On Learning and Covering Structured Distributions

by

Gautam Kamath

B.S., Cornell University (2012)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

Author . . . .
Department of Electrical Engineering and Computer Science
August 29, 2014

Certified by . . . .
Constantinos Daskalakis
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

On Learning and Covering Structured Distributions

by

Gautam Kamath

Submitted to the Department of Electrical Engineering and Computer Science
on August 29, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

We explore a number of problems related to learning and covering structured distributions.

Hypothesis Selection: We provide an improved and generalized algorithm for selecting a good candidate distribution from among competing hypotheses. Namely, given a collection of N hypotheses containing at least one candidate that is ε-close to an unknown distribution, our algorithm outputs a candidate which is $O(\varepsilon)$-close to the distribution. The algorithm requires $O(\log N/\varepsilon^2)$ samples from the unknown distribution and $O(N \log N/\varepsilon^2)$ time, which improves previous such results (such as the Scheffé estimator) from a quadratic dependence of the running time on N to quasilinear. Given the wide use of such results for the purpose of hypothesis selection, our improved algorithm implies immediate improvements to any such use.

Proper Learning Gaussian Mixture Models: We describe an algorithm for properly learning mixtures of two single-dimensional Gaussians without any separability assumptions. Given $\tilde{O}(1/\varepsilon^2)$ samples from an unknown mixture, our algorithm outputs a mixture that is ε-close in total variation distance, in time $\tilde{O}(1/\varepsilon^5)$. Our sample complexity is optimal up to logarithmic factors, and significantly improves upon both Kalai et al. [40], whose algorithm has a prohibitive dependence on $1/\varepsilon$, and Feldman et al. [33], whose algorithm requires bounds on the mixture parameters and depends pseudo-polynomially in these parameters.

Covering Poisson Multinomial Distributions: We provide a sparse ε-cover for the set of Poisson Multinomial Distributions. Specifically, we describe a set of $n^{O(k^3)}(k/\varepsilon)^{\mathrm{poly}(k/\varepsilon)}$ distributions such that any Poisson Multinomial Distribution of size n and dimension k is ε-close to a distribution in the set. This is a significant sparsification over the previous best-known ε-cover due to Daskalakis and Papadimitriou [24], which is of size $n^{f(k,1/\varepsilon)}$, where f is polynomial in $1/\varepsilon$ and exponential in k. This cover also implies an algorithm for learning Poisson Multinomial Distributions with a sample complexity which is polynomial in $k, 1/\varepsilon$ and $\log n$.

Thesis Supervisor: Constantinos Daskalakis
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

First off, thank you to my advisor, Costis Daskalakis, for introducing me to this amazing field of study. Your support and advice has guided me through times both easy and hard. I look forward to the road ahead.

Thank you to all my collaborators on research projects past and present, including Jayadev Acharya, Christina Brandt, Clément Canonne, Costis Daskalakis, Ilias Diakonikolas, Bobby Kleinberg, Jerry Li, Nicole Immorlica, Christos Tzamos. I'm incredibly fortunate to consistently find myself surrounded by such a talented and tireless group of researchers.

Extra special thanks to Bobby Kleinberg, my undergraduate research advisor. From nominating me as a sophomore for the Experience Theory Project at University of Washington, to adding to your immense list of responsibilities by taking me on as a research advisee, to trusting me to present our result at STOC when I was just a wet-behind-the-ears undergraduate, you've always believed in me, even at times I found it hard to believe in myself.

Thank you to all the members of the MIT Theory group for helping make the 5th and 6th floors of Stata feel like home. In particular, thanks to Pablo Azar, Arturs Backurs, Adam Bouland, Aloni Cohen, Michael Forbes, Themis Gouleakis, Daniel Grier, Jerry Li, Ludwig Schmidt, Adrian Vladu and Henry Yuen. Special thanks to Aaron Sidford for your efforts in bolstering the sense of community within the group, which include bringing us together for some Not So Great Ideas. Extra special thanks to Ilya Razenshteyn for being a true Russian friend, in every sense of the phrase.

Thank you to all my non-Theory MIT friends for the occasional diversions from work and excellent company. I'll single out Sara Achour, Ariel Anders, Pavel Chvykov, Eva Golos, Michal Grzadkowski, Twan Koolen, Albert Kwon and Andrew Sabisch, with whom I've shared many fond memories. With adventures like karaoke in K-Town, boating in Boston Harbor, nightlife in New York, and chilling in Cambridge, you've helped make my life a little bit more interesting.

Special thanks Clément Canonne, who isn't in either of the two aforementioned groups, yet blends perfectly with both. Whether it's discussing property testing via instant messages, hosting get-togethers at your place, or providing unconventional refuge for weary travellers in New York, you've consistently been there. Thanks for making both my summers in Cambridge much more fun.

Thank you to my other non-MIT friends, for helping me keep in touch with the real world and occasionally get away from MIT, whether that means Montreal, the Berkshires, Walden Pond, or the world of Dota 2. To name a few of you, thanks to Laura Castrale, Luke Chan, Dominick Grochowina, Taylor Helsel, Rocky Li and Michael Wu.

Thanks to the all the administrative assistants in the Theory group for helping make things happen, in particular, Be Blackburn, Joanne Hanley, and Nina Olff. Without your help, we would never have great traditions like the Theory Retreat and Theory Lunch.

Finally, thank you to my parents Markad and Padma, my brother Anand, and my sister-in-law Archana. I can't fathom where I would be without your unwavering love and support.

Contents

Chapter 1

Introduction

Distribution learning is one of the oldest problems in statistics. Given sample access to a probability distribution, the goal is to output a hypothesis which is close to the original distribution. Over the years, a number of methods have been proposed for this problem, including histograms [47], kernel methods [52], maximum likelihood [35], and metric entropy [42, 43]. These classical methods generally focus solely on sample efficiency: minimizing our hypothesis' error given a particular number of samples. As computer scientists, we care about optimizing both sample and time complexity.

There is good news and bad news. The good news is that we can learn an arbitrary discrete distribution over $[N]$ to ε-accuracy at the cost of $O\left(\frac{N}{\varepsilon^2}\right)$ samples and time steps. The bad news is that sometimes we need something more efficient. Often, N is prohibitively large, and we require algorithms which are sublinear in the support size. The even worse news: this problem is impossible for arbitrary continuous distributions – no finite-sample algorithm can even determine if a distribution is discrete or continuous.

Another classic statistical problem is covering. Given a class of probability distributions $\mathcal{C}$, we wish to output a finite set of distributions $\mathcal{X}$ such that for every distribution in $\mathcal{C}$, there exists a distribution in $\mathcal{X}$ which is close to it. Naturally, we would like to minimize the size of this set. This time, the size of an ε-cover for discrete distributions over $[N]$ is $\left(\frac{N}{4\varepsilon}\right)^N$. The cover size is exponential in N, and thus it quickly becomes enormous.

These results are clearly insufficient – we have to make some additional assumptions. One approach is to give more power to the algorithm by changing its access to the probability distribution. Some models which have been studied include allowing the algorithm to draw samples conditioned on a subset of the domain [11, 12], or being able to query the PDF or CDF of the distribution [13].

Another approach is to make assumptions on the *structure* of the distribution. We could assume the distribution is of a particular shape. For example, Birgé's classic result [8] shows that a monotone distribution over $[N]$ can be learned to ε-accuracy at the cost of $O\left(\frac{\log N}{\varepsilon^3}\right)$ samples and time steps – an exponential improvement in the complexity! Other sublinear results have been shown for distributions which are k-modal [19, 21], k-flat [14], or piecewise polynomial [15]. We could also assume the distribution is a mixture of simple distributions, such as a mixture of Gaussians [46, 16, 5, 54, 3, 10, 40, 45, 7, 38, 22] or mixtures of product distributions [41, 34]. Finally, we could assume that the distribution is a sum of simple distributions, such as Bernoulli [20, 25], categorical [24], or integer-valued [18] random variables. For instance, consider a sum of N independent (but not necessarily identical) Bernoulli random variables. Although the support is $[N]$, we can ε-learn it with only $\tilde{O}(1/\varepsilon^3)$ samples and ε-cover it with a set of size $N^2 + N \cdot (1/\varepsilon)^{O(\log^2(1/\varepsilon))}$.

It is clear from these cases that we gain immense statistical and computational power by exploiting the structure of a distribution. We present some new results on learning and covering structured distributions. In particular, we present a learning result for mixtures of Gaussians, a covering result for Poisson Multinomial distributions, and a tool for hypothesis selection. Not only do these results improve on the prior state of the art, we also believe that the tools and techniques may be applied to a variety of other problems.

1.1 Structure of the Thesis

Chapter 2 introduces a tool for selecting a hypothesis from a collection of candidate hypotheses. It provides the guarantee that, if at least one candidate hypothesis is

$O(\varepsilon)$-close to an unknown target distribution, it will return a hypothesis which is $O(\varepsilon)$-close to the target distribution. The running time of this algorithm is quasilinear in the number of hypotheses. This provides a generic tool for converting from a covering result to a learning result.

Chapter 3 leverages this tool to give an algorithm for properly learning mixtures of two single-dimensional Gaussians. The sample complexity is optimal up to logarithmic factors, and the time complexity improves significantly upon prior work on this problem.

These two chapters appeared as “Faster and Sample Near-Optimal Algorithms for Proper Learning Mixtures of Gaussians” in the Proceedings of The 27th Conference on Learning Theory (COLT 2014) [22] and are based on joint work with Costis Daskalakis.

Chapter 4 applies a recent central limit theorem by Valiant and Valiant to provide a sparse cover for the class of Poisson Multinomial distributions. In particular, our result shows that every Poisson Multinomial distribution is ε-close to the sum of several discretized Gaussians and a sparse Poisson Multinomial distribution. The size of our cover is significantly smaller than the prior best-known cover, and combined with our hypothesis selection tool, implies a sample-efficient learning algorithm for Poisson Multinomial distributions.

This chapter is based on joint work with Costis Daskalakis and Christos Tzamos.

1.2 Preliminaries

We start with some preliminaries which will be relevant to multiple chapters of this thesis.

Let $\mathcal{N}(\mu, \sigma^2)$ represent the univariate Gaussian distribution with mean $\mu \in \mathbb{R}$, variance $\sigma^2 \in \mathbb{R}$, and probability density function

$$\mathcal{N}(\mu, \sigma^2, x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Let $\mathcal{N}(\mu, \Sigma)$ represent the k-variate Gaussian distribution with mean $\mu \in \mathbb{R}^k$, variance $\Sigma \in \mathbb{R}^{k \times k}$, and probability density function

$$\mathcal{N}(\mu, \Sigma, x) = \frac{1}{(2\pi)^k |\Sigma|} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right).$$

The total variation distance between two probability measures P and Q on a σ-algebra F is defined by

$$d_{\mathrm{TV}}(P, Q) = \sup_{A \in F} |P(A) - Q(A)| = \frac{1}{2}\|P - Q\|_1.$$

Definition 1. *Two probability measures P and Q are* ε-close *if $d_{\mathrm{TV}}(P, Q) \leq \varepsilon$.*

Definition 2. *A set of distributions $\mathcal{Q}$ is an* ε-cover *for a set of distributions $\mathcal{P}$ if for each $P \in \mathcal{P}$, there exists a $Q \in \mathcal{Q}$ such that P and Q are ε-close.*

We will use the Data Processing Inequality for Total Variation Distance (see part (iv) of Lemma 2 of [48] for the proof). Our statement of the inequality is taken from [18].

Lemma 1 (Data Processing Inequality for Total Variation Distance)**.** *Let X, X' be two random variables over a domain Ω. Fix any (possibly randomized) function F on Ω (which may be viewed as a distribution over deterministic functions on Ω) and let $F(X)$ be the random variable such that a draw from $F(X)$ is obtained by drawing independently x from X and f from F and then outputting $f(x)$ (likewise for $F(X')$). Then we have*

$$d_{\mathrm{TV}}\left(F(X), F(X')\right) \leq d_{\mathrm{TV}}\left(X, X'\right).$$

1.2.1 Gridding

We will encounter settings where we have bounds L and R on an unknown value X such that $L \leq X \leq R$, and wish to obtain an estimate $\hat{X}$ such that $(1-\varepsilon)X \leq \hat{X} \leq (1+\varepsilon)X$. Gridding is a common technique to generate a list of candidates that is guaranteed to contain such an estimate.

Fact 2. *Candidates of the form $L + k\varepsilon L$ define an additive grid with at most $\frac{1}{\varepsilon}\left(\frac{R-L}{L}\right)$ candidates.*

Fact 3. *Candidates of the form $L(1+\varepsilon)^k$ define a multiplicative grid with at most $\frac{1}{\log(1+\varepsilon)}\log\left(\frac{R}{L}\right)$ candidates.*

We also encounter scenarios where we require an additive estimate $X - \varepsilon \leq \hat{X} \leq X + \varepsilon$.

Fact 4. *Candidates of the form $L + k\varepsilon$ define an absolute additive grid with $\frac{1}{\varepsilon}(R-L)$ candidates.*

Chapter 2

Hypothesis Selection

2.1 Introduction

The goal of this chapter is to present a hypothesis selection algorithm, `FastTournament`, which is given sample access to a target distribution X and several hypotheses distributions $H_1, \ldots, H_N$, together with an accuracy parameter $\varepsilon > 0$, and is supposed to select a hypothesis distribution from $\{H_1, \ldots, H_N\}$. The desired behavior is this: if at least one distribution in $\{H_1, \ldots, H_N\}$ is ε-close to X in total variation distance, we want that the hypothesis distribution selected by the algorithm is $O(\varepsilon)$-close to X. We provide such an algorithm whose sample complexity is $O(\frac{1}{\varepsilon^2} \log N)$ and whose running time $O(\frac{1}{\varepsilon^2} N \log N)$, *i.e. quasi-linear in the number of hypotheses*, improving the running time of the state of the art (predominantly the Scheffé-estimate based algorithm in [31]) quadratically.

We develop our algorithm in full generality, assuming that we have sample access to the distributions of interest, and without making any assumptions about whether they are continuous or discrete, and whether their support is single- or multi-dimensional. All our algorithm needs is sample access to the distributions at hand, together with a way to compare the probability density/mass functions of the distributions, encapsulated in the following definition. In our definition, $H_i(x)$ is the probability mass at x if H_i is a discrete distribution, and the probability density at x if H_i is a continuous distribution. We assume that H_1 and H_2 are either both discrete or both continuous, and that, if they are continuous, they have a density function.

