New Models and Algorithms for Data Analysis

BY

Benjamin Fish
B.A., Pomona College, 2013
M.S., University of Illinois at Chicago, 2015

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:

Lev Reyzin, Chair and Advisor
Avrim Blum, Toyota Technological Institute at Chicago
Dhruv Mubayi
Robert Sloan, Department of Computer Science
György Turán

To my parents, without whom this thesis would not have been possible.

Acknowledgments

I would like to thank my advisor Lev Reyzin for his tireless support, encouragement, advice, thoughtfulness, optimism, and opinions throughout my time at the University of Illinois at Chicago. Many thanks also to György Turán, as supporting, patient, and kind as a mentor could be.

Thanks to my entire committee, which in addition to Lev Reyzin and György Turán, consisted of Avrim Blum, Dhruv Mubayi, and Robert Sloan. I would like to acknowledge Avrim Blum for pointing out a mistake in an earlier draft.

Thanks also to Rajmonda Caceres, Benjamin Rubinstein, and Suresh Venkatasubramanian for not only graciously hosting me during summers, but giving me the opportunity to broaden my academic horizons.

A special thanks to Yi Huang, Jeremy Kun, and Ádám D. Lelkes for spending countless hours working with me on our many projects. And to all the other graduate students who supported me throughout graduate school.

Finally, a big thanks to my entire family.

In the course of writing this thesis, I was supported by the following grants and fellowships: Army Research Office grant #66497-NS, NSF grant IIS-1526379, and the NSF EAPSI fellowship.

Contribution of Authors

Chapter 3 represents the preprint [27], co-authored with Lev Reyzin and Benjamin I. P. Rubinstein. Chapter 4 represents the manuscript [26], co-authored with Lev Reyzin. Chapter 5 represents the manuscript [25], co-authored with Yi Huang and Lev Reyzin. This work also appears in the thesis of Yi Huang [38].

All content in these chapters, including introduction, formulation of definitions, theorems, algorithms, experiments, and writing of the various manuscripts were done jointly with the co-authors.

Table of Contents

List of Figures

List of Tables

Summary

In this thesis, we introduce and analyze new models and new algorithms for problems in data analysis. Data analysis has become increasingly important as the quantity and quality of available data for machine learning has greatly increased. This means that many new challenges and constraints for data analysis have arisen. In this thesis, we discuss how to overcome a few of these challenges.

In the first part of this thesis, we tackle the problems that arise when data analysis is adaptive, so that past inferences guide future inquiries into the same dataset. A recent line of work in the theory community has established mechanisms that provide low generalization error even on adaptive queries. Building on this, we show how sampling techniques can be used to provably guarantee validity while speeding up analysis over previous work.

We describe mechanisms that provide a polynomial speed-up per query over previous mechanisms, without needing to increase the total amount of data needed for low generalization error. We prove that this speed-up holds for arbitrary low-sensitivity queries, and then show how this can be applied to speed up adaptively-made convex optimization queries. We also provide a method for achieving statistically-meaningful responses even when the mechanism is only allowed to see a constant number of samples from the data per query.

In the second part of this thesis, we analyze the problem of learning with label proportions. Here, the training data is unlabeled, and only the proportions of examples receiving each label are given. The goal is to learn a hypothesis that predicts the proportions of labels on the distribution underlying the sample.

We resolve foundational questions regarding the computational complexity of learning in this setting. We formalize a simple version of the setting, and we compare the computational complexity of learning in this model to classical PAC learning. We also demonstrate a non-trivial hypothesis class that is efficiently PAC learnable but cannot be learned from labels efficiently under natural assumptions. We also give an algorithm that demonstrates the feasibility of learning under well-behaved distributions.

In the final part of this thesis, we investigate how to reconstruct social networks from voting data. In particular, given a voting model that considers social network structure, we aim to find the network that best explains the agents' votes. We study two plausible voting models, one edge-centric and the other vertex-centric.

For these models, we give algorithms and lower bounds, characterizing cases where network recovery is possible and where it is computationally difficult. We also test our algorithms on United States Senate data.

Despite the similarity of the two models, we show that their respective network recovery problems differ in complexity and involve distinct algorithmic challenges. Moreover, the networks produced when working under these models can also differ significantly. These results indicate that great care should be exercised when choosing a voting model for network recovery tasks.

1

Introduction

The demands we place on data analysis are high: we must make predictions, generalize to unseen data, and infer structure found within the data. We must classify and cluster, analyze and interact. We must do this all while using as little data and time as possible. We may have additional constraints, like dealing with adversarial processes, handling missing data, or obeying to privacy constraints. Yet while machine learning techniques have become increasingly effective, they often lack the ability to handle such constraints. Much has been made of such techniques to handle large data using increasing amounts of available training data and processing power, but no matter how much processing power we may have at

our disposal or how much data we throw at the problem, current machine learning often fails at dealing with additional constraints of the type we will discuss in this thesis. In this thesis, we discuss three particular constraints on our data analysis, introducing both appropriate models to discuss such constraints and algorithms to overcome the challenges posed by those constraints.

1.1 Adaptivity in data analysis

The first constraint is a central one in data analysis: Tools for data analysis must be able to handle adaptivity, simply because that is how much of data science is performed in practice. Consider the following sequence of events: A data analyst tests out their sophisticated classifier, say a neural net, on a validation set. It receives high empirical loss, which if we assume the data comes i.i.d. from some unknown distribution, is guaranteed to generalize to that distribution (with high probability, assuming that there's sufficient data). This means that the high empirical loss implies high true loss. Because the analyst is looking for a neural net with lower error, the analyst may modify the architecture or hyperparameters of the neural net. But when the analyst goes to measure the error of the modified classifier, the empirical error may no longer necessarily generalize to the distribution because the classifier and the validation set are no longer independent from each other.

This is an unfortunate case where overfitting—the bane of data analysis—naturally arises due to the adaptivity of the process, where the data analyst adapted their actions to previous conclusions on the data. In this thesis, we

ask for new algorithmic techniques for guaranteeing generalization in the face of adaptivity using a model of Dwork et al. [20]. In particular, we focus on faster algorithms, enabling data analysts to spend less time and effort avoiding overfitting.

In this framework, we must answer queries about an unknown distribution using only an i.i.d. sample from that distribution, where the queries may be arbitrarily adaptive (so that a query may depend not only on all previous queries but all answers received in answer to those queries). Examples of queries include asking for the loss of a classifier, asking for the classifier amongst a class with the minimum loss, or asking how well-clustered example points are on average. Traditional approaches to guaranteeing adaptivity, especially in the statistics literature, focus on a particular sequence of queries the data analyst may adapt. For example, the analyst may perform a round of feature selection followed by a round of regression on the selected variables [19]. This approach does not necessarily succeed even when the order of queries changes, even if the types of queries themselves remain the same. Meanwhile, traditional approaches in the theory of machine learning focus on being able to answer any queries from a class of queries of bounded complexity. In the case when the queries ask for the loss of a binary classifier, this becomes classes with bounded VC dimension (see Chapter 2).

However, these techniques fail in this adaptive setting for classes of unbounded VC dimension (or the like), which may be undesirable in many settings, e.g. exploratory data analysis, where the data analyst does not necessarily ask queries from such a class. To remedy this, Dwork et al. [20] introduce a new elegant

method that relies on guaranteeing differential privacy. Formally defined in Chapter 2, guaranteeing differential privacy usually means adding noise to the responses. This may be counterintuitive because noise might seem to make the responses less accurate, but it instead does the opposite because differential privacy acts as a notion of stability. Stability is a widely used property of learning algorithms to guarantee generalization [68]. In particular, Bassily et al. [4], building on the work of Dwork et al. [20], show that for responses to be accurate with respect to the underlying distribution, it suffices for responses to be 1) close to the empirical estimate of the query and 2) differentially private.

In this thesis, we focus on mechanisms that provide responses that also have this guarantee using differential privacy, but are significantly faster. In previous work, the mechanisms took at least linear time in the sample size per query. We improve this running time, allowing the data analyst to ask significantly more queries for fixed amount of time spent. This involves examining fewer samples to compute the response to each query. We go on to show that meaningful mechanisms are still possible even if the algorithm only has time to examine just a constant number of samples per query. We also show how to speed up convex optimization queries, which ask for the minimizer of the expected loss for a given convex loss function.

1.2 The connection between examples and labels

In the previous section, we assumed the data was complete—we could measure the loss of a classifier on a validation set precisely because that validation set contained both the examples and their labels. Unfortunately, this is not always

going to be the case. One possibility is that we have the examples, and the set of labels, but we don't know which label is attached to which example. For instance, if each example is information on a voter and the labels are which candidate that voter prefers, then a preference poll, of the sort conducted before an American presidential election, is exactly a set of labels, e.g. "40% of Americans prefer Candidate A over Candidate B." Voter roles and associated data give information on the voters themselves but often not which voter has which preference. How can we infer which candidate will win, or infer how voters reach such a conclusion, without knowing which input is attached to which label?

A related setting is Multiple Instance Learning (MIL), first introduced by Dietterich et al. [17], where the goal is to classify bags of examples with unobserved labels, and a bag is labeled positively by a boolean 'or' function: if any example in the bag is labeled positively, the bag is as well. The goal in MIL is to label new bags with whether any example in the bag is labeled positively. While a notion of learning that does not have as input individual labels but merely statistics about groups of examples, it does not capture voting or other similar settings.

To address this, Kück and de Freitas [16] and, independently, Chen et al. [13] introduce problems where the goal is to learn from other group statistics besides the one used in MIL. Kück and de Freitas adopt a Bayesian approach borrowed from the MIL setting to learn an instance-level classifier. Meanwhile, Chen et al. generalize to the case where examples' data are also aggregated in addition to the labels. In both of these, the goal is to learn an instance-level classifier: the classifier should get the labels of individual examples correct. On the other

hand, in this thesis we only desire a classifier that achieves a weaker goal, namely one that predicts the correct proportion of labels amongst the instances. This is because, as we will see, our goal is already difficult to provably achieve.

In order to give a theory for how difficult this goal is to achieve, we will use a formal model of learning. Neither Kück and de Freitas nor Chen et al. give such a model [13, 16]. However, there has also been some theoretical work. Quandrianto et al. [62] give certain convergence bounds for their proposed algorithm, for example. The only model of learning (which includes a formalized goal for which guarantees exist) for this setting that has already been proposed is in the work of Yu et al. [83]. This model is a multi-bag model (multiple groups of examples) that assumes bags are independent from each other, but examples within a bag are not independent of each other. To represent the case where examples from a sample are independent of each other (as may be the case for polling data), we introduce our own model of learning.

In this thesis, we formalize a PAC-style model of learning that captures this voting problem, and problems of this ilk. (PAC will be defined formally in Chapter 2). We then give a series of results that explore to what degree learning is computationally feasible in this model.

1.3 Inferring networks

In the voting example in the previous section, we have ignored that decisions about voting are typically not made in isolation. Indeed, classical machine learning assumes that each example's label depends only on that label, and the goal is

to learn a mapping between example and label. On the other hand voters, for example, discuss how to vote with the people around them in their social or voting network. So the labels of the examples (each a voter) depend on other examples. Here, the goal changes from inferring a mapping between voter and label to inferring an entire network that describes dependencies of the data: Which voters talk to which other voters? In this thesis, we want to infer this information solely from the kind of data we typically have, namely the votes of all of the voters, but not side information like what the votes are about or any affiliation the voters may have. We formalize this by assuming there is some graph, unknown to the learner, where an edges between two voters represent the fact that they communicate in order to decide their vote. The goal is to learn this graph from the final votes of the voters.

Here, we infer the network from vertex attributes (in this case the voters' votes). In a broad sense, there has been a sizeable amount of work inferring network structure from vertex attributes. This includes the related task of link prediction, which assumes that some of the network is already known, and attempts to infer the missing edges. One typical approach is to define a measure of similarity between nodes, such as the number of common neighbors they share, but there are many approaches [49, 73].

One common approach to network inference, where we typically assume no knowledge of edges is given, also measures the pairwise similarity of nodes, in particular correlation between their attributes. This type of network is known as a 'correlation network,' naturally. The goal then becomes determining the right

threshold over which correlation is deemed significant and an edge is recorded. This is usually done by simple statistical tests, or is determined by trial and error [10] but these methods typically do not come with any guarantees on the quality of the inferred network.

Another approach is to assume the existence of a parametric model that relates observed attributes with the underlying network. The network may then be inferred with a maximum likelihood approach. This includes everything from inferring graphical models [31, 34] to inferring an epidemiological network from observed infections [28, 56]. Again, these are typically hard problems to solve exactly, and heuristics are frequently substituted.

In contrast, in this thesis, while we do assume the existence of such a parametric model, we are able to prove guarantees on the quality of the observed network. To do this, we use correlations between attributes, thereby proving the efficacy of a correlation network for inference. We do this to both efficiently find the maximum likelihood graph and the underlying graph, assuming that such a graph exists.

However, our focus is on showing that such models that relate the observed attributes to the network, even very simple ones, are not robust, in the sense that small changes to the model can have drastic changes to both the computational feasibility of provably learning the voting network and the structure of the resulting inferred network. We therefore show not only positive results guaranteeing the quality of an efficient algorithm, but several settings in which no such algorithm is possible.

1.4 Organization of this thesis

The following is an outline for the remainder of this thesis:

- In Chapter 2, we provide introductions to the areas this thesis covers, including differential privacy, learning theory, and computational complexity theory.
- In Chapter 3, we analyze faster algorithms for adaptive data analysis.
- In Chapter 4, we provide and analyze a model of learning from label proportions.
- In Chapter 5, we show how to learn social networks from voting data.

While Chapter 2 provides some of the basic definitions, many of which may be found in standard textbooks, we also review background and previous work specific to each chapter within that chapter.

2

Background

In this section, we go over some of the central tools needed for this thesis. This includes tools from data analysis and machine learning. We start with learning theory.

2.1 Learning theory: generalization in learning

In this thesis, we are acutely interested in generalizing our results to unseen data in the context of learning theory, so we give a short introduction to statistical learning theory here. For a longer introduction, see Mohri et al. [53]. Loosely, generalization means when we have access to a sample S of size m drawn i.i.d. from

an unknown distribution D over a domain X, the conclusion we draw from S, say the value $f(S)$ for $f : X^m \to \mathbb{R}$, should be close to $\mathbb{E}_{S\sim D^m}[f(S)]$.

For example, we may care about the error of a hypothesis $h : X \to \{0, 1\}$ with respect to a labeler $c : X \to \{0, 1\}$. The empirical *risk* (alternatively, *error* or *loss*) of such a hypothesis is $\mathcal{L}_S(h) := \frac{1}{m}\sum_{x\in S} \mathbb{1}_{h(x)\neq c(x)}$. Then the generalization error of h is $\mathcal{L}_D(h) := \mathbb{E}_{S\sim D^n}[\mathcal{L}_S(h)]$.

One standard formalization, called Probably Approximately Correct (PAC) learning, characterizes what it means to successfully learn such a hypothesis. A learning task is parameterized by a *concept class* H (alternatively, *hypothesis class*), a set of *hypotheses* $h : X \to \{0, 1\}$. The learner is given a set of examples S drawn from an unknown distribution D where each example is labeled by a *target* function $c : X \to \{0, 1\}$, taking the form $(x, c(x))$. The goal is to find a hypothesis h with low generalization error $\mathcal{L}_D(h)$ with high probability over the randomness of the sample. So that the learning problem may be computationally tractable, we assume that the representation of elements of X may be computed in time $O(n)$, and similarly for each $c \in H$ it has finite representation size, denoted size(c). We may now define PAC learning:

Definition 1 (PAC learning [75])**.** *A concept class H is* PAC learnable *if there exists an algorithm $\mathcal{A}$ and a polynomial poly$(\cdot, \cdot, \cdot, \cdot)$ such that for any $\epsilon, \delta > 0$, for any distribution D on X, for any target $c \in H$, on an input labeled sample of size $m \geq$ poly$(1/\epsilon, 1/\delta, n,$ size$(c))$, $\mathcal{A}$ produces a hypothesis h such that*

$$\mathbb{P}_{S\sim D^m}[\mathcal{L}_D(h) \leq \epsilon] \geq 1 - \delta.$$

Furthermore, if $\mathcal{A}$ *runs in time* $poly(1/\epsilon, 1/\delta, n, size(c))$*, then* H *is said to be* efficiently *PAC learnable.*

In this thesis, we will only be considered with efficient PAC learnability (and may drop the word 'efficient' for convenience). Furthermore, we will primarily be concerned with *proper* PAC learning, where the returned hypothesis must come from the original concept class.

For what kind of classes is PAC learning possible? Ignoring computational efficiency, there is a nice combinatorial property of concept classes which exactly characterizes learnability. First, we say a finite set $S \subset X$ is *shattered* by a hypothesis class H if the restriction of H to $S = \{x_1, \ldots, x_m\}$ (that is, $\{(c(x_1), c(x_2), \ldots, c(x_m)) : c \in H\}$) is the set of all functions from S to $\{0, 1\}$.

Definition 2 (VC dimension [76])**.** *The VC dimension of a hypothesis class* H*, denoted* $\mathrm{VC}(H)$*, is the maximal size of a set* $S \subset X$ *that can be shattered by* H*. If not such maximal size exists, then we define* $VC(H) = \infty$*.*

The algorithm to PAC learn with finite VC dimension is commonly known as *empirical risk minimization*, which simply returns the hypothesis in the hypothesis class with the smallest empirical risk. This tactic works because of Occam's razor:

Theorem 3 (Occam's Razor [9])**.** *For every* $h \in H$*, for any distribution* D *and* $\delta > 0$*, we have with probability at least* $1 - \delta$ *over a sample of size* m*:*

$$|\mathcal{L}_D(h) - \mathcal{L}_S(h)| \leq O\left(\frac{1}{\delta}\sqrt{\frac{\mathrm{VC}(H)\log(m/\mathrm{VC}(H))}{m}}\right).$$

This thesis, however, will also be concerned with computational efficiency, wherein finite VC dimension is not sufficient for efficient PAC learning.

2.2 Differential privacy

Differential privacy was invented by Dwork et al. [21] as a semantic notion of algorithmic privacy. The idea is that an adversary should not gains significantly more knowledge about your personal information from a database if you choose to include your data in the data set versus if you choose not to include your data in the data set.

Definition 4 (Differential privacy)**.** *Let $\mathcal{M} : X^n \to Z$ a randomized algorithm. We call $\mathcal{M}$* (ϵ, δ)-differentially private *if for every two samples $S, S' \in X^n$ differing on one instance, and every measurable $z \subset Z$,*

$$\mathbb{P}[\mathcal{M}(S) \in z] \leq e^{\epsilon} \cdot \mathbb{P}[M(S') \in z] + \delta.$$

If $\mathcal{M}$ is $(\epsilon, 0)$-private, we may simply call it ϵ-private.

Differential privacy comes with several guarantees useful for developing new mechanisms.

Proposition 5 (Adaptive composition; [23])**.** *Given parameters $0 < \epsilon < 1$ and $\delta > 0$, to ensure $(\epsilon, k\delta' + \delta)$-privacy over k adaptive mechanisms, it suffices that each mechanism is (ϵ', δ')-private, where $\epsilon' = \frac{\epsilon}{2\sqrt{2k \log(1/\delta)}}$.*

We also have a post-processing guarantee:

Lemma 6 (Post-processing; [22]). *Let $\mathcal{M} : X^n \to Z$ be an (ϵ, δ)-private mechanism and $f : Z \to Z'$ a (possibly randomized) algorithm. Then $f \circ \mathcal{M}$ is (ϵ, δ)-private.*

In this thesis, we use two well-established differentially-private mechanisms: the Laplace and exponential mechanisms.

The Laplace mechanism provides a way to output any real-valued function, with noise calibrated to sensitivity. The ℓ_1 *sensitivity* of a function $f : X^n \to \mathbb{R}^k$ is $\Delta f := \max_{d(S,S')=1} \|f(S) - f(S')\|_1$, where $d(S, S')$ is the number of elements on which S and S' differ. X is the arbitrary domain from which examples come, and f takes as input a sample from X.

In the Laplace mechanism, we add noise drawn from a carefully picked distribution to $f(S)$. That distribution is the Laplace distribution, whose probability density function with parameter b is $\text{Lap}_b(x) := \frac{1}{2b} e^{-|x|/b}$.

Definition 7 (Laplace mechanism [21]). *Given privacy parameter ϵ and $f : X^n \to \mathbb{R}^k$, the Laplace mechanism is then just*

$$\mathcal{M}(S) := f(S) + (Y_1, \ldots, Y_k),$$

where each Y_i is i.i.d. drawn from $Lap_{\Delta f/\epsilon}$.

The Laplace mechanism is indeed a private mechanism. Moreover, it is also an accurate one, in the following sense:

Proposition 8 (21). *The Laplace mechanism $\mathcal{M}$ is ϵ-private. In addition, for*

any $\delta > 0$,

$$\mathbb{P}\left[\|f(S) - \mathcal{M}(S)\|_\infty \geq \frac{\Delta f \ln(k/\delta)}{\epsilon}\right] \leq \delta.$$

The exponential mechanism is a generalization of the Laplace mechanism to arbitrary domains. In the Laplace mechanism, it is straight-forward to define accuracy: the output of the mechanism must be close to the true value of f. For arbitrary domains $\mathcal{R}$, we must instead be given a utility function $u : X^n \times \mathcal{R} \to \mathbb{R}$, which says for any given sample and possible element to output, how desirable is it to output this element. We define sensitivity of a utility function as

$$\Delta u := \max_{r \in \mathcal{R}} \max_{d(S,S')=1} |u(S, r) - u(S', r)|.$$

The exponential mechanism outputs $r \in \mathcal{R}$ with probability proportional to its utility, so that we're more likely to output more useful elements, but still maintain privacy.

Definition 9 (Exponential mechanism [52])**.** *The exponential mechanism outputs an element* $r \in \mathcal{R}$ *with probability*

$$\frac{\exp\left(\frac{\epsilon u(S,r)}{2\Delta u}\right)}{\sum_{r' \in \mathcal{R}} \exp\left(\frac{\epsilon u(S,r')}{2\Delta u}\right)}.$$

The exponential mechanism is also ϵ-private [52].

For a more complete accounting of differential privacy, see Dwork and Roth [22].

2.3 Complexity theory

To discuss computational efficiency, we use standard notions from computational complexity theory, so we provide a very brief summary here.

We use as our model computation the Turing machine, which we will not define precisely here. For a precise definition, as well as a much more complete introduction to complexity theory, we refer the reader to Arora and Barak [3]. Briefly, a Turing machine consists of a tape on which symbols are written and a list of of instructions, along with a read/write head that looks at the current symbol and uses the appropriate instruction to change the symbol and move along the tape. The idea is that we can measure how long an algorithm takes on a Turing machine by the number of symbols it reads before stopping on a given input.

