \frac{E[X^2]}{E[X^0]} \cdot \frac{E[X^4]}{E[X^2]} \cdots \frac{E[X^{2k}]}{E[X^{2k-2}]}$$

$$\leq \frac{1}{(2k)!} (E[X]^2 + \sigma^2)(E[X]^2 + 3\sigma^2) \cdots (E[X]^2 + (2k - 1)\sigma^2)$$

$$\leq \frac{1}{(2k)!} \sum_{l=0}^{k} E[X]^{2l} \sigma^{2(k-l)} \binom{k}{l} (2k - 1)(2k - 3) \cdots (2l + 1)$$

$$= \frac{1}{(2k)!} \sum_{l=0}^{k} E[X]^{2l} \frac{(2k)!}{(2l)!(k-l)!}$$

$$= \sum_{l=0}^{k} \frac{1}{(2l)!} \frac{E[X]^{2l}}{(k-l)!} (\sigma^2/2)^{k-l}$$

$$= [\lambda^{2k}] \exp(\lambda E[X] + \lambda^2 \sigma^2/2).$$

In going from the second to the third line, we have expanded the product and grouped terms, upper bounding the sum of all products of $k - l$ of the $\sigma^2$ coefficients by \( \binom{k}{l} \) times the largest of them. From the third to the fourth lines, we have expanded \( \binom{k}{l} = \frac{k(k-1)\cdots(l+1)}{(k-l)!} \) and doubled the terms in the numerator to interchange with the odd terms, adding an extra factor of $2 \sigma^2$ term.
The odd terms are similar:

\[
\lambda^{2k+1} \mathbb{E}[\exp(\lambda X)] = \frac{1}{(2k+1)!} \mathbb{E}[X^{2k+1}] = \frac{1}{(2k+1)!} \frac{\mathbb{E}[X^3]}{\mathbb{E}[X]} \cdots \frac{\mathbb{E}[X^{2k+1}]}{\mathbb{E}[X^{2k-1}]} \\
\leq \frac{1}{(2k+1)!} \mathbb{E}[X]^{(2k+1)} + \mathbb{E}[X^2] + 2 \sigma^2 \cdots ((k+1)^2 + 2k \sigma^2)
\]

\[
\leq \frac{1}{(2k+1)!} \sum_{l=0}^{k} \mathbb{E}[X]^{2l+1} \sigma^{2(k-l)} \binom{k}{l} (2k+1)(2k-1) \cdots (2l+3)
\]

\[
= \frac{1}{(2k+1)!} \sum_{l=0}^{k} \mathbb{E}[X]^{2l+1} \sigma^{2(k-l)} \frac{(2k+1)!}{(2l+1)!(k-l)!}
\]

\[
= \sum_{l=0}^{k} \frac{1}{(2l+1)!} \mathbb{E}[X]^{2l+1} \frac{1}{(k-l)!} \sigma^{2(k-l)}
\]

\[
= \lambda^{2k+1} \exp(\lambda \mathbb{E}[X] + \lambda^2 \sigma^2 / 2).
\]

The only other major difference is that between the third and fourth lines, we also replaced the numerators of \(2k, 2k-2, \ldots, 2\) with the corresponding larger values of \(2k+1, 2k-1, \ldots, 3\). Summing all of the terms, which are positive because \(\lambda > 0\), yields the desired inequality. \(\square\)

This is the core of the proof. To apply this to the beta distribution, we just have to check the condition on the raw moments, which are given by (2.6). This amounts to checking the following technical claim:

**Lemma 26.** For any nonnegative integer \(j\) and \(\alpha, \beta > 0\),

\[
\frac{\alpha + j}{\alpha + \beta + j} \cdot \frac{\alpha + j + 1}{\alpha + \beta + j + 1} \leq \left( \frac{\alpha}{\alpha + \beta} \right)^2 + \frac{j + 1}{2(\alpha + \beta + 1)}.
\]