Definition 3. *Let H_1 and H_2 be probability distributions over some set $\mathcal{D}$. A* PDF comparator *for H_1, H_2 is an oracle that takes as input some $x \in \mathcal{D}$ and outputs 1 if $H_1(x) > H_2(x)$, and 0 otherwise.*

Our hypothesis selection algorithm is summarized in the following statement:

Theorem 1. *There is an algorithm* `FastTournament`$(X, \mathcal{H}, \varepsilon, \delta)$*, which is given sample access to some distribution X and a collection of distributions $\mathcal{H} = \{H_1, \ldots, H_N\}$ over some set $\mathcal{D}$, access to a PDF comparator for every pair of distributions $H_i, H_j \in \mathcal{H}$, an accuracy parameter $\varepsilon > 0$, and a confidence parameter $\delta > 0$. The algorithm makes $O\left(\frac{\log 1/\delta}{\varepsilon^2} \cdot \log N\right)$ draws from each of $X, H_1, \ldots, H_N$ and returns some $H \in \mathcal{H}$ or declares "failure." If there is some $H^* \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H^*, X) \leq \varepsilon$ then with probability at least $1 - \delta$ the distribution H that* `FastTournament` *returns satisfies $d_{\mathrm{TV}}(H, X) \leq 512\varepsilon$. The total number of operations of the algorithm is $O\left(\frac{\log 1/\delta}{\varepsilon^2}\left(N \log N + \log^2 \frac{1}{\delta}\right)\right)$. Furthermore, the expected number of operations of the algorithm is $O\left(\frac{N \log N/\delta}{\varepsilon^2}\right)$.*

The proof of Theorem 1 is given in Section 2.4, while the preceding sections build the required machinery for the construction.

Remark 5. *A slight modification of our algorithm provided in Section 2.5 admits a worst-case running time of $O\left(\frac{\log 1/\delta}{\varepsilon^2}\left(N \log N + \log^{1+\gamma} \frac{1}{\delta}\right)\right)$, for any desired constant $\gamma > 0$, though the approximation guarantee is weakened based on the value of γ. See Corollary 14 and its proof in Section 2.5.*

Comparison to Other Hypothesis Selection Methods: The skeleton of the hypothesis selection algorithm of Theorem 1 as well as the improved one of Corollary 14, is having candidate distributions compete against each other in a tournament-like fashion. This approach is quite natural and has been commonly used in the literature; see e.g. Devroye and Lugosi ([29, 30] and Chapter 6 of [31]), Yatracos [55], as well as the recent papers of Daskalakis et al. [20] and Chan et al. [14]. The hypothesis selection algorithms in these works are significantly slower than ours, as their running times have quadratic dependence on the number N of hypotheses, while our dependence is quasi-linear. Furthermore, our setting is more general than prior work, in that we only require sample access to the hypotheses and a PDF comparator. Previous algorithms required knowledge of (or ability to compute) the probability assigned by every pair of hypotheses to their Scheffé set—this is the subset of the support where one hypothesis has larger PMF/PDF than the other, which is difficult to compute in general, even given explicit descriptions of the hypotheses.

Recent independent work by Acharya et al. [1, 2] provides a hypothesis selection algorithm, based on the Scheffé estimate in Chapter 6 of [31]. Their algorithm performs a number of operations that is comparable to ours. In particular, the expected running time of their algorithm is also $O\left(\frac{N \log N/\delta}{\varepsilon^2}\right)$, but our worst-case running time has better dependence on δ. Our algorithm is not based on the Scheffé estimate, using instead a specialized estimator provided in Lemma 6. Their algorithm, described in terms of the Scheffé estimate, is not immediately applicable to sample-only access to the hypotheses, or to settings where the probabilities on Scheffé sets are difficult to compute.

2.2 Choosing Between Two Hypotheses

We start with an algorithm for choosing between two hypothesis distributions. This is an adaptation of a similar algorithm from [20] to continuous distributions and sample-only access.

Lemma 6. *There is an algorithm* `ChooseHypothesis`$(X, H_1, H_2, \varepsilon, \delta)$*, which is given sample access to distributions* X, H_1, H_2 *over some set* $\mathcal{D}$*, access to a PDF comparator for* H_1, H_2*, an accuracy parameter* $\varepsilon > 0$*, and a confidence parameter* $\delta > 0$*. The algorithm draws* $m = O(\log(1/\delta)/\varepsilon^2)$ *samples from each of* X, H_1 *and* H_2*, and either returns some* $H \in \{H_1, H_2\}$ *as the winner or declares a "draw." The total number of operations of the algorithm is* $O(\log(1/\delta)/\varepsilon^2)$*. Additionally, the output satisfies the following properties:*

1. *If $d_{\mathrm{TV}}(X, H_1) \le \varepsilon$ but $d_{\mathrm{TV}}(X, H_2) > 8\varepsilon$, the probability that H_1 is not declared winner is $\le \delta$;*

2. *If $d_{\mathrm{TV}}(X, H_1) \le \varepsilon$ but $d_{\mathrm{TV}}(X, H_2) > 4\varepsilon$, the probability that H_2 is declared winner is $\le \delta$;*

3. *The analogous conclusions hold if we interchange H_1 and H_2 in Properties 1 and 2 above;*

4. *If $d_{\mathrm{TV}}(H_1, H_2) \le 5\varepsilon$, the algorithm declares a "draw" with probability at least $1 - \delta$.*

Proof. We set up a competition between H_1 and H_2, in terms of the following subset of $\mathcal{D}$:

$$\mathcal{W}_1 \equiv \mathcal{W}_1(H_1, H_2) := \{w \in \mathcal{D} \mid H_1(w) > H_2(w)\}.$$

In terms of $\mathcal{W}_1$ we define $p_1 = H_1(\mathcal{W}_1)$ and $p_2 = H_2(\mathcal{W}_1)$. Clearly, $p_1 > p_2$ and $d_{\mathrm{TV}}(H_1, H_2) = p_1 - p_2$. The competition between H_1 and H_2 is carried out as follows:

```
1a. Draw $m = O\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$ samples $s_1, \ldots, s_m$ from $X$, and let
    $\hat{\tau} = \frac{1}{m}|\{i \mid s_i \in \mathcal{W}_1\}|$ be the fraction of them that fall inside $\mathcal{W}_1$.
1b. Similarly, draw $m$ samples from $H_1$, and let $\hat{p}_1$ be the fraction of them
    that fall inside $\mathcal{W}_1$.
1c. Finally, draw $m$ samples from $H_2$, and let $\hat{p}_2$ be the fraction of them
    that fall inside $\mathcal{W}_1$.
 2. If $\hat{p}_1 - \hat{p}_2 \le 6\varepsilon$, declare a draw. Otherwise:
 3. If $\hat{\tau} > \hat{p}_1 - 2\varepsilon$, declare $H_1$ as winner and return $H_1$; otherwise,
 4. if $\hat{\tau} < \hat{p}_2 + 2\varepsilon$, declare $H_2$ as winner and return $H_2$; otherwise,
 5. Declare a draw.
```

Notice that, in Steps 1a, 1b and 1c, the algorithm utilizes the PDF comparator for distributions H_1 and H_2. The correctness of the algorithm is a consequence of the following claim.

Claim 7. *Suppose that $d_{\mathrm{TV}}(X, H_1) \leq \varepsilon$. Then:*

1. *If $d_{\mathrm{TV}}(X, H_2) > 8\varepsilon$, then the probability that the competition between H_1 and H_2 does not declare H_1 as the winner is at most $6e^{-m\varepsilon^2/2}$;*

2. *If $d_{\mathrm{TV}}(X, H_2) > 4\varepsilon$, then the probability that the competition between H_1 and H_2 returns H_2 as the winner is at most $6e^{-m\varepsilon^2/2}$.*

The analogous conclusions hold if we interchange H_1 and H_2 in the above claims. Finally, if $d_{\mathrm{TV}}(H_1, H_2) \leq 5\varepsilon$, the algorithm will declare a draw with probability at least $1 - 6e^{-m\varepsilon^2/2}$.

Proof of Claim 7: Let $\tau = X(\mathcal{W}_1)$. The Chernoff bound (together with a union bound) imply that, with probability at least $1 - 6e^{-m\varepsilon^2/2}$, the following are simultaneously true: $|p_1 - \hat{p}_1| < \varepsilon/2$, $|p_2 - \hat{p}_2| < \varepsilon/2$, and $|\tau - \hat{\tau}| < \varepsilon/2$. Conditioning on these:

- If $d_{\mathrm{TV}}(X, H_1) \leq \varepsilon$ and $d_{\mathrm{TV}}(X, H_2) > 8\varepsilon$, then from the triangle inequality we get that $p_1 - p_2 = d_{\mathrm{TV}}(H_1, H_2) > 7\varepsilon$, hence $\hat{p}_1 - \hat{p}_2 > p_1 - p_2 - \varepsilon > 6\varepsilon$. Hence, the algorithm will go beyond Step 2. Moreover, $d_{\mathrm{TV}}(X, H_1) \leq \varepsilon$ implies that $|\tau - p_1| \leq \varepsilon$, hence $|\hat{\tau} - \hat{p}_1| < 2\varepsilon$. So the algorithm will stop at Step 3, declaring H_1 as the winner of the competition between H_1 and H_2.

- If $d_{\mathrm{TV}}(X, H_2) \leq \varepsilon$ and $d_{\mathrm{TV}}(X, H_1) > 8\varepsilon$, then as in the previous case we get from the triangle inequality that $p_1 - p_2 = d_{\mathrm{TV}}(H_1, H_2) > 7\varepsilon$, hence $\hat{p}_1 - \hat{p}_2 > p_1 - p_2 - \varepsilon > 6\varepsilon$. Hence, the algorithm will go beyond Step 2. Moreover, $d_{\mathrm{TV}}(X, H_2) \leq \varepsilon$ implies that $|\tau - p_2| \leq \varepsilon$, hence $|\hat{\tau} - \hat{p}_2| < 2\varepsilon$. So $\hat{p}_1 > \hat{\tau} + 4\varepsilon$. Hence, the algorithm will not stop at Step 3, and it will stop at Step 4 declaring H_2 as the winner of the competition between H_1 and H_2.

- If $d_{\mathrm{TV}}(X, H_1) \leq \varepsilon$ and $d_{\mathrm{TV}}(X, H_2) > 4\varepsilon$, we distinguish two subcases. If $\hat{p}_1 - \hat{p}_2 \leq 6\varepsilon$, then the algorithm will stop at Step 2 declaring a draw. If $\hat{p}_1 - \hat{p}_2 > 6\varepsilon$, the algorithm proceeds to Step 3. Notice that $d_{\mathrm{TV}}(X, H_1) \leq \varepsilon$ implies that $|\tau - p_1| \leq \varepsilon$, hence $|\hat{\tau} - \hat{p}_1| < 2\varepsilon$. So the algorithm will stop at Step 3, declaring H_1 as the winner of the competition between H_1 and H_2.

- If $d_{\mathrm{TV}}(X, H_2) \leq \varepsilon$ and $d_{\mathrm{TV}}(X, H_1) > 4\varepsilon$, we distinguish two subcases. If $\hat{p}_1 - \hat{p}_2 \leq 6\varepsilon$, then the algorithm will stop at Step 2 declaring a draw. If $\hat{p}_1 - \hat{p}_2 > 6\varepsilon$, the algorithm proceeds to Step 3. Notice that $d_{\mathrm{TV}}(X, H_2) \leq \varepsilon$ implies that $|\tau - p_2| \leq \varepsilon$, hence $|\hat{\tau} - \hat{p}_2| < 2\varepsilon$. Hence, $\hat{p}_1 > \hat{p}_2 + 6\varepsilon \geq \hat{\tau} + 4\varepsilon$, so the algorithm will not stop at Step 3 and will proceed to Step 4. Given that $|\hat{\tau} - \hat{p}_2| < 2\varepsilon$, the algorithm will stop at Step 4, declaring H_2 as the winner of the competition between H_1 and H_2.

- If $d_{\mathrm{TV}}(H_1, H_2) \leq 5\varepsilon$, then $p_1 - p_2 \leq 5\varepsilon$, hence $\hat{p}_1 - \hat{p}_2 \leq 6\varepsilon$. So the algorithm will stop at Step 2 declaring a draw.

□

□

2.3 The Slow Tournament

We proceed with a hypothesis selection algorithm, `SlowTournament`, which has the correct behavior, but whose running time is suboptimal. Again we proceed similarly to [20] making the approach robust to continuous distributions and sample-only access. `SlowTournament` performs pairwise comparisons between all hypotheses in $\mathcal{H}$, using the subroutine `ChooseHypothesis` of Lemma 6, and outputs a hypothesis that never lost to (but potentially tied with) other hypotheses. The running time of the algorithm is quadratic in $|\mathcal{H}|$, as all pairs of hypotheses are compared. `FastTournament`, described in Section 2.4, organizes the tournament in a more efficient manner, improving the running time to quasilinear.

Lemma 8. *There is an algorithm* `SlowTournament`$(X, \mathcal{H}, \varepsilon, \delta)$*, which is given sample access to some distribution* X *and a collection of distributions* $\mathcal{H} = \{H_1, \ldots, H_N\}$ *over* *some set $\mathcal{D}$, access to a PDF comparator for every pair of distributions $H_i, H_j \in \mathcal{H}$, an accuracy parameter $\varepsilon > 0$, and a confidence parameter $\delta > 0$. The algorithm makes $m = O(\log(N/\delta)/\varepsilon^2)$ draws from each of $X, H_1, \ldots, H_N$ and returns some $H \in \mathcal{H}$ or declares "failure." If there is some $H^* \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H^*, X) \leq \varepsilon$ then with probability at least $1 - \delta$ the distribution H that* `SlowTournament` *returns satisfies $d_{\mathrm{TV}}(H, X) \leq 8\varepsilon$. The total number of operations of the algorithm is $O\left(N^2 \log(N/\delta)/\varepsilon^2\right)$.*

Proof. Draw $m = O(\log(2N/\delta)/\varepsilon^2)$ samples from each of $X, H_1, \ldots, H_N$ and, using the same samples, run

$$\texttt{ChooseHypothesis}\left(X, H_i, H_j, \varepsilon, \frac{\delta}{2N}\right),$$

for every pair of distributions $H_i, H_j \in \mathcal{H}$. If there is a distribution $H \in \mathcal{H}$ that was never a loser (but potentially tied with some distributions), output any such distribution. Otherwise, output "failure."

We analyze the correctness of our proposed algorithm in two steps. First, suppose there exists $H^* \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H^*, X) \leq \varepsilon$. We argue that, with probability at least $1 - \frac{\delta}{2}$, H^* never loses a competition against any other $H' \in \mathcal{H}$ (so the tournament does not output "failure"). Consider any $H' \in \mathcal{H}$. If $d_{\mathrm{TV}}(X, H') > 4\varepsilon$, by Lemma 6 the probability that H^* is not declared a winner or tie against H' is at most $\frac{\delta}{2N}$. On the other hand, if $d_{\mathrm{TV}}(X, H') \leq 4\varepsilon$, the triangle inequality gives that $d_{\mathrm{TV}}(H^*, H') \leq 5\varepsilon$ and, by Lemma 6, the probability that H^* does not draw against H' is at most $\frac{\delta}{2N}$. A union bound over all N distributions in $\mathcal{H}$ shows that with probability at least $1 - \frac{\delta}{2}$, the distribution H^* never loses a competition.