A *decision problem*, or *language*, is a subset of $\{0,1\}^*$, the set of all finite binary strings, and a language is *decided* by a Turing machine when that machine accepts an input string if and only if it is in the language. A *complexity class* is a set of languages, typically specified by a constraint on resources available to the Turing machines. In this thesis we will primarily be concerned with classes constrained by the amount of available time.

Definition 10 (P and NP)**.** *A language L is in* $\mathrm{TIME}(T(n))$ *for $T : \mathbb{N} \to \mathbb{N}$ if a Turing machine decides L in $O(T(n))$ time. We define* P *as* $\cup_{c=1}^{\infty} \mathrm{TIME}(n^c)$.

A language L is in NP *if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M, in the sense above, such that for every $x \in \{0,1\}^*$, $x \in L$ if and only if there is some $u \in \{0,1\}^{p(|x|}$ such that $M(x,u) = 1$, i.e. M accepts (x,u).*

We also consider the class of languages in NP that are at least as hard as any problem in NP, called NP-hard. That is, there is a reduction which says that if an NP-hard language can be decided by some Turing machine, then every problem in NP can be decided using that algorithm, plus only polynomial additional time. That no NP-hard language is in P is the famous conjecture P $\neq$ NP. We will also use a variant of this conjecture, that RP $\neq$ NP, where RP is the class of NP problems where if the correct answer is no, then the polynomial-time Turing machine always rejects but if the answer is yes, then it accepts with probability at least 1/2.

In addition to decision problems, we are also interested in the running time for *counting* problems, where the goal is to output the number of solutions to a problem rather than just if there is one or not.

Definition 11 (#P)**.** *A function $f : \{0,1\}^* \to \mathbb{N}$ is in* #P *if there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$,*

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} : M(x,y) = 1\}|.$$

Defined analogously to NP-hardness, we can define #P-hardness as the #P problems at least as hard as any #P problem. The analogous conjecture to P vs. NP for #P is that the class of counting problems that can be solve in polynomial time FP is not equal to #P.

2.4 Graphs

One of the structures we will be interested in learning from data is a graph, so we briefly review a few basic definitions here. A *graph* $G = (V, E)$ consists of a vertex set V of size n, and an edge set E of m edges. Each edge $e = (u, v)$ is a tuple consisting of two vertices $u, v \in V$, wherein u and v are called *adjacent*. In an *undirected* graph, the kind of graph we focus on, (u, v) is treated as an unordered pair. A standard representation of a graph is the *adjacency matrix*, an $n \times n$ matrix indexed by the vertices, where an entry (u, v) is 1 if $(u, v) \in E$, and 0 otherwise.

The *neighborhood* $\Gamma(v)$ of a vertex v is the set of vertices adjacent to v. The *degree* of a vertex is the size of its neighborhood. A *complete* graph is a graph where every vertex has maximum degree. An (induced) *subgraph* G' of G on a subset of vertices $V' \subseteq V$ is the graph with vertex set V' and has edge (u, v) if and only if (u, v) is an edge of G. So a *clique* in a graph is a subset of vertices whose induced subgraph is complete.

3

Sublinear-Time Adaptive Data Analysis

3.1 Introduction

The field of data analysis seeks out statistically valid conclusions from data: inferences that generalize to an underlying distribution rather than specialize to the data sample at hand. As a result, classical proofs of statistical efficiency have focused on independence assumptions on data with a pre-determined sequence of analyses [48]. In practice, most data analysis is adaptive: previous inferences

This chapter is based on the preprint Fish et al. [27].

inform future analysis. This adaptivity is nigh impossible to avoid when multiple scientists contribute work to an area of study using the same or similar data sets. Unfortunately, adaptivity may lead to 'false discovery,' where the dependence on past analysis may create pervasive overfitting—also known as 'the garden of forking paths' or 'p hacking' [30]. While basing each analysis on new data drawn from the same distribution might appear an appealing solution, repeated data collection and analysis time can be prohibitively costly.

There has been much recent progress in minimizing the amount of data needed to draw generalizable conclusions, without having to make any assumptions about the type of adaptations used by the data analysis. However, the results in this burgeoning field of adaptive data analysis have ignored bootstrapping and related sampling techniques, even though these have enjoyed widespread and successful use in practice in a variety of settings [47, 81], including in adaptive settings [32]. This is a gap that not only points to an unexplored area of theoretical study, but also opens up the possibility of creating substantially faster algorithms for answering adaptively generated queries.

In this chapter, we aim to do just this: we develop strong theoretical results that are significantly faster than previous approaches, and we open up a host of interesting open problems at the intersection of sublinear-time algorithm design and this important new field. For example, sublinear-time algorithms are a necessary component to establish non-trivial results in property testing. We also enable the introduction of anytime algorithms in adaptive data analysis, by defining mechanisms that provide guarantees on accuracy when the time allotted

is restricted.

As in previous literature, a mechanism $\mathcal{M}$ is given an i.i.d. sample S of size n from an unknown distribution D over a finite space X, and is supplied queries of the form $q : D \to \mathbb{R}$. After each query, the mechanism must respond with an answer a that is close to $q(D)$ up to a parameter α with high probability. Furthermore, each query may be adaptive: The query may depend on the previous queries and answers to those queries.

In previous work, the mechanisms execute in $\Omega(n)$ time per query. In this work, we introduce mechanisms that deliver an exponential improvement on this bound. Remarkably, we show that these results come at almost no tradeoff—we can obtain these improvements in running time and yet use essentially the same sample sizes.

3.1.1 Motivation and results

Our results are summarized in Table 3.1. Our first result, in Section 3.3, is a method to answer low-sensitivity queries (defined in Section 3.2) that still has $n = \tilde{O}(\sqrt{k}/\alpha^2)$ sample complexity (as in previous work) but takes only $\tilde{O}(\log^2(k)/\alpha^2)$ time per query instead of $\tilde{O}(n)$ time per query as in previous approaches (Theorem 18). Moreover, our mechanism to answer a query is simple: given a database S, we first sample ℓ points i.i.d. from S, compute the empirical mean of q on that subsample, and then add Laplacian noise, which guarantees a *differentially-private* mechanism. The intuition behind this approach is that sampling offers two benefits: it can decrease computation time while simultaneously

query type	sample complexity		samples per query	
	previous work [4]	this work	previous work [4]	this work
low-sensitivity queries (Section 3.3)	$\tilde{O}\left(\frac{\sqrt{k}}{\alpha^2}\right)$	$\tilde{O}\left(\frac{\sqrt{k}}{\alpha^2}\right)$	$\tilde{O}\left(\frac{\sqrt{k}}{\alpha^2}\right)$	$\tilde{O}\left(\frac{\log(k)}{\alpha^2}\right)$
sampling counting queries (Section 3.4)	—	$\tilde{O}\left(\log\left(\frac{k}{\alpha}\right)\right)$	—	$\tilde{O}\left(\frac{\log(k)}{\alpha^2}\right)$
convex optimization (Section 3.6)	$\tilde{O}\left(\frac{\sqrt{dk}}{\alpha^2}\right)$	$\tilde{O}\left(\frac{d^{3/2}\sqrt{k}}{\alpha^5}\right)$	$\tilde{O}\left(\frac{dk}{\alpha^4}\right)$	$\tilde{O}\left(\frac{d^2\log(k)}{\alpha^5}\right)$
strongly convex optimization (Section 3.6)	$\tilde{O}\left(\frac{\sqrt{dk}}{\alpha^{3/2}}\right)$	$\tilde{O}\left(\frac{d^{3/2}\sqrt{k}}{\alpha^{5/2}}\right)$	$\tilde{O}\left(\frac{dk}{\alpha^3}\right)$	$\tilde{O}\left(\frac{d^2\log(k)}{\alpha^3}\right)$

query type	iterations per query	
	previous work [4]	this work
convex optimization (Section 3.6)	$\tilde{O}\left(\frac{dk}{\alpha^4}\right)$	$\tilde{O}\left(\frac{\log(k)}{\alpha^2}\right)$
strongly convex optimization (Section 3.6)	$\tilde{O}\left(\frac{dk}{\alpha^3}\right)$	$\tilde{O}\left(\frac{\log(k)}{\alpha}\right)$

Table 3.1: Summary of our results. k is the number of queries and α is the accuracy rate. Dependence on the probability of failure has been suppressed for ease of reading. Above the double line are our more general results and below are their applications to convex optimization. Note that this table does not show the slightly different assumptions made in previous work versus this work for convex optimization. For more precise definitions, see Section 3.2.

boosting privacy. Privacy yields a strong notion of stability, which in turn allows us to reduce the computation time without sacrificing accuracy.

In particular, this mechanism takes only $\tilde{O}(\log^2(k)/\alpha^2)$ time per query and a sample size of $\ell = \tilde{O}(\log(k)/\alpha^2)$, all while matching the established sample complexity bound $\tilde{O}(\sqrt{k}/\alpha^2)$. Even in the non-adaptive case, it must take $\Omega(\log(k)/\alpha^2)$ samples to answer k such queries [4]. This means our results are tight in ℓ and come close to matching the best known lower bound for the time complexity of answering such queries adaptively, which is simply $\Omega(\log(k)/\alpha^2)$. We show that this holds both when using uniform sampling with and without replacement.

While sampling in this manner requires examining $\ell = \tilde{O}(\log(k)/\alpha^2)$ samples per query, an analyst may wish to control the number of samples used. For example, the analyst might seek the answer to a counting query using a very small number of sample points from the database, even just a single sample point. The above methods cannot handle this case gracefully because when ℓ is sufficiently small, the guarantees on accuracy (using Definition 13 below) become trivial—we get only that $\alpha = O(1)$, which any mechanism will satisfy. Instead, we want the mechanism to have to return a statistically-meaningful reply even if $\ell = 1$. Indeed, the empirical answer to such a query is $\{0, 1\}$-valued, while a response using Laplacian noise will not be.

To address these issues, we consider an 'honest' setting where the mechanism must always yield a plausible reply to each query (Section 3.4). This is analogous to the honest version [82] of the statistical query (SQ) setting for learning [6, 44],

or the 1-STAT oracle for optimization [24]. Thus we introduce *sampling counting queries*, which imitate the process of an analyst requesting the value of a query on a single random sample. This allows for greater control over how long each query takes, in addition to greater control over the outputs. Namely, we require that for a query of the form $q : X \to \{0, 1\}$, the mechanism must output a $\{0, 1\}$-valued answer that is accurate in expectation. We show how to answer queries of this form by sampling a single point x from S and then applying a simple differentially-private algorithm to $q(x)$ that has not been used in adaptive data analysis prior to this work (Theorem 24). In Section 3.5, we compare sampling counting queries to counting queries.

Finally, to demonstrate the applicability of our general results, we use them as a black-box technique to obtain improved bounds for convex optimization (Section 3.6). In particular, we introduce a procedure for adaptive gradient descent that uses our sampling mechanism for low sensitivity queries to compute gradients. For answering k convex optimization queries, we improve the per-query sample complexity of $O(\sqrt{k})$ from Bassily et al. [4] to $O(\log k)$ in this work. That is, while the overall sample complexity does not improve over previous work, the number of samples that need to be examined for each query does. We also similarly decrease the number of iterations of gradient descent per query. (Note, however, 4 make slightly different assumptions about the loss function than we do.) While a nontrivial advance on its own, this contribution also points to the applicability of our general methods.

3.1.2 Previous work

Previous work in this area has focused on finding accurate mechanisms with low sample complexity (the size of S) for a variety of queries and settings [4, 18, 20, 63, 70]. Most applicable to our work is that of [4] who consider, among others, *low-sensitivity queries*, which are merely any function of X^n whose output does not change much when the input is perturbed (for a more precise definition, see below). If the queries are nonadaptive, then only roughly $\log(k)/\alpha^2$ samples are needed to answer k such queries. And if the queries are adaptive but the mechanism simply outputs the empirical estimate of q on S, then the sample complexity is much worse—order k/α^2 instead.

In this chapter, we will focus only on computationally efficient mechanisms. It is not necessarily obvious that it is possible to achieve a smaller sample complexity for an efficient mechanism in the adaptive case, but Bassily et al. [4], building on the work of Dwork et al. [20], provide a mechanism with sample complexity $n = \tilde{O}(\sqrt{k}/\alpha^2)$ to answer k low-sensitivity queries. Furthermore, for efficient mechanisms, this bound is tight in k [71]. Bassily et al. [4] also show how to efficiently answer *convex optimization queries*, which ask for the minimizer of a convex loss function, using a (private) gradient descent algorithm of Bassily et al. [5].

This literature shows that the key to finding such mechanisms with this quadratic improvement over the naive method is finding stable mechanisms: those whose output does not change too much when the sample is changed by a single element. Much of this literature leverages differential privacy [4, 18, 20, 70], which offers a

strong notion of stability. This work uses differentially-private mechanisms after sampling, as we are acutely interested in the impact on privacy when sampling. In both theory and practice, sampling in settings where privacy matters has long been deemed useful [42, 43, 45].

In our setting, we need an efficient uniform sampling method that not only maintains privacy, but actually boosts it. In particular, for an ϵ-private mechanism on a database of size n, we want to show that if you sample ℓ points uniformly and efficiently from those n points, and then apply the same mechanism, the result is $O\left(\frac{\ell}{n}\epsilon\right)$-private.

Fortunately, folklore has it that sampling boosts privacy, implicitly in Kasiviswanathan et al. [43], and certainly explicitly in the work of Lin et al. [50], who show that in their particular setting sampling without replacement boosts privacy to the degree we require. We note that their proof method easily generalizes to arbitrary domains and ϵ-private mechanisms. In addition, Bun et al. [11] show that sampling with replacement also boosts privacy.

3.2 Model and preliminaries

In the adaptive data analysis setting we consider, a (possibly stateful) mechanism $\mathcal{M}$ that is given an i.i.d. sample S of size n from an unknown distribution D over a finite space X. The mechanism $\mathcal{M}$ must answer queries from a stateful adversary $\mathcal{A}$. These queries are adaptive: $\mathcal{A}$ outputs a query q_i, to which the mechanism returns a response a_i, and the outputs of $\mathcal{A}$ and $\mathcal{M}$ may depend on all queries $q_1, \ldots, q_{i-1}$ and responses $a_1, \ldots, a_{i-1}$.

3.2.1 Low-sensitivity queries and optimization queries

In this work, the first type of query we consider is a *low-sensitivity query*, which is specified by a function $q : X^n \to \mathbb{R}$ with the property that for all samples $S, S' \in X^n$ where S and S' differ by at most one element, we have $|q(S) - q(S)'| \leq 1/n$, where we define $q(D) = \mathbb{E}_{S \sim D^n}[q(S)]$. (We can generalize to Δ-sensitive queries where $|q(S) - q(S)'| \leq \Delta$, but for simplicity we state all of our results with $\Delta = 1/n$.) We can now define the accuracy of $\mathcal{M}$.

Definition 12. *A mechanism $\mathcal{M}$ is said to be* (α, β)-accurate *over a sample S on low-sensitivity queries $q_1, \ldots, q_k$ if for its responses $a_1, \ldots, a_k$ we have*

$$\mathbb{P}_{\mathcal{M},\mathcal{A}}\left[\max_i |q_i(S) - a_i| \leq \alpha\right] \geq 1 - \beta.$$

The key requirement is stronger. Namely, we seek accuracy over the unknown distribution.

Definition 13. *A mechanism $\mathcal{M}$ is* (α, β)-accurate *over distribution D on low-sensitivity queries $q_1, \ldots, q_k$, if for its responses $a_1, \ldots, a_k$ we have*

$$\mathbb{P}_{\mathcal{M},\mathcal{A}}\left[\max_i |q_i(D) - a_i| \leq \alpha\right] \geq 1 - \beta.$$

In this work, we not only desire (α, β)-accuracy but we also want to consider the time per query taken by $\mathcal{M}$. In this work, we assume we will have oracle access to q, which will compute $q(x)$ for a sample point x in unit time (and also $q(S)$ in at most $O(|S|)$ time). This is not a strong assumption: If the queries can

be computed efficiently, then this can add only at most a poly-log factor overhead in n and $|X|$ (as long as we only compute q on a roughly $\log(n)$ size sample, which will turn out to be exactly the case).

We also consider optimization queries. In convex optimization, we have a loss function $\mathcal{L} : X^n \times \Theta \to \mathbb{R}$ defined over a convex set $\Theta \subseteq \mathbb{R}^d$ and a sample from X^n drawn from a distribution D, and the goal is to output $\theta \in \Theta$ that minimizes the expected loss, i.e. such a query is defined as

$$q(D) := \arg\min_{x \in \Theta} \mathbb{E}_{S \sim D^n}[\mathcal{L}(S, x)].$$

We measure accuracy of the response a_i by the expected regret: A mechanism is (α, β)-accurate on optimization queries each specified by a loss function $\mathcal{L}_i$ with respect to a distribution D if

$$\mathbb{P}_{\mathcal{M},\mathcal{A}}\left[\max_i \mathbb{E}_S\left[\mathcal{L}_i(S, a_i) - \min_{x \in \Theta} \mathcal{L}_i(S, x)\right] \leq \alpha\right] \geq 1 - \beta.$$

We will assume that $\mathcal{L}$ is convex in x. We will also consider the special case when $\mathcal{L}$ is strongly convex in x. A function $\mathcal{L}$ is *H-strongly convex* if for all x, y in Θ,

$$\mathcal{L}(y) \geq \mathcal{L}(x) + \langle \nabla \mathcal{L}(x), y - x \rangle + \frac{H}{2}\|y - x\|_2^2.$$

3.2.2 Counting queries and sampling counting queries

In this work we consider the special case of *counting queries*, which ask the question "What proportion of the data satisfies property q?" Counting queries are

a simple and important restriction of low-sensitivity queries [8, 12, 70]. More formally, a counting query is specified by a function $q : X \to \{0,1\}$, where $q(S) = \frac{1}{|S|}\sum_{x\in S} q(x)$ and $q(D) = \mathbb{E}_{x\sim D}[q(x)]$. As in the low-sensitivity setting, an answer to a counting query must be close to $q(D)$ (Definition 13).

This means, however, that answers will not necessarily be counts themselves, nor meaningful in settings where we require ℓ to be small, i.e. very few samples from the database. To this end, we introduce *sampling counting queries*. A sampling counting query (SCQ) is again specified by a function $q : X \to \{0,1\}$, but this time the mechanism $\mathcal{M}$ must return an answer $a \in \{0,1\}$. Given these restricted responses, we want such a mechanism to act like what would happen if $\mathcal{A}$ were to take a single random sample point x from D and evaluate $q(x)$. The average value the mechanism returns (over the coins of the mechanism) should be close to the expected value of q. More precisely, we want the following:

Definition 14. *A mechanism $\mathcal{M}$ is (α,β)-accurate on distribution D for k sampling counting queries q_i if for all states of $\mathcal{M}$ and $\mathcal{A}$, when $\mathcal{M}$ is given an i.i.d. sample S from D,*

$$\mathbb{P}_{S,\mathcal{M},\mathcal{A}}\left[\max_i |\mathbb{E}_{\mathcal{M}}[\mathcal{M}(q_i)] - q_i(D)| \leq \alpha\right] \geq 1-\beta.$$

We also define (α,β)-accuracy on a sample S from D analogously. Again, our requirement is that $\mathcal{M}$ be (α,β)-accurate with respect to the unknown distribution D, this time using only around $\log(n)$ time per query (and a constant number of samples per query).

3.2.3 The transfer theorem

A key method of Bassily et al. [4] for answering queries adaptively is a 'transfer theorem,' which states that if a mechanism is both accurate on a sample and differentially private, then it will be accurate on the sample's generating distribution.

Theorem 15 (4)**.** *Let $\mathcal{M}$ be a mechanism that on input sample $S \sim D^n$ answers k adaptively chosen low-sensitivity queries, is $(\frac{\alpha}{64}, \frac{\alpha\beta}{32})$-private for some $\alpha, \beta > 0$ and $(\frac{\alpha}{8}, \frac{\alpha\beta}{16})$-accurate on S. Then $\mathcal{M}$ is (α, β)-accurate on D.*

Their 'monitoring algorithm' proof technique involves a thought experiment in which an algorithm, called the monitor, assesses how accurately an input mechanism replies to an adversary, and remembers the query it does the worst on. It repeats this process some T times, and outputs the query that the mechanism does the worst on over all T rounds. Since the mechanism is private, so too is the monitor; and since privacy implies stability, this will ensure that the accuracy of the worst query is not too bad. For more details see Bassily et al. [4].

In order to prove our own transfer theorem for SCQ's, we will use some of the tools they developed. First, for a monitoring algorithm $\mathcal{W}$, the expected value of the outputted query on the sample will be close to its expected value over the distribution—formalizing a connection between privacy and stability.

Lemma 16 (4)**.** *Let $\mathcal{W} : (X^n)^T \to Q \times [T]$ be (ϵ, δ)-private where Q is the class of low-sensitivity queries. Let $S_i \sim D^n$ for each of $i \in [T]$ and $\mathbf{S} = \{S_1, \ldots, S_T\}$.*

Then

$$|\mathbb{E}_{\mathbf{S},\mathcal{W}}[q(D)|(q,t)=\mathcal{W}(\mathbf{S})]-\mathbb{E}_{\mathbf{S},\mathcal{W}}[q(S_t)|(q,t)=\mathcal{W}(\mathbf{S})]|\leq e^{\epsilon}-1+T\delta.$$

We will also use a convenient form of accuracy bound for the exponential mechanism.

Lemma 17 (4)**.** *Let $\mathcal{R}$ be a finite set, $f:\mathcal{R}\to\mathbb{R}$ a function, and $\eta>0$. Define a random variable X on $\mathcal{R}$ by $\mathbb{P}[X=r]=e^{\eta f(r)}/C$, where $C=\sum_{r\in\mathcal{R}}e^{\eta f(r)}$. Then $\mathbb{E}[f(X)]\geq\max_{r\in\mathcal{R}}f(r)-\frac{1}{\eta}\log|\mathcal{R}|$.*

3.3 Fast mechanisms for low-sensitivity queries

In this section, we provide simple and fast mechanisms for answering low-sensitivity queries. Our mechanism $\mathcal{M}$ for answering low-sensitivity queries comprises: Given a data set S of size n and query q, sample some ℓ points uniformly at random from S (with or without replacement), and call this new set S_ℓ. Then the mechanism returns $q(S_\ell)+\mathrm{Lap}\left(\frac{1}{\ell\epsilon''}\right)$, where $\mathrm{Lap}(b)$ refers to the zero-mean Laplacian distribution with scale parameter b, and ϵ'' is a carefully chosen privacy setting.

Algorithm 1 Fast mechanism for low-sensitivity queries

Parameters: Sub-sample size ℓ, target privacy parameters (ϵ,δ), number of queries k

Input: Sample S, query q

$S_\ell:=\{s_1,\ldots,s_\ell\}$, where $s_i\sim S$ uniformly at random (with or without replacement).

$\epsilon'':=\frac{cn\epsilon}{\ell\sqrt{k\log(1/\delta)}}$ (c a constant)

return $q(S_\ell)+\mathrm{Lap}\left(\frac{1}{\ell\epsilon''}\right)$.