**Proof.** We induct on \(j\), starting with two base cases: \(j = 0\) and \(j = 1\).
**Base Case.** For $j = 0$, we have

\[
\frac{\alpha}{\alpha + \beta} \cdot \frac{\alpha + 1}{\alpha + \beta + 1} - \left( \frac{\alpha}{\alpha + \beta} \right)^2 = \frac{\frac{\beta}{(\alpha + \beta)(\alpha + \beta + 1)}}{(\alpha + \beta)^2/4} \leq \frac{1}{(\alpha + \beta)^2(\alpha + \beta + 1)} = \frac{1}{4(\alpha + \beta + 1)} < \frac{1}{2(\alpha + \beta + 1)},
\]

where we used the AM-GM inequality $\sqrt{\alpha \beta} \leq (\alpha + \beta)/2$. Now, if $j = 1$, we have

\[
\frac{\alpha + 1}{\alpha + \beta + 1} \cdot \frac{\alpha + 2}{\alpha + \beta + 2} - \left( \frac{\alpha}{\alpha + \beta} \right)^2 = \frac{(\alpha + 1)(\alpha + 2)(\alpha + \beta)^2 - \alpha^2(\alpha + \beta + 1)(\alpha + \beta + 2)}{(\alpha + \beta)^2(\alpha + \beta + 1)(\alpha + \beta + 2)} = \frac{(\alpha + 1)(\alpha + \beta)^2(\alpha + \beta + 2)}{(\alpha + \beta)^2(\alpha + \beta + 1)(\alpha + \beta + 2)}
\]

\[
= \frac{3(\alpha + 2)(\alpha + \beta)^2 - \alpha^2(3(\alpha + \beta) + 2)}{(\alpha + \beta)^2(\alpha + \beta + 1)(\alpha + \beta + 2)} = \frac{3\alpha\beta(\alpha + \beta) + 4\alpha\beta + 2\beta^2}{(\alpha + \beta)^3 + 2(\alpha + \beta)^2} \leq \frac{1}{(\alpha + \beta)^2(\alpha + \beta + 1)(\alpha + \beta + 2)} = \frac{1}{\alpha + \beta + 1},
\]

as desired.

**Inductive Step.** For the inductive step, take $j \geq 2$. Then

\[
\frac{\alpha + j}{\alpha + \beta + j} \cdot \frac{\alpha + j + 1}{\alpha + \beta + j + 1} - \left( \frac{\alpha + j - 1}{\alpha + \beta + j - 1} \cdot \frac{\alpha + j}{\alpha + \beta + j} \right) = \frac{\alpha + j}{\alpha + \beta + j} \cdot \frac{\alpha + j + 1}{\alpha + \beta + j + 1} - \frac{\alpha + j - 1}{\alpha + \beta + j - 1} \cdot \frac{\alpha + j}{\alpha + \beta + j}
\]

\[
\leq \left( \frac{\alpha + j}{\alpha + \beta + j} \cdot \frac{\alpha + j + 1}{\alpha + \beta + j + 1} - \left( \frac{\alpha}{\alpha + \beta} \right)^2 \right) \leq \frac{1}{2(\alpha + \beta + j - 1)} \leq \frac{1}{2(\alpha + \beta + 1)} \leq \frac{1}{j + 1} \leq \frac{j + 1}{2(\alpha + \beta + 1)} + \frac{1}{2(\alpha + \beta + 1)} = \frac{j + 1}{2(\alpha + \beta + 1)},
\]

as desired.
where we applied the inductive hypothesis to get the last line.

Therefore, we’ve shown that for $\lambda > 0$,

$$\mathbb{E}[e^{\lambda X}] \leq \exp \left( \frac{\lambda \alpha}{\alpha + \beta} + \frac{\lambda^2}{4(\alpha + \beta + 1)} \right).$$

However, to prove subgaussianity, we need this for all $\lambda \in \mathbb{R}$. To prove it for $\lambda < 0$, we utilize the symmetry of the beta distribution: $\text{Beta}(\beta, \alpha) = 1 - \text{Beta}(\alpha, \beta)$. Therefore, we’ve in fact also shown that for $\lambda > 0$,

$$\mathbb{E}[e^{\lambda(1-X)}] \leq \exp \left( \frac{\lambda \beta}{\alpha + \beta} + \frac{\lambda^2}{4(\alpha + \beta + 1)} \right).$$

Dividing both sides by $e^\lambda$, we immediately get

$$\mathbb{E}[e^{-\lambda X}] \leq \exp \left( - \frac{\lambda \alpha}{\alpha + \beta} + \frac{\lambda^2}{4(\alpha + \beta + 1)} \right),$$

so the desired bound holds for $\lambda < 0$ as well.