We next argue that with probability at least $1 - \frac{\delta}{2}$, every distribution $H \in \mathcal{H}$ that never loses must be 8ε-close to X. Fix a distribution H such that $d_{\mathrm{TV}}(X, H) > 8\varepsilon$. Lemma 6 implies that H loses to H^* with probability at least $1 - \delta/2N$. A union bound gives that with probability at least $1 - \frac{\delta}{2}$, every distribution H such that $d_{\mathrm{TV}}(X, H) > 8\varepsilon$ loses some competition.

Thus, with overall probability at least $1 - \delta$, the tournament does not output "failure" and outputs some distribution H such that $d_{\mathrm{TV}}(H, X) \leq 8\varepsilon$. □

2.4 The Fast Tournament

We prove our main result of this chapter, providing a quasi-linear time algorithm for selecting from a collection of hypothesis distributions $\mathcal{H}$ one that is close to a target distribution X, improving the running time of `SlowTournament` from Lemma 8. Intuitively, there are two cases to consider. Collection $\mathcal{H}$ is either dense or sparse in distributions that are close to X. In the former case, we show that we can subsample $\mathcal{H}$ before running `SlowTournament`. In the latter case, we show how to set-up a two-phase tournament, whose first phase eliminates all but a sub linear number of hypotheses, and whose second phase runs `SlowTournament` on the surviving hypotheses. Depending on the density of $\mathcal{H}$ in distributions that are close to the target distribution X, we show that one of the aforementioned strategies is guaranteed to output a distribution that is close to X. As we do not know a priori the density of $\mathcal{H}$ in distributions that are close to X, and hence which of our two strategies will succeed in finding a distribution that is close to X, we use both strategies and run a tournament among their outputs, using `SlowTournament` again.

Proof of Theorem 1: Let p be the fraction of the elements of $\mathcal{H}$ that are 8ε-close to X. The value of p is unknown to our algorithm. Regardless, we propose two strategies for selecting a distribution from $\mathcal{H}$, one of which is guaranteed to succeed whatever the value of p is. We assume throughout this proof that N is larger than a sufficiently large constant, otherwise our claim follows directly from Lemma 8.

S1: Pick a random subset $\mathcal{H}' \subseteq \mathcal{H}$ of size $\lceil 3\sqrt{N} \rceil$, and run `SlowTournament`$(X, \mathcal{H}', 8\varepsilon, e^{-3})$ to select some distribution $\tilde{H} \in \mathcal{H}'$.

Claim 9. *The number of samples drawn by* S1 *from each of the distributions in* $\mathcal{H} \cup \{X\}$ *is* $O(\frac{1}{\varepsilon^2}\log N)$*, and the total number of operations is* $O(\frac{1}{\varepsilon^2} N \log N)$*. Moreover, if* $p \in [\frac{1}{\sqrt{N}}, 1]$ *and there is some distribution in* $\mathcal{H}$ *that is* ε*-close to* X*, then the distribution* $\tilde{H}$ *output by* S1 *is* 64ε*-close to* X*, with probability at least* $9/10$.

Proof of Claim 9: The probability that $\mathcal{H}'$ contains no distribution that is 8ε-close to X is at most

$$(1-p)^{\lceil 3\sqrt{N}\rceil} \leq e^{-3}.$$

If $\mathcal{H}'$ contains at least one distribution that is 8ε-close to X, then by Lemma 8 the distribution output by $\texttt{SlowTournament}(X, \mathcal{H}', 8\varepsilon, e^{-3})$ is 64ε-close to X with probability at least $1 - e^{-3}$. From a union bound, it follows that the distribution output by S1 is 64ε-close to X, with probability at least $1 - 2e^{-3} \geq 9/10$. The bounds on the number of samples and operations follow from Lemma 8. □

S2: There are two phases in this strategy:

- **Phase 1:** This phase proceeds in $T = \lfloor \log_2 \frac{\sqrt{N}}{2} \rfloor$ iterations, $i_1, \ldots, i_T$. Iteration i_ℓ takes as input a subset $\mathcal{H}_{i_{\ell-1}} \subseteq \mathcal{H}$ (where $\mathcal{H}_{i_0} \equiv \mathcal{H}$), and produces some $\mathcal{H}_{i_\ell} \subset \mathcal{H}_{i_{\ell-1}}$, such that $|\mathcal{H}_{i_\ell}| = \left\lceil \frac{|\mathcal{H}_{i_{\ell-1}}|}{2} \right\rceil$, as follows: randomly pair up the elements of $\mathcal{H}_{i_{\ell-1}}$ (possibly one element is left unpaired), and for every pair (H_i, H_j) run $\texttt{ChooseHypothesis}(X, H_i, H_j, \varepsilon, 1/3N)$. We do this with a small caveat: instead of drawing $O(\log(3N)/\varepsilon^2)$ fresh samples (as required by Lemma 6) in every execution of `ChooseHypothesis` (from whichever distributions are involved in that execution), we draw $O(\log(3N)/\varepsilon^2)$ samples from each of $X, H_1, \ldots, H_N$ once and for all, and reuse the same samples in all executions of `ChooseHypothesis`.

- **Phase 2:** Given the collection $\mathcal{H}_{i_T}$ output by Phase 1, we run $\texttt{SlowTournament}(X, \mathcal{H}_{i_T}, \varepsilon, 1/4)$ to select some distribution $\hat{H} \in \mathcal{H}_{i_T}$. (We use fresh samples for the execution of `SlowTournament`.)

Claim 10. *The number of samples drawn by* S2 *from each of the distributions in* $\mathcal{H} \cup \{X\}$ *is* $O(\frac{1}{\varepsilon^2}\log N)$*, and the total number of operations is* $O(\frac{1}{\varepsilon^2} N \log N)$*. Moreover, if* $p \in (0, \frac{1}{\sqrt{N}}]$ *and there is some distribution in* $\mathcal{H}$ *that is* ε*-close to* X*, then the distribution* $\hat{H}$ *output by* S2 *is* 8ε*-close to* X*, with probability at least* $1/4$.

Proof of Claim 10: Suppose that there is some distribution $H^* \in \mathcal{H}$ that is ε-close to X. We first argue that with probability at least $\frac{1}{3}$, $H^* \in \mathcal{H}_{i_T}$. We show this in two steps:

(a) Recall that we draw samples from $X, H_1, \ldots, H_N$ before Phase 1 begins, and reuse the same samples whenever required by some execution of `ChooseHypothesis` during Phase 1. Fix a realization of these samples. We can ask the question of what would happen if we executed `ChooseHypothesis`$(X, H^*, H_j, \varepsilon, 1/3N)$, for some $H_j \in \mathcal{H} \setminus \{H^*\}$ using these samples. From Lemma 6, it follows that, if H_j is farther than 8ε-away from X, then H^* would be declared the winner by `ChooseHypothesis`$(X, H^*, H_j, \varepsilon, 1/3N)$, with probability at least $1 - 1/3N$. By a union bound, our samples satisfy this property simultaneously for all $H_j \in \mathcal{H} \setminus \{H^*\}$ that are farther than 8ε-away from X, with probability at least $1 - 1/3$. Henceforth, we condition that our samples have this property.

(b) Conditioning on our samples having the property discussed in (a), we argue that $H^* \in \mathcal{H}_{i_T}$ with probability at least $1/2$ (so that, with overall probability at least $1/3$, it holds that $H^* \in \mathcal{H}_{i_T}$). It suffices to argue that, with probability at least $1/2$, in all iterations of Phase 1, H^* is not matched with a distribution that is 8ε-close to X. This happens with probability at least:

$$(1-p)(1-2p)\cdots(1-2^{T-1}p) \geq 2^{-2p\sum_{i=0}^{T-1} 2^i} = 2^{-2p(2^T-1)} \geq 1/2.$$

Indeed, given the definition of p, the probability that H^* is not matched to a distribution that is 8ε-close to X is at least $1 - p$ in the first iteration. If this happens, then (because of our conditioning from (a)), H^* will survive this iteration. In the next iteration, the fraction of surviving distributions that are 8ε-close to X and are different than H^* itself is at most $2p$. Hence, the probability that H^* is not matched to a distribution that is 8ε-close to X is at least $1 - 2p$ in the second iteration, etc.

Now, conditioning on $H^* \in \mathcal{H}_{i_T}$, it follows from Lemma 8 that the distribution $\hat{H}$ output by `SlowTournament`$(X, \mathcal{H}_{i_T}, \varepsilon, 1/4)$ is 8ε-close to X with probability at least $3/4$.

Hence, with overall probability at least $1/4$, the distribution output by S2 is 8ε-close to X.

The number of samples drawn from each distribution in $\mathcal{H} \cup \{X\}$ is clearly $O(\frac{1}{\varepsilon^2} \log N)$, as Phase 1 draws $O(\frac{1}{\varepsilon^2} \log N)$ samples from each distribution and, by Lemma 8, Phase 2 also draws $O(\frac{1}{\varepsilon^2} \log N)$ samples from each distribution.

The total number of operations is bounded by $O(\frac{1}{\varepsilon^2} N \log N)$. Indeed, Phase 1 runs `ChooseHypothesis` $O(N)$ times, and by Lemma 6 and our choice of $1/3N$ for the confidence parameter of each execution, each execution takes $O(\log N/\varepsilon^2)$ operations. So the total number of operations of Phase 1 is $O(\frac{1}{\varepsilon^2} N \log N)$. On the other hand, the size of $\mathcal{H}_{i_T}$ is at most $\frac{2^{\lceil \log_2 N \rceil}}{2^T} = \frac{2^{\lceil \log_2 N \rceil}}{2^{\lfloor \log_2 \frac{\sqrt{N}}{2} \rfloor}} \leq 8\sqrt{N}$. So by Lemma 8, Phase 2 takes $O(\frac{1}{\varepsilon^2} N \log N)$ operations. □

Given strategies S1 and S2, we first design an algorithm which has the stated worst-case number of operations. The algorithm $\texttt{FastTournament}_A$ works as follows:

1. Execute strategy S1 $k_1 = \log_2 \frac{2}{\delta}$ times, with fresh samples each time. Let $\tilde{H}_1, \ldots, \tilde{H}_{k_1}$ be the distributions output by these executions.

2. Execute strategy S2 $k_2 = \log_4 \frac{2}{\delta}$ times, with fresh samples each time. Let $\hat{H}_1, \ldots, \hat{H}_{k_2}$ be the distributions output by these executions.

3. Set $\mathcal{G} \equiv \{\tilde{H}_1, \ldots, \tilde{H}_{k_1}, \hat{H}_1, \ldots, \hat{H}_{k_2}\}$. Execute $\texttt{SlowTournament}(X, \mathcal{G}, 64\varepsilon, \delta/2)$.

Claim 11. $\texttt{FastTournament}_A$ *satisfies the properties described in the statement of Theorem 1, except for the bound on the expected number of operations.*

Proof of Claim 11: The bounds on the number of samples and operations follow immediately from our choice of k_1, k_2, Claims 9 and 10, and Lemma 8. Let us justify the correctness of the algorithm. Suppose that there is some distribution in $\mathcal{H}$ that is ε-close to X. We distinguish two cases, depending on the fraction p of distributions in $\mathcal{H}$ that are ε-close to X:

- $p \in [\frac{1}{\sqrt{N}}, 1]$: In this case, each execution of S1 has probability at least $9/10$ of outputting a distribution that is 64ε-close to X. So the probability that none of

$\tilde{H}_1, \ldots, \tilde{H}_{k_1}$ is 64ε-close to X is at most $(\frac{1}{10})^{k_1} \le \delta/2$. Hence, with probability at least $1-\delta/2$, $\mathcal{G}$ contains a distribution that is 64ε-close to X. Conditioning on this, `SlowTournament`$(X, \mathcal{G}, 64\varepsilon, \delta/2)$ will output a distribution that is 512ε-close to X with probability at least $1-\delta/2$, by Lemma 8. Hence, with overall probability at least $1-\delta$, the distribution output by `FastTournament` is 512ε-close to X.

- $p \in (0, \frac{1}{\sqrt{N}}]$: This case is analyzed analogously. With probability at least $1-\delta/2$, at least one of $\hat{H}_1, \ldots, \hat{H}_{k_2}$ is 8ε-close to X (by Claim 10). Conditioning on this, `SlowTournament`$(X, \mathcal{G}, 64\varepsilon, \delta/2)$ outputs a distribution that is 512ε-close to X, with probability at least $1-\delta/2$ (by Lemma 8). So, with overall probability at least $1-\delta$, the distribution output by `FastTournament` is 512ε-close to X.

□

We now describe an algorithm which has the stated expected number of operations. The algorithm `FastTournament`$_B$ works as follows:

1. Execute strategy S1, let $\tilde{H}_1$ be the distribution output by this execution.
2. Execute strategy S2, let $\tilde{H}_2$ be the distribution output by this execution.
3. Execute `ChooseHypothesis`$(X, \tilde{H}_i, H, 64\varepsilon, \delta/N^3)$ for $i \in \{1,2\}$ and all $H \in \mathcal{H}$. If either $\tilde{H}_1$ or $\tilde{H}_2$ never loses, output that hypothesis. Otherwise, remove $\tilde{H}_1$ and $\tilde{H}_2$ from $\mathcal{H}$, and repeat the algorithm starting from step 1, unless $\mathcal{H}$ is empty.

Claim 12. `FastTournament`$_B$ *satisfies the properties described in the statement of Theorem 1, except for the worst-case bound on the number of operations.*

Proof of Claim 12: We note that we will first draw $O(\log(N^3/\delta)/\varepsilon^2)$ from each of $X, H_1, \ldots, H_N$ and use the same samples for every execution of `ChooseHypothesis` to avoid blowing up the sample complexity. Using this fact, the sample complexity is as claimed.

We now justify the correctness of the algorithm. Since we run `ChooseHypothesis` on a given pair of hypotheses at most once, there are at most N^2 executions of this algorithm. Because each fails with probability at most $\frac{\delta}{N^3}$, by the union bound, the probability that any execution of `ChooseHypothesis` ever fails is at most δ, so all executions succeed with probability at least $1 - \frac{\delta}{N}$. Condition on this happening for the remainder of the proof of correctness. In Step 3 of our algorithm, we compare some $\tilde{H}$ with every other hypothesis. We analyze two cases:

- Suppose that $d_{\mathrm{TV}}(X, \tilde{H}) \leq 64\varepsilon$. By Lemma 6, $\tilde{H}$ will never lose, and will be output by $\texttt{FastTournament}_B$.

- Suppose that $d_{\mathrm{TV}}(X, \tilde{H}) > 512\varepsilon$. Then by Lemma 6, $\tilde{H}$ will lose to any candidate H' with $d_{\mathrm{TV}}(X, H') \leq 64\varepsilon$. We assumed there exists at least one hypothesis with this property in the beginning of the algorithm. Furthermore, by the previous case, if this hypothesis were selected by S1 or S2 at some prior step, the algorithm would have terminated; so in particular, if the algorithm is still running, this hypothesis could not have been removed from $\mathcal{H}$. Therefore, $\tilde{H}$ will lose at least once and will not be output by $\texttt{FastTournament}_B$.