We may now state our main theorem for mechanism $\mathcal{M}$, using suitable values for ϵ, δ, and ℓ.

Theorem 18. *When* $\ell \geq \frac{2\log(4k/\beta)}{\alpha^2}$ *for* k *low-sensitivity queries,*

1. $\mathcal{M}$ *takes* $\tilde{O}\left(\frac{\log(k)\log(k/\beta)}{\alpha^2}\right)$ *time per query.*
2. $\mathcal{M}$ *is* (α, β)*-accurate (on the distribution) so long as* $n = \Omega\left(\frac{\sqrt{k}\log k \cdot \log^{3/2}(\frac{1}{\alpha\beta})}{\alpha^2}\right)$.

Sampling with replacement takes $O(\log n)$ time per sample, for a total of $O(\ell \log n)$ time over ℓ samples. This suffices to prove part 1) for the values of ℓ and n given. It is also the case that sampling without replacement may take $O(\log n)$ time per sample, for a total of $O(\ell \log n)$ time over ℓ samples, in several settings. Again, this is sufficient, but may come at the cost of space complexity, e.g. by keeping track of which elements have not been chosen so far [80]. Alternatively, there are methods that enjoy optimal space complexity at the cost of worst-case running times, as in rejection sampling [77].

To prove part 2), we must establish that sampling boosts privacy. If sampling before a ϵ-private mechanism were to only deliver $O(\epsilon)$ instead of $O(\frac{\ell}{n}\epsilon)$ privacy then we would need $\ell > \frac{2\sqrt{2k\log(1/\delta)}\log(2k/\beta)}{\alpha\epsilon}$, which would be undesirable: ℓ then becomes the size of the entire database and sampling yields no time savings over computing $q(S)$ exactly. Fortunately, sampling can boost privacy:

Proposition 19 (Adapted from 50)**.** *Given a mechanism* $\mathcal{P} : X^\ell \to Y$, $\mathcal{M}$ *will be the mechanism that does the following: Sample uniformly at random* without *replacement* ℓ *points from an input sample* $S \in X^n$ *of size* n*, and call this set* S_ℓ.

Output $\mathcal{P}(S_\ell)$. *Then if* $\mathcal{P}$ *is* ϵ*-private, then* $\mathcal{M}$ *is* $\log(1 + \frac{\ell}{n}(e^\epsilon - 1)) = O\left(\frac{\ell}{n} \cdot \epsilon\right)$ *private for* $\ell \geq 1$.

For convenience, we provide a proof, based on [50], that sampling without replacement boosts privacy.

Proposition 19. Let S and S' be samples of size that differ only on the kth index. Consider the set of all subsamples of the indices $[n]$ of size ℓ:

$$\mathcal{R} = \{\pi : \pi = \{i_1, \ldots, i_k\} \subset [n]\}.$$

Under uniform sampling without replacement, we choose uniformly at random a subsample π from $\mathcal{R}$. For any index k, either k is in π or π differs in exactly one element from some subsample that includes k. In particular, any π not including k is distance one away from exactly ℓ subsamples with k: namely, the subsamples that replace each element of π with k. Abusing notation, we will identify the subsample of indices with the corresponding subsample of S (and likewise with S'), so that $\mathbb{P}[\mathcal{P}(\pi) = z|S]$ refers to the probability that mechanism $\mathcal{P}$ outputs z given the subsample π of S. Then for any output z in Y, where $d(\pi, \pi') = 1$

denotes two subsamples differing in exactly one element,

$$\begin{aligned}\mathbb{P}[\mathcal{M}(S) = z|S] &= \frac{1}{|\mathcal{R}|}\sum_{\pi\in\mathcal{R}}\mathbb{P}[\mathcal{P}(\pi) = z|S] \\ &= \frac{1}{|\mathcal{R}|}\left(\sum_{\pi\in\mathcal{R}:k\in\pi}\mathbb{P}[\mathcal{P}(\pi) = z|S] + \frac{1}{\ell}\sum_{\pi\in\mathcal{R}:k\in\pi}\sum_{\pi':k\notin\pi,d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S]\right) \\ &= \frac{1}{|\mathcal{R}|}\sum_{\pi\in\mathcal{R}:k\in\pi}\left(\mathbb{P}[\mathcal{P}(\pi) = z|S] + \frac{1}{\ell}\sum_{\pi':k\notin\pi,d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S]\right).\end{aligned}$$

That is,

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{M}(S) = z|S]}{\mathbb{P}[\mathcal{M}(S') = z|S']} &= \frac{\sum_{\pi\in\mathcal{R}:k\in\pi}\left(\mathbb{P}[\mathcal{P}(\pi) = z|S] + \frac{1}{\ell}\sum_{\pi':k\notin\pi,d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S]\right)}{\sum_{\pi\in\mathcal{R}:k\in\pi}\left(\mathbb{P}[\mathcal{P}(\pi) = z|S'] + \frac{1}{\ell}\sum_{\pi':k\notin\pi,d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S']\right)} \\ &\leq \max_{\pi\in\mathcal{R}:k\in\pi}\frac{\mathbb{P}[\mathcal{P}(\pi) = z|S] + \frac{1}{\ell}\sum_{\pi':k\notin\pi,d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S]}{\mathbb{P}[\mathcal{P}(\pi) = z|S'] + \frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi,\pi')=1}\mathbb{P}[\mathcal{P}(\pi') = z|S']},\end{aligned}$$

where we bound the ratio of sums by the maximum ratio. Name π^* the maximizer of this ratio. Now we bound the numerator and denominator via privacy:

Fix $\pi \in \mathcal{R}$ with $k \in \pi$. Firstly, we have $\mathbb{P}[\mathcal{P}(\pi) = z|S] \leq e^{\epsilon}\mathbb{P}[\mathcal{P}(\pi) = z|S']$. Secondly, for π' such that $k \notin \pi'$ but $d(\pi, \pi') = 1$ we have $\mathbb{P}[\mathcal{P}(\pi) = z|S] \leq e^{\epsilon}\mathbb{P}[\mathcal{P}(\pi') = z|S]$. Finally, $\mathbb{P}[\mathcal{P}(\pi') = z|S] = \mathbb{P}[\mathcal{P}(\pi') = z|S']$ since $k \notin \pi'$. Thus

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{M}(S)=z|S]}{\mathbb{P}[\mathcal{M}(S')=z|S']} &\leq \frac{\mathbb{P}[\mathcal{P}(\pi^*)=z|S]+\frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi^*,\pi')=1}\mathbb{P}[\mathcal{P}(\pi')=z|S]}{\mathbb{P}[\mathcal{P}(\pi^*)=z|S']+\frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi^*,\pi')=1}\mathbb{P}[\mathcal{P}(\pi')=z|S]} \\
&= 1+\frac{\mathbb{P}[\mathcal{P}(\pi^*)=z|S]-\mathbb{P}[\mathcal{P}(\pi^*)=z|S']}{\mathbb{P}[\mathcal{P}(\pi^*)=z|S']+\frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi^*,\pi')=1}\mathbb{P}[\mathcal{P}(\pi')=z|S]} \\
&\leq 1+\frac{\mathbb{P}[\mathcal{P}(\pi^*)=z|S]-e^{-\epsilon}\mathbb{P}[\mathcal{P}(\pi^*)=z|S]}{e^{-\epsilon}\mathbb{P}[\mathcal{P}(\pi^*)=z|S]+\frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi^*,\pi')=1}e^{-\epsilon}\mathbb{P}[\mathcal{P}(\pi^*)=z|S]} \\
&= 1+\frac{1-e^{-\epsilon}}{e^{-\epsilon}+\frac{1}{\ell}\sum_{\pi':k\notin\pi',d(\pi^*,\pi')=1}e^{-\epsilon}} \\
&= 1+\frac{\ell}{n}(e^{\epsilon}-1),
\end{aligned}$$

where the second inequality uses privacy and the last equality follows from the fact that there are $n-\ell$ such π' where $k\notin\pi'$ but $d(\pi^*,\pi')=1$. □

Sampling with replacement also boosts privacy:

Proposition 20 (11)**.** *Given a mechanism $\mathcal{P}: X^\ell \to Y$, $\mathcal{M}$ will be the mechanism that does the following: Sample uniformly at random* with *replacement ℓ points from an input sample $S\in X^n$ of size n, and call this set S_ℓ. Output $\mathcal{P}(S_\ell)$. Then if $\mathcal{P}$ is ϵ-private, then $\mathcal{M}$ is $\frac{6\epsilon\ell}{n}$-private for $\ell\geq 1$.*

We may now return to the main theorem:

Proof of Theorem 18. Since the Laplace mechanism receives a sample S_ℓ of size ℓ, output a_q can be bounded with the standard accuracy result for the Laplace mechanism ensuring ϵ''-privacy for any $\epsilon''>0$:

$$\mathbb{P}[|a_q-q(S_\ell)|\geq\alpha/2]\leq e^{-\frac{\alpha\epsilon''\ell}{2}}.$$

We can bound this above by $\frac{\beta}{2k}$ provided $\epsilon'' \geq \frac{\log(2k/\beta)}{\ell\alpha}$; and this follows from a Chernoff bound

$$\mathbb{P}[|q(S_\ell) - q(S)| \geq \alpha/2] \leq e^{-\frac{\alpha^2\ell}{2}}.$$

Once again we can bound this above by $\frac{\beta}{2k}$ so long as $\ell \geq \frac{2\log(4k/\beta)}{\alpha^2}$.

Thus we have, for all q, $\mathbb{P}[|a_q - q(S)| \geq \alpha] \leq \mathbb{P}[|a_q - q(S_\ell)| \geq \alpha/2] + \mathbb{P}[|q(S_\ell) + q(S)| \geq \alpha/2] \leq \beta/k$. The union bound immediately yields (α, β)-accuracy over all k queries. From Proposition 19, we also have $\left(\frac{\ell}{n}\epsilon''\right)$-privacy, where $\frac{\ell}{n}\epsilon'' = \frac{\log(2k/\beta)}{n\alpha}$. Equivalently, we have ϵ'-privacy when $n \geq \frac{\log(2k/\beta)}{\epsilon'\alpha}$. With adaptive composition (Proposition 5), we can answer k queries with (ϵ, δ)-privacy when $\epsilon' = \frac{\epsilon}{2\sqrt{2k\log(1/\delta)}}$, resulting in (α, β)-accuracy and (ϵ, δ)-privacy on S so long as $n > \frac{2\sqrt{2k\log(1/\delta)}\log(2k/\beta)}{\alpha\epsilon}$. The proof is concluded by applying Theorem 15. □

We also have a version of this theorem in expectation, which will require a new version of the transfer theorem, stated now:

Theorem 21. *Consider any possibility for the simulation between $\mathcal{A}$ and $\mathcal{M}$ up to the first $t-1$ rounds. Denoting the expectation while conditioning on any such possibility $E_{t-1}[\cdot]$, we have for any round $i \geq t$, if $\mathcal{M}$ is $(\alpha/8, \alpha/4)$-private for $\alpha \leq 1$, and $E_{t-1,S,\mathcal{M},\mathcal{A}}[|q_i(S) - a_i|] \leq \alpha/2$, then*

$$E_{t-1,S,\mathcal{M},\mathcal{A}}[|q_i(D) - a_i|] \leq \alpha.$$

Proof. Suppose by way of contradiction that $\mathbb{E}_{t-1,S,\mathcal{M},\mathcal{A}}[|q_i(D) - a_i|] > \alpha$. Note the monitor $\mathcal{W}$, given in Algorithm 2, simply outputs q_i, conditioned on $q_1, \ldots, q_{t-1}$

Algorithm 2 Monitor $\mathcal{W}$

Parameters: Mechanisms $\mathcal{M}$ and $\mathcal{A}$, index i, and initial sequence of queries $q_1, \ldots, q_{t-1}$ and responses $a_1, \ldots, a_{t-1}$

Input: Sample S

Set the internal states of $\mathcal{M}(S)$ and $\mathcal{A}$ to be what they would be if the resulting simulation had produced $q_1, \ldots, q_{t-1}$ and $a_1, \ldots, a_{t-1}$.

Now simulate $\mathcal{M}(S)$ and $\mathcal{A}$ interacting starting in those states for $i - t + 1$ rounds. Let $q_t, \ldots, q_i$ be the resulting queries.

return q_i.

and $a_1, \ldots, a_{t-1}$ being the initial sequence of queries and responses, so

$$\begin{aligned}|\mathbb{E}_{S,\mathcal{W}}[q(D) - q(S)|q = \mathcal{W}(S)]| &= |\mathbb{E}_{t-1,S,\mathcal{M},\mathcal{A}}[q_i(S) - q_i(D)]| \\ &\geq |\mathbb{E}_{t-1,S,\mathcal{M},\mathcal{A}}[q_i(D) - a_i]| - |\mathbb{E}_{t-1,S,\mathcal{M},\mathcal{A}}[q_i(S) - a_i]| \\ &> \alpha - \alpha/2 = \alpha/2.\end{aligned}$$

Since the monitor $\mathcal{W}$ only outputs q_i, which is post-processing from a private mechanism $\mathcal{M}$, $\mathcal{W}$ remains $(\alpha/8, \alpha/4)$-private. Therefore by Lemma 16, $|\mathbb{E}_{S,\mathcal{W}}[q(D) - q(S)|q = \mathcal{W}(S)]| \leq e^{\epsilon} - 1 + \delta \leq \alpha/2$ with the above values of ϵ and δ for $\alpha \leq 1$. □

We can now state how our fast mechanism $\mathcal{M}$ performs with respect to this notion of accuracy.

Corollary 22. *With respect to any possible simulation between $\mathcal{A}$ and $\mathcal{M}$ up to the first $t - 1$ rounds, for any $i \geq t$,*

$$\mathbb{E}_{t-1,S,\mathcal{A},\mathcal{M}}[|a_i - q_i(D)|] \leq \tilde{O}\left(\frac{k^{1/4}}{\sqrt{n}} + \frac{1}{\sqrt{\ell}}\right).$$

This follows as in the proof of Theorem 18, this time using

$$\mathbb{E}_{t-1,S,\mathcal{A},\mathcal{M}}[|a_i - q_i(S)|] \leq \frac{\sqrt{k\log(1/\delta)}}{n\epsilon} + \frac{1}{\sqrt{\ell}}.$$

3.4 Sampling counting queries

We now turn to sampling counting queries. Unlike in the previous section, we cannot leverage an existing transfer theorem, so instead we establish a new one.

Theorem 23. *Let $\mathcal{M}$ be a mechanism that on input sample $S \sim D^n$ answers k adaptively chosen sampling counting queries, is $(\frac{\alpha}{64}, \frac{\alpha\beta}{16})$-private for some $\alpha, \beta > 0$ and $(\alpha/2, 0)$-accurate on S. Suppose further that $n \geq \frac{1024\log(k/\beta)}{\alpha^2}$. Then $\mathcal{M}$ is (α, β)-accurate on D.*

This allows us to answer sampling counting queries:

Theorem 24. *There is a mechanism $\mathcal{M}$ that satisfies the following:*

1. *$\mathcal{M}$ takes $\tilde{O}\left(\log\left(\frac{k\log(\frac{1}{\beta})}{\alpha}\right)\right)$ time per query.*
2. *$\mathcal{M}$ is (α, β)-accurate on k sampling counting queries, where*

$$n \geq \Omega\left(\max\left(\frac{\sqrt{k\log(\frac{1}{\alpha\beta})}}{\alpha^2}, \frac{\log(k/\beta)}{\alpha^2}\right)\right).$$

We prove our transfer theorem using the following monitoring algorithm, which takes as input T sample sets, and outputs a query with probability proportional to how far away the query is on the sample as opposed to the distribution.

Algorithm 3 Monitor with exponential mechanism $\mathcal{W}_D$

Parameters: Mechanisms $\mathcal{M}$ and $\mathcal{A}$, distribution D
Input: Set of samples $\mathbf{S} = \{S_1, \ldots, S_T\}$
  for t in $[T]$ **do**
    Simulate $\mathcal{M}(S_t)$ and $\mathcal{A}$ interacting.
    Let $q_{t,1}, \ldots, q_{t,k}$ be the queries of $\mathcal{A}$.
  end for
  Let $\mathcal{R} := \{(q_{t,i}, t)\}_{t\in[T], i\in[k]}$.
  Abusing notation, for each t and $i \in [k]$, consider the corresponding element $r_{t,i}$ of $\mathcal{R}$ and define the utility of $r_{t,i}$ as $u(\mathbf{S}, r_{t,i}) = |q_{t,i}(S_t) - q_{t,i}(D)|$.
  return $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\epsilon \cdot n \cdot u(\mathbf{S},r)}{2}\right)$.

Lemma 25. *If $\mathcal{M}$ is (ϵ, δ)-private for k queries, then $\mathcal{W}_D$ is $(2\epsilon, \delta)$-private.*

The idea is that $\mathcal{R}$ represents post-processing from the differentially-private $\mathcal{M}$, and outputting an element from $\mathcal{R}$ is achieved with the exponential mechanism, making the monitor $\mathcal{W}_D$ private.

Lemma 25. A single perturbation to $\mathbf{S}$ can only change one S_t, for some t. Then since $\mathcal{M}$ on S_t is (ϵ, δ)-private, $\mathcal{M}$ remains (ϵ, δ)-private over the course of the T simulations. Since $\mathcal{A}$ uses only the outputs of $\mathcal{M}$, $\mathcal{A}$ is just post-processing $\mathcal{M}$, and therefore it is (ϵ, δ)-private as well: releasing all of $\mathcal{R}$ remains (ϵ, δ)-private.

Since the sensitivity of u is $\Delta = 1/n$, the monitor is just using the exponential mechanism to release some $r \in \mathcal{R}$, which is ϵ-private. The standard composition theorem completes the proof. □

Given that the monitor is private, we can now bound the probability that the query that the monitor outputs on the sample are far away from the distribution

on both sides, if $\mathcal{M}$ is not accurate, by using both Lemmas 16 and 17, yielding the transfer theorem given in Theorem 23.

Theorem 23. Consider the results for simulating T times the interaction between $\mathcal{M}$ and $\mathcal{A}$. Suppose for the sake of contradiction that $\mathcal{M}$ is not (α, β)-accurate on D. Then for every i in $[k]$ and t in T, since $|\mathbb{E}_{\mathcal{M}}[\mathcal{M}(q_{t,i})] - q(S_t)| \leq \alpha/2$, we have

$$\mathbb{P}_{S_t,\mathcal{M},\mathcal{A}}\left[\max_i |q_{t,i}(S_t) - q_{t,i}(D)| > \alpha/2\right] > \beta.$$

Call some q and t that achieves the maximum $|q(S_t) - q(D)|$ over the T independent rounds of $\mathcal{M}$ and $\mathcal{A}$ interacting, as $\mathcal{W}_D$ does (Algorithm 3), by q_w and t_w. Since each round t is independent, the probability that $|q_w(S_{t_w}) - q_w(D)| \leq \alpha/2$ is then no more than $(1-\beta)^T$. Then using Markov's inequality immediately grants us that

$$\mathbb{E}_{\mathbf{S},\mathcal{W}_D}\left[|q_w(S_{t_w}) - q_w(D)|\right] > \frac{\alpha}{2}(1 - (1-\beta)^T). \tag{3.1}$$

Let $\Gamma = \mathbb{E}_{\mathbf{S},\mathcal{W}_D}\left[|q^*(S_{t^*}) - q^*(D)| : (q^*, t^*) = \mathcal{W}_D(\mathbf{S})\right]$.

Setting $f(r) = u(\mathbf{S}, r)$, Lemma 17 implies that under the exponential mechanism, we have

$$\begin{aligned}&\mathbb{E}[|q^*(S_{t^*}) - q^*(D)| : (q^*, t^*) = \mathcal{W}_D(\mathbf{S})] \\ &\geq |q_w(S_{t_w}) - q_w(D)| - \frac{2}{\epsilon n}\log(kT).\end{aligned}$$

Taking the expected value of both sides with respect to $\mathbf{S}$ and the randomness of

the rest of $\mathcal{W}_D$, we obtain

$$\begin{aligned}\Gamma &\geq \mathbb{E}_{\mathbf{S},\mathcal{W}_D}[|q_w(S_{t_w}) - q_w(D)|] - \frac{2}{\epsilon n}\log(kT) \\ &> \frac{\alpha}{2}(1-(1-\beta)^T) - \frac{2}{\epsilon n}\log(kT), \end{aligned} \tag{3.2}$$

which follows from employing Equation (3.1). On the other hand, suppose that $\mathcal{M}$ is (ϵ, δ)-private for some $\epsilon, \delta > 0$. Then by Lemma 25, $\mathcal{W}_D$ is $(2\epsilon, \delta)$-private, and then in turn Lemma 16 implies that

$$\Gamma \leq e^{2\epsilon} - 1 + T\delta. \tag{3.3}$$

We will now ensure $\Gamma \geq \alpha/8$, via (3.2), and $\Gamma \leq \alpha/8$, via (3.3), yielding a contradiction. Set $T = \lfloor \frac{1}{\beta} \rfloor$ and $\delta = \frac{\alpha\beta}{16}$. Then

$$e^{2\epsilon} - 1 + T\delta \leq e^{2\epsilon} - 1 + \alpha/16 \leq \alpha/8$$

when $e^{2\epsilon} - 1 \leq \alpha/16$, which in turn is satisfied when $\epsilon \leq \alpha/64$, since $0 \leq \alpha \leq 1$.

On the other side, $1 - (1-\beta)^{\lfloor \frac{1}{\beta} \rfloor} \geq 1/2$. Then it suffices to set ϵ such that $\frac{2}{\epsilon n}\log(kT) \leq \alpha/8$. Thus we need ϵ such that

$$\frac{16\log(k/\beta)}{\alpha n} \;\leq\; \epsilon \;\leq\; \alpha/64.$$

Such an ϵ exists, since we explicitly required $n \geq \frac{1024\log(k/\beta)}{\alpha^2}$. □

With a transfer theorem in hand, we now introduce a private mechanism that

is accurate on a sample for answering sampling counting queries.

Lemma 26 (SCQ mechanism)**.** *For $\epsilon \leq 1$, There is an (ϵ, δ)-private mechanism to release k SSQs that is $(\alpha, 0)$-accurate, for $\alpha \leq 1/2$, with respect to a fixed sample S of size n so long as $n > \frac{2\sqrt{2k\log(1/\delta)}}{\alpha\epsilon}$.*

Proof. We design a mechanism $\mathcal{M}$ to release a $(\alpha, 0)$-accurate SCQ for $n > \frac{1}{\alpha\epsilon}$ and then use Proposition 5. The mechanism is simple: sample x i.i.d. from S. Then release $q(x)$ with probability $1-\alpha$ and $1-q(x)$ with probability α. Let $i = \sum_{x\in S} q(x)$. Then $\mathbb{E}_{\mathcal{M}}[\mathcal{M}(q)] = \frac{(1-\alpha)i+\alpha(n-i)}{n} = \frac{i}{n} + \alpha\left(\frac{n-2i}{n}\right)$, so $\frac{i}{n} - \alpha \leq \mathbb{E}_{\mathcal{M}}[\mathcal{M}(q)] \leq \frac{i}{n} + \alpha$, implying that $\mathcal{M}$ is $(\alpha, 0)$-accurate on S.