This method also applies to other random variables with easy-to-compute raw moments. For ease of discussion, this definition will be helpful:

**Definition.** A random variable $X$ is $\sigma^2$-upper subgaussian if for all $\lambda > 0$, $\mathbb{E}[e^{\lambda X}] \leq e^{\lambda^2 \sigma^2/2}$. Similarly, $X$ is $\sigma^2$-lower subgaussian if this bound holds for all $\lambda < 0$.

**Proposition 27.** If $X$ is $\sigma^2$-upper (resp. lower) subgaussian, then $\mathbb{P}[X > \epsilon] \leq e^{-\epsilon^2/(2\sigma^2)}$ (resp. $\mathbb{P}[X < -\epsilon] \leq e^{-\epsilon^2/(2\sigma^2)}$) for all $\epsilon > 0$.

**Proof.** This is the standard application of Markov’s inequality:

$$\mathbb{P}[X \geq \epsilon] \leq e^{-\lambda \epsilon} \mathbb{E}[e^{\lambda X}] \leq e^{\lambda^2 \sigma^2/2 - \lambda \epsilon},$$

for any $\lambda > 0$. Setting $\lambda = \epsilon/\sigma^2$ yields the desired result. The lower subgaussian bound is identical.
That these upper bounds on the moments of $X$ only imply bounds on the upper tail of the distribution of $X$ is perhaps not surprising, since large moments distinguish the upper tail much more than they do the lower tail.

We can see that this method works well whenever we can bound ratios of raw moments of an uncentered random variable in this fashion. For a second example, consider the Chi distribution, a distribution on $[0, \infty)$, which measures the Euclidean norm of a standard $k$-dimensional Gaussian. Its moments are given by (see [32])

$$E[X^j] = 2^{j/2} \frac{\Gamma((k + j)/2)}{\Gamma(k/2)} \Rightarrow E[X] = \sqrt{2} \frac{\Gamma((k + 1)/2)}{\Gamma(k/2)}$$

$$E[X^2] = k$$

$$E[X^{j+2}] = (k + j)E[X^j].$$

**Proposition 28.** The Chi distribution $\chi_k$ is $1$-upper subgaussian.

**Proof.** To apply Lemma 25, we already have $E[X^2] = k$ and $\frac{E[X^{j+2}]}{E[X^j]} = k + j = (k - 1) + (j + 1)$, so it only remains to show that $E[X]^2 \geq k - 1$. Because the Gamma function is log-convex, $\frac{\Gamma((k + 1)/2)}{\Gamma(k/2)} > \frac{\Gamma(k/2)}{\Gamma((k-1)/2)}$, so

$$E[X]^2 = 2 \frac{\Gamma((k + 1)/2)^2}{\Gamma(k/2)^2} > 2 \frac{\Gamma((k + 1)/2)}{\Gamma(k/2)} \cdot \frac{\Gamma(k/2)}{\Gamma((k - 1)/2)} = 2 \frac{\Gamma((k + 1)/2)}{\Gamma((k - 1)/2)} = k - 1,$$

as desired. \qed

Of course, this is not particularly surprising, given that the probability density function of the $\chi$ distribution is proportional to $x^{k-1}e^{-x^2/2}$. Still, this example illustrates the close relationship between bounds on such ratios of moments and having a subgaussian upper tail.

### C.2 Dirichlet Projects to Beta on Counting Queries

In this section, we complete the proof of Theorem 13 by showing that the projection of the Dirichlet distribution onto a counting query is a beta distribution.
Proof. Call the counting query $v \in \{0,1\}^k$. By relabeling coordinates, we can suppose that $v_1 = \cdots = v_l = 1$ and $v_{l+1} = \cdots = v_k = 0$ for some $2 \leq l \leq k - 1$ (if all $v_i = 0$ or all $v_i = 1$ the dot product is always 0 or 1 respectively). We first transform the simplex of possible $\tilde{p}$ in a fairly common way by considering the partial sums $s_1 = p_1, s_2 = p_1 + p_2, \ldots, s_{k-1} = 1 - p_k$. Then the simplex is given by

$$\Delta_k' = \{s_1, \ldots, s_{k-1} \in \mathbb{R} : 0 \leq s_1 \leq s_2 \leq \cdots \leq s_{k-1} \leq 1\}.$$ 