The correctness of our algorithm follows from the second case above. Indeed, if the algorithm outputs a distribution $\tilde{H}$, it must be the case that $d_{\mathrm{TV}}(X, \tilde{H}) \leq 512\varepsilon$. Moreover, we will not run out of hypotheses before we output a distribution. Indeed, we only discard a hypothesis if it was selected by S1 or S2 and then lost at least once in Step 3. Furthermore, in the beginning of our algorithm there exists a distribution H such that $d_{\mathrm{TV}}(X, H) \leq 64\varepsilon$. If ever selected by S1 or S2, H will not lose to any distribution in Step 3, and the algorithm will output a distribution. If it is not selected by S1 or S2, H won't be removed from $\mathcal{H}$.

We now reason about the expected running time of our algorithm. First, consider the case when all executions of `ChooseHypothesis` are successful, which happens with probability $\geq 1 - \frac{\delta}{N}$. If either S1 or S2 outputs a distribution such that $d_{\mathrm{TV}}(X, \tilde{H}) \leq 64\varepsilon$, then by the first case above it will be output by $\texttt{FastTournament}_B$. If this happened with probability at least p independently in every iteration of our algorithm, then the number of iterations would be stochastically dominated by a geometric random variable with parameter p, so the expected number of rounds would be upper bounded by $\frac{1}{p}$. By Claims 9 and 10, $p \geq \frac{1}{4}$, so, when `ChooseHypothesis` never fails, the expected number of rounds is at most 4. Next, consider when at least one execution of `ChooseHypothesis` fails, which happens with probability $\leq \frac{\delta}{N}$. Since $\texttt{FastTournament}_B$ removes at least one hypothesis in every round, there are at most N rounds. Combining these two cases, the expected number of rounds is at most $(1 - \frac{\delta}{N})4 + \frac{\delta}{N}N \leq 5$.

By Claims 9 and 10 and Lemma 6, each round requires $O(N \log N + N \log N/\delta)$ operations. Since the expected number of rounds is $O(1)$, we obtain the desired bound on the expected number of operations. □

In order to obtain all the guarantees of the theorem simultaneously, our algorithm `FastTournament` will alternate between steps of $\texttt{FastTournament}_A$ and $\texttt{FastTournament}_B$, where both algorithms are given an error parameter equal to $\frac{\delta}{2}$. If either algorithm outputs a hypothesis, `FastTournament` outputs it. By union bound and Claims 11 and 12, both $\texttt{FastTournament}_A$ and $\texttt{FastTournament}_B$ will be correct with probability at least $1 - \delta$. The worst-case running time is as desired by Claim 11 and since interleaving between steps of the two tournaments will multiply the number of steps by a factor of at most 2. We have the expected running time similarly, by Claim 12.

□

2.5 Faster Slow and Fast Tournaments

In this section, we describe another hypothesis selection algorithm. This algorithm is faster than `SlowTournament`, though at the cost of a larger constant in the approximation factor. In most reasonable parameter regimes, this algorithm is slower than `FastTournament`, and still has a larger constant in the approximation factor. Regardless, we go on to show how it can be used to improve upon the worst-case running time of `FastTournament`.

Lemma 13. *For every constant $\gamma > 0$, there exists an algorithm* $\texttt{RecursiveSlowTournament}_\gamma(X, \mathcal{H}, \varepsilon, \delta)$*, which is given sample access to some distribution X and a collection of distributions $\mathcal{H} = \{H_1, \ldots, H_N\}$ over some set $\mathcal{D}$, access to a PDF comparator for every pair of distributions $H_i, H_j \in \mathcal{H}$, an accuracy parameter $\varepsilon > 0$, and a confidence parameter $\delta > 0$. The algorithm makes $m = O(\log(N/\delta)/\varepsilon^2)$ draws from each of $X, H_1, \ldots, H_N$ and returns some $H \in \mathcal{H}$ or declares "failure." If there is some $H^* \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H^*, X) \leq \varepsilon$ then with probability at least $1 - \delta$ the distribution H that* `RecursiveSlowTournament` *returns satisfies $d_{\mathrm{TV}}(H, X) \leq O(\varepsilon)$. The total number of operations of the algorithm is $O\left(N^{1+\gamma} \log(N/\delta)/\varepsilon^2\right)$.*

Proof. For simplicity, assume that $\sqrt{N}$ is an integer. (If not, introduce into $\mathcal{H}$ multiple copies of an arbitrary $H \in \mathcal{H}$ so that $\sqrt{N}$ becomes an integer.) Partition $\mathcal{H}$ into $\sqrt{N}$ subsets, $\mathcal{H} = \mathcal{H}_1 \sqcup \mathcal{H}_2 \sqcup \ldots \sqcup \mathcal{H}_{\sqrt{N}}$ and do the following:

1. Set $\delta' = \delta/2$, draw $O(\log(\sqrt{N}/\delta')/\varepsilon^2)$ samples from X and, using the same samples, run $\texttt{SlowTournament}(X, \mathcal{H}_i, \varepsilon, \delta')$ from Lemma 8 for each i;

2. Run $\texttt{SlowTournament}(X, \mathcal{W}, 8\varepsilon, \delta')$, where $\mathcal{W}$ are the distributions output by `SlowTournament` in the previous step. If $\mathcal{W} = \emptyset$ output "failure".

Let us call the above algorithm $\texttt{SlowTournament}^{\bigotimes 1}(X, \mathcal{H}, \varepsilon, \delta)$, before proceeding to analyze its correctness, sample and time complexity. Suppose there exists a distribution $H \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H, X) \leq \varepsilon$. Without loss of generality, assume that $H \in \mathcal{H}_1$. Then, from Lemma 8, with probability at least $1-\delta'$, $\texttt{SlowTournament}(X, \mathcal{H}_1, \varepsilon, \delta')$ will output a distribution H' such that $d_{\mathrm{TV}}(H', X) \leq 8\varepsilon$. Conditioning on this and applying Lemma 8 again, with conditional probability at least $1 - \delta'$ $\texttt{SlowTournament}(X, \mathcal{W}, 8\varepsilon, \delta')$ will output a distribution H'' such that $d_{\mathrm{TV}}(H'', X) \leq 64\varepsilon$. So with overall probability at least $1-\delta$, $\texttt{SlowTournament}^{\bigotimes 1}(X, \mathcal{H}, \varepsilon, \delta)$ will output a distribution that is 64ε-close to X. The number of samples that the algorithm draws from X is $O(\log(N/\delta)/\varepsilon^2)$, and the running time is

$$\sqrt{N} \times O\left(N \log(N/\delta')/\varepsilon^2\right) + O\left(N \log(N/\delta')/(8\varepsilon)^2\right) = O\left(N^{3/2} \log(N/\delta)/\varepsilon^2\right).$$

So, compared to `SlowTournament`, $\texttt{SlowTournament}^{\otimes 1}$ has the same sample complexity asymptotics and the same asymptotic guarantee for the distance from X of the output distribution, but the exponent of N in the running time improved from 2 to 3/2.

For $t = 2, 3, \ldots$, define $\texttt{SlowTournament}^{\otimes t}$ by replacing `SlowTournament` by $\texttt{SlowTournament}^{\otimes t-1}$ in the code of $\texttt{SlowTournament}^{\otimes 1}$. It follows from the same analysis as above that as t increases the exponent of N in the running time gets arbitrarily close to 1. In particular, in one step an exponent of $1+\alpha$ becomes an exponent of $1+\alpha/2$. So for some constant t, $\texttt{SlowTournament}^{\otimes t}$ will satisfy the requirements of the theorem. □

As a corollary, we can immediately improve the running time of `FastTournament` at the cost of the constant in the approximation factor. The construction and analysis is nearly identical to that of `FastTournament`. The sole difference is in step 3 of $\texttt{FastTournament}_A$ - we replace `SlowTournament` with $\texttt{RecursiveSlowTournament}_\gamma$.

Corollary 14. *For any constant $\gamma > 0$, there is an algorithm $\texttt{FastTournament}_\gamma(X, \mathcal{H}, \varepsilon, \delta)$, which is given sample access to some distribution X and a collection of distributions $\mathcal{H} = \{H_1, \ldots, H_N\}$ over some set $\mathcal{D}$, access to a PDF comparator for every pair of distributions $H_i, H_j \in \mathcal{H}$, an accuracy parameter $\varepsilon > 0$, and a confidence parameter $\delta > 0$. The algorithm makes $O\left(\frac{\log 1/\delta}{\varepsilon^2} \cdot \log N\right)$ draws from each of $X, H_1, \ldots, H_N$ and returns some $H \in \mathcal{H}$ or declares "failure." If there is some $H^* \in \mathcal{H}$ such that $d_{\mathrm{TV}}(H^*, X) \le \varepsilon$ then with probability at least $1-\delta$ the distribution H that* `SlowTournament` *returns satisfies $d_{\mathrm{TV}}(H, X) \le O(\varepsilon)$. The total number of operations of the algorithm is $O\left(\frac{\log 1/\delta}{\varepsilon^2}(N \log N + \log^{1+\gamma}\frac{1}{\delta})\right)$. Furthermore, the expected number of operations of the algorithm is $O\left(\frac{N \log N/\delta}{\varepsilon^2}\right)$.*

Chapter 3

Proper Learning Gaussian Mixture Models

3.1 Introduction

Learning mixtures of Gaussian distributions is one of the most fundamental problems in Statistics, with a multitude of applications in the natural and social sciences, which has recently received considerable attention in Computer Science literature. Given independent samples from an unknown mixture of Gaussians, the task is to 'learn' the underlying mixture.

In one version of the problem, 'learning' means estimating the *parameters* of the mixture, that is the mixing probabilities as well as the parameters of each constituent Gaussian. The most popular heuristic for doing so is running the EM algorithm on samples from the mixture [28], albeit no rigorous guarantees are known for it in general.

A line of research initiated by Dasgupta [16, 5, 54, 3, 10] provides rigorous guarantees under separability conditions: roughly speaking, it is assumed that the constituent Gaussians have variation distance bounded away from 0 (indeed, in some cases, distance exponentially close to 1). This line of work was recently settled by a triplet of breakthrough results [40, 45, 7], establishing the polynomial solvability of the problem under minimal separability conditions for the parameters to be recoverable in the first place: for any $\varepsilon > 0$, polynomial in n and $1/\varepsilon$ samples from a mixture of n-dimensional Gaussians suffice to recover the parameters of the mixture in $\mathrm{poly}(n, 1/\varepsilon)$ time.

While these results settle the polynomial solvability of the problem, they serve more as a proof of concept in that their dependence on $1/\varepsilon$ is quite expensive.[1] Very recently, [38] presented a new algorithm (and a matching lower bound) for estimating the parameters of a mixture of 2 Gaussians. This algorithm has a much milder dependence on $1/\varepsilon$, though their definition of parameter estimation is slightly different from the usual one.

A weaker goal for the learner of Gaussian mixtures is this: given samples from an unknown mixture, find *any* mixture that is close to the unknown one, for some notion of closeness. This PAC-style version of the problem [41] was pursued by Feldman et al. [33] who obtained efficient learning algorithms for mixtures of n-dimensional, axis-aligned Gaussians. Given $\mathrm{poly}(n, 1/\varepsilon, L)$ samples from such mixture, their algorithm constructs a mixture whose KL divergence to the sampled one is at most ε. Unfortunately, the sample and time complexity of their algorithm depends polynomially on a (priorly known) bound L, determining the range of the means and variances of the constituent Gaussians in every dimension.[2] In particular, the algorithm has pseudo-polynomial dependence on L where there should not be any dependence on L at all [40, 45, 7].

Finally, yet a weaker goal for the learner would be to construct *any distribution* that is close to the unknown mixture. In this *non-proper* version of the problem the learner is not restricted to output a Gaussian mixture, but can output any (representation of a) distribution that is close to the unknown mixture. For this problem, recent results of Chan et al. [15] provide algorithms for single-dimensional mixtures, whose sample complexity has near-optimal dependence on $1/\varepsilon$. Namely, given $\tilde{O}(1/\varepsilon^2)$

[1] For example, the single-dimensional algorithm in the heart of [40] has sample and time complexity of $\Theta(1/\varepsilon^{300})$ and $\Omega(1/\varepsilon^{1377})$ respectively (even though the authors most certainly did not intend to optimize their constants).

[2] In particular, it is assumed that every constituent Gaussian in every dimension has mean $\mu \in [-\mu_{\max}, \mu_{\max}]$ and variance $\sigma^2 \in [\sigma^2_{\min}, \sigma^2_{\max}]$ where $\mu_{\max}\sigma_{\max}/\sigma_{\min} \leq L$.

samples from a single-dimensional mixture, they construct a piecewise polynomial distribution that is ε-close in total variation distance.

Inspired by this recent progress on non-properly learning single-dimensional mixtures, our goal in this paper is to provide sample-optimal algorithms that *properly learn*. We obtain such algorithms for mixtures of two single-dimensional Gaussians. Namely,

Theorem 2. *For all $\varepsilon, \delta > 0$, given $\tilde{O}(\log(1/\delta)/\varepsilon^2)$ independent samples from an arbitrary mixture F of two univariate Gaussians we can compute in time $\tilde{O}(\log^3(1/\delta)/\varepsilon^5)$ a mixture F' such that $d_{\mathrm{TV}}(F, F') \leq \varepsilon$ with probability at least $1 - \delta$. The expected running time of this algorithm is $\tilde{O}(\log^2(1/\delta)/\varepsilon^5)$.*

We note that learning a univariate mixture often lies at the heart of learning multivariate mixtures [40, 45], so it is important to understand this fundamental case.

Discussion. Note that our algorithm makes no separability assumptions about the constituent Gaussians of the unknown mixture, nor does it require or depend on a bound on the mixture's parameters. Also, because the mixture is single-dimensional it is not amenable to the techniques of [39]. Moreover, it is easy to see that our sample complexity is optimal up to logarithmic factors. Indeed, a Gaussian mixture can trivially simulate a Bernoulli distribution as follows. Let Z be a Bernoulli random variable that is 0 with probability $1 - p$ and 1 with probability p. Clearly, Z can be viewed as a mixture of two Gaussian random variables of 0 variance, which have means 0 and 1 and are mixed with probabilities $1 - p$ and p respectively. It is known that $1/\varepsilon^2$ samples are needed to properly learn a Bernoulli distribution, hence this lower bound immediately carries over to Gaussian mixtures.

Approach. Our algorithm is intuitively quite simple, although some care is required to make the ideas work. First, we can guess the mixing weight up to additive error $O(\varepsilon)$, and proceed with our algorithm pretending that our guess is correct. Every guess will result in a collection of candidate distributions, and the final step of our algorithm is a tournament that will select, from among all candidate distributions produced in the course of our algorithm, a distribution that is ε-close to the unknown mixture, if such a distribution exists. To do this we will make use of the hypothesis selection algorithm described in the previous chapter. It is also worth noting that the tournament based approach of [33] cannot be used for our purposes in this paper as it would require a priorly known bound on the mixture's parameters and would depend pseudopolynomially on this bound.