Now let S' differ from S on one element x, where $q(x) = 0$ but for $x' \in S'$, $q(x') = 1$. Consider

$$\frac{\mathbb{P}[\mathcal{M}(S) = 1]}{\mathbb{P}[\mathcal{M}(S') = 1]} = \frac{(1-\alpha)\frac{i+1}{n} + \alpha(\frac{n-i+1}{n})}{(1-\alpha)\frac{i}{n} + \alpha(\frac{n-i}{n})} = 1 + \frac{1-2\alpha}{i - 2\alpha i + \alpha n},$$

for $i = 0$ to $i = n-1$. The other cases are similar. Note this is at least 1 since $1 - 2\alpha \geq 0$. Thus it suffices to show when this is upper-bounded by e^{ϵ}. By computing the partial derivative with respect to i, it is easy to see that it suffices to consider the cases when $i = 0$ or $i = n-1$. When $i = 0$,

$$\log\left(\frac{\mathbb{P}[\mathcal{M}(S) = 1]}{\mathbb{P}[\mathcal{M}(S') = 1]}\right) \leq \frac{1-2\alpha}{\alpha n} \leq \frac{1}{\alpha n} \leq \epsilon$$

when $n \geq \frac{1}{\epsilon\alpha}$. When $i = n - 1$,

$$\log\left(\frac{\mathbb{P}[\mathcal{M}(S) = 1]}{\mathbb{P}[\mathcal{M}(S') = 1]}\right) \leq \frac{1 - 2\alpha}{n(1-\alpha) - (1-2\alpha)} \leq \epsilon$$

when $n \geq \frac{(1-2\alpha)(\epsilon+1)}{(1-\alpha)\epsilon}$ but because $\frac{1-2\alpha}{1-\alpha} \leq 1$, it suffices to set $n \geq 1 + \frac{1}{\epsilon}$. The proof is completed by noting that $\frac{1}{\epsilon\alpha} \geq 1 + \frac{1}{\epsilon}$ because $\epsilon \leq 1$. □

We now use this mechanism to answer sampling counting queries.

Theorem 24. We use the mechanism of Lemma 26. This gives an (ϵ, δ)-private mechanism that is $(\alpha/2, 0)$-accurate so long as $n \geq \frac{4\sqrt{2k\log(1/\delta)}}{\alpha\epsilon}$. Setting ϵ and δ as required by Theorem 23 implies that we need $n \geq \Omega\left(\sqrt{k\log(\frac{1}{\alpha\beta})}/\alpha^2\right)$. Note to use Theorem 23 we also need $n \geq \Omega\left(\log(k/\beta)/\alpha^2\right)$. The sample complexity bound follows. This mechanism samples a single random point, which takes $O(\log(n))$ time, completing the proof. □

3.5 Comparing counting and sampling counting queries

How do our mechanisms for counting queries and sampling counting queries compare to each other? Can we use a mechanism for SCQ's to simulate a mechanism for counting queries, or vice-versa? We now show that the natural approach to simulate a counting query with SCQ's results in an extra $O(1/\alpha)$ factor (although it does enjoy a slightly better dependence on k). This represents a $O(1/\alpha)$ overhead in order to ensure that the mechanism returns meaningful results for all sample sizes ℓ.

Proposition 27. *Using ℓ SCQ's to estimate each counting query is an (α, β)-accurate mechanism for k counting queries if $\ell \geq \frac{2\log(4k/\beta)}{\alpha^2}$ and*

$$n = \Omega\left(\frac{\sqrt{k\log k}\log^{3/2}(\frac{1}{\alpha\beta})}{\alpha^3}\right).$$

Proof. The mechanism, for each query q, will query the SCQ mechanism $\mathcal{M}$ described in Section 3.4 ℓ times with the query q, and return the average, call this a_q. Note that $\mathbb{E}[a_q] = \mathbb{E}[\mathcal{M}(q)]$. Since each SCQ is independent of each other, a Chernoff bound gives $\mathbb{P}[|a_q - \mathbb{E}[a_q]| \geq \alpha/2] \leq 2e^{-\ell\alpha^2/2} \leq \beta/2k$ when $\ell \geq \frac{2\log(4k/\beta)}{\alpha^2}$. Using Theorem 24, as long as $n = \Omega\left(\frac{\sqrt{k\ell}\log(\frac{1}{\alpha\beta})}{\alpha^2}\right)$, we have that $\mathbb{P}[\max_q |\mathbb{E}[\mathcal{M}(q)] - q(D)| \geq \alpha/2] \leq \beta/2$, over all $k\ell$ queries. Then the union bound implies that

$$\begin{aligned}\mathbb{P}[\max_q |a_q - q(D)| \geq \alpha] &\leq \mathbb{P}[\max_q |a_q - \mathbb{E}[\mathcal{M}(q)]| + |\mathbb{E}[\mathcal{M}(q)] - q(D)| \geq \alpha] \\ &\leq \beta/2 + \beta/2 \leq \beta.\end{aligned}$$

Plugging in ℓ into the above expression for n completes the proof. □

Meanwhile, it is possible to use a mechanism for counting queries to attempt to answer SCQ's, but it has higher sample complexity than the mechanism for SCQ's proposed above. Indeed, there is the naive approach that ignores time constraints by first computing $q(S)$ exactly, adding noise to obtain a value $\tilde{a_q}$, and then returning 1 with probability $\tilde{a_q}$ and 0 otherwise. For this mechanism we obtain an (ϵ, δ)-private mechanism to release k SCQ's that is (α, β)-accurate with

respect to a fixed sample S of size n so long as

$$n > \frac{2\sqrt{2k\log(1/\delta)}\log(1/\beta)}{\alpha\epsilon},$$

which is strictly worse than the mechanism for SCQ's we actually use. This motivates our approach to SCQ's.

3.6 An application to convex optimization

Our mechanism for convex optimization will take advantage of our fast mechanism for low-sensitivity queries. We will perform straightforward gradient descent but we calculate each coordinate of each gradient via the mechanism described in Section 3.3. We'll use this mechanism to obtain an approximation of the gradient via the query $q_{t-1}(S) := \nabla\mathcal{L}(S, x_{t-1})^{(i)}$. The mechanism, recall, takes a random subsample S_ℓ and adds independent noise which we'll call b, so that $\tilde{\nabla}\mathcal{L}(S, x_{t-1})^{(i)} := \nabla\mathcal{L}(S_\ell, x_{t-1})^{(i)} + b_{i,t-1}$. When clear, we abbreviate $\tilde{\nabla}\mathcal{L}(S, x_t)$ as $\tilde{\nabla}\mathcal{L}(x_t)$, or just $\tilde{\nabla}_t$.

Algorithm 4 Gradient descent with an adaptive mechanism for gradients

Parameters: Mechanism $\mathcal{M}$, learning rate η
Input: number of rounds T, initial point x_0
  for t in $[T]$ **do**
    $q_{t-1,i}(S) := \nabla\mathcal{L}(S, x_{t-1})^{(i)}$
    $\tilde{\nabla}\mathcal{L}(S, x_{t-1}) := (\mathcal{M}(q_{t-1,i}, S))_{\{i\}}$
    $x_t := x_{t-1} - \eta\tilde{\nabla}\mathcal{L}(S, x_{t-1})$
  end for
  return $\frac{1}{T}\sum x_t$.

We first show that the expected excess risk $\mathbb{E}_{S,\mathcal{M},\mathcal{A}}[\mathcal{L}(S, x) - \min_{x\in\Theta}\mathcal{L}(S, x)]$ for

x the output of Algorithm 4 is small, for strongly convex functions. Here, we assume the gradient $\nabla\mathcal{L}(S, x)$ is a low sensitivity function in each of the coordinates of S.

Theorem 28. *Let $\mathcal{L}$ be differentiable, H-strongly convex, let $\nabla\mathcal{L}$ be low sensitivity, and for any $S' \subset S$ and $x \in \Theta$, $E[\|\nabla\mathcal{L}(S', x)\|^2] \leq G^2$. Then there is a mechanism that answers k such optimization queries each with expected excess risk α if $n = \tilde{O}\left(\frac{d^{3/2}\sqrt{k}}{\alpha^{5/2}}\right)$ in $\tilde{O}\left(\frac{d^2}{\alpha^3}\right)$ samples per query and $\tilde{O}\left(\frac{1}{\alpha}\right)$ iterations of gradient descent per query.*

Bounding regret here is similar to typical analyses, but is complicated by one major difference: A typical assumption in stochastic gradient descent is that the oracle returning the oracle for the gradient is unbiased, so that $\mathbb{E}[\tilde{\nabla}\mathcal{L}] = \nabla\mathcal{L}$ (e.g. in Shamir and Zhang [69]), whereas here $\mathbb{E}[\tilde{\nabla}\mathcal{L}]$ is only guaranteed to be close to the true gradient $\mathcal{L}$. We take advantage of (strong) convexity to show that for sufficiently large sample size, gradient descent still converges sufficiently quickly.

Proof. In order to answer k optimization queries, we use our low-sensitivity oracle to get each component of $\nabla\mathcal{L}$, for a total of $R = k \cdot T \cdot d$ rounds, where T is the number of iterations per optimization (Algorithm 4). For each optimization query, we now bound regret. As is standard, we pick $x^* = \arg\min_{x\in\Theta} \mathcal{L}(x)$ to plug in to strong convexity to get, rearranging,

$$\mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] \leq \mathbb{E}[\langle \nabla_t, x_t - x^* \rangle] - \frac{H}{2}\mathbb{E}[\|x_t - x^*\|^2].$$

Again following the standard analysis,

$$\begin{aligned}\|x_{t+1} - x^*\|^2 = \|\Pi(x_t - \eta_t\tilde{\nabla}_t) - x^*\| &\leq \|x_t - \eta_t\tilde{\nabla}_t - x^*\|^2 \\ &\leq \|x_t - x^*\|^2 + \eta_t^2\|\tilde{\nabla}_t\|^2 - 2\eta_t\langle\tilde{\nabla}_t, x_t - x^*\rangle.\end{aligned}$$

In other words,

$$\langle\tilde{\nabla}_t, x_t - x^*\rangle \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta_t} + \frac{\eta_t}{2}\|\tilde{\nabla}_t\|^2.$$

Moreover, we can upper-bound $\mathbb{E}[\|\tilde{\nabla}_t\|^2]$ since $\tilde{\nabla}_t = \nabla\mathcal{L}(S_\ell, x_t) + b_t$, where b_t is the noise vector.

$$\begin{aligned}\mathbb{E}[\|\tilde{\nabla}_t\|^2] &= \mathbb{E}[\|\nabla\mathcal{L}(S_\ell, x_t)\|^2] + \mathbb{E}[\|b_t\|^2] + 2\mathbb{E}[\langle\nabla\mathcal{L}(S_\ell, x_t), b_t\rangle] \\ &\leq G^2 + 2d\sigma^2 = G^2 + \frac{2dk\log(1/\alpha')}{n^2\alpha'^2},\end{aligned}$$

where σ^2 is the variance of the noise. Note $\mathbb{E}[\langle\nabla\mathcal{L}(S_\ell, x_t), b_t\rangle] = 0$ because b_t is independent of both S_ℓ and x_t.

Now, using the bounds on our oracle, we upper-bound $\langle\nabla_t, x_t - x^*\rangle$ using $\langle\tilde{\nabla}_t, x_t - x^*\rangle$.

Using $\mathbb{E}_{t-1}[\cdot]$ to denote the expectation conditioned on all of the previous $t-1$ iterations, the promise of our mechanism (Corollary 22) is that we can guarantee that for each coordinate i, $\mathbb{E}_{t-1}[\nabla_t^{(i)}] \leq \mathbb{E}_{t-1}[\tilde{\nabla}_t^{(i)}] + \alpha'$, where

$$\alpha' = \tilde{O}\left(\frac{R^{1/4}}{\sqrt{n}} + \frac{1}{\sqrt{\ell}}\right).$$

Then

$$\mathbb{E}[\langle \nabla_t, x_t - x^* \rangle] = \sum_i \mathbb{E}[\mathbb{E}_{t-1}[\nabla_t^{(i)}(x_t - x^*)^{(i)}]]$$

$$\leq \sum_i \mathbb{E}[\mathbb{E}_{t-1}[(\tilde{\nabla}_t^{(i)} + \alpha')(x_t - x^*)^{(i)}]] = \mathbb{E}[\langle \tilde{\nabla}_t, x_t - x^* \rangle] + \alpha' \mathbb{E}\left[\sum_i (x_t - x^*)^{(i)}\right]$$

$$\leq \mathbb{E}[\langle \tilde{\nabla}_t, x_t - x^* \rangle] + \alpha' \mathbb{E}[\|x_t - x^*\|_1] \leq \mathbb{E}[\langle \tilde{\nabla}_t, x_t - x^* \rangle] + \alpha' \sqrt{d}\, \mathbb{E}[\|x_t - x^*\|_2].$$

The first equality conditions on the first $t-1$ rounds and then expands the inner product. The first inequality follows because once we condition on the first $t-1$ rounds, ∇_t and x_t are independent, so we can use the mechanism's guarantee. $\tilde{\nabla}_t$ and x_t are also independent when conditioned on the first $t-1$ rounds, from which the second equality follows. The last inequality follows from Cauchy-Schwartz.

Note further that $\mathbb{E}[\|x_t - x^*\|_2] \leq 1 + \mathbb{E}[\|x_t - x^*\|_2^2]$, simply because either $\|x_t - x^*\|_2 \leq 1$ or $\|x_t - x^*\|_2 < \|x_t - x^*\|_2^2$. Thus

$$\mathbb{E}[\langle \nabla_t, x_t - x^* \rangle] \leq \mathbb{E}[\langle \tilde{\nabla}_t, x_t - x^* \rangle] + \alpha' \sqrt{d} + \alpha' \sqrt{d} \mathbb{E}[\|x_t - x^*\|^2].$$

Thus we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] \\
&\leq \sum_{t=1}^{T} \left(\frac{(1+\alpha'\sqrt{d})E[\|x_t - x^*\|^2] - \mathbb{E}[\|x_{t+1} - x^*\|^2]}{2\eta_t} - \frac{H}{2}\mathbb{E}[\|x_t - x^*\|^2] \right. \\
&\qquad \left. + \frac{\eta_t}{2}\left(G^2 + \frac{2dk\log(1/\alpha')}{n^2\alpha'^2}\right) + \alpha'\sqrt{d}\right) \\
&\leq \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}[\|x_t - x^*\|^2]\left(\frac{1+\alpha'\sqrt{d}}{\eta_t} - \frac{1}{\eta_{t-1}} - H\right) \\
&\qquad + \left(\frac{G^2}{2} + \frac{dk\log(1/\alpha')}{n^2\alpha'^2}\right)\left(\sum_{t=1}^{T}\eta_t\right) + \alpha'\sqrt{d}T.
\end{aligned}
$$

Now if we set $\eta_t = \frac{2}{Ht}$, then $\frac{1+\alpha'\sqrt{d}}{\eta_t} - \frac{1}{\eta_{t-1}} - H \leq 0$ when $\alpha'\sqrt{d} \leq 1/t$.

Then setting $\alpha'\sqrt{d} \leq \frac{1}{T}$, the average loss is

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] &\leq \frac{2}{HT}\left(\frac{G^2}{2} + \frac{dk\log(1/\alpha')}{n^2\alpha'^2}\right)\sum_{t=1}^{T} 1/t + \alpha'\sqrt{d} \\
&= \frac{G^2}{H}\cdot\frac{1+\log(T)}{T} + \frac{2dk\log(1/\alpha')}{Hn^2\alpha'^2}\cdot\frac{1+\log(T)}{T} + \alpha'\sqrt{d}.
\end{aligned}
$$

Thus to show that the average loss is no more than α requires us to provide that $\frac{G^2}{H}\cdot\frac{1+\log(T)}{T} \leq \alpha/3$, or $T = \tilde{O}\left(\frac{G^2}{\alpha H}\right)$. We also require $\alpha'\sqrt{d} \leq 1/T$, $\alpha'\sqrt{d} \leq \alpha/2$, and $\frac{2dk\log(1/\alpha')}{Hn^2\alpha'^2}\cdot\frac{1+\log(T)}{T} \leq \alpha/3$. Thus it suffices so that $n = \tilde{O}\left(\frac{G^5}{H^{5/2}}\cdot\frac{d^{3/2}\sqrt{k}}{\alpha^{5/2}}\right)$ and $\ell = \tilde{O}\left(\frac{G^4}{H^2}\cdot\frac{d}{\alpha^2}\right)$. Finally, the number of samples used per optimization query is $T\cdot d\cdot\ell = \tilde{O}\left(\frac{G^6}{H^3}\cdot\frac{d^2}{\alpha^3}\right)$. □

As is standard, we can boost this to a high-probability result by running the

gradient-descent algorithm $\log(k/\beta)$ times and choosing the best run among them.

Corollary 29. *This mechanism is (α, β)-accurate for k optimization queries, with the same assumptions as in Theorem 28, when $n = \tilde{O}\left(\frac{d^{3/2}\sqrt{k}\log(k/\beta)}{\alpha^{5/2}}\right)$ in $\tilde{O}\left(\frac{d^2\log(k/\beta)}{\alpha^3}\right)$ samples per query and $\tilde{O}\left(\frac{\log(k/\beta)}{\alpha}\right)$ iterations of gradient descent per query.*

We now give the equivalent result when the loss function is only guaranteed to be convex and not strongly convex.

Theorem 30. *Let $\mathcal{L}$ be differentiable and convex, let $\nabla\mathcal{L}$ be low sensitivity, for any $S' \subset S$ and $x \in \Theta$, $\mathbb{E}[\|\nabla\mathcal{L}(S', x)\|^2] \leq G^2$, and finally, for any $x, y \in \Theta$, $\|x - y\|^2 \leq D^2$. Then there is a mechanism that answers k such optimization queries each with expected excess loss α if $n = \tilde{O}\left(\frac{d^{3/2}\sqrt{k}}{\alpha^5}\right)$ in $\tilde{O}\left(\frac{d^2}{\alpha^5}\right)$ samples per query and $\tilde{O}\left(\frac{1}{\alpha^2}\right)$ iterations of gradient descent per query.*

Proof. The proof is very similar to that of the proof of Theorem 28, using the same algorithm, except now we only have

$$\mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] \leq \mathbb{E}[\langle \nabla_t, x_t - x^* \rangle].$$

But as before, we have

$$\mathbb{E}[\langle \nabla_t, x_t - x^* \rangle] \leq \mathbb{E}[\langle \tilde{\nabla}_t, x_t - x^* \rangle] + \alpha'\sqrt{d} + \alpha'\sqrt{d}\ \mathbb{E}[\|x_t - x^*\|^2],$$

$$\langle \tilde{\nabla}_t, x_t - x^* \rangle \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta_t} + \frac{\eta_t}{2}\|\tilde{\nabla}_t\|^2,$$

and

$$\mathbb{E}[\|\tilde{\nabla}_t\|^2] \leq G^2 + \frac{2dk\log(1/\alpha')}{n^2\alpha'^2}.$$

Then

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] \\
&\leq \sum_{t=1}^{T} \left(\frac{(1+\alpha'\sqrt{d})\mathbb{E}[\|x_t - x^*\|^2] - \mathbb{E}[\|x_{t+1} - x^*\|^2]}{2\eta_t} \right. \\
&\qquad \left. + \frac{\eta_t}{2}\left(G^2 + \frac{2dk\log(1/\alpha')}{n^2\alpha'^2}\right) + \alpha'\sqrt{d}\right) \\
&\leq \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}[\|x_t - x^*\|^2]\left(\frac{1+\alpha'\sqrt{d}}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \\
&\quad + \left(\frac{G^2}{2} + \frac{dk\log(1/\alpha')}{n^2\alpha'^2}\right)\left(\sum_{t=1}^{T}\eta_t\right) + \alpha'\sqrt{d}\cdot T. \\
&\leq \frac{D^2}{2\eta_T} + \frac{D^2\alpha'\sqrt{d}}{2}\sum_{t=1}^{T}\frac{1}{\eta_t} + \left(\frac{G^2}{2} + \frac{dk\log(1/\alpha')}{n^2\alpha'^2}\right)\left(\sum_{t=1}^{T}\eta_t\right) + \alpha'\sqrt{d}\cdot T,
\end{aligned}$$

where the last inequality comes from upper-bounding $\|x_t - x^*\|^2$ by the diameter, and collapsing the telescoping series. Set $\eta_t = \frac{D}{G\sqrt{t}}$. This gives the average loss as

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\mathcal{L}(x_t) - \mathcal{L}(x^*)] &\leq \frac{DG}{2\sqrt{T}} + \frac{DG\alpha'\sqrt{dT}}{2} + \frac{DG}{2\sqrt{T}} + \frac{Ddk\log(1/\alpha')\sqrt{T}}{Gn^2\alpha'^2} + \alpha'\sqrt{d} \\
&= \frac{DG}{\sqrt{T}} + \frac{DG\alpha'\sqrt{dT}}{2} + \frac{Ddk\log(1/\alpha')\sqrt{T}}{Gn^2\alpha'^2} + \alpha'\sqrt{d}.
\end{aligned}$$

It suffices to show that each of these four terms are upper-bounded by $\alpha/4$, in which case we require $T \geq O(\frac{D^2G^2}{\alpha^2})$, $n \geq \tilde{O}\left(\frac{D^5G^5d^{3/2}\sqrt{k}}{\alpha^5}\right)$, and $\ell \geq \tilde{O}\left(\frac{D^3G^3d}{\alpha^3}\right)$.

Thus the number of samples used per query is $T \cdot d \cdot \ell = \tilde{O}\left(\frac{D^5 G^5 d^2}{\alpha^5}\right)$. □

Again, we can give the high-probability version:

Corollary 31. *This mechanism is (α, β)-accurate for k optimization queries, with the same assumptions as in Theorem 30, when $n = \tilde{O}\left(\frac{d^{3/2}\sqrt{k}\log(k/\beta)}{\alpha^5}\right)$ in $\tilde{O}\left(\frac{d^2 \log(k/\beta)}{\alpha^5}\right)$ samples per query and $\tilde{O}\left(\frac{\log(k/\beta)}{\alpha^2}\right)$ iterations of gradient descent per query.*

3.7 Conclusion

In this chapter, we have introduced new faster mechanisms that take advantage of sampling's simultaneous ability to boost privacy while decreasing running time. Using this sampling approach as a black box, we show its applicability to create fast algorithms for adaptive convex optimization. This approach has several upsides, including the fact that the only the black box needs access to the sample; the rest of the algorithm performing gradient descent does not. As importantly, both for convex optimization and low-sensitivity queries, we improve the running time over previous work from order $\sqrt{k}$ to $\log(k)$, all without increasing the sample complexity above $\tilde{O}\left(\sqrt{k}\right)$.