In this notation, $\tilde{p} \cdot \tilde{v} = t$ corresponds to $p_1 + \ldots + p_l = s_l = t$. The probability density of $\tilde{p} \cdot \tilde{v}$ at $t$ is therefore proportional to the $(k - 2)$-dimensional volume with respect to the Dirichlet distribution of this slice, or

$$\mathbb{P}[v \cdot p = t] \propto \int_{\Delta_k'(s_l=t)} s_1^{\alpha_l-1}(s_2-s_1)^{\alpha_2-1} \cdots (1-s_{k-1})^{\alpha_{k-1}} ds_1 \cdots ds_{l-1} ds_{l+1} \cdots ds_{k-1} \quad \text{or}$$

$$= \int_{0 \leq s_1 \leq \cdots \leq s_{l-1} \leq t} s_1^{\alpha_l-1} \cdots (t-s_{l-1})^{\alpha_l-1} ds_1 \cdots ds_{l-1} \times$$

$$\int_{t \leq s_{l+1} \leq \cdots \leq s_{k-1} \leq 1} (s_{l+1} - t)^{\alpha_{l+1}-1} \cdots (1-s_{k-1})^{\alpha_{k-1}} ds_{l+1} \cdots ds_{k-1}$$

$$= \int_{0 \leq s_1' \leq \cdots \leq s_{l-1}' \leq 1} (ts_1')^{\alpha_l-1} \cdots (t-ts_{l-1}')^{\alpha_l-1} d(ts_1') \cdots d(ts_{l-1}') \times$$

$$\int_{0 \leq s_{l+1}' \leq \cdots \leq s_{k-1}' \leq 1} ((1-t)s_{l+1}')^{\alpha_{l+1}-1} \cdots ((1-t)(1-s_k'))^{\alpha_{k-1}}$$

$$= t^{\alpha_l + \cdots + \alpha_{l-1}} (1-t)^{\alpha_{l+1} + \cdots + \alpha_k-1},$$

where we have substituted $s_i = ts_i'$ for $i \leq l$ and $s_j = (1-t)s_j' + t$ for $j > l$. Pulling out the factors of $t$ and $1-t$, the remainder no longer depends on $t$. After normalizing, this is exactly the Beta($\alpha_1 + \cdots + \alpha_l, \alpha_{l+1} + \cdots + \alpha_k$) distribution. Therefore, by Theorem 10, this distribution is $\frac{1}{2(\alpha_1 + \cdots + \alpha_k + 1)}$-subgaussian, as desired. \hfill \square
Appendix D

Parameter Settings for Empirical CV Results

The results of Figures 3-1, 3-2, 3-3, and 3-4 were obtained in an R notebook with the following parameters:

- **Steep Learning Curve**
  - Algorithm $A$: Linear Models (1m)
  - Universe $\mathcal{X} \times \mathcal{Y}$: $\mathcal{X} = \mathbb{R}^{20}$, $\mathcal{Y} = \mathbb{R}$.
  - Distribution $\bar{p}$: All $x$-coordinates and the $y$ coordinate are independent Gaussians with mean 0 and variance 1.
  - Data set size $n$: 50.
  - Loss function $\ell$: Squared error, $\ell(y, y') = (y - y')^2$.
  - Minimum training set size $t$: 30.

- **Nearly Flat Learning Curve**
  - Algorithm $A$: Random Forests (randomForest)
  - Universe $\mathcal{X} \times \mathcal{Y}$: $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = \mathbb{R}$.
  - Distribution $\bar{p}$: $x \sim N(0,1)$, $y \sim N(x,1)$. 

105
- Data set size $n$: 50.
- Loss function $\ell$: Squared error, $\ell(y, y') = (y - y')^2$.

- Full Computational Budget Permutations: $\sigma_1$ is a random permutation of $[50]$, and the rest are all possible cyclic shifts of $\sigma_1$.

- Small Computational Budget Permutations: $\sigma_1$ is a random permutation of $[50]$, and the rest are cyclic shifts of $\sigma_1$ by multiples of 5.

- Holdout set size: 10000.

- Number of iterations: 1000.

We expect linear models on a high dimensional example to have a fairly steep learning curve as it must learn to estimate all of the coefficients. By contrast, random forests on a low-dimensional model will already have done most of its learning by training set size 6, and will only have marginal improvements with additional data points, leading to a nearly flat learning curve.

The data set size of 50 was chosen as a tradeoff between allowing for a range of training set sizes and limiting computation time to allow for more results. The distributions were chosen for simplicity and to limit the effect of outliers, and the minimum training set sizes were chosen to avoid sizes that are problematic for the respective algorithms (e.g. rank-deficient linear models).
Bibliography


108