Tuning the number of samples according to the guessed mixing weight, we proceed to draw samples from the unknown mixture, expecting that some of these samples will fall sufficiently close to the means of the constituent Gaussians, where the closeness will depend on the number of samples drawn as well as the unknown variances. We guess which sample falls close to the mean of the constituent Gaussian that has the smaller value of σ/w (standard deviation to mixing weight ratio), which gives us the second parameter of the mixture. To pin down the variance of this Gaussian, we implement a natural idea. Intuitively, if we draw samples from the mixture, we expect that the constituent Gaussian with the smallest σ/w will determine the smallest distance among the samples. Pursuing this idea we produce a collection of variance candidates, one of which truly corresponds to the variance of this Gaussian, giving us a third parameter.

At this point, we have a complete description of one of the component Gaussians. If we could remove this component from the mixture, we would be left with the remaining unknown Gaussian. Our approach is to generate an empirical distribution of the mixture and "subtract out" the component that we already know, giving us an approximation to the unknown Gaussian. For the purposes of estimating the two parameters of this unknown Gaussian, we observe that the most traditional estimates of location and scale are unreliable, since the error in our approximation may cause probability mass to be shifted to arbitrary locations. Instead, we use robust statistics to obtain approximations to these two parameters.

The empirical distribution of the mixture is generated using the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [32]. With $O(1/\varepsilon^2)$ samples from an *arbitrary* distribution, this algorithm generates an ε-approximation to the distribution (with respect to the Kolmogorov metric). Since this result applies to arbitrary distributions, it generates a hypothesis that is weak, in some senses, including the choice of distance metric. In particular, the hypothesis distribution output by the DKW inequality is discrete, resulting in a total variation distance of 1 from a mixture of Gaussians (or any other continuous distribution), regardless of the accuracy parameter ε. Thus, we consider it to be interesting that such a weak hypothesis can be used as a tool to generate a stronger, proper hypothesis. We note that the Kolmogorov distance metric is not special here - an approximation with respect to other reasonable distance metrics may be substituted in, as long as the description of the hypothesis is efficiently manipulable in the appropriate ways.

We show that, for any target total variation distance ε, the number of samples required to execute the steps outlined above in order to produce a collection of candidate hypotheses one of which is ε-close to the unknown mixture, as well as to run the tournament to select from among the candidate distributions are $\tilde{O}(1/\varepsilon^2)$. The running time is $\tilde{O}(1/\varepsilon^5)$.

Comparison to Prior Work on Learning Gaussian Mixtures. In comparison to the recent breakthrough results [40, 45, 7], our algorithm has near-optimal sample complexity and much milder running time, where these results have quite expensive dependence of both their sample and time complexity on the accuracy ε, even for single-dimensional mixtures.[3] On the other hand, our algorithm has weaker guarantees in that we properly learn but do not do parameter estimation. In some sense, our result is incomparable to [38]. Their algorithm performs parameter learning on mixtures of two Gaussians at the cost of a higher time and sample complexity than our algorithm. Unlike the results mentioned before, the exponents in their algorithm's complexity are reasonably small constants. However, while in most settings parameter learning implies proper learning, [38] uses a slightly different definition of parameter learning which does not. In comparison to [33], our algorithm requires no bounds

[3] For example, compared to [40] we improve by a factor of at least 150 the exponent of both the sample and time complexity.

on the parameters of the constituent Gaussians and exhibits no pseudo-polynomial dependence of the sample and time complexity on such bounds. On the other hand, we learn with respect to the total variation distance rather than the KL divergence. Finally, compared to [14, 15], we properly learn while they non-properly learn and we both have near-optimal sample complexity.

Recently and independently, Acharya et al. [1] have also provided algorithms for properly learning spherical Gaussian mixtures. Their primary focus is on the high dimensional case, aiming at a near-linear sample dependence on the dimension. Our focus is instead on optimizing the dependence of the sample and time complexity on ε in the one-dimensional case.

In fact, [1] also study mixtures of k Gaussians in one dimension, providing a proper learning algorithm with near-optimal sample complexity of $\tilde{O}\left(k/\varepsilon^2\right)$ and running time $\tilde{O}_k\left(1/\varepsilon^{3k+1}\right)$. Specializing to two single-dimensional Gaussians ($k = 2$), their algorithm has near-optimal sample complexity, like ours, but is slower by a factor of $O(1/\varepsilon^2)$ than ours. We also remark that, through a combination of techniques from our paper and theirs, a proper learning algorithm for mixtures of k Gaussians can be obtained, with near-optimal sample complexity of $\tilde{O}\left(k/\varepsilon^2\right)$ and running time $\tilde{O}_k\left(1/\varepsilon^{3k-1}\right)$, improving by a factor of $O(1/\varepsilon^2)$ the running time of their k-Gaussian algorithm. Roughly, this algorithm creates candidate distributions in which the parameters of the first $k-1$ components are generated using methods from [1], and the parameters of the final component are determined using our robust statistical techniques, in which we "subtract out" the first $k-1$ components and robustly estimate the mean and variance of the remainder.

3.2 Preliminaries

The univariate half-normal distribution with parameter σ^2 is the distribution of $|Y|$ where Y is distributed according to $\mathcal{N}(0, \sigma^2)$. The CDF of the half-normal distribution is

$$F(\sigma, x) = \operatorname{erf}\left(\frac{x}{\sigma\sqrt{2}}\right),$$

where $\mathrm{erf}(x)$ is the error function, defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,\mathrm{d}t.$$

We also make use of the complement of the error function, $\mathrm{erfc}(x)$, defined as $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$.

A Gaussian mixture model (GMM) of distributions $\mathcal{N}_1(\mu_1, \sigma_1^2), \ldots, \mathcal{N}_n(\mu_n, \sigma_n^2)$ has PDF

$$f(x) = \sum_{i=1}^{n} w_i \mathcal{N}(\mu_i, \sigma_i^2, x),$$

where $\sum_i w_i = 1$. These w_i are referred to as the mixing weights. Drawing a sample from a GMM can be visualized as the following process: select a single Gaussian, where the probability of selecting a Gaussian is equal to its mixing weight, and draw a sample from that Gaussian. In this paper, we consider mixtures of two Gaussians, so $w_2 = 1 - w_1$. We will interchangeably use w and $1 - w$ in place of w_1 and w_2.

For simplicity in the exposition of our algorithm, we make the standard assumption (see, e.g., [33, 40]) of infinite precision real arithmetic. In particular, the samples we draw from a mixture of Gaussians are real numbers, and we can do exact computations on real numbers, e.g., we can exactly evaluate the PDF of a Gaussian distribution on a real number.

3.2.1 Bounds on Total Variation Distance for GMMs

We recall a result from [18]:

Proposition 15 (Proposition B.4 of [18])**.** *Let $\mu_1, \mu_2 \in \mathbb{R}$ and $0 \le \sigma_1 \le \sigma_2$. Then*

$$d_{\mathrm{TV}}(\mathcal{N}(\mu_1, \sigma_1^2), \mathcal{N}(\mu_2, \sigma_2^2)) \le \frac{1}{2}\left(\frac{|\mu_1 - \mu_2|}{\sigma_1} + \frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2}\right).$$

The following proposition provides a bound on the total variation distance between two GMMs in terms of the distance between the constituent Gaussians.

Proposition 16. *Suppose we have two GMMs X and Y, with PDFs $w\mathcal{N}_1 + (1-w)\mathcal{N}_2$*

and $\hat{w}\hat{\mathcal{N}_1} + (1-\hat{w})\hat{\mathcal{N}_2}$ *respectively. Then* $d_{\mathrm{TV}}(X,Y) \leq |w-\hat{w}| + w d_{\mathrm{TV}}(\mathcal{N}_1, \hat{\mathcal{N}_1}) + (1-w)d_{\mathrm{TV}}(\mathcal{N}_2, \hat{\mathcal{N}_2})$.

Proof. We use $d_{\mathrm{TV}}(P,Q)$ and $\frac{1}{2}\|P-Q\|_1$ interchangeably in the cases where P and Q are not necessarily probability distributions. Let $\mathcal{N}_i = \mathcal{N}(\mu_i, \sigma_i^2)$ and $\hat{\mathcal{N}_i} = \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$. By triangle inequality,

$$d_{\mathrm{TV}}(\hat{w}\hat{\mathcal{N}_1}+(1-\hat{w})\hat{\mathcal{N}_2}, w\mathcal{N}_1+(1-w)\mathcal{N}_2) \leq d_{\mathrm{TV}}(\hat{w}\hat{\mathcal{N}_1}, w\mathcal{N}_1)+d_{\mathrm{TV}}((1-\hat{w})\hat{\mathcal{N}_2}, (1-w)\mathcal{N}_2)$$

Inspecting the first term,

$$\frac{1}{2}\left\|w\mathcal{N}_1 - \hat{w}\hat{\mathcal{N}_1}\right\|_1 = \frac{1}{2}\left\|w\mathcal{N}_1 - w\hat{\mathcal{N}_1} + w\hat{\mathcal{N}_1} - \hat{w}\hat{\mathcal{N}_1}\right\|_1 \leq w d_{\mathrm{TV}}(\mathcal{N}_1, \hat{\mathcal{N}_1}) + \frac{1}{2}|w-\hat{w}|,$$

again using the triangle inequality. A symmetric statement holds for the other term, giving us the desired result. □

Combining these propositions, we obtain the following lemma:

Lemma 17. *Let* X *and* Y *by two GMMs with PDFs* $w_1\mathcal{N}_1 + w_2\mathcal{N}_2$ *and* $\hat{w}_1\hat{\mathcal{N}_1} + \hat{w}_2\hat{\mathcal{N}_2}$ *respectively, where* $|w_i - \hat{w}_i| \leq O(\varepsilon)$, $|\mu_i - \hat{\mu}_i| \leq O(\frac{\varepsilon}{w_i})\sigma_i \leq O(\varepsilon)\sigma_i$, $|\sigma_i - \hat{\sigma}_i| \leq O(\frac{\varepsilon}{w_i})\sigma_i \leq O(\varepsilon)\sigma_i$, *for all* i *such that* $w_i \geq \frac{\varepsilon}{25}$. *Then* $d_{\mathrm{TV}}(X,Y) \leq \varepsilon$.

3.2.2 Kolmogorov Distance

In addition to total variation distance, we will also use the Kolmogorov distance metric.

Definition 4. *The* Kolmogorov distance *between two probability measures with CDFs* F_X *and* F_Y *is* $d_{\mathrm{K}}(F_X, F_Y) = \sup_{x\in\mathbb{R}} |F_X(x) - F_Y(x)|$.

We will also use this metric to compare general functions, which may not necessarily be valid CDFs.

We have the following fact, stating that total variation distance upper bounds Kolmogorov distance [37].

Fact 18. $d_{\mathrm{K}}(F_X, F_Y) \leq d_{\mathrm{TV}}(f_X, f_Y)$

Fortunately, it is fairly easy to learn with respect to the Kolmogorov distance, due to the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [32].

Theorem 3. *([32],[44]) Suppose we have n IID samples $X_1, \ldots X_n$ from a probability distribution with CDF F. Let $F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{X_i \leq x\}}$ be the empirical CDF. Then $\Pr[d_{\mathrm{K}}(F, F_n) \geq \varepsilon] \leq 2e^{-2n\varepsilon^2}$. In particular, if $n = \Omega((1/\varepsilon^2) \cdot \log(1/\delta))$, then $\Pr[d_{\mathrm{K}}(F, F_n) \geq \varepsilon] \leq \delta$.*

3.2.3 Representing and Manipulating CDFs

We will need to be able to efficiently represent and query the CDF of probability distributions we construct. This will be done using a data structure we denote the *n-interval partition* representation of a distribution. This allows us to represent a discrete random variable X over a support of size $\leq n$. Construction takes $\tilde{O}(n)$ time, and at the cost of $O(\log n)$ time per operation, we can compute $F_X^{-1}(x)$ for $x \in [0, 1]$.

Using this data structure and Theorem 3, we can derive the following proposition:

Proposition 19. *Suppose we have $n = \Theta(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ IID samples from a random variable X. In $\tilde{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right)$ time, we can construct a data structure which will allow us to convert independent samples from the uniform distribution over $[0, 1]$ to independent samples from a random variable $\hat{X}$, such that $d_{\mathrm{K}}\left(F_X, F_{\hat{X}}\right) \leq \varepsilon$ with probability $1 - \delta$.*

We provide more details on the construction of this data structure. An n-interval partition is a set of disjoint intervals which form a partition of $[0, 1]$, each associated with an element of the support. A value $x \in [0, 1]$ is mapped to the support element associated to the interval which contains x. This data structure is constructed by mapping each support element to an interval of width equal to the probability of that element. This data structure is queried in $O(\log n)$ time by performing binary search on the left endpoints of the intervals. To avoid confusion with intervals that represent elements of the σ-algebra of the distribution, we refer to the intervals that are stored in the data structure as *probability intervals*.

We note that, if we are only concerned with sampling, the order of the elements of the support is irrelevant. However, we will sort the elements of the support in order to perform efficient modifications later.

At one point in our learning algorithm, we will have a candidate which correctly describes one of the two components in our mixture of Gaussians. If we could "subtract out" this component from the mixture, we would be left with a single Gaussian – in this setting, we can efficiently perform parameter estimation to learn the other component. However, if we naively subtract the probability densities, we will obtain negative probability densities, or equivalently, non-monotonically increasing CDFs. To deal with this issue, we define a process we call monotonization. Intuitively, this will shift negative probability density to locations with positive probability density. We show that this preserves Kolmogorov distance and that it can be implemented efficiently.

Definition 5. *Given a bounded function $f : \mathbb{R} \to \mathbb{R}$, the* monotonization *of f is $\hat{f}$, where $\hat{f}(x) = \sup_{y \leq x} f(x)$.*

We argue that if a function is close in Kolmogorov distance to a monotone function, then so is its monotonization.

Proposition 20. *Suppose we have two bounded functions F and G such that $d_{\mathrm{K}}(F, G) \leq \varepsilon$, where F is monotone non-decreasing. Then $\hat{G}$, the monotonization of G, is such that $d_{\mathrm{K}}(F, \hat{G}) \leq \varepsilon$.*

Proof. We show that $|F(x) - \hat{G}(x)| \leq \varepsilon$ holds for an arbitrary point x, implying that $d_{\mathrm{K}}(F, \hat{G}) \leq \varepsilon$. There are two cases: $F(x) \geq \hat{G}(x)$ and $F(x) < \hat{G}(x)$.

If $F(x) \geq \hat{G}(x)$, using the fact that $\hat{G}(x) \geq G(x)$ (due to monotonization), we can deduce $|F(x) - \hat{G}(x)| \leq |F(x) - G(x)| \leq \varepsilon$.

If $F(x) < \hat{G}(x)$, consider an infinite sequence of points $\{y_i\}$ such that $G(y_i)$ becomes arbitrarily close to $\sup_{y \leq x} G(x)$. By monotonicity of F, we have that $|F(x) - \hat{G}(x)| \leq |F(y_i) - G(y_i)| + \delta_i \leq \varepsilon + \delta_i$, where $\delta_i = |\hat{G}(x) - G(y_i)|$. Since δ_i can be taken arbitrarily small, we have $|F(x) - \hat{G}(x)| \leq \varepsilon$. □

We will need to efficiently compute the monotonization in certain settings, when subtracting one monotone function from another.