There is still much future work to be done. In what other adaptive settings can sampling help as much as it does in this work? Sub-linear time algorithms are frequently required for a variety of problems, such as property testing or large-data environments. How can fast algorithms for adaptive analysis be developed in these types of settings?

4

On the Complexity of Learning from Label Proportions

4.1 Introduction

In this chapter, we investigate the complexity of the learning problem of estimating the proportion of labels for a given set of instances. For example, this problem

This chapter is based on the manuscript Fish and Reyzin [26]. Copyright held by the IJCAI Organization.

appears when predicting the proportion of votes for a given candidate [16]; correctly predicting how each individual votes is not required, only which candidate will win. Variants of this problem also appear in many other domains, including in consumer marketing [13], medicine and other health domains [37, 79], image processing [16], physical processes [55], fraud detection [64], manufacturing [72], and voting networks [25].

In classical PAC learning, we are given labeled data instances from a distribution, and in the idealized case, must find a function that labels all of the data consistent with the observations. In less constrained settings, the goal is to find a function of low error, or at least of error as low as possible on the data presented to the algorithm. There is substantial literature on classical PAC learning outside the scope of this dissertation; see e.g. [67] for a survey. Once the classifier is found, it is easy to find the proportion of instances with a given label by invoking the classifier on the instances. Algorithms for estimating the proportion of labels with labeled data have been introduced before, for example by Iyer et al. [39].

However, getting instances with attached labels, as assumed in classical PAC learning, is often difficult. Sometimes this is due to limits on the measurement process [37, 16, 55, 72]. At other times, before datasets are released, labels are purposely detached from their instances in order to maintain privacy [13, 64, 79]. Instead, only the proportion of labels are given for a group of sample instances. For example, in estimating who will win an election, pre-election polls only release the percentage of people planning to vote for a given candidate. Quadrianto et al. [62] give several other examples where the only data available is of this form.

The goal is then to learn a classifier from a hypothesis class that is able to correctly predict the proportions of labels from a hidden distribution using a training set which consists of a set of instances and the proportions of labels of that set of instances. This is the learning-theoretic problem we formalize and tackle in this chapter. The proportion of labels may be inferred by first finding a classifier that predicts the labels for each instance [58, 62, 64, 84]. Alternatively, Iyer et al. [40] propose inferring the proportion of labels directly.

Yu et al. [83] introduce a version of a model for learning from label proportions. In their model, each bag of examples comes with the proportions of each label in that bag, and each bag is drawn i.i.d. from a distribution over bags. They give some of the first sample complexity guarantees. Another approach is where the examples are drawn i.i.d., but the bags may be an arbitrary partition of the examples, as in [64, 72]. Compared to these 'bag' models, our model of learning from label proportions corresponds to the 'one-bag case' with binary labels, where each example is drawn i.i.d. from an arbitrary distribution. However, as we demonstrate, this model is already interesting to study. We formalize this as a PAC-like learning model, which allows us to compare the difficult of learning a hypothesis class in classical PAC learning to learning a hypothesis class in this model.

In particular, we give the following results, including the first computational hardness results for learning label proportions. After formally defining the model in Section 4.2, we show in Section 4.3 that the classes of hypotheses (with finite VC dimension) that are learnable from label proportions are a subset of the

classes that are PAC learnable. We then go on to show that learning from label proportions is strictly harder than PAC learning in Section 4.4, by showing that under natural assumptions, parity functions are not efficiently learnable from label proportions. Finally, in Section 4.5 we give some positive results indicating cases where it is possible to PAC learn from label proportions. We also show that n-dimensional half-spaces over the boolean cube are learnable from label proportions under the uniform distribution.

4.2 Model and Sample Complexity

For a distribution D over the domain of a function c, call $c(D)$ the resulting distribution over the range of c. For c a function $\{0,1\}^n \to \{0,1\}$, we will call p_c the percentage of positive labels in this distribution, i.e. $c(D)(1)$. For a given sample, we call the percentage labeled positively as $\hat{p_c}$. Where clear, we will abbreviate these as p and $\hat{p}$ respectively.

In this setting, each example x drawn from D has a hidden label $c(x)$, but the learning algorithm does not get to see examples with labels. Instead, the algorithm only gets to see the set of unlabeled examples S and $\hat{p}$, the percentage of S labeled positively by c. The goal is to find a function h in a hypothesis class H such that p_c should be close to p_h with high probability.

Definition 32. *A class of functions H is* PAC learnable from label proportions *if there is an efficient algorithm A such that for every target function c in H, any distribution D over $\{0,1\}^n$, and for any $\epsilon, \delta > 0$, given $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$*

examples drawn i.i.d. from D and $\hat{p}$, returns a hypothesis h in H such that

$$\mathbb{P}[|p_c - p_h| \leq \epsilon] \geq 1 - \delta.$$

We call this form of learning "**PAC learning from label proportions**." In general, we may consider agnostic or improper versions of this PAC model. However, improper learning from the class of all functions here is very easy: We can efficiently learn with a sample complexity that only depends on ϵ and δ:

Observation 33. *The sample complexity for improper PAC learning from label proportions is $O\left(\frac{\ln(1/\delta)}{\epsilon^2}\right)$.*

Proof outline. In improper learning, it is easy to find a function h^* so that not only does $\hat{p}_{h^*} = \hat{p}$, but also $p_{h^*} = \hat{p}$: e.g. h^* may be a randomized function that on any input returns 1 with probability $\hat{p}$ and 0 otherwise. Then $p_{h^*} = \hat{p}$ and a Chernoff bound implies that $\hat{p}$ is close to p. □

For example, if the task is to predict the proportion of votes for a given candidate using only a single poll, *improper* learning in this model is easy simply by virtue of the fact that $\hat{p}$ is an unbiased estimator for p. However, the hypothesis h^* described above will not be a realistic model of voting. So proper learning corresponds to finding a realistic model of voting, one which describes a relationship between examples and labels, that also predicts the proportion of votes correctly. For this reason, for the remainder of this chapter, we will only consider *proper* PAC learning from label proportions.

Definition 32 is a distribution-free setting, but when the distribution is known, sample complexity also may be independent of the VC-dimension.

Observation 34. *Let D be a known distribution. Let*

$$\beta = \inf_{\substack{h,h' \in H: \\ h \neq h'}} |p_h - p_{h'}|.$$

Then the sample complexity for PAC learning from label proportions the hypothesis class H is $O\left(\frac{\ln(1/\delta)}{\beta^2}\right)$.

Proof outline. Here, we can use another Chernoff bound to get that with high probability, $\hat{p}$ is within $\beta/2$ of p_c, for c the target hypothesis. But the definition of β implies that there is exactly one value p_{c^*} in $\{p_c : c \in H\}$ such that $\hat{p}$ is closer to p_{c^*} than any other value in $\{p_c : c \in H\}$. Then with high probability $p_c = p_{c^*}$. Thus an algorithm may output any h such that $p_h = p_{c^*}$. □

This analysis of the distribution-free setting only considers sample complexity and not computational complexity. In Section 4.5, we will give an example where we can efficiently PAC learn from label proportions under the uniform distribution.

We may still wish to bound the sample complexity of PAC learning from label proportions in the setting where the distribution is arbitrary. Following the proof of the equivalent bounds in PAC learning under an arbitrary loss function (see Chapter 6 of Shalev-Shwartz and Ben-David [67]), we can show the same bounds also hold here. Namely, we can use the VC dimension of a hypothesis class H to bound generalization error. We denote this quantity by $\text{VC}(H)$. In particular, we have:

Theorem 35 (Occam's razor)**.** *For target function $c \in H$, with probability at least $1 - \delta$, for all $h \in H$,*

$$|p_c - p_h| \leq |\hat{p}_c - \hat{p}_h| + O\left(\frac{1}{\delta}\sqrt{\frac{\mathrm{VC}(H)\log(m/\mathrm{VC}(H))}{m}}\right).$$

4.3 Comparing Our Model to Classical PAC

The definition of PAC learning from label proportions makes it harder to learn a class on one hand (by unlinking input from label) but easier on the other hand (by making the loss function easier to satisfy). So it may not be obvious what the relationship with PAC is.

In this section, we show that the hypothesis classes that may be efficiently learned in PAC from label proportions is a subset of the classes that may be efficiently learned in PAC. Corollary 38 then implies it is a strict subset.

Theorem 36. *Suppose* NP $\neq$ RP*. Then if a hypothesis class H with finite VC dimension is efficiently learnable from label proportions, it is also efficiently (proper) PAC learnable.*

Proof. Let H be learnable from label proportions by some efficient oracle A, and f the polynomial sample size required by this oracle. We now give an efficient algorithm for PAC learning H. Given $\epsilon, \delta > 0$, draw m samples from the unknown distribution D, with m to be determined later. Call the set S of unique inputs $x_1, \ldots, x_m$ and their labels $c(x_1), \ldots, c(x_m)$ for hidden target function c. Let k be the number of positive labels $\sum_j c(x'_j)$. Define a new distribution D' as the

following:

$$D'(x) = \left\{ \begin{array}{ll} \frac{m}{km+m-k} & \text{if } x \in S \text{ and } c(x) = 1 \\ \frac{1}{km+m-k} & \text{if } x \in S \text{ and } c(x) = 0 \\ 0 & \text{otherwise} \end{array} \right\}.$$

Let $\epsilon' = 1/(2m^2)$ and $\delta' = \delta$. Draw $m' = f(1/\epsilon', 1/\delta')$ samples x'_j from D' and label each as $c(x'_j)$. We give to the oracle as input ϵ', δ', and the examples x'_j, along with the proportion of positive labels $\hat{p} = \frac{k}{m'}$. Then with probability at least $1 - \delta$ the oracle returns a hypothesis c^* such that

$$|p_{c^*} - p_c| < \frac{1}{2m^2}.$$

The smallest non-zero probability mass in D', however, is

$$\frac{1}{km + m - k} \geq \frac{1}{m^2},$$

minimized when $k = m$. Thus $p_{c^*} = p_c$.

We now show that $c^* = c$ when restricted to the points $x_1, \ldots, x_m$. Suppose there is a point x_i such that $c^*(x_i) \neq c(x_i)$ where $c(x_i) = 1$. Then in order to have $p_{c^*} = p_c$ while $c^*(x_i) = 0$, at least m points labeled 0 by c must be labeled positively by c^*, since D' places (proportional to) m weight on positively labeled points and only unit weight on negative points. This is a contradiction, as there

are only m total points. Similarly, if $c(x_i) = 0$ and $c^*(x_i) = 1$, there must be m points labeled 0 by c^* that are labeled 1 by c, but again there are only m distinct points. Thus c and c^* must agree on all m points, i.e. c^* has zero empirical error.

All that remains is to check that we need only a polynomial sample size to use Occam's razor (Theorem 35). This only requires $\mathrm{VC}(H) = \mathrm{poly}(1/\epsilon, 1/\delta, n, \mathrm{size}(c))$. If H is finite, recall that $\mathrm{VC}(H) \leq \log |H|$. But since $\mathrm{size}(c) \geq \log |H|$, we certainly have $\mathrm{VC}(H) = \mathrm{poly}(\mathrm{size}(c))$. If H is infinite, then the size of the input and classifiers are unbounded, so $\mathrm{VC}(H)$ needs only to be polynomial in $1/\epsilon$ and $1/\delta$. Then the assumption that $\mathrm{VC}(H)$ is finite implies that $\mathrm{VC}(H)$ is in fact a constant, and therefore the bound given by Occam's razor is indeed a polynomial. □

4.4 Hardness of Learning from Label Proportions

A natural question to ask is if there are classes with small VC dimension that are hard to learn. We now show that this is the case for parity functions on the first k bits of the input. This will imply under a natural assumption on the hardness of PAC learning noisy parities that efficiently learning from label proportions is strictly harder than efficiently PAC learning.

Recall in (white-label) noisy PAC learning, each label in the training data is flipped with unknown rate η. We assume the algorithm is given as input some η', where $\eta \leq \eta' < 1/2$ and must only take time polynomial in $\frac{1}{1-2\eta'}$. Noisy PAC learning parity functions under the uniform distribution is presumed to be hard. Blum et al. [7] give an $2^{O(n/\log n)}$) algorithm, which is the best-current bound.

We now find a specific distribution where PAC learning from label proportions

is hard in this sense for parities:

Theorem 37. *For a hypothesis c, Let D_c be the the distribution over $\{0,1\}^n$ that places $\frac{\eta}{2^{n-1}}$ weight on the examples labeled 0 and $\frac{1-\eta}{2^{n-1}}$ weight on examples labeled 1.*

PAC learning parities from label proportions under D_c is as least as hard as PAC learning unknown parity c with η white-label noise under the uniform distribution.

Proof. We use an oracle for PAC learning parities from label proportions under D_c to noisy-PAC learn parities. We get as input η', parameters ϵ and δ, and some m examples x_i, with m to be determined later, with noisy labels $\tilde{\ell}_i$. When $\tilde{\ell}_i = 1$, with probability η, the true label $\ell_i = 0$ and otherwise $\ell_i = 1$. We may assume that the unknown parity c is non-trivial. Then under the uniform distribution over $\{0,1\}^n$, for any such parity function, there are 2^{n-1} points labeled 1 and 2^{n-1} points labeled 0. For any point labeled 0, the probability that it was drawn from the uniform distribution is $\frac{1}{2^{n-1}}$ and the probability that its label was flipped to 1 was η. Then the probability that an example had $\tilde{\ell}_i = 1$ but $\ell_i = 0$ is $\frac{\eta}{2^{n-1}}$ and similarly if $\ell_i = 1$ the probability is $\frac{1-\eta}{2^{n-1}}$. Note that this is exactly the distribution D_c. So if the oracle for PAC learning parities from label proportions is given just the examples where $\tilde{\ell}_i = 1$, the oracle will receive i.i.d. samples from D_c. We will also give to the oracle $\epsilon' = \frac{1/2-\eta'}{2}$ and $\delta' = \delta/3$. The expected proportion of these examples given to the oracle is $1-\eta$, but we do not know the true labels nor do we know η. So instead, we will invoke this oracle $M+1$ times, with the proportion given to the oracle as each of $0, 1/M, \ldots, 1$, where $M = \sum_i \tilde{\ell}_i$, i.e. the number of

training examples with noisy label $\tilde{\ell}_i = 1$ *.

If the oracle returns the correct parity c, then it should agree in expectation with the noisy labels $\tilde{\ell}_i$ on all but η of the examples. For an incorrect parity c', by the orthonormality of the parity functions, the expected disagreement is $1/2$. For h the output of the oracle, if smaller than an $\frac{\eta'+1/2}{2}$ fraction of the noisy labels $\tilde{\ell}_i$ disagree with the corresponding label $h(x_i)$, then we return the hypothesis. Otherwise, we repeat with the next invocation of the oracle.

Let f be the polynomial sample bound for the oracle for PAC learning from label proportions . First, we need to make sure that the oracle receives at least $f(1/\epsilon', 1/\delta')$ examples except with probability at most $\delta/3$. In expectation, $m/2$ of the examples x_i will have $\tilde{\ell}_i = 1$. Using a Chernoff bound,

$$\mathbb{P}\left[\left|\sum_i \tilde{\ell}_i - m/2\right| > m/4\right] \leq 2e^{-m/8}.$$

So the oracle will receive at least $\frac{1}{4}m$ examples (and no more than $\frac{3}{4}m$ examples) except with probability no more than $\delta/3$ so long as $m > 8\log(6/\delta)$. This then means that we require $m > 4 \cdot f(1/\epsilon', 1/\delta')$ so that $M \geq f(1/\epsilon', 1/\delta')$.

Now we need to verify that when the proportion given to the oracle is the correct proportion $\hat{p}_c$, the oracle will return c except with probability at most $\delta/3$. The oracle is guaranteed to return a parity h such that except with probability $\delta' = \delta/3$,

$$|p_h - p_c| \leq \epsilon' = \frac{1/2 - \eta'}{2}.$$

*The oracle is undefined when the proportion of positive labels is not the true value $\hat{p}$. We may assume that the oracle returns an arbitrary hypothesis in this case.

Using the definition of D_c, $p_c = 1 - \eta$. If $h \neq c$, then $p_h = 1/2$ again by orthonormality. But then

$$|p_h - p_c| = |1/2 - \eta| > \frac{1/2 - \eta'}{2},$$

so it must be the case that $h = c$. Thus at least one of the invocations of the oracle will return the correct parity.

So it remains to show that we will succeed at returning this parity. If the oracle returns an incorrect parity h, again using a Chernoff bound,

$$\begin{aligned}\mathbb{P}\left[\left|\frac{\sum_i \mathbb{1}_{h(x_i)\neq\tilde{\ell}_i}}{m} - 1/2\right| \geq \frac{1/2 - \eta'}{2}\right] &\leq 2e^{-\frac{m(1/2-\eta')^2}{2}} \\ &< \frac{1}{M+1} \cdot \frac{\delta}{3}\end{aligned}$$

when

$$m = \Omega\left(\frac{\log(\frac{M}{\delta})}{(1/2 - \eta')^2}\right) = \Omega\left(\frac{\log\left(\frac{1}{(1/2-\eta')\delta}\right)}{(1/2 - \eta')^2}\right)$$

because $M \leq \frac{3}{4}m$, where $\mathbb{1}_A$ is the indicator function that is 1 if A is true and 0 otherwise. This implies that for an incorrect hypothesis, whose expected fraction of disagreements with the noisy labels is $1/2$, the empirical fraction is at least $\frac{\eta'+1/2}{2}$, the threshold we had set. Similarly, for the correct hypothesis, where the expected fraction of disagreements is $\eta < \eta'$, the empirical fraction of disagreements is no more than $\frac{\eta'+1/2}{2}$ except with probability at most $\frac{1}{M+1} \cdot \frac{\delta}{3}$. This means that all of the tests of the hypothesis succeeds except with probability at most

$\delta/3$. Then setting

$$m = \Omega\left(\max\left(\left(\frac{\log\left(\frac{1}{(1/2-\eta')\delta}\right)}{(1/2-\eta')^2}\right), 4 \cdot f(1/\epsilon', 1/\delta')\right)\right)$$

suffices so that, with the union bound, the total probability of failure is no more than δ, as required. □

Consider parity functions on the first k bits, which have VC dimension equal to k. There is no known algorithm for noisy PAC learning parity functions on the first k bits when $k = \omega(\log n \log\log n)$. It is conjectured that there is no efficient algorithm for PAC-learning noisy parity that runs in time $o(2^{\sqrt{n}})$, which would imply hardness of noisy PAC learning parities on the first k bits for $k = \omega(\log^2 n)$. Calling this the 'noisy parity assumption,' Theorem 37 implies the following:

Corollary 38. *Under the noisy parity assumption, there is no efficient algorithm for PAC learning label proportions of parities on the first k bits for $k = \omega(\log^2 n)$.*

This means there are hypothesis classes with VC dimension $\omega(\log^2 n)$ that aren't PAC learnable from label proportions.

4.5 Classes PAC Learnable from Label Proportions

Call d the VC dimension of a given hypothesis class H. In Section 4.4, we showed that if d is a fractional power, H is hard to learn. We also gave examples with d as small as $\log n$ that are hard to learn, under stronger complexity assumptions. On the other hand, as long as labelings in a given hypothesis class are efficiently enu-

merable, then finite classes H are certainly PAC learnable from label proportions in time $|H|$. Or instead, by enumerating only distinct hypotheses on the sample, assuming that this is efficient, learning can be achieved in m^d time using Sauer's lemma. This immediately implies that all such classes with constant d are learnable from label proportions. We now show that not all classes with $d = \Omega(\log n)$ are hard to learn.

Consider the following hypothesis class which only allows hypotheses whose positive labels are close to each other:

$$H_k = \{h : \{1, \ldots, 2^n\} \to \{0, 1\} : \max_{h(i)=h(j)=1} |i - j| \leq k\}.$$

There are still exponentially many functions and $VC(H_k) = k$. H_k was shown to be hard in Section 4.4. However, for H_k, this is not the case:

Observation 39. *PAC learning H_k from label proportions has an $O(2^k m)$ time algorithm.*

Order the m examples in $\{1, \ldots, 2^n\}$, and for each length k subset, of which there are $m-k+1$ of them, check all 2^k possible labelings. Now when $k = O(\log n)$, this is a polynomial-time algorithm for learning H_k from label proportions even though the VC dimension is not constant.

In the classical PAC setting, when it is hard to learn under an arbitrary distribution, it is often still valuable to show that learning can still be done in special cases, such as the uniform distribution. We now give an example, namely halfspaces, where it is easy to learn under the uniform distribution.

The idea to find a half-space that classifies the given proportion $\hat{p}$ positively is to take a random half-space through the origin, and then move it in the direction of its normal vector, and stop when the half-space classifies the input p proportion of the sample positively. With high probability, this will be possible because no two points in the sample will be projected to the same point on the normal vector.

Proposition 40. *The class of half-spaces in n dimensions is learnable from label proportions under the uniform distribution over $\{0,1\}^n$.*

Proof. Since the VC-dimension of half-spaces is linear in n by Radon's theorem [53], using Theorem 35 it certainly suffices to be able to efficiently find a half-space h such that $\hat{p_h} = p$ with high probability. Consider a hyper-plane P of dimension $n-1$ through the origin and v a normal vector defining P.

First, we show that for a randomly chosen vector v, no two points in $\{0,1\}^n$ project more than exponentially close to each other (in terms of n) on v. This allows us to use only a polynomial number of bits to represent each projected point. Consider an arbitrary pair of points x and y in $\{0,1\}^n$ and consider the line ℓ that passes through these two points. If v and ℓ are perpendicular, then x and y will project onto the same point on v. More generally, we can find the maximum obtuse angle between v and ℓ such that the two points so that the points project exponentially close together on v. Any closer, and we will not have enough bits to distinguish between the projection of x and y. Namely, for a pair of points distance d apart, using the Taylor approximation for $\sin(x)$, the difference between $\pi/2$ and this maximum angle is no more than $O\left(\frac{1}{d2^{\omega(n^c)}}\right)$ for constant c. Since the points come from $\{0,1\}^n$, $d \geq 1$, and there are $O(2^{n^2})$ such pairs of

points, so the total angle from which a uniformly-random vector v may not be chosen is at most $O\left(\frac{2^{n^2}}{2^{\omega(n^c)}}\right)$, an exponentially small probability. Thus, with high probability, no two points in $\{0,1\}^n$ project to the same point on v, or project more than exponentially close to each other on v.

Given m examples, setting m to be polynomial in n insures with high probability that all examples are distinct, and therefore no two examples project more than exponentially lose to each other on v. Since $\hat{p}_c = i/m$ for some $i \in \{0, 1, \ldots, m\}$, we need to find a plane parallel to P such that the corresponding linear threshold function classifies i of the sample points positively. For each pair of consecutive projected points cv and $c'v$ on v for real number c and c', consider the half-space given by the plane defined by the points $p \in \mathbb{R}^n$ satisfying $v\left(p - \left(\frac{c+c'}{2}\right)v\right) = 0$, so that these two points are classified differently by the half-space. Thus one of these half-spaces (or the half-spaces classifying all points positively or negatively) will have $\hat{p}_h = i/m$ since no two points in the sample project onto the same point on v. □

While we have shown that it is strictly harder to PAC learn from label proportions than to PAC learn, introducing noise to the models changes the relationship between these two models. For example, PAC learning parities with unknown η white-label noise is hard under the uniform distribution, as discussed above, but PAC learning parities from label proportions with white-label noise is easy under the uniform distribution. In our model, that means each label is flipped i.i.d. with probability some unknown η, and the proportion of noisy positive labels $\hat{p}^\eta$ is given as input instead, but otherwise the learning requirement remains stays the

same.

Observation 41. *The class of parities is learnable from label proportions under the uniform distribution and unknown η white-label noise.*

Proof. Let p_c^η be the proportion of positive labels under η noise and parity c. Note p_c^η is always

$$(1-\eta)p_c + \eta(1-p_c) = p_c(1-2\eta) + \eta,$$

but for any non-trivial parity c, $p_c = 1/2$, so $p_c^\eta = 1/2$. Then Observation 34 implies that we may distinguish efficiently the trivial parity from the non-trivial parities and in the case that $p_c^\eta = 1/2$ we may return any non-trivial parity. □

4.6 Conclusion

In this chapter we formalized a model for learning a hypothesis class by only examples drawn from a distribution and the proportion of them receiving each label, with the goal of finding a hypothesis that matches these statistics on the underlying distribution, and we focused on the binary label setting.

We give some initial results into a learning theory for this task, including that if a class with finite VC dimension is efficiently learnable from label proportions, it is automatically also efficiently properly PAC learnable. Moreover, we exhibit a class with non-trivial VC dimension that is hard to learn in our model, under natural assumptions about the hardness of PAC learning noisy parities. We give examples where it is possible to efficiently PAC learn from label proportions, which may be surprising given that this is a low-information setting, including half-spaces under

the uniform distribution.

These results are for the binary setting and only for the 'one bag' version of the problem. We leave for future work the analysis of the case where there is more than one bag of examples and each bag's proportion of labels is given. For that case, and in other similar settings where the learner is given more information, we expect there to be more positive algorithmic results.

5

Recovering Social Networks by Observing Votes

5.1 Introduction

One approach to investigating voting data assumes that agents' votes are independent of one another, conditioned on some underlying (sometimes probabilistic) model of ground truth. This is usually an unrealistic assumption, leading to a more

This chapter is based on the manuscript Fish et al. [25].

recent line of inquiry which asks how the social network structure of the voters affects the relationship between votes. Each agent in a social network expresses a position (votes for or against a bill, prefers one brand over another, etc.) that is influenced by their social connections. In this view, it is possible to detect the organization and evolution of voting communities by looking at the social network structure. The literature on congressional and political voting networks focuses on detecting community structure, partisanship, and evolutionary dynamics [1, 41, 51, 60, 78, 85], while the literature on idea propagation investigates how to best maximize the spread of ideas through a population [14, 46].

However, it is often not necessarily clear how to build this social network graph. For example, Macon et al. give a few different variants on how to define the social network of the voters of the United Nations [51]. In this approach, different graphs may reveal different aspects of the social network structure.

This corresponds neatly with a typical view in social choice theory that votes are manifestations of subjective preferences. At the other extreme, a voter votes according to a noisy estimate of the ground-truth qualities of the possible choices on which he or she is voting. While both are over-simplified extremes, it is useful to consider the extremes in order to investigate their consequences [15].

In this chapter, as in previous work, we assume there is a fixed probabilistic model which is used to determine the relationship between initial preferences for the possible choices and how each individual ends up voting for those choices. This probabilistic model takes into account the social network structure in which the voters are embedded.

In this approach, it is typically assumed that the social network of the voters is known. The goal is then to find the correct choice from votes, as tackled by Conitzer and then others [15, 61, 74]. This can be made more difficult depending on the structure of a social network, which may enforce the wrong choice by aggregating individual opinions over dense subgraphs, leading voters with low degree to possibly change their mind to the majority view of the subgraph.

In practice, the social network is usually not known and it is not necessarily clear how to infer the graph. In this chapter, we tackle the problem of inferring the social network from the votes alone. We discuss two similar but distinct voting models in the vein of Conitzer [15], and show how to recover the graph given the votes under these voting models and under several notions of what it means to recover the graph. We show that your ability to learn the graph from the votes is highly dependent on the underlying voting model - in some settings, it is computationally hard to do so but not in others. Moreover, we demonstrate that the resulting learned graphs can differ significantly depending on which underlying voting model is assumed.

5.2 Models and results

We give results for two similar models: an edge-centric model, which Conitzer calls the *independent conversation model* [15], and a vertex-centric model, introduced in this chapter, which we will call the *common neighbor model*.

Similar to some existing models, the common neighbor model is, for instance, equivalent to the "deterministic binary majority process" run for one step (where

the initial assignment is random). This process was examined by Goles and Olivos [33] and related work, e.g. by Frischknecht et al. [29], and it has been used in the press to illustrate the disproportionate influence of certain voters [65]. The models we consider herein also resemble settings in multiple previous works, e.g. by Grabisch and Rusinowska [35], Grandi et al. [36], and Schwind et al. [66].

In both of our models, there is an unknown simple undirected graph G on n vertices. Each vertex is an agent, who can vote "-1" or "1". Both models describe how each agent votes in one round of voting. We consider m rounds of voting and in each round every vertex votes, leading to a sequence of vote sets $V^{[m]} = V_1, \ldots, V_m$, where each V_i is the set of votes from all voters. The problem is to recover G from $V^{[m]}$.

First, we define the independent conversation model, a two-step process where edges represent conversations between voting agents, and each agent votes according to the majority outcome of his conversations.

Definition 42 (**independent conversation model**). *First, each edge flips a coin i.i.d. that with probability p is 1 and with probability $(1-p)$ is -1. Then each vertex votes according to the majority of its adjacent edges' preferences. If there is a tie, then it is broken in favor of voting 1 with probability q and -1 with probability $1-q$.*

This process is depicted for a particular graph in Figure 5.1. Note that the set of votes V_i only includes the final votes, not the initial preferences.

The common neighbor model is similar, except here the initial preferences are on the vertices, not the edges:

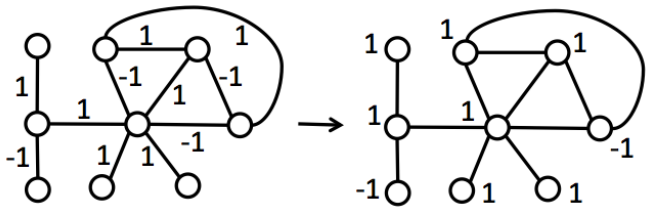


Figure 5.1: Left: the outcome of pairwise "conversations" between connected neighbors. Right: the resulting votes. For simplicity, the edge probabilities are not depicted.

Definition 43 (**common neighbor model**). *Each vertex initially flips a coin i.i.d. that with probability* p *is* 1 *and with probability* $1-p$ *is* -1*. Then each vertex votes* 1 *if more adjacent vertices' initial preferences were* 1 *then* -1 *and vice versa. If there is a tie, then it is broken in favor of voting* 1 *with some probability* q *and* -1 *with probability* $1-q$*.*

This process is illustrated in Figure 5.2.

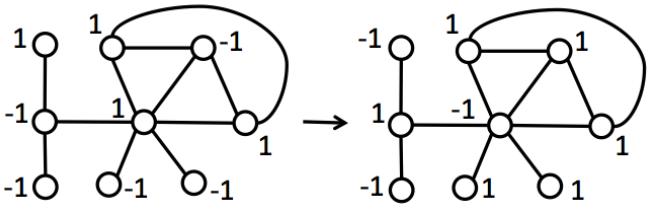


Figure 5.2: Left: the initial preferences of the nodes. Right: the resulting votes. For simplicity, the preference probabilities are not depicted.

It is straightforward to see how they are different. In the independent conversation model, two vertices' votes are independent of each other if and only if they do not share an edge, while in the common neighbor model, they are independent if and only if they have no common neighbors.

Our main contribution consists of algorithms to recover the hidden graph G from the votes, lower bounds, and experiments for both models.

Our results span a few different notions of what it means to recover the unknown graph. First, we ask whether there exists a polynomial-time algorithm that succeeds with high probability (given only a polynomial number of votes in the number of voters) in finding the unknown graph G when the votes were drawn from G. The algorithm must take time polynomial in both the number of votes given as input and the number of vertices. We refer to this as *exact learning*. We show the following:

Result 44. *In the independent conversation model, there is a polynomial-time algorithm that exactly learns the unknown graph when $p = 1/2$ (with high probability). Moreover, for constant $p \neq 1/2$, an exponential number of votes are required to exactly learn the graph. (See **Observation 50** and **Theorem 51**.)*

Our algorithm is a statistical test for edges between pairs of vertices by calculating sample covariances of their votes, which here measures how likely it is that two voters vote the same way. This is very similar to how voting networks are often constructed in the literature [2, 51, 78]. This result can then be seen as a formal motivation for why this type of method should be used.

Result 45. *In the common neighbor model, no algorithm can exactly recover the unknown graph. (See **Observation 57**.)*

The above two results motivate us to consider other notions of what it means to recover the graph because the graph is generally not recoverable efficiently here. Moreover, in a setting where there is not necessarily a ground-truth graph from which the votes were drawn, we are still interested in finding a graph that explains the votes.

We make this precise by asking for the maximum likelihood estimator (MLE) graph, that graph that maximizes the probability of the observed votes over the distribution of votes induced by a fixed voting model. As is standard for the maximum likelihood estimator, we assume that the prior over graphs is uniform. We refer to this as *maximum likelihood learning*. Under this model of learning, votes do not necessarily need to come from any particular hidden graph. Nevertheless, the goal is to produce the MLE graph under a voting model for a set of input votes regardless of where the votes came from.

Result 46. *In the independent conversation model, there is a polynomial-time bounded-probability random reduction from a* #P*-complete problem to finding the likelihood of the MLE graph. However, if enough votes are drawn from a hidden graph G when $p = 1/2$, there is a polynomial-time algorithm that finds the MLE graph on the votes with high probability. (See* ***Theorem 54*** *and* ***Theorem 56****.)*

This lower bound is an indication that computing the likelihood of the MLE graph is difficult, since if there were an efficient algorithm to compute this quantity then there would be an efficient algorithm to solve a #P-complete problem that succeeds with high probability.

On the other hand, merely trying to find the MLE graph is possible, at least if there is enough votes given as input (specifically, for n voters, order n^4 votes suffices).

In the common neighbor model, we investigate a third approach to finding a graph that explains the votes. Given that we recover graphs in the independent conversation model using covariances between votes, it is natural to ask whether

it is possible to find a graph whose expected covariances are close to the observed covariances in the common neighbor model. We show that even if you were to know the expected covariances exactly, it would still be computationally difficult to find such a graph.

Result 47. *For the common neighbor model, finding a graph with given expected covariances between votes is at least as hard as recovering an adjacency matrix from its square. Moreover, a generalization of this problem, namely the generalized squared adjacency problem, is NP-hard. (See* ***Observation 58*** *and* ***Theorem 60****.)*

The squared adjacency problem and its generalized version are defined as follows:

Definition 48. *The input for the squared adjacency matrix problem is a matrix B, and the decision problem asks if there is an adjacency matrix A of a simple graph such that $A^2 = B$.*

Definition 49. *The input for the generalized squared adjacency problem is a collection of n^2 sets S_{ij}, and the decision problem asks if there is a simple graph whose adjacency matrix is A such that $A^2_{ij} \in S_{ij}$ for each entry A^2_{ij} of A^2.*

To the best of our knowledge, the squared adjacency matrix problem is not known to be NP-hard nor is it known to be in P. It is a difficult open problem in its own right, and other versions of it have been proven NP-hard [54]. It is equivalent to the special case of the generalized version where the set sizes are exactly one.

5.3 The independent conversation model

In this section, we show when there is a algorithm to recover the hidden graph. We will also show that it is hard to find the likelihood of the maximum likelihood graph for an input sequence of votes. Before we present those two results, we start with the following observation:

Observation 50. *For constant $p \neq 1/2$, under the independent conversation model, it takes exponentially many votes to distinguish with high probability between the complete graph and the complete graph minus an edge.*

This follows directly from the fact that in both the complete graph and the complete graph minus an edge, if $p \neq 1/2$, with exponentially high probability every voter will vote 1. Our only hope is that it becomes possible to recover the graph G when $p = 1/2$, which we show to be the case.

5.3.1 An algorithm for $p = 1/2$

In this section, we prove the following:

Theorem 51. *Let $p = q = 1/2$. For any graph G on n vertices and $\delta > 0$, if $m = \Omega\left(n^2\left(\ln n + \ln \frac{1}{\delta}\right)\right)$ votes are drawn from G under the independent conversation model, there is a polynomial-time algorithm that will recover G with probability at least $1 - \delta$.*[*]

[*]This result actually remains true for arbitrary values of q, but we restrict this theorem to $q = 1/2$ to simplify the proof in this version.

Let $X_u \in \{1, -1\}$ be the random variable representing the outputted vote of vertex u, so $X_u = 1$ if u votes 1 and -1 otherwise. Now consider two vertices u and v. The votes of u and v are independent if and only if (u, v) is not an edge. This yields a natural approach to determining if (u, v) is an edge of G: measure the sample covariance between the votes of u and v and if this covariance is sufficiently far away from zero, there must be an edge.

To formalize this, we need to calculate the covariance between X_u and X_v if there is an edge between them:

Lemma 52. *For any edge (u, v) of G, let d_u and d_v be the degrees of u and v. For convenience, let $\rho = (1-2p)q + p$. Then $Cov(X_u, X_v)$ is*

$$\begin{cases} 4\rho^2 \binom{d_u-1}{\frac{d_u-2}{2}} \binom{d_v-1}{\frac{d_v-2}{2}} (p(1-p))^{\frac{d_u+d_v-2}{2}}, & \text{even } d_u, d_v \\ 4\rho \binom{d_u-1}{\frac{d_u-2}{2}} \binom{d_v-1}{\frac{d_v-1}{2}} (p(1-p))^{\frac{d_u+d_v-1}{2}}, & \text{even } d_u, \text{ odd } d_v \\ 4\rho \binom{d_u-1}{\frac{d_u-1}{2}} \binom{d_v-1}{\frac{d_v-2}{2}} (p(1-p))^{\frac{d_u+d_v-1}{2}}, & \text{odd } d_u, \text{ even } d_v \\ 4 \binom{d_u-1}{\frac{d_u-1}{2}} \binom{d_v-1}{\frac{d_v-1}{2}} (p(1-p))^{\frac{d_u+d_v}{2}}, & \text{odd } d_u, d_v. \end{cases}$$

Proof. Consider an edge $(u, v) \in E(G)$. Since u and v vote independently given the vote of the edge (u, v), we will write the probability that each of these vertices vote 1 given the edge vote.

Namely, call $P_u^1 = \mathbb{P}(X_u = 1 | \text{edge } (u, v) \text{ votes 1})$ and $P_u^{-1} = \mathbb{P}(X_u = 1 | \text{edge } (u, v) \text{ votes -1})$ and similarly P_v^1, P_v^{-1} the analogous probabilities for v.

We can write the covariance in terms of these four probabilities: $Cov(X_u, X_v) = 4p(1-p)(P_u^1 - P_u^{-1})(P_v^1 - P_v^{-1})$.

To show this, it suffices to write the covariance as a function of the joint proba-

bilities $P(X_u = 1, X_v = 1)$, etc., and then write each joint probability as a function of the probabilities that a vertex votes 1 given that how the adjacent edge votes. For example, by conditioning on the vote of edge (u, v),

$$P(X_u = X_v = 1) = pP_u^1 P_v^1 + (1-p)P_u^{-1}P_v^{-1}.$$

The others are similar.

To complete the proof, all we need are formulae for P_u^1 and P_u^{-1} (P_v^1 and P_v^{-1} are calculated analogously). This is done by choosing edges to form the majority vote of u's neighborhood.

Recall $d(u) - 1$ and $d(v) - 1$ are the degrees of u and v, respectively, minus 1 (in order to discount the edge (u, v)). We then have

$$P_u^1 = \begin{cases} \sum_{i=0}^{\frac{d(u)-2}{2}} \binom{d(u)-1}{i}(1-p)^i p^{d(u)-1-i} & \text{even } d(u) \\ + q\binom{d(u)-1}{\frac{d(u)}{2}}(1-p)^{\frac{d(u)}{2}} p^{\frac{d(u)-2}{2}}, & \\ \sum_{i=0}^{\frac{d(u)-1}{2}} \binom{d(u)-1}{i}(1-p)^i p^{d(u)-1-i}, & \text{odd } d(u). \end{cases}$$

and

$$P_u^{-1} = \begin{cases} \sum_{i=0}^{\frac{d(u)-4}{2}} \binom{d(u)-1}{i}(1-p)^i p^{d(u)-1-i} & \text{even } d(u) \\ + q\binom{d(u)-1}{\frac{d(u)-2}{2}}(1-p)^{\frac{d(u)-2}{2}} p^{\frac{d(u)}{2}}, & \\ \sum_{i=0}^{\frac{d(u)-3}{2}} \binom{d(u)-1}{i}(1-p)^i p^{d(u)-1-i}, & \text{odd } d(u), \end{cases}$$

The statement of the lemma then follows. □

In the case where $p = 1/2$, the covariance will be sufficiently large; namely that it will be $\Omega(1/n)$, where n is the number of vertices of G:

Corollary 53. *When $p = 1/2$ and (u, v) is an edge of G,*

$$Cov(X_u, X_v) \geq \frac{1}{2\pi} \frac{1}{\sqrt{d_u d_v}} \geq \frac{1}{2\pi n}.$$

Proof. We simplify the formula for the covariance derived in Lemma 52 by giving lower bounds for the central binomial coefficients, from which the result immediately follows: For any positive integer k, the central binomial coefficient(s) satisfy

$$\binom{k}{\lceil \frac{k-1}{2} \rceil} \geq \frac{2^k}{\sqrt{\pi k}}.$$

These lower bounds follow from Sterling's approximation $k! = \sqrt{2\pi k} \left(\frac{k}{e}\right)^k \left(1 + O\left(\frac{1}{k}\right)\right)$. □

Note this lower bound was only polynomial in $1/n$ because $p = 1/2$; otherwise, the exponential term $(p(1-p))^n$, for p constant, ensures that the covariance goes to 0 exponentially quickly in n.

We are now ready to prove Theorem 51, which uses the Hoeffding bound to establish that the sample covariance converges quickly enough to its expectation, which if there is an edge is given in Lemma 52 and if there is no edge is just 0.

of Theorem 51. Recall that in the independent conversation model, we are given m votes $X_{u,i} \overset{i.i.d.}{\sim} X_u$ for $i = 1, \ldots, m$ and all u in G, where X_u is the $\{-1, 1\}$-valued random variable found by taking the majority vote of the initial votes of u's neighborhood.

For $p = q = 1/2$, $\mathbb{E}(X_u) = 0$ for each vertex u, which means that $Cov(X_v, X_u) =$

$\mathbb{E}(X_u X_v)$. This means that the sample covariance between u and v is

$$C_{u,v}^m = \frac{1}{m}\sum_i^m X_{u,i}X_{v,i}.$$

The algorithm to recover G from the m votes is straightforward: For each pair of vertices u, v, calculate the sample covariance $C_{u,v}^m$. If $C_{u,v}^m > \frac{1}{4\pi n}$, then the algorithm claims there is an edge between u and v, and otherwise, the algorithm claims there is no such edge. We call this the *covariance test*. It suffices to show that the probability that the covariance test is wrong is low. Using Corollary 53, we get for edge (u, v),

$$\mathbb{P}\left(C_{u,v}^m < \frac{1}{4\pi n}\right) \leq \mathbb{P}\left(C_{u,v}^m - \mathbb{E}\left(C_{u,v}^m\right) < -\frac{1}{4\pi n}\right),$$

and for $(u, v) \notin E(G)$,

$$\mathbb{P}\left(C_{u,v}^m > \frac{1}{4\pi n}\right) = \mathbb{P}\left(C_{u,v}^m - \mathbb{E}\left(C_{u,v}^m\right) > \frac{1}{4\pi n}\right).$$

By the Hoeffding bound each of these two terms is bounded above by $e^{-\frac{cm}{n^2\pi^2}}$ for some constant c.

Let G' be the network inferred by the above algorithm. Then the probability that G' is not G is no more than

$$\sum_{u,v\in E} \mathbb{P}\left(C_{u,v}^m < \frac{1}{4\pi n}\right) + \sum_{u,v\notin E} \mathbb{P}\left(C_{u,v}^m > \frac{1}{4\pi n}\right),$$

which is bounded from above by $\binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}$.

Hence, for any $\delta > 0$, setting $m = \Omega\left(n^2\left(\ln n + \ln \frac{1}{\delta}\right)\right)$ suffices so that $\binom{n}{2} e^{-\frac{cm}{n^2\pi^2}} < \delta$. □

5.3.2 Moving from exact learning to maximum likelihood learning

In the previous section, we showed that exact learning is possible when $p = q = 1/2$. We now show that it is possible to not only exactly learn the graph, but also find the maximum likelihood graph for the votes when $p = q = 1/2$, assuming we are given enough data. Recall the maximum likelihood graph (which we will also refer to as the MLE graph) is the graph that maximizes the probability of the observed votes over the distribution of votes.

Theorem 54. *Let $p = q = 1/2$. For any graph G on n vertices and $\delta > 0$, if $m = \Omega\left(n^2\left(n^2 + \ln \frac{1}{\delta}\right)\right)$ votes are drawn from G under the independent conversation model, there is a polynomial-time algorithm that will find the maximum likelihood graph on the drawn votes with probability at least $1 - \delta$.*

In other words, if the votes really do come from a hidden graph and we are given order n^4 votes, we can find the maximum likelihood graph. Specifically, what we show is that if you are given this many votes from a hidden graph, then the MLE graph *is* the hidden graph with high probability. The proof of Theorem 54 then follows from applying Theorem 51, i.e. using the covariance test to find the hidden graph, which is the MLE graph. This moves a statement about exact learning to a statement about maximum likelihood learning.

We now prove (Lemma 55) that the MLE graph is the hidden graph for a sufficiently large set of votes drawn from a hidden graph. Indeed, we show something stronger: the hidden graph will be more likely (under this model) than any other graph by an arbitrarily large factor α (where the number of votes given as input needs to increase logarithmically in α). This stronger result will also be needed for the proof of Theorem 56.