Proposition 21. *Suppose we have access to the n-interval partition representation of a CDF F. Given a monotone non-decreasing function G, we can compute the n-interval partition representation of the monotonization of $F - G$ in $O(n)$ time.*

Proof. Consider the values in the n-interval partition of F. Between any two consecutive values v_1 and v_2, F will be flat, and since G is monotone non-decreasing, $F - G$ will be monotone non-increasing. Therefore, the monotonization of $F - G$ at $x \in [v_1, v_2)$ will be the maximum of $F - G$ on $(-\infty, v_1]$. The resulting monotonization will also be flat on the same intervals as F, so we will only need to update the probability intervals to reflect this monotonization.

We will iterate over probability intervals in increasing order of their values, and describe how to update each interval. We will need to keep track of the maximum value of $F - G$ seen so far. Let m be the maximum of $F - G$ for all $x \leq v$, where v is the value associated with the last probability interval we have processed. Initially, we have the value $m = 0$. Suppose we are inspecting a probability interval with endpoints $[l, r]$ and value v. The left endpoint of this probability interval will become $\hat{l} = m$, and the right endpoint will become $\hat{r} = r - G(v)$. If $\hat{r} \leq \hat{l}$, the interval is degenerate, meaning that the monotonization will flatten out the discontinuity at v - therefore, we simply delete the interval. Otherwise, we have a proper probability interval, and we update $m = \hat{r}$.

This update takes constant time per interval, so the overall time required is $O(n)$.

□

We provide a lemma that we will apply to "subtract out" one of the components.

Lemma 22. *Suppose we have access to the n-interval partition representation of a CDF F, and that there exists a weight w and CDFs G and H such that $d_{\mathrm{K}}\left(H, \frac{F - wG}{1 - w}\right) \leq \varepsilon$. Given w and G, we can compute the n-interval partition representation of a distribution $\hat{H}$ such that $d_{\mathrm{K}}(H, \hat{H}) \leq \varepsilon$ in $O(n)$ time.*

Proof. First, by assumption, we know that $\frac{1}{1-w} d_{\mathrm{K}}((1-w)H, F - wG) \leq \varepsilon$. By Proposition 21, we can efficiently compute the monotonization of $F - wG$ - name this $(1-w)\hat{H}$. By Proposition 20, we have that $\frac{1}{1-w} d_{\mathrm{K}}((1-w)H, (1-w)\hat{H}) \leq \varepsilon$. Renormalizing the distributions gives the desired approximation guarantee.

To justify the running time of this procedure, we must also argue that the normalization can be done efficiently. To normalize the distribution $(1-w)\hat{H}$, we make another $O(n)$ pass over the probability intervals and multiply all the endpoints by $\frac{1}{r^*}$, where r^* is the right endpoint of the rightmost probability interval. We note that r^* will be exactly $1-w$ because the value of $F - wG$ at ∞ is $1-w$, so this process results in the distribution $\hat{H}$. □

3.2.4 Robust Statistics

We use two well known robust statistics, the median and the interquartile range. These are suited to our application for two purposes. First, they are easy to compute with the n-interval partition representation of a distribution. Each requires a constant number of queries of the CDF at particular values, and the cost of each query is $O(\log n)$. Second, they are robust to small modifications with respect to most metrics on probability distributions. In particular, we will demonstrate their robustness on Gaussians when considering distance with respect to the Kolmogorov metric.

Lemma 23. *Let $\hat{F}$ be a distribution such that $d_{\mathrm{K}}(\mathcal{N}(\mu, \sigma^2), \hat{F}) \leq \varepsilon$, where $\varepsilon < \frac{1}{8}$. Then $med(\hat{F}) \triangleq \hat{F}^{-1}(\frac{1}{2}) \in [\mu - 2\sqrt{2}\varepsilon\sigma, \mu + 2\sqrt{2}\varepsilon\sigma]$.*

Proof. We will use x to denote the median of our distribution, where $\hat{F}(x) = \frac{1}{2}$. Since $d_{\mathrm{K}}(F, \hat{F}) \leq \varepsilon$, $F(x) \leq \frac{1}{2} + \varepsilon$. Using the CDF of the normal distribution, we obtain $\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right) \leq \frac{1}{2} + \varepsilon$. Rearranging, we get $x \leq \mu + \sqrt{2}\mathrm{erf}^{-1}(2\varepsilon)\sigma \leq \mu + 2\sqrt{2}\varepsilon\sigma$, where the inequality uses the Taylor series of erf^{-1} around 0 and Taylor's Theorem. By symmetry of the Gaussian distribution, we can obtain the corresponding lower bound for x. □

Lemma 24. *Let $\hat{F}$ be a distribution such that $d_{\mathrm{K}}(\mathcal{N}(\mu, \sigma^2), \hat{F}) \leq \varepsilon$, where $\varepsilon < \frac{1}{8}$. Then $\frac{IQR(\hat{F})}{2\sqrt{2}erf^{-1}(\frac{1}{2})} \triangleq \frac{\hat{F}^{-1}(\frac{3}{4}) - \hat{F}^{-1}(\frac{1}{4})}{2\sqrt{2}erf^{-1}(\frac{1}{2})} \in \left[\sigma - \frac{5}{2erf^{-1}(\frac{1}{2})}\varepsilon\sigma, \sigma + \frac{7}{2erf^{-1}(\frac{1}{2})}\varepsilon\sigma\right]$.*

Proof. First, we show that

$$F^{-1}\left(\frac{3}{4}\right) \in \left[\mu + \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2}\right)\sigma - \frac{5\sqrt{2}}{2}\sigma\varepsilon, \mu + \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2}\right)\sigma + \frac{7\sqrt{2}}{2}\sigma\varepsilon\right].$$

Let $x = F^{-1}\left(\frac{3}{4}\right)$. Since $d_{\mathrm{K}}(F, \hat{F}) \leq \varepsilon$, $F(x) \leq \frac{3}{4} + \varepsilon$. Using the CDF of the normal distribution, we obtain $\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right) \leq \frac{3}{4} + \varepsilon$. Rearranging, we get $x \leq \mu + \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2} + 2\varepsilon\right)\sigma \leq \mu + \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2}\right)\sigma + \frac{7\sqrt{2}}{2}\varepsilon\sigma$, where the inequality uses the Taylor series of erf^{-1} around $\frac{1}{2}$ and Taylor's Theorem. A similar approach gives the desired lower bound.

By symmetry, we can obtain the bounds

$$F^{-1}\left(\frac{1}{4}\right) \in \left[\mu - \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2}\right)\sigma - \frac{7\sqrt{2}}{2}\sigma\varepsilon, \mu - \sqrt{2}\text{erf}^{-1}\left(\frac{1}{2}\right)\sigma + \frac{5\sqrt{2}}{2}\sigma\varepsilon\right].$$

Combining this with the previous bounds and rescaling, we obtain the lemma statement. □

3.2.5 Outline of the Algorithm

We can decompose our algorithm into two components: generating a collection of candidate distributions containing at least one candidate with low statistical distance to the unknown distribution (Theorem 4), and identifying such a candidate from this collection (Theorem 1).

Generation of Candidate Distributions: In Section 3.3, we deal with generation of candidate distributions. A *candidate distribution* is described by the parameter set $(\hat{w}, \hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2)$, which corresponds to the GMM with PDF $f(x) = \hat{w}\mathcal{N}(\hat{\mu}_1, \hat{\sigma}_1^2, x) + (1 - \hat{w})\mathcal{N}(\hat{\mu}_2, \hat{\sigma}_2^2, x)$. As suggested by Lemma 17, if we have a candidate distribution with sufficently accurate parameters, it will have low statistical distance to the unknown distribution. Our first goal will be to generate a collection of candidates that contains at least one such candidate. Since the time complexity of our algorithm depends on the size of this collection, we wish to keep it to a minimum.

At a high level, we sequentially generate candidates for each parameter. In particular, we start by generating candidates for the mixing weight. While most of these will be inaccurate, we will guarantee to produce at least one appropriately accurate candidate $\hat{w}^*$. Then, for each candidate mixing weight, we will generate candidates for the mean of one of the Gaussians. We will guarantee that, out of the candidate means we generated for $\hat{w}^*$, it is likely that at least one candidate $\hat{\mu}_1^*$ will be sufficiently close to the true mean for this component. The candidate means that were generated for other mixing weights have no such guarantee. We use a similar sequential approach to generate candidates for the variance of this component. Once we have a description of the first component, we simulate the process of subtracting it from the mixture, thus giving us a single Gaussian, whose parameters we can learn. We can not immediately identify which candidates have inaccurate parameters, and they serve only to inflate the size of our collection.

At a lower level, our algorithm starts by generating candidates for the mixing weight followed by generating candidates for the mean of the component with the smaller value of $\frac{\sigma_i}{w_i}$. Note that we do not know which of the two Gaussians this is. The solution is to branch our algorithm, where each branch assumes a correspondence to a different Gaussian. One of the two branches is guaranteed to be correct, and it will only double the number of candidate distributions. We observe that if we take n samples from a single Gaussian, it is likely that there will exist a sample at distance $O(\frac{\sigma}{n})$ from its mean. Thus, if we take $\Theta(\frac{1}{w_i \varepsilon})$ samples from the mixture, one of them will be sufficiently close to the mean of the corresponding Gaussian. Exploiting this observation we obtain candidates for the mixing weight and the first mean as summarized by Lemma 27.

Next, we generate candidates for the variance of this Gaussian. Our specific approach is based on the observation that given n samples from a single Gaussian, the minimum distance of a sample to the mean will likely be $\Theta(\frac{\sigma}{n})$. In the mixture, this property will still hold for the Gaussian with the smaller $\frac{\sigma_i}{w_i}$, so we extract this statistic and use a grid around it to generate sufficiently accurate candidates for σ_i. This is Lemma 29.

At this point, we have a complete description of one of the component Gaussians.

Also, we can generate an empirical distribution of the mixture, which gives an adequate approximation to the true distribution. Given these two pieces, we update the empirical distribution by removing probability mass contributed by the known component. When done carefully, we end up with an approximate description of the distribution of the unknown component. At this point, we extract the median and the interquartile range (IQR) of the resulting distribution. These statistics are robust, so they can tolerate error in our approximation. Finally, the median and IQR allow us to derive the last mean and variance of our distribution. This is Lemma 31.

Putting everything together, we obtain the following result whose proof is in Section 3.3.5.

Theorem 4. *For all $\varepsilon, \delta > 0$, given $\log(1/\delta) \cdot O(1/\varepsilon^2)$ independent samples from an arbitrary mixture F of two univariate Gaussians, we can generate a collection of $\log(1/\delta) \cdot \tilde{O}(1/\varepsilon^3)$ candidate mixtures of two univariate Gaussians, containing at least one candidate F' such that $d_{\mathrm{TV}}(F, F') < \varepsilon$ with probability at least $1 - \delta$.*

Candidate Selection: In view of Theorem 4, to prove our main result it suffices to select from among the candidate mixtures some mixture that is close to the unknown mixture. In Chapter 2, we described a tournament-based algorithm (Theorem 1) for identifying a candidate which has low statistical distance to the unknown mixture, which is used to conclude the proof of Theorem 2.

3.3 Generating Candidate Distributions

By Proposition 16, if one of the Gaussians of a mixture has a negligible mixing weight, it has a negligible impact on the mixture's statistical distance to the unknown mixture. Hence, the candidate means and variances of this Gaussian are irrelevant. This is fortunate, since if $\min(w, 1-w) << \varepsilon$ and we only draw $\tilde{O}(1/\varepsilon^2)$ samples from the unknown mixture, as we are planning to do, we have no hope of seeing a sufficient number of samples from the low-weight Gaussian to perform accurate statistical tests for it. So for this section we will assume that $\min(w, 1-w) \geq \Omega(\varepsilon)$ and we will deal with the other case separately.

3.3.1 Generating Mixing Weight Candidates

The first step is to generate candidates for the mixing weight. We can obtain a collection of $O(\frac{1}{\varepsilon})$ candidates containing some $\hat{w}^* \in [w - \varepsilon, w + \varepsilon]$ by simply taking the set $\{t\varepsilon \mid t \in \left[\frac{1}{\varepsilon}\right]\}$.

3.3.2 Generating Mean Candidates

The next step is to generate candidates for the mean corresponding to the Gaussian with the smaller value of $\frac{\sigma_i}{w_i}$. Note that, a priori, we do not know whether $i = 1$ or $i = 2$. We try both cases, first generating candidates assuming they correspond to μ_1, and then repeating with μ_2. This will multiply our total number of candidate distributions by a factor of 2. Without loss of essential generality, assume for this section that $i = 1$.

We want a collection of candidates containing $\hat{\mu}_1^*$ such that $\mu_1 - \varepsilon\sigma_1 \leq \hat{\mu}_1^* \leq \mu_1 + \varepsilon\sigma_1$.

Proposition 25. *Fix $i \in \{1, 2\}$. Given $\frac{20\sqrt{2}}{3w_i\varepsilon}$ samples from a GMM, there will exist a sample $\hat{\mu}_i^* \in \mu_i \pm \varepsilon\sigma_i$ with probability $\geq \frac{99}{100}$.*

Proof. The probability that a sample is from $\mathcal{N}_i$ is w_i. Using the CDF of the half-normal distribution, given that a sample is from $\mathcal{N}_i$, the probability that it is at a distance $\leq \varepsilon\sigma_i$ from μ_i is $\mathrm{erf}\left(\frac{\varepsilon}{\sqrt{2}}\right)$.

If we take a single sample from the mixture, it will satisfy the desired conditions with probability at least $w_i\mathrm{erf}\left(\frac{\varepsilon}{\sqrt{2}}\right)$. If we take $\frac{20\sqrt{2}}{3w_i\varepsilon}$ samples from the mixture, the probability that some sample satisfies the conditions is at least

$$1 - \left(1 - w_i\mathrm{erf}\left(\frac{\varepsilon}{\sqrt{2}}\right)\right)^{\frac{20\sqrt{2}}{3w_i\varepsilon}} \geq 1 - \left(1 - w_i \cdot \frac{3}{4}\frac{\varepsilon}{\sqrt{2}}\right)^{\frac{20\sqrt{2}}{3w_i\varepsilon}} \geq 1 - e^{-5} \geq \frac{99}{100}$$

where the first inequality is by noting that $\mathrm{erf}(x) \geq \frac{3}{4}x$ for $x \in [0, 1]$. □

Proposition 26. *Fix $i \in \{1, 2\}$. Suppose $w_i - \varepsilon \leq \hat{w}_i \leq w_i + \varepsilon$, and $w_i \geq \varepsilon$. Then $\frac{2}{\hat{w}_i} \geq \frac{1}{w_i}$.*

Proof. $w_i \geq \varepsilon$ implies $w_i \geq \frac{w_i+\varepsilon}{2}$, and thus $\frac{2}{\hat{w}_i} \geq \frac{2}{w_i+\varepsilon} \geq \frac{1}{w_i}$. □

We use these facts to design a simple algorithm: for each candidate $\hat{w}_1$ (from Section 3.3.1), take $\frac{40\sqrt{2}}{3\hat{w}_1\varepsilon}$ samples from the mixture and use each of them as a candidate for μ_1.