The statement of this will need some notation: For any graph G on n vertices, let $\mathcal{V}_G$ be the distribution over a set of n votes induced by G under the independent conversation model for $p = q = 1/2$. For convenience we will denote the m-product distribution $\mathcal{V}_G \times \ldots \times \mathcal{V}_G$ as $\mathcal{V}_G^{[m]}$. That is, for any vote $V \in \{-1, 1\}^n$, $\mathbb{P}_{\mathcal{V}_G}(V) = \mathbb{P}(V|G)$ is the probability mass of V under $\mathcal{V}_G$. Similarly, for a sequence of votes $V^{[m]}$, $\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(V^{[m]}\right) = \mathbb{P}\left(V^{[m]}|G\right)$ is the probability mass of $V^{[m]}$ under $\mathcal{V}_G^{[m]}$.

Lemma 55. *For $\delta > 0$, $\alpha > 1$,*

$$\mathbb{P}_{V^{[m]} \sim \mathcal{V}_G^{[m]}}\left(\mathbb{P}\left(V^{[m]}|G\right) \leq \alpha \max_{G' \neq G} \mathbb{P}\left(V^{[m]}|G'\right)\right) < \delta$$

for $m = \Omega\left(n^2\left(n^2 + \ln\frac{1}{\delta} + \ln\alpha\right)\right)$.

Proof. Fix $\alpha > 1$ and denote by E the event that

$$\frac{\max_{G' \neq G} \mathbb{P}(V^{[m]}|G')}{\mathbb{P}(V^{[m]}|G)} \geq \frac{1}{\alpha}.$$

(If $\mathbb{P}(V^{[m]}|G) = 0$, then this event occurs, so we can safely assume the converse.) The idea of this proof is that we will show the probability of E happening is small by conditioning on what the vote sequence $V^{[m]}$ looks like when drawn from G.

Specifically, in the proof of Theorem 51, we show that the covariance test would have successfully found G with high probability, so we condition on this happening.

Fix a graph $G' \neq G$. We will want to show that the probability that $V^{[m]}$ is pulled from G' (instead of G) is sufficiently small. The covariance test failed if $V^{[m]}$ were pulled from G': the covariance test returned G on $V^{[m]}$ instead of G'. And again, the probability that the covariance test failed is low, as showed in the proof of Theorem 51, so the probability that $V^{[m]}$ is pulled from G' must be small.

We denote the set of vote sequences for which the covariance test returns G by Φ_G. Using this notation, we condition on $V^{[m]}$ being in Φ_G or not and then get an immediate upper bound:

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}(E) \leq \mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E|V^{[m]} \in \Phi_G\right) + \mathbb{P}_{\mathcal{V}_G^{[m]}}\left(V^{[m]} \notin \Phi_G\right).$$

We then bound each of these two terms. The probability that the covariance test failed on $V^{[m]}$ is small: By inspecting the proof of Theorem 51, we have that for some constant c,

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(V^{[m]} \notin \Phi_G\right) \leq \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}. \tag{5.1}$$

Otherwise, the covariance test succeeded and we condition on $V^{[m]} \in \Phi_G$. We now show that

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E|V^{[m]} \in \Phi_G\right) \leq \alpha\left(2^{\binom{n}{2}} - 1\right)\binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}. \tag{5.2}$$

Markov's inequality gives

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(\frac{\max_{G'\neq G}\mathbb{P}\left(V^{[m]}|G'\right)}{\mathbb{P}\left(V^{[m]}|G\right)} \geq \frac{1}{\alpha}\middle| V^{[m]} \in \Phi_G\right)$$
$$\leq$$
$$\alpha \cdot \mathbb{E}_{\mathcal{V}_G^{[m]}}\left(\frac{\max_{G'\neq G}\mathbb{P}\left(V^{[m]}|G'\right)}{\mathbb{P}\left(V^{[m]}|G\right)}\middle| V^{[m]} \in \Phi_G\right).$$

It is then enough to expand this expected value using the definition to get that

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}\left(E|V^{[m]} \in \Phi_G\right) \leq \alpha \sum_{V^{[m]}\in\Phi_G} \max_{G'\neq G} \mathbb{P}\left(V^{[m]}|G'\right).$$

Now we group the terms of the sum by which graph $G' \neq G$ maximizes the probability $\mathbb{P}\left(V^{[m]}|G'\right)$. There may be many terms in the sum that any one graph G' maximizes, but certainly each vote sequence associated with each term is in Φ_G. There are of course $2^{\binom{n}{2}} - 1$ such graphs, so

$$\begin{aligned}\sum_{V^{[m]}\in\Phi_G} \max_{G'\neq G} \mathbb{P}\left(V^{[m]}|G'\right) &\leq \sum_{G':G'\neq G} \sum_{V^{[m]}\in\Phi_G} \mathbb{P}\left(V^{[m]}|G'\right) \\ &= \left(2^{\binom{n}{2}} - 1\right)\mathbb{P}\left(V^{[m]} \in \Phi_G|G'\right).\end{aligned}$$

If $V^{[m]}$ were in Φ_G but $V^{[m]}$ was pulled from G', then the covariance test has failed at returning G'. So

$$\mathbb{P}\left(V^{[m]} \in \Phi_G|G'\right) \leq \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}},$$

implying Equation 5.2. Combining Equations 5.1 and 5.2, we get

$$\mathbb{P}_{\mathcal{V}_G^{[m]}}(E) \leq \alpha 2^{\binom{n}{2}} \binom{n}{2} e^{-\frac{cm}{n^2\pi^2}}.$$

For $\mathbb{P}_{\mathcal{V}_G^{[m]}}(E)$ to be upper-bounded by $\delta > 0$, it suffices to set

$$m = \Omega\left(n^2\left(n^2 + \ln\frac{1}{\delta} + \ln\alpha\right)\right).$$

□

5.3.3 Hardness of computing the MLE

As we have seen, when $p = q = 1/2$, distinguishing between graphs can be done in polynomial time. This might give hope that, in this case, computing the likelihood of the MLE graph, given a set of votes, may be easy. That is, given a graph G which is the maximum likelihood graph for a set of input votes $V^{[m]}$ over G, we wish to compute $\mathbb{P}(V^{[m]}|G)$. Alas, we give hardness results indicating this is not easy to do.

We reduce from Conitzer's problem of computing $\mathbb{P}(V^*|G)$, where V^* is a vote produced by a given graph G [15].† He shows that this is problem is #P-hard by reducing from counting the number of perfect matchings in a bipartite graph. Surprisingly, our proof of this hardness result uses the easiness of finding the MLE graph in polynomial time in the case when $p = q = 1/2$. Namely, we use

†While the problem Conitzer considers is slightly different than computing $\mathbb{P}(V^*|G)$, in the case where $p = 1/2$, his problem reduces to computing $\mathbb{P}(V^*|G)$.

Lemma 55 to be able to say when the input G is the maximum likelihood graph for a set of votes $V^{[m]}$, which in turn says when the oracle will successfully compute $\mathbb{P}(V^{[m]}|G)$. Formally, we prove the following theorem:

Theorem 56. *There is a randomized polynomial-time oracle reduction from computing the MLE of the maximum likelihood graph from a sequence of votes with high probability to counting the number of perfect matchings in a balanced bipartite graph.*

sketch. It suffices to consider the case where $p = q = 1/2$. Instead of directly reducing from the #P-hard problem of counting the number of perfect matchings in a balanced bipartite graph, we reduce from the #P-hard problem of computing $\mathbb{P}(V^*|G)$ given a graph G and vote V^* on n voters under the independent conversation model.

The idea of the proof is going to be to build a sequence of votes $V^{[m]}$ whose MLE we know to be the input G, and then compute

$$\mathbb{P}(V^*|G) = \frac{\mathbb{P}(V^{[m]}, V^*|G)}{\mathbb{P}(V^{[m]}|G)}.$$

Our oracle will give us the values of the right-hand side. This approach will work if $\mathbb{P}(V^*|G) \neq 0$.

So we first test for the case if $\mathbb{P}(V^*|G) = 0$. Conitzer provides a way to do this for a similar problem when the vertices of the graph have all odd degree: his reduction is from the maximum weighted b-matching problem, which we can adapt to the so-called "c-capacitated" version that we need [15, 59].

Else, $P(V^*|G) \neq 0$. We draw a sequence of votes $V^{[m]} \overset{i.i.d.}{\sim} \mathcal{V}_G \times \ldots \times \mathcal{V}_G$. Lemma 55 immediately implies that G will be the MLE for $V^{[m]}$ with failure probability less than $\delta/2$ when $m = \Omega(n^2(n^2 + \ln \frac{2}{\delta}))$. In other words, with just $\Omega(n^4)$ votes we will successfully query the oracle for $\mathbb{P}(V^{[m]}|G)$ with high probability.

It suffices to ensure that with high probability we will also successfully query the oracle for the $m+1$-length sequence $V^{[m]}, V^*$. Recall that since $\mathbb{P}(V^*|G)$ is the sum, over all satisfying edge votes, of the quantity $\left(\frac{1}{2}\right)^{|E(G)|}$, where $E(G)$ is the edge set of G and $p = 1/2$. There must be at least one satisfying edge-vote assignment since $\mathbb{P}(V^*|G) \neq 0$, so $\mathbb{P}(V^*|G) \geq \left(\frac{1}{2}\right)^{|E(G)|}$. In addition, again by Lemma 55, for any $G' \neq G, \frac{\mathbb{P}(V^{[m]}|G)}{\mathbb{P}(V^{[m]}|G')} > \alpha$ with failure probability no more than $\delta/2$ when $m = \Omega(n^2(n^2 + \ln \frac{2}{\delta} + \ln \alpha))$. Then for any $G' \neq G$,

$$\begin{aligned}\mathbb{P}(V^{[m]}, V^*|G') &< \left(\frac{1}{\alpha}\mathbb{P}(V^{[m]}|G)\right)\left(2^{|E(G)|}\mathbb{P}(V^*|G)\right) \\ &= \frac{2^{|E(G)|}}{\alpha}\mathbb{P}(V^{[m]}, V^*|G).\end{aligned}$$

Setting $\alpha = \Omega(e^{n^2})$ suffices to ensure that $\alpha > 2^{|E(G)|}$. Thus setting

$$m = \Omega(n^2(n^2 + \ln \frac{2}{\delta} + \ln \alpha))$$

as above for this setting of α, a query to the oracle for $\mathbb{P}(V^{[m]}, V^*|G)$ will fail with probability less than $\delta/2$. Setting δ to be, say, $\Theta(\frac{1}{2^n})$, yields that $m = \Omega(n^4)$. The oracle reduction, once it tests for the existence of at least one valid edge

vote, simply consists of drawing m votes from G and then querying the oracle for $\mathbb{P}(V^{[m]}, V^*|G)$ and $\mathbb{P}(V^{[m]}|G)$. The reduction then succeeds with probability at least $1 - \delta$. □

5.4 The common neighbor model

We now turn our attention to the common neighbor model. Again, we ask if it is possible to recover G by seeing only polynomially many votes. In general, it is not possible to recover G at all, let alone with only polynomially many votes:

Observation 57. *Under the common neighbor model, no algorithm can distinguish between two different perfect matchings.*

If G is a matching between the vertices, each vertex will vote how its neighbor votes, meaning that each vertex votes i.i.d. with probability p regardless. Thus there is no way to distinguish between different matchings.

5.4.1 Recovering A^2 from covariances

Given the impossibility of recovering the graph, we relax the problem to the following: Find a graph that is likely to produce the given votes in the sense that the expected covariances of this graph should be as close as possible (under some norm) to the covariances of the observed votes. This problem is motivated by the algorithm for the independent conversation model which finds a graph whose expected covariances match the measured covariances.

Yet even if we were to know the *expected* covariances of the input votes, finding a graph whose expected covariances are close to those input covariances remains challenging:

Observation 58. *For the common neighbor model, finding a graph with given expected vote covariances is at least as hard as recovering an adjacency matrix from its square.*

To prove this observation, it suffices to show that the expected covariances are a function solely of the entries of A^2. Then recovering A from A^2 consists of using the entries of A^2 to compute the expected covariances, at which point the adjacency matrix of a graph with those covariances will be exactly A. The i, jth entry of A^2 is the number of length-two paths between i and j, so it is enough to write the covariances of a graph in terms of the following: For $\Gamma(v)$ the neighborhood of a vertex v, denote $d_{uv} = |\Gamma(v) \cap \Gamma(u)|$, $d_u = |\Gamma(u) \setminus (\Gamma(v) \cap \Gamma(u))|$, and d_v analogously. The covariances are a function of d_u, d_v, and d_{uv}. For the sake of simplicity we will assume that $|\Gamma(u)|$ and $|\Gamma(v)|$ are odd, but it is straightforward to modify the formula given below in the cases when they are not.

Lemma 59. *Assume $|\Gamma(u)|$ and $|\Gamma(v)|$ are odd. For $p = 1/2$,*

$$Cov(X_u, X_v) = \frac{1}{2^{d_{uv}-2}} \left(\sum_{k=0}^{d_{uv}} \binom{d_{uv}}{k} P_{u,v}(k) P_{v,u}(k) \right) - 1,$$

where, for $\theta_{u,v} = (d_{uv} + d_u + 1)/2 - k$,

$$P_{u,v}(k) = \begin{cases} \frac{1}{2^{d_u}} \sum_{i=\theta_{u,v}}^{d_u} \binom{d_u}{i} & \text{if } 0 \leq \theta_{u,v} \leq d_u \\ 0 & \text{if } \theta_{u,v} > d_u \\ 1 & \text{if } \theta_{u,v} \leq 0. \end{cases}$$

Proof. Let X_u represent vertex u's vote. When $p = 1/2$, $E[X_u] = 0$, and $P(X_u = X_v = 1) = P(X_u = X_v = 0)$, so the covariance $Cov(X_u, X_v)$ is

$$E[X_u X_v] = 2P(X_u = X_v) - 1 = 4P(X_u = X_v = 1) - 1.$$

To determine $P(X_u = X_v = 1)$, we condition on the number of common neighbors that voted 1:

Assuming some k common neighbors vote 1, in order for u to vote 1, u needs an additional $\frac{d_{uv}+d_u+1}{2} - k$ neighbors to vote 1. If k is already at least $\frac{d_{uv}+d_u+1}{2}$, then the probability of voting 1 is already 1; on the other hand if there aren't enough remaining vertices to vote 1, then the probability is 0. This yields $P_{u,v}(k)$ as the probability that u votes 1 given that k common neighbors of u and v voted 1.

Now we can write $P(X_u = X_v = 1)$ as

$$\sum_{k=0}^{d_{uv}} \binom{d_{uv}}{k} p^k (1-p)^{d_{uv}-k} P_{u,v}(k) P_{v,u}(k),$$

completing the proof. □

When recovering a graph from a sequence of input votes, we are not even given

the expected covariances of the input votes. Instead we can calculate the measured covariances, from which we can determine A^2. At this point, we have a function inversion problem on our hands: We can find A^2 merely by recovering these d_{uv}'s and d_u's from the covariances, but given that the formula given in Lemma 59 is not closed, this is not trivial. Since there are only polynomially many possible values for d_{uv} and d_u, we can simply try all values to find the covariance closest to the observed value. However, there may be covariances that are exponentially close to each other, making it impossible to distinguish between these values for given d_{uv}, d_u. In this case the values recovered for the entries of A^2 may not be unique, which in the worst case leads to the generalized squared adjacency problem. Even in the case when we recover unique values, it still reduces to the squared adjacency problem.

While the squared adjacency problem is open, we show the following:

Theorem 60. *The generalized squared adjacency problem is* NP*-hard.*

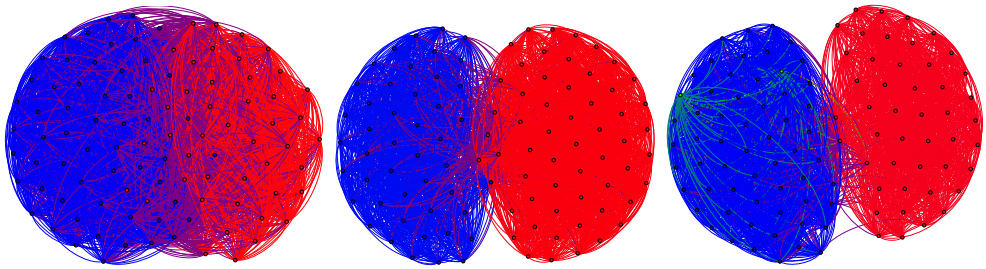
Proof. The reduction is from CLIQUE, which asks if there is a clique of size k on the input graph. Given a graph $G = (V, E)$ and an integer k, we construct a set system $\{S_{ij}\}$ with $(n+1)^2$ sets, where $n = |V|$. We then show that G has a clique of size k if and only if there is a graph $G' = (V \cup \{v\}, E')$ such that the i, jth entry of $A(G')^2$ is in $S_{i,j}$, where $A(G')$ is the adjacency matrix of G'. The

$(n+1)^2$-sized set system $\{S_{i,j}\}$ is defined as follows:

$$S_{ij} = \begin{cases} \{0,1\} & \text{if } i \neq j \text{ and } i,j \neq v \text{ and } (i,j) \in E(G) \\ \{0\} & \text{if } i \neq j \text{ and } i,j \neq v \text{ and } (i,j) \notin E(G) \\ \{0\} & \text{if } i = v \text{ and } j \neq v \\ \{0\} & \text{if } j = v \text{ and } i \neq v \\ \{k\} & \text{if } i = j = v \\ \{0,1\} & \text{if } i = j \text{ and } i,j \neq v \end{cases}$$

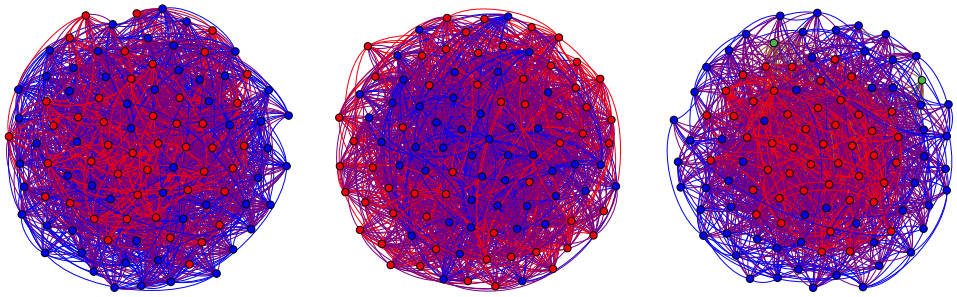
Assume there is a clique of size k in G. Then G' is defined as follows: Denote the vertex set of the clique in G by C. G' will have an edge between v and all members of C, and no other edges. It is straightforward to check that the i,jth entry of $A(G')^2$ is in S_{ij} by noting that the diagonal entries of $A(G')^2$ are the vertices' degrees and the off-diagonal entries counts the number of common neighbors.

In the other direction, assume there is such a graph G' whose squared adjacency matrix satisfies the constraints imposed by the set system $\{S_{i,j}\}$. In this case, the clique of size k in G will be exactly the neighborhood of v in G' (not including v itself). Call $N(v)$ the neighborhood of v in G'. Note the degree of v in G' must be k, by definition of S_{vv}, i.e. $|N(v)| = k$. Consider a distinct pair of vertices i,j in $N(v)$. The vertices i and j have at least one common neighbor in G', namely v, because both are in $N(v)$, meaning that $A(G')^2_{ij} \geq 1$. But if (i,j) is not an edge in G then $S_{ij} = \{0\}$ by the definition of S_{ij}, a contradiction, forcing $N(v)$ to be a clique as required. □



(a) 101st Congress (1989-1990) **(b)** 106th Congress (1999-2000) **(c)** 113th Congress (2013-2014)

Figure 5.3: Graphs of the US Senate for three congressional terms under the independent conversation model. Democrats are colored blue, Republicans are red, and Independents are green.



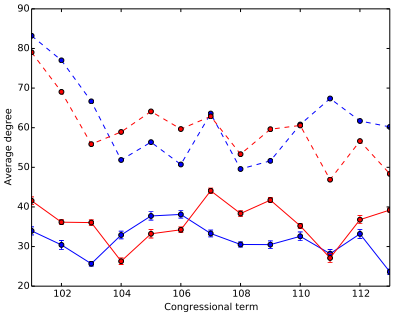
(a) 101st Congress (1989-1990) **(b)** 106th Congress (1999-2000) **(c)** 113th Congress (2013-2014)

Figure 5.4: Graphs of the US Senate for three congressional terms under the common neighbor model. Democrats are colored blue, Republicans are in red, and Independents are in green.

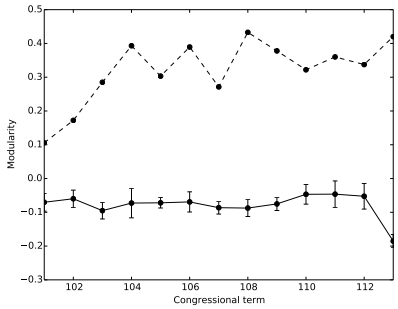
5.4.2 A heuristic approach

Unlike in the independent conversation model, we have no efficient algorithm for producing the social network under the common neighbor model. Hence, we employ a heuristic to find a graph that satisfies or comes close to satisfying the constraints imposed on it by the measured covariances.

Because of the computational hardness of this problem, we propose a heuristic approach to learn networks under the common neighbor model. This heuristic will



(a) Average degree. The blue and red lines are the average degree of the Democrats and the Republicans, respectively.



(b) Modularity between Democrats and Republicans. (Independents are included with the party with which they caucus.)

Figure 5.5: Data for the 101st-113th Congress. Dashed and solid lines are statistics for the independent conversation model and common neighbor model, respectively. Error bars represent one standard deviation, over 20 trials.

be used in our experimental results in Section 5.5. Our heuristic, Algorithm 5, finds those pairs of vertices whose current expected covariance (assuming $p = 1/2$) is farthest away from the measured covariance and modifies the graph to decrease that gap.

The local changes we want to make clearly cannot just consist of adding/removing single edges: Say the covariance between a given pair of vertices needs to go up and those vertices' neighborhoods are currently empty. The only way to increase the covariance is to add at least two edges: (i, v) and (v, j) for some other vertex v. The natural compromise is then to add or remove the minimal number of edges (either one or two) to change the covariance, as seen in Algorithm 6.

Algorithm 5 Common neighbor heuristic

Input: $\{\hat{c}_{ij}\}$, measured covariances between voters i and j, and T, the number of iterations to run.

$G \sim \mathcal{G}(n, 1/2)$, $G_{best} := \emptyset$

$\{c_{ij}\} := \{\sigma(i,j)\}$ # calculate expected covariances

for 0 to T **do**

$i, j := \text{argmax}_{i,j} |c_{ij} - \hat{c}_{i,j}|$

$G := \text{Modify}G(G, i, j, \hat{c}_{ij}, c_{ij})$

$\{c_{ij}\} := \{\sigma(i,j)\}$ # update the expected covariances

if $\sum_{i,j} c_{ij} - \hat{c}_{ij}$ is smallest so far **then** $G_{best} := G$

end for

return G_{best}

5.5 Experimental Results

In this section, we test our algorithms on United States Senate roll call votes. We examine each two-year congressional session as one voting network. Each Senate member is an agent who either votes for the bill in question, against, or does not vote (either because the senator served only part of the term or because the senator just didn't vote on the bill), yielding votes from the set $\{-1, 0, 1\}$.

Obviously, our models are simplifications — they don't take into account evolution of opinion, nor do they take into account the possibility of anti-correlated voters. Even assuming that either model is representative, when presented real data, the parameter p is not given, as is assumed above. The algorithms we present assume that $p = 1/2$, which is not necessarily the case. Finally, we are given a fixed amount of data, independent of the number of voters.

Despite these limitations, for the independent conversation model, our covariance test, which forms the basis for Theorem 51, results in intuitive behavior.

Algorithm 6 Modify$G$

Input: Graph G; vertices i, j; $\hat{c}_{ij}$, c_{ij} the measured and expected covariances between i and j.

$unconnected := V(G) \setminus (\Gamma(i) \cup \Gamma(j) \cup \{i, j\})$

$cn := \Gamma(i) \cap \Gamma(j)$

if $c_{ij} - \hat{c}_{ij} > 0$ **then**

randomize among whichever of these are available:

1: x := random(i, j), y := random$(unconnected)$

add edge (x, y) to G

2: x := random(i, j), y := random(cn).

delete edge (x, y) from G

3: y := random(cn)

delete edges (i, y) and (j, y) from G

else if $c_{ij} - \hat{c}_{ij} < 0$ **then**

randomize among whichever of these are available:

1: y := random$(unconnected)$

add edges (i, y) and (j, y) to G

2: y := random$(\Gamma(i) \setminus cn)$

add (i, y) or delete (j, y) from G randomly

3: y := random$(\Gamma(j) \setminus cn)$

add (j, y) or delete (i, y) from G randomly

end if

return G

where random(.) selects an element of its input u.a.r.

Note that while our model assumes binary votes, our covariance test is general enough to handle such votes — covariance is calculated between the $\{-1, 0, 1\}$-valued votes and the threshold remains the same as in the original covariance test.[‡] Examples of the results from this covariance test on the US Senate are shown in Figure 5.3. Given the highly structured nature of these graphs, it is possible to recover senators' places on the left/right political spectrum, but since this is not the focus of this chapter, we do not go into any further detail here.

For the common neighbor model, we use Algorithm 5. Examples of results of this heuristic run on US Senate data are shown in Figure 5.4. Graphs under this model appear to be very different from those found using the covariance test under the independent conversation model.

To demonstrate these marked differences, in Figure 5.5 we give modularity values and average degrees of Democrats and Republicans under both models for the period 1989-2014 (corresponding to the 101st through 113th Congresses). Modularity is a standard measure of the amount of division between communities [57]. Both average degree and modularity are much higher under the independent conversation model (dashed lines in Figure 5.5) than under the common neighbor model (solid lines). Since the heuristic is randomized, we average these statistics over twenty graphs, each of which is an independent run of the heuristic with 100,000 rounds.

[‡]We do, however, use the unbiased sample covariance instead of the biased sample covariance, as the assumption that $p = 1/2$ no longer necessarily holds, despite the analysis of the algorithm assuming it.

5.6 Conclusion

In this chapter we derive algorithms and lower bounds for recovering graphs from their vertices' votes under two distinct models. We also present experiments on the U.S. Senate voting network. In the independent conversation model, we show when the graph is recoverable using only a polynomial number of votes. However, if we want to instead take a maximum likelihood approach to recovering graphs, then the task becomes computationally hard.

The common neighbor model, on the other hand, leads to significantly different results. Not only is it impossible to recover the graph using only polynomially many votes, finding a graph whose votes' covariances are close to the observed covariances leads to having to solve a hard problem (the generalized squared adjacency problem).

This implies that these models really are very different from each other, despite their very similar definitions. This is strong evidence that much care needs to be taken when choosing voting models for network inference. Experiments on U.S. Senate roll call data support this conclusion.

Cited Literature

[1] Alessia Amelio and Clara Pizzuti. Analyzing voting behavior in Italian Parliament: Group cohesion and evolution. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012*, pages 140–146. IEEE, 2012.

[2] Clio Andris, David Lee, Marcus J. Hamilton, Mauro Martino, Christian E. Gunning, and John Armistead Selden. The rise of partisanship and supercooperators in the US House of Representatives. *PLoS ONE*, 10(4), 2015.

[3] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[4] Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1046–1059, 2016.

[5] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 464–473, 2014.

[6] Avrim Blum, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 253–262, 1994.

[7] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[8] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 609–618, 2008.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[10] Ivan Brugere, Brian Gallagher, and Tanya Y. Berger-Wolf. Network structure inference, A survey: Motivations, methods, and applications. *arXiv preprint arXiv:1610.00782*, 2016.

[11] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 634–649, 2015.

[12] Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the 46th Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 1–10, 2014.

[13] Bee-Chung Chen, Lei Chen, Raghu Ramakrishnan, and David R. Musicant. Learning from aggregate views. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 3, 2006.

[14] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[15] Vincent Conitzer. The maximum likelihood approach to voting on social networks. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 1482–1487. IEEE, 2013.

[16] Nando de Freitas and Hendrik Kück. Learning about individuals from group statistics. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, UAI '05, Edinburgh, Scotland, July 26-29, 2005*, pages 332–339, 2005.

[17] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[18] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems 28, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2350–2358, 2015.

[19] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[20] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 117–126, 2015.

[21] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006*, pages 265–284, 2006.

[22] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[23] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 51–60, 2010.

[24] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Srinivas Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM*, 64(2):8:1–8:37, 2017.

[25] Benjamin Fish, Yi Huang, and Lev Reyzin. Recovering social networks by observing votes. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 376–384, 2016. http://dl.acm.org/citation.cfm?id=2936980.

[26] Benjamin Fish and Lev Reyzin. On the complexity of learning from label proportions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1675–1681, 2017.

[27] Benjamin Fish, Lev Reyzin, and Benjamin I. P. Rubinstein. Sublinear-time adaptive data analysis. *arXiv preprint arXiv:1709.09778*, 2017.

[28] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[29] Silvio Frischknecht, Barbara Keller, and Roger Wattenhofer. Convergence in (social) influence networks. In *Distributed Computing*, pages 433–446. Springer, 2013.

[30] Andrew Gelman and Eric Loken. The statistical crisis in science data-dependent analysis—a "garden of forking paths"—explains why many statistically significant comparisons don't hold up. *American Scientist*, 102(6):460–465, 2014.

[31] Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2002.

[32] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 595–604, 2011.

[33] Eric Goles and Jorge Olivos. Periodic behaviour of generalized threshold functions. *Discrete Mathematics*, 30(2):187–189, 1980.

[34] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *TKDD*, 5(4):21:1–21:37, 2012.

[35] Michel Grabisch and Agnieszka Rusinowska. A model of influence in a social network. *Theory and Decision*, 69(1):69–96, 2008.

[36] Umberto Grandi, Emiliano Lorini, and Laurent Perrussel. Propositional opinion diffusion. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 989–997, 2015.

[37] Jerónimo Hernández-González, Iñaki Inza, and José Antonio Lozano. Learning bayesian network classifiers from label proportions. *Pattern Recognition*, 46(12):3425–3440, 2013.

[38] Yi Huang. *Problems in Learning under Limited Resources and Information*. PhD thesis, University of Illinois at Chicago, 2017.

[39] Arun Shankar Iyer, J. Saketha Nath, and Sunita Sarawagi. Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 530–538, 2014.

[40] Arun Shankar Iyer, J. Saketha Nath, and Sunita Sarawagi. Privacy-preserving class ratio estimation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 925–934, 2016.

[41] Aleks Jakulin, Wray Buntine, Timothy M La Pira, and Holly Brasher. Analyzing the US Senate in 2003: Similarities, clusters, and blocs. *Political Analysis*, 17(3):291–310, 2009.

[42] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? Personalized differential privacy. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pages 1023–1034, 2015.

[43] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 531–540, 2008.

[44] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[45] Georgios Kellaris and Stavros Papadopoulos. Practical differential privacy via grouping and smoothing. *Proceedings of the VLDB Endowment*, 6(5):301–312, 2013.

[46] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[47] Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I. Jordan. The big data bootstrap. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[48] Jason D. Lee, Dennis L. Sun, Yuekai Sun, and Jonathan E. Taylor. Exact post-selection inference, with application to the lasso. *Annals of Statististics*, 44(3):907–927, 06 2016.

[49] David Liben-Nowell and Jon M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[50] Bing-Rong Lin, Ye Wang, and Shantanu Rane. On the benefits of sampling in privacy preserving statistical analysis on distributed databases. *arXiv preprint arXiv:1304.4613*, 2013.

[51] Kevin T Macon, Peter J Mucha, and Mason A Porter. Community structure in the united nations general assembly. *Physica A: Statistical Mechanics and its Applications*, 391(1):343–361, 2012.

[52] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103, 2007.

[53] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.

[54] Rajeev Motwani and Madhu Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.

[55] David R. Musicant, Janara M. Christensen, and Jamie F. Olson. Supervised learning by training on aggregate outputs. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 252–261, 2007.

[56] Seth A. Myers and Jure Leskovec. On the convexity of latent social network inference. In *Advances in Neural Information Processing Systems 23: 24th*

Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada., pages 1741–1749, 2010.

[57] Mark E.J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[58] Giorgio Patrini, Richard Nock, Tiberio Caetano, and Paul Rivera. (almost) no label no cry. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 190–198, 2014.

[59] Michal Penn and Moshe Tennenholtz. On multi-object auctions and matching theory: Algorithmic aspects. In *Graph Theory, Combinatorics and Algorithms*, pages 173–188. Springer, 2005.

[60] Mason A Porter, Peter J Mucha, Mark E.J. Newman, and Andrew J Friend. Community structure in the United States House of Representatives. *Physica A: Statistical Mechanics and its Applications*, 386(1):414–438, 2007.

[61] Ariel D. Procaccia, Nisarg Shah, and Eric Sodomka. Ranked voting on social networks. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2040–2046, 2015.

[62] Novi Quadrianto, Alexander J. Smola, Tibério S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.

[63] Ryan M. Rogers, Aaron Roth, Adam D. Smith, and Om Thakkar. Max-information, differential privacy, and post-selection hypothesis testing. In *57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 487–494, 2016.

[64] Stefan Rüping. SVM classifier estimation from group probabilities. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 911–918, 2010.

[65] Kevin Schaul. A quick puzzle to tell whether you know what people are thinking, October 2015. [The Washington Post; posted online 09-October-2015].

[66] Nicolas Schwind, Katsumi Inoue, Gauvain Bourgne, Sébastien Konieczny, and Pierre Marquis. Belief revision games. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1590–1596, 2015.

[67] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[68] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

[69] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 71–79, 2013.

[70] Thomas Steinke and Jon Ullman. Between pure and approximate differential privacy. In *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2015.

[71] Thomas Steinke and Jonathan Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Proceedings of the 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 1588–1628, 2015.

[72] Marco Stolpe and Katharina Morik. Learning from label proportions by optimizing cluster model selection. In *Proceedings of the Joint European*

Conference on Machine Learning and Knowledge Discovery in Databases, ECML-PKDD 2011, Athens, Greece, September 5-9, 2011, pages 349–364, 2011.

[73] Benjamin Taskar, Ming Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 659–666, 2003.

[74] Alan Tsang, John A. Doucette, and Hadi Hosseini. Voting with social influence: Using arguments to uncover ground truth. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1841–1842, 2015.

[75] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[76] V.N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[77] Jeffrey Scott Vitter. Faster methods for random sampling. *Communications of the ACM*, 27(7):703–718, 1984.

[78] Andrew Scott Waugh, Liuyi Pei, James H. Fowler, Peter J. Mucha, and Mason A. Porter. Party polarization in Congress: A network science approach. *arXiv preprint arXiv:0907.3509*, 2009.

[79] Janusz Wojtusiak, Katherine Irvin, Aybike Birerdinc, and Ancha V. Baranova. Using published medical results and non-homogenous data in rule learning. In *10th International Conference on Machine Learning and Applications and Workshops, ICMLA 2011, Honolulu, Hawaii, USA, December 18-21, 2011. Volume 2: Special Sessions and Workshop*, pages 84–89, 2011.

[80] C. K. Wong and Malcolm C. Easton. An efficient method for weighted sampling without replacement. *SIAM Journal of Computing*, 9(1):111–113, 1980.

[81] Houping Xiao, Jing Gao, Qi Li, Fenglong Ma, Lu Su, Yunlong Feng, and Aidong Zhang. Towards confidence in the truth: A bootstrapping based truth discovery approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1935–1944, 2016.

[82] Ke Yang. On learning correlated Boolean functions using statistical queries. In *Proceedings of the 12th International Conference on Algorithmic Learning Theory ALT 2001, Washington, DC, USA, November 25-28, 2001*, pages 59–76, 2001.

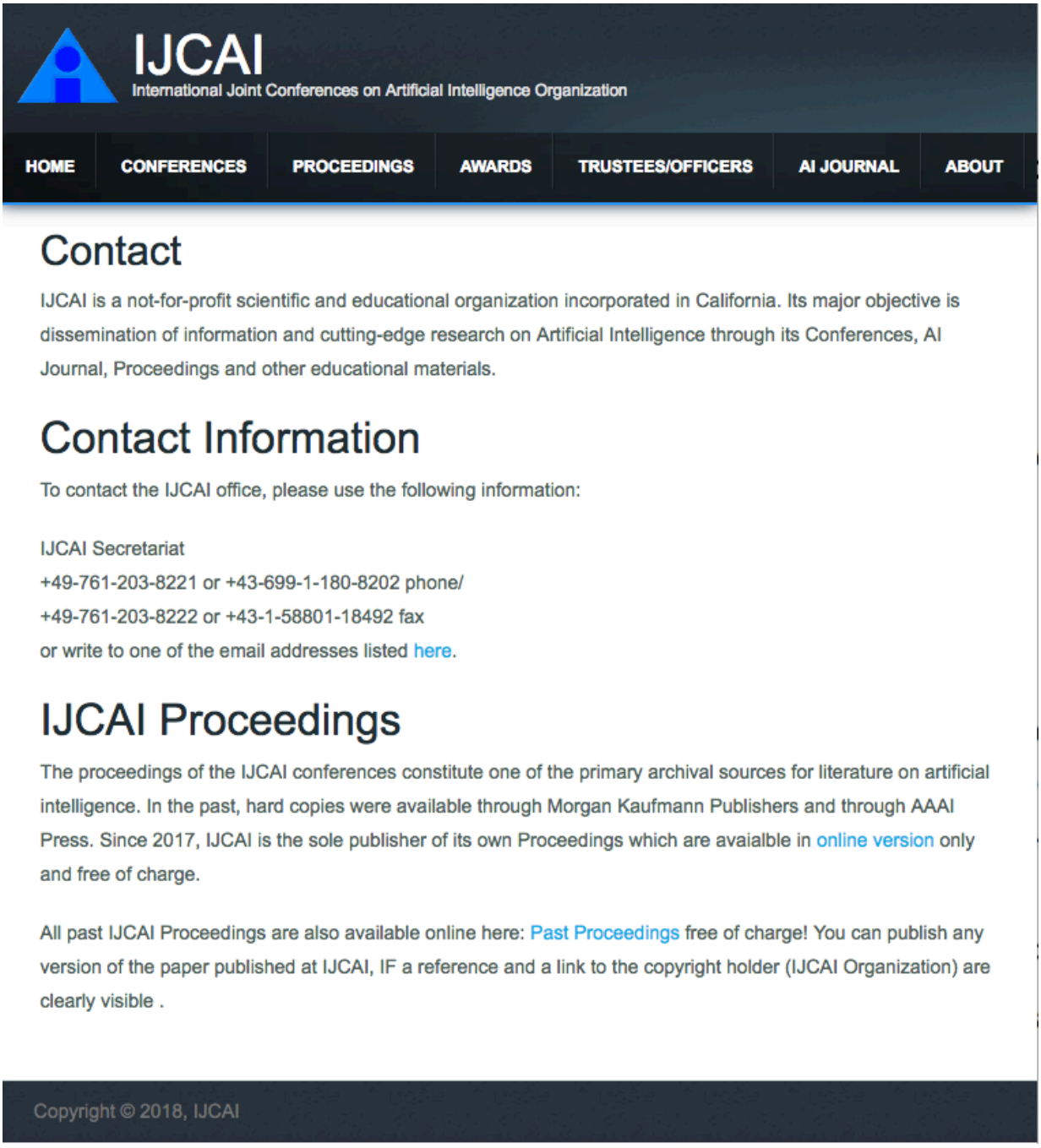
[83] Felix X. Yu, Krzysztof Choromanski, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014.

[84] Felix X. Yu, Dong Liu, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. $\propto$-SVM for learning with label proportions. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 504–512, 2013.

[85] Yan Zhang, A.J. Friend, Amanda L. Traud, Mason A. Porter, James H. Fowler, and Peter J. Mucha. Community structure in Congressional cosponsorship networks. *Physica A: Statistical Mechanics and its Applications*, 387(7):1705–1712, 2008.

Appendix

This appendix contains reproductions of statements from the publishers' websites detailing the use policies that allow the original publications to be reproduced in this thesis.


IJCAI
International Joint Conferences on Artificial Intelligence Organization
HOME
CONFERENCES
PROCEEDINGS
AWARDS
TRUSTEES/OFFICERS
AI JOURNAL
ABOUT
Contact
IJCAI is a not-for-profit scientific and educational organization incorporated in California. Its major objective is dissemination of information and cutting-edge research on Artificial Intelligence through its Conferences, AI Journal, Proceedings and other educational materials.
Contact Information
To contact the IJCAI office, please use the following information:
IJCAI Secretariat
+49-761-203-8221 or +43-699-1-180-8202 phone/
+49-761-203-8222 or +43-1-58801-18492 fax
or write to one of the email addresses listed here.
IJCAI Proceedings
The proceedings of the IJCAI conferences constitute one of the primary archival sources for literature on artificial intelligence. In the past, hard copies were available through Morgan Kaufmann Publishers and through AAAI Press. Since 2017, IJCAI is the sole publisher of its own Proceedings which are avaialble in online version only and free of charge.
All past IJCAI Proceedings are also available online here: Past Proceedings free of charge! You can publish any version of the paper published at IJCAI, IF a reference and a link to the copyright holder (IJCAI Organization) are clearly visible .
Copyright © 2018, IJCAI

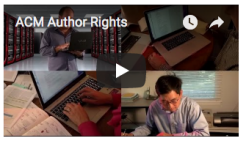
ACM Information for Authors

Author Rights | FAQ

ACM Author Rights

ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession



CHOOSE

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works.

POST

Authors can post the accepted, peer-reviewed version prepared by the author-known as the "pre-print"-to the following sites, with a DOI pointer to the Definitive Version of Record in the ACM Digital Library.

- On Author's own Home Page *and*
- On Author's Institutional Repository *and*
- In any repository legally mandated by the agency funding the research on which the work is based *and*
- On any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise profit from serving articles.

DISTRIBUTE

Authors can post an *Author-Izer* link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library

- On the Author's own Home Page *or*
- In the Author's Institutional Repository.

REUSE

Authors can reuse any portion of their own work in a new work of *their own* (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is *not* the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

- Commercially produced course-packs that are *sold* to students require permission and possibly a fee.

CREATE

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

RETAIN

Authors retain all *perpetual rights* laid out in the ACM Author Rights and Publishing Policy, including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM

Have more questions? Check out the FAQ.

back to top

Vita

Education

Ph.D., University of Illinois at Chicago, Chicago, Illinois, 2018

M.S., University of Illinois at Chicago, Chicago, Illinois, 2015

B.A., Pomona College, Claremont, California, 2013

Research and Teaching Experience

Research Assistant, University of Illinois at Chicago, 2016, 2017.

Visiting Scholar, University of Utah, 2017.

Teaching Assistant, University of Illinois at Chicago, 2013 – 2017.

Visiting Researcher, University of Melbourne, Melbourne, Australia, 2016.

Research Intern, MIT Lincoln Laboratory, Lexington, MA, 2014 – 2016.

Papers

Benjamin Fish, Lev Reyzin, and Benjamin I.P. Rubinstein. Sub-Linear Time Adaptive Data Analysis. Submitted.

Benjamin Fish and Lev Reyzin. On the Complexity of Learning from Label Proportions. *International Joint Conference on Artificial Intelligence.*

Benjamin Fish and Rajmonda S. Caceres. A task-driven approach to time scale detection in dynamic networks. *Workshop on Mining and Learning with Graphs*, 2017.

Benjamin Fish and Lev Reyzin. Open Problem: Meeting Times for Learning Random Automata. *Conference on Learning Theory*, 2017.

Benjamin Fish, Rahul Kushwaha, and György Turán. Betweenness centrality profiles in trees. *Journal of Complex Networks*, Volume 5, Issue 5, October 2017.

Benjamin Fish, Yi Huang, and Lev Reyzin. Recovering Social Networks by Observing Votes. *Autonomous Agents and Multiagent Systems International Conference*, 2016.

Benjamin Fish, Jeremy Kun, and Ádám D. Lelkes. A Confidence-Based Approach for Balancing Fairness and Accuracy. *SIAM Intl. Conference on Data Mining*.

Benjamin Fish and Rajmonda S. Caceres. Handling Oversampling in Dynamic Networks Using Link Prediction. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Benjamin Fish, Jeremy Kun, Ádám D. Lelkes, Lev Reyzin, and György Turán. On the Computational Complexity of MapReduce. *In Proceedings of the 29th International Symposium on Distributed Computing*.

Benjamin Fish, Jeremy Kun, and Ádám D. Lelkes. Fair boosting: A case study. *ICML 2015 Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2015.

Ghassan Sarkis, Shahriar Shahriari, and the Pomona College Undergraduate Research Circle. Zero-sum Flows of the Linear Lattice. *Finite Fields and their Application*, 2015.

Benjamin Fish, Robert W. McGrail, James Belk, Solomon Garber, and Japheth Wood. CSPs and Connectedness: P/NP Dichotomy for Idempotent, Right Quasigroups. *16th Intl. Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2014.

Ghassan Sarkis, Shahriar Shahriari, and the Pomona College Undergraduate Research Circle. Diamond Free Subsets in the Linear Lattices. *Order*, 2013.

Advised by Rena Levitt. The Word Problem in Quandles. B.A. Thesis, 2013.

Advised by Ran Libeskind-Hadas. The Cophylogeny Reconstruction Problem. B.A. Thesis, 2013.

Benjamin Fish, Ammar Khan, Nabil Hajj Chehade, Chieh Chien, and Greg Pottie. Feature selection based on mutual information for human activity recognition. *IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, 2012.