We now examine how many candidate pairs $(\hat{w}, \hat{\mu}_1)$ we generated. Naively, since $\hat{w}_i$ may be as small as $O(\varepsilon)$, the candidates for the mean will multiply the size of our collection by $O\left(\frac{1}{\varepsilon^2}\right)$. However, we note that when $\hat{w}_i = \Omega(1)$, then the number of candidates for μ_i is actually $O\left(\frac{1}{\varepsilon}\right)$. We count the number of candidate triples $(\hat{w}, \hat{\mu}_1)$, combining with previous results in the following:

Lemma 27. *Suppose we have sample access to a GMM with (unknown) parameters $(w, \mu_1, \mu_2, \sigma_1, \sigma_2)$. Then for any $\varepsilon > 0$ and constants $c_w, c_m > 0$, using $O(\frac{1}{\varepsilon^2})$ samples from the GMM, we can generate a collection of $O\left(\frac{\log \varepsilon^{-1}}{\varepsilon^2}\right)$ candidate pairs for (w, μ_1). With probability $\geq \frac{99}{100}$, this will contain a pair $(\hat{w}^*, \hat{\mu}_1^*)$ such that $\hat{w}^* \in w \pm O(\varepsilon)$, $\hat{\mu}_1^* \in \mu_1 \pm O(\varepsilon)\sigma_1$.*

Proof. Aside from the size of the collection, the rest of the conclusions follow from Propositions 25 and 26.

For a given $\hat{w}$, the number of candidates $\hat{\mu}_1$ we consider is $\frac{40\sqrt{2}}{3\hat{w}\varepsilon}$. We sum this over all candidates for $\hat{w}$, namely, $\varepsilon, 2\varepsilon, \ldots, 1-\varepsilon$, giving us

$$\sum_{t=1}^{\frac{1}{\varepsilon}-1} \frac{40\sqrt{2}}{3k\varepsilon^2} = \frac{40\sqrt{2}}{3\varepsilon^2} H_{\frac{1}{\varepsilon}-1} = O\left(\frac{\log \varepsilon^{-1}}{\varepsilon^2}\right)$$

where H_n is the nth harmonic number. □

This implies that we can generate $O\left(\frac{1}{\varepsilon^2}\right)$ candidate triples, such that at least one pair simultaneously describes w and μ_1 to the desired accuracy.

3.3.3 Generating Candidates for a Single Variance

In this section, we generate candidates for the variance corresponding to the Gaussian with the smaller value of $\frac{\sigma_i}{w_i}$. We continue with our guess of whether $i = 1$ or $i = 2$

from the previous section.

Again, assume for this section that $i = 1$. The basic idea is that we will find the closest point to $\hat{\mu}_1$. We use the following property (whose proof is deferred to Appendix A) to establish a range for this distance, which we can then grid over.

We note that this lemma holds in scenarios more general than we consider here, including $k > 2$ and when samples are drawn from a distribution which is only close to a GMM, rather than exactly a GMM.

Lemma 28. *Let c_1 and c_2 be constants as defined in Proposition 51, and $c_3 = \frac{c_1}{9\sqrt{2}c_2}$. Consider a mixture of k Gaussians f, with components $\mathcal{N}(\mu_1, \sigma_1^2), \ldots, \mathcal{N}(\mu_k, \sigma_k^2)$ and weights $w_1, \ldots, w_k$, and let $j = \arg\min_i \frac{\sigma_i}{w_i}$. Suppose we have estimates for the weights and means for all $i \in [1, k]$:*

- *$\hat{w}_i$, such that $\frac{1}{2}\hat{w}_i \le w_i \le 2\hat{w}_i$*
- *$\hat{\mu}_i$, such that $|\hat{\mu}_i - \mu_i| \le \frac{c_3}{2k}\sigma_j$*

Now suppose we draw $n = \frac{9\sqrt{\pi}c_2}{2\hat{w}_j}$ samples $X_1, \ldots, X_n$ from a distribution $\hat{f}$, where $d_{\mathrm{K}}(f, \hat{f}) \le \delta = \frac{c_1}{2n} = \frac{c_1}{9\sqrt{\pi}c_2}\hat{w}_j$. Then $\min_i |X_i - \hat{\mu}_j| \in [\frac{c_3}{2k}\sigma_j, (\sqrt{2} + \frac{c_3}{2k})\sigma_j]$ with probability $\ge \frac{9}{10}$.

Summarizing what we have so far,

Lemma 29. *Suppose we have sample access to a GMM with parameters $(w, \mu_1, \mu_2, \sigma_1, \sigma_2)$, where $\frac{\sigma_1}{w} \le \frac{\sigma_2}{1-w}$. Furthermore, we have estimates $\hat{w}^* \in w \pm O(\varepsilon)$, $\hat{\mu}_1^* \in \mu_1 \pm O(\varepsilon)\sigma_1$. Then for any $\varepsilon > 0$, using $O(\frac{1}{\varepsilon})$ samples from the GMM, we can generate a collection of $O\left(\frac{1}{\varepsilon}\right)$ candidates for σ_1. With probability $\ge \frac{9}{10}$, this will contain a candidate $\hat{\sigma}_1^*$ such that $\hat{\sigma}_1^* \in (1 \pm O(\varepsilon))\sigma_1$.*

Proof. Let Y be the nearest sample to $\hat{\mu}_1$. From Lemma 28, with probability $\ge \frac{9}{10}$, $|Y - \hat{\mu}_1| \in [\frac{c_3}{4}\sigma_1, (\sqrt{2} + \frac{c_3}{4})\sigma_1]$.

We can generate candidates by rearranging the bounds to obtain

$$\frac{Y}{\sqrt{2} + \frac{c_3}{4}} \le \sigma_1 \le \frac{Y}{\frac{c_3}{4}}$$

Applying Fact 3 and noting that $\frac{R}{L} = O(1)$, we conclude that we can grid over this range with $O(\frac{1}{\varepsilon})$ candidates. □

3.3.4 Learning the Last Component Using Robust Statistics

At this point, our collection of candidates must contain a triple $(\hat{w}^*, \hat{\mu}_1^*, \hat{\sigma}_1^*)$ which are sufficiently close to the correct parameters. Intuitively, if we could remove this component from the mixture, we would be left with a distribution corresponding to a single Gaussian, which we could learn trivially. We will formalize the notion of "component subtraction," which will allow us to eliminate the known component and obtain a description of an approximation to the CDF for the remaining component. Using classic robust statistics (the median and the interquartile range), we can then obtain approximations to the unknown mean and variance. This has the advantage of a single additional candidate for these parameters, in comparison to $O(\frac{1}{\varepsilon})$ candidates for the previous mean and variance.

Our first step will be to generate an approximation of the overall distribution. We will do this only once, at the beginning of the entire algorithm. Our approximation is with respect to the Kolmogorov distance. Using the DKW inequality (Theorem 3) and Proposition 19, we obtain a $O(\frac{1}{\varepsilon^2})$-interval partition representation of $\hat{H}$ such that $d_{\mathrm{K}}(\hat{H}, H) \leq \varepsilon$, with probability $\geq 1 - \delta$ using $O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ time and samples (where H is the CDF of the GMM).

Next, for each candidate $(\hat{w}, \hat{\mu}_1, \hat{\sigma}_1)$, we apply Lemma 22 to obtain the $O(\frac{1}{\varepsilon^2})$-interval partition of the distribution with the known component removed, i.e., using the notation of Lemma 22, let H be the CDF of the GMM, F is our DKW-based approximation to H, w is the weight $\hat{w}$, and G is $\mathcal{N}(\hat{\mu}_1, \hat{\sigma}_1^2)$. We note that this costs $O(\frac{1}{\varepsilon^2})$ for each candidate triple, and since there are $\tilde{O}(\frac{1}{\varepsilon^3})$ such triples, the total cost of such operations will be $\tilde{O}(\frac{1}{\varepsilon^5})$. However, since the tournament we will use for selection of a candidate will require $\tilde{\Omega}(\frac{1}{\varepsilon^5})$ anyway, this does not affect the overall runtime of our algorithm.

The following proposition shows that, when our candidate triple is $(w^*, \mu_1^*, \sigma_1^*)$, the distribution that we obtain after subtracting the known component out and rescaling is close to the unknown component.

Proposition 30. *Suppose there exists a mixture of two Gaussians $F = w\mathcal{N}(\mu_1, \sigma_1^2) + (1-w)\mathcal{N}(\mu_2, \sigma_2^2)$ where $O(\varepsilon) \le w \le 1 - O(\varepsilon)$, and we have $\hat{F}$, such that $d_{\mathrm{K}}(\hat{F}, F) \le O(\varepsilon)$, $\hat{w}^*$, such that $|\hat{w}^* - w| \le O(\varepsilon)$, $\hat{\mu}_1^*$, such that $|\mu_1^* - \mu_1| \le O(\varepsilon)\sigma_1$, and $\hat{\sigma}_1^*$, such that $|\sigma_1^* - \sigma_1| \le O(\varepsilon)\sigma_1$.*

Then $d_{\mathrm{K}}\left(\mathcal{N}(\mu_2, \sigma_2^2), \frac{\hat{F} - \hat{w}^*\mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2})}{1-\hat{w}^*}\right) \le \frac{O(\varepsilon)}{1-w}$.

Proof.

$$
\begin{aligned}
& d_{\mathrm{K}}\left(\mathcal{N}(\mu_2, \sigma_2^2), \frac{\hat{F} - \hat{w}^*\mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2})}{1-\hat{w}^*}\right) \\
& = \frac{1}{1-\hat{w}} d_{\mathrm{K}}(\hat{w}^*\mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2}) + (1-\hat{w}^*)\mathcal{N}(\mu_2, \sigma_2^2), \hat{F}) \\
& \le \frac{1}{1-\hat{w}} (d_{\mathrm{K}}(\hat{w}^*\mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2}) + (1-\hat{w}^*)\mathcal{N}(\mu_2, \sigma_2^2), F) + d_{\mathrm{K}}(F, \hat{F})) \\
& \le \frac{1}{1-\hat{w}} (d_{\mathrm{TV}}(\hat{w}^*\mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2}) + (1-\hat{w}^*)\mathcal{N}(\mu_2, \sigma_2^2), F) + O(\varepsilon)) \\
& \le \frac{1}{1-\hat{w}} (|w - \hat{w}| + d_{\mathrm{TV}}(\mathcal{N}(\mu_1, \sigma_1), \mathcal{N}(\hat{\mu}_1^*, \hat{\sigma}_1^{*2})) + O(\varepsilon)) \\
& \le \frac{O(\varepsilon)}{1-\hat{w}} \\
& \le \frac{O(\varepsilon)}{1-w}
\end{aligned}
$$

The equality is a rearrangement of terms, the first inequality is the triangle inequality, the second inequality uses Fact 18, and the third and fourth inequalities use Propositions 15 and 16 respectively. □

Since the resulting distribution is close to the correct one, we can use robust statistics (via Lemmas 23 and 24) to recover the missing parameters. We combine this with previous details into the following Lemma.

Lemma 31. *Suppose we have sample access to a GMM with parameters $(w, \mu_1, \mu_2, \sigma_1, \sigma_2)$, where $\frac{\sigma_1}{w} \le \frac{\sigma_2}{1-w}$. Furthermore, we have estimates $\hat{w}^* \in w \pm O(\varepsilon)$, $\hat{\mu}_1^* \in \mu_1 \pm O(\varepsilon)\sigma_1$, $\hat{\sigma}_1^* \in (1 \pm O(\varepsilon))\sigma_1$. Then for any $\varepsilon > 0$, using $O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ samples from the GMM, with probability $\ge 1 - \delta$, we can generate candidates $\hat{\mu}_2^* \in \mu_2 \pm O\left(\frac{\varepsilon}{1-w}\right)\sigma_2$ and $\hat{\sigma}_2^* \in \left(1 \pm O\left(\frac{\varepsilon}{1-w}\right)\right)\sigma_2$.*

Proof. The proof follows the sketch outlined above. We first use Proposition 19 to construct an approximation $\hat{F}$ of the GMM F. Using Proposition 30, we see that $d_{\mathrm{K}}\left(\mathcal{N}(\mu_2,\sigma_2^2), \frac{\hat{F}-\hat{w}^*\mathcal{N}(\hat{\mu}_1^*,\hat{\sigma}_1^{*2})}{1-\hat{w}^*}\right) \le \frac{O(\varepsilon)}{1-w}$. By Lemma 22, we can compute a distribution $\hat{H}$ such that $d_{\mathrm{K}}(\mathcal{N}(\mu_2,\sigma_2^2),\hat{H}) \le \frac{O(\varepsilon)}{1-w}$. Finally, using the median and interquartile range and the guaranteed provided by Lemmas 23 and 24, we can compute candidates $\hat{\mu}_2^* \in \mu_2 \pm O\left(\frac{\varepsilon}{1-w}\right)\sigma_2$ and $\hat{\sigma}_2^* \in \left(1 \pm O\left(\frac{\varepsilon}{1-w}\right)\right)\sigma_2$ from $\hat{H}$, as desired.

□

3.3.5 Putting It Together

At this point, we are ready to prove our main result on generating candidate distributions.

Proof of Theorem 4: We produce two lists of candidates corresponding to whether $\min(w, 1-w) = \Omega(\varepsilon)$ or not:

- In the first case, combining Lemmas 27, 29, and 31 and taking the Cartesian product of the resulting candidates for the mixture's parameters, we see that we can obtain a collection of $O\left(\frac{\log \varepsilon^{-1}}{\varepsilon^3}\right)$ candidate mixtures. With probability $\ge \frac{4}{5}$, this will contain a candidate $(\hat{w}^*, \hat{\mu}_1^*, \hat{\mu}_2^*, \hat{\sigma}_1^*, \hat{\sigma}_2^*)$ such that $\hat{w} \in w \pm O(\varepsilon)$, $\hat{\mu}_i \in \mu_i \pm O(\varepsilon)\sigma_i$ for $i = 1,2$, and $\hat{\sigma}_i \in (1 \pm O(\varepsilon))\sigma_i$ for $i = 1,2$. Note that we can choose the hidden constants to be as small as necessary for Lemma 17, and thus we can obtain the desired total variation distance.

  Finally, note that the number of samples that we need for the above to hold is $O(1/\varepsilon^2)$. For this, it is crucial that we *first* draw a sufficient $O(1/\varepsilon^2)$ samples from the mixture (specified by the worse requirement among Lemmas 27, 29, and 31), and *then* execute the candidate generation algorithm outlined in Lemmas 27, 29, and 31. In particular, we do not want to redraw samples for every branching of this algorithm.

  Finally, to boost the success probability, we repeat the entire process $\log_5 \delta^{-1}$ times and let our collection of candidate mixutres be the union of the collections from these repetitions. The probability that none of these collections contains a suitable candidate distribution is $\leq \left(\frac{1}{5}\right)^{\log_5 \delta^{-1}} \leq \delta$.

- In the second case, i.e. when one of the weights, w.l.o.g. w_2, is $O(\varepsilon)$, we set $\hat{w}_1 = 1$ and we only produce candidates for (μ_1, σ_1^2). Note that this scenario fits into the framework of Lemmas 23 and 24. Our mixture F is such that $d_{\mathrm{K}}(F, \mathcal{N}(\mu_1, \sigma_1^2)) \leq d_{\mathrm{TV}}(F, \mathcal{N}(\mu_1, \sigma_1^2)) \leq O(\varepsilon)$. By the DKW inequality (Theorem 3), we can use $O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta})$ samples to generate the empirical distribution, which gives us a distribution $\hat{F}$ such that $d_{\mathrm{TV}}(\hat{F}, \mathcal{N}(\mu_1, \sigma_1^2)) \leq O(\varepsilon)$ (by triangle inequality), with probability $\geq 1 - \delta$. From this distribution, using the median and interquartile range and the guarantees of Lemmas 23 and 24, we can extract $\hat{\mu}_1^*$ and $\hat{\sigma}_1^*$ such that $|\hat{\mu}_1^* - \mu_1| \leq O(\varepsilon)\sigma_1$ and $|\hat{\sigma}_1^* - \sigma_1| \leq O(\varepsilon)\sigma_1$. Thus, by Lemma 17, we can achieve the desired total variation distance.

□

3.4 Proof of Theorem 2

Finally, we conclude with the proof of our main learning result.

Proof of Theorem 2: Theorem 2 is an immediate consequence of Theorems 4 and 1. Namely, we run the algorithm of Theorem 4 to produce a collection of Gaussian mixtures, one of which is within ε of the unknown mixture F. Then we use `FastTournament` of Theorem 1 to select from among the candidates a mixture that is $O(\varepsilon)$-close to F. For the execution of `FastTournament`, we need a PDF comparator for all pairs of candidate mixtures in our collection. Given that these are described with their parameters, our PDF comparators evaluate the densities of two given mixtures at a challenge point x and decide which one is largest. We also need sample access to our candidate mixtures. Given a parametric description $(w, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$ of a mixture, we can draw a sample from it as follows: first draw a uniform $[0, 1]$ variable whose value compared to w determines whether to sample from $\mathcal{N}(\mu_1, \sigma_1^2)$ or $\mathcal{N}(\mu_2, \sigma_2^2)$ in the second step; for the second step, use the Box-Muller transform [9] to obtain sample from either $\mathcal{N}(\mu_1, \sigma_1^2)$ or $\mathcal{N}(\mu_2, \sigma_2^2)$ as decided in the first step. □

3.5 Open Problems

There are a number of interesting directions for further study:

- **Are there faster algorithms for proper learning mixtures of 2 Gaussians in 1 dimension?** Our algorithm has near-optimal sample complexity of $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$, but the time complexity of $\tilde{O}\left(\frac{1}{\varepsilon^5}\right)$ still has room for improvement. Can it be lowered to the best-known lower bound of $O\left(\frac{1}{\varepsilon^2}\right)$?

- **What is the time complexity of proper learning mixtures of k Gaussians in 1 dimension?** As shown in [45], performing parameter estimation for a mixture of k Gaussians requires a number of samples which is exponential in k, even in 1 dimension. However, as demonstrated in [1], proper learning doesn't have the same restriction – they provide an algorithm for proper learning mixtures of k Gaussians in 1 dimension whose sample complexity has a polynomial dependence on k. Unfortunately, the running time of their algorithm is still exponential in k. Is it possible to break through this exponential dependence on the number of components, or is it required for proper learning?

- **How efficiently can we learn mixtures of Gaussians in high dimensions?** While it has been shown that mixtures of Gaussians are efficiently learnable in high dimensions [40, 45, 7], there are still many things we don't know. Do there exist practically efficient algorithms for this task, in any learning setting? In which circumstances can we avoid an exponential dependence on the number of components? There have been a number of recent works which study specific cases of this problem [39, 38, 1, 4], including mixtures of spherical Gaussians or when the covariance matrices are known and identical, but it is clear we still have much to learn.

Chapter 4

Covering Poisson Multinomial Distributions

4.1 Introduction

A Poisson Multinomial distribution of size n and dimension k is the distribution of the sum

$$X = \sum_{i=1}^{n} X_i,$$

where $X_1, \ldots X_n$ are independent k-dimensional categorical random variables (i.e., distributions over the k basis vectors). This generalizes both the Multinomial distribution (in which all X_i are identically distributed) and the Poisson Binomial distribution (which corresponds to the special case $k = 2$).

Our main result of this chapter is the following cover theorem:

Theorem 5. *The set of Poisson Multinomial distributions of size n and dimension k can be ε-covered by a set of size $n^{O(k^3)} \left(\frac{k}{\varepsilon}\right)^{\mathrm{poly}\left(k, \frac{1}{\varepsilon}\right)}$.*

Using only Theorem 5 and Theorem 1, we can also obtain the following learning result:

Corollary 32. *There exists an algorithm with* $\mathrm{poly}\left(\log n, k, \frac{1}{\varepsilon}\right)$ *sample complexity which learns Poisson Multinomial distributions. The running time of this algorithm* *is* $n^{O(k^3)} \left(\frac{k}{\varepsilon}\right)^{\mathrm{poly}(k,1/\varepsilon)}$.

There is a large body of work on Poisson Binomial distributions, including results on approximation, learning, covering and testing (i.e., see [20, 25] and the references contained therein). Comparatively, to our knowledge, there is far less prior work on Poisson Multinomial distributions. Daskalakis and Papadimitriou initiated the study of covers for Poisson Multinomial distributions, motivated by the efficient computation of approximate Nash equilibria for anonymous games [24]. Very recently, Valiant and Valiant proved a central limit theorem for this class [53], which we use as a key component in our analysis. In the realm of probability theory, there are some results by Roos for approximating a Poisson Multinomial distribution with a multivariate Poisson distribution or finite signed measures [49, 50, 51].

Approach. Our analysis will show that any Poisson Multinomial distribution is close to the sum of a small number of discretized Gaussians and a sparse Poisson Multinomial distribution. Since the resulting distribution has relatively few parameters (and the minimum eigenvalues of the covariance matrices of these Gaussians are sufficiently large), we can generate a sparse cover for this set by gridding over these parameters.

At the heart of our approximation lies a central limit theorem by Valiant and Valiant [53], which approximates a Poisson Multinomial distribution with a discretized Gaussian. There are two issues with a naive application of this result: the accuracy of the approximation decreases as we increase n or decrease the minimum eigenvalue of the covariance matrix of the distribution. The main technical challenge of this work lies in avoiding these two penalties.

To mitigate the latter cost, we apply a rounding scheme to the parameters of our distribution. We shift the parameters such that they are either equal to 0 or sufficiently far from 0, while simultaneously approximately preserving our mean vector. This results in each categorical random variable having a large variance in any direction which it is non-zero. Partitioning the categorical random variables into sets based on their non-zero directions and ignoring the zeroes, we get that the minimum variance of each set is large.

To avoid the cost based on the value of n, we repeatedly partition and sort the categorical random variables into bins. A bin will have the property that this cost is negligible compared to the variance of the collection in the bin, so we can apply the central limit theorem. We note that there will be a small number of categorical random variables which do not fall into a bin that has this property – these leftover variables result in the "sparse Poisson Multinomial distribution" component of our cover.

The above approximation results a cover which contains the sum of many discretized Gaussians. In order to reduce the size of our cover, we merge many Gaussians into a single distribution. It is well known that the sum of two Gaussians has the same distribution as a single Gaussian whose parameters are equal to the sum of the parameters of the two components. The same is not true for discretized Gaussians, and we must quantify the error induced by this merging operation.

Comparison with Prior Work. The main prior work on this problem is in [24, 27]. They produce a cover of size $n^{O(f(k,1/\varepsilon))}$, where f is polynomial in $1/\varepsilon$ and exponential in k^3. In comparison, our cover is of size $n^{O(k^3)}(k/\varepsilon)^{O(k/\varepsilon)}$, which significantly reduces the exponent of n and decouples the dependence on n and $1/\varepsilon$. However, theirs is a proper cover (i.e., the class of covering distributions is a subclass of the class we are trying to cover), while ours is improper (we cover it with a different class of distributions).

As mentioned before, the case when $k = 2$ is also called the Poisson Binomial distribution. These distributions have been studied extensively [20, 25, 17, 26], and as a result, we have a much richer understanding of them. The best known cover is proper, and is of size $n^2 + n \cdot (1/\varepsilon)^{O(\log^2(1/\varepsilon))}$, which has a much milder exponent for n and is exponential in only $\log^2(1/\varepsilon)$. Furthermore, there exist learning algorithms for this class which have sample complexity independent of n and a time complexity which is polynomial in $\log n$. In light of these results, we believe that sparser covers and more efficient algorithms exist for the Poisson Multinomial case.

4.2 Preliminaries

We start by formally defining a Poisson Multinomial distribution.

Definition 6. *A* Categorical Random Variable *(CRV) of dimension k is a random variable that takes values in $\{e_1, \ldots, e_k\}$ where e_j is the k-dimensional unit vector along direction j. $\pi(i)$ is the probability of observing e_i.*

Definition 7. *A* Poisson Multinomial Distribution *(PMD) is given by the law of the sum of n independent but not necessarily identical categorical random variables of dimension k. A PMD is parameterized by a nonnegative matrix $\pi \in [0,1]^{n \times k}$ each of whose rows sum to 1 is denoted by M^π, and is defined by the following random process: for each row $\pi(i, \cdot)$ of matrix π interpret it as a probability distribution over the columns of π and draw a column index from this distribution; return a row vector recording the total number of samples falling into each column (the histogram of the samples). We will refer to n and k as the* size *and* dimension *of the PMD, respectively.*

We note that a sample from a PMD is redundant – given $k-1$ coordinates of a sample, we can recover the final coordinate by noting that the sum of all k coordinates is n. For instance, while a Binomial distribution is over a support of size 2, a sample is 1-dimensional since the frequency of the other coordinate may be inferred given the parameter n. With this inspiration in mind, we provide the following definitions:

Definition 8. *A* Truncated Categorical Random Variable *of dimension k is a random variable that takes values in $\{0, e_1, \ldots, e_{k-1}\}$ where e_j is the $(k-1)$-dimensional unit vector along direction j, and 0 is the $(k-1)$ dimensional zero vector. $\rho(0)$ is the probability of observing the zero vector, and $\rho(i)$ is the probability of observing e_i.*

While we will approximate the Multinomial distribution with Gaussian distributions, it does not make sense to compare discrete distributions with continuous distributions, since the total variation distance is always 1. As such, we must discretize the Gaussian distributions. We will use the notation $\lfloor x \rceil$ to say that x is rounded to the nearest integer (with ties being broken arbitrarily). If x is a vector, we round each coordinate independently to the nearest integer.

Definition 9. *A* Generalized Multinomial Distribution *(GMD) is given by the law of the sum of n independent but not necessarily identical truncated categorical random variables of dimension k. A GMD is parameterized by a nonnegative matrix $\rho \in [0,1]^{n\times(k-1)}$ each of whose rows sum to at most 1 is denoted by G^{ρ}, and is defined by the following random process: for each row $\rho(i,\cdot)$ of matrix ρ interpret it as a probability distribution over the columns of ρ – including, if $\sum_{j=1}^{k}\rho(i,j) < 1$, an "invisible" column 0 – and draw a column index from this distribution; return a row vector recording the total number of samples falling into each column (the histogram of the samples). We will refer to n and k as the* size *and* dimension *of the GMD, respectively.*

We note that a PMD corresponds to a GMD where the "invisible" column is the zero vector, and thus the definition of GMDs is more general than that of PMDs. However, whenever we refer to a GMD in this chapter, it will explicitly have a non-zero invisible column – in particular, as we will see later, all entries in the invisible column will be at least $\frac{1}{k}$.

Definition 10. *The k-dimensional* Discretized Gaussian Distribution *with mean μ and covariance matrix Σ, denoted $\lfloor\mathcal{N}(\mu,\Sigma)\rceil$, is the distribution with support $\mathbb{Z}^k$ obtained by picking a sample according to the k-dimensional Gaussian $\mathcal{N}(\mu,\Sigma)$, then rounding each coordinate to the nearest integer.*

Definition 11. *The k-dimensional* Detruncated Discretized Gaussian Distribution *with mean μ, covariance matrix Σ, size n, and position i, denoted $\lfloor\mathcal{N}(\mu,\Sigma,n,i)\rceil$, is the distribution with support $\mathbb{Z}^k$ obtained by drawing a sample via the following process: draw a sample from the $(k-1)$-dimensional Gaussian $\mathcal{N}(\mu,\Sigma)$, round each coordinate to the nearest integer to obtain a vector $(x_1,\ldots,x_{k-1})$, and return the vector $(x_1,\ldots,x_{i-1},n-\sum_{j=1}^{k-1}x_j,x_i,\ldots,x_{k-1})$.*

We will use the following form of Chernoff/Hoeffding bounds:

Lemma 33 (Chernoff/Hoeffding)**.** *Let $Z_1,\ldots,Z_m$ be independent random variables* *with $Z_i \in [0,1]$ for all i. Then, if $Z = \sum_{i=1}^n Z_i$ and $\gamma \in (0,1)$,*

$$\Pr[|Z - E[Z]| \geq \gamma E[Z]] \leq 2\exp(-\gamma^2 E[Z]/3).$$

4.2.1 Covariance Matrices of Truncated Categorical Random Variables

First, recall the definition of a symmetric diagonally dominant matrix.

Definition 12. *A matrix A is* symmetric diagonally dominant *(SDD) if $A^T = A$ and $A_{ii} \geq \sum_{j\neq i} |A_{ij}|$ for all i.*

As a tool, we will use this corollary of the Gershgorin Circle Theorem [36] which follows since all eigenvalues of a symmetric matrix are real.

Proposition 34. *Given an SDD matrix A with positive diagonal entries, the minimum eigenvalue of A is at least $\min_i A_{ii} - \sum_{j\neq i} |A_{ij}|$.*

Proposition 35. *The minimum eigenvalue of the covariance matrix Σ of a truncated CRV is at least $\rho(0)\min_i \rho(i)$.*

Proof. The entries of the covariance matrix are

$$\begin{aligned}\Sigma_{ij} &= E[x_i x_j] - E[x_i]E[x_j] \\ &= \begin{cases} \rho(i) - \rho(i)^2 & \text{if } i = j \\ -\rho(i)\rho(j) & \text{else} \end{cases}\end{aligned}$$

We note that Σ is SDD, since $\sum_{j\neq i} |\Sigma_{ij}| = \rho(i)\sum_{j\neq i}\rho(j) = \rho(i)(1-\rho(i)-\rho(0)) \leq \rho(i)(1-\rho(i)) = \Sigma_{ii}$. Thus, applying Proposition 34, we see that the minimum eigenvalue of Σ is at least $\min_i \rho(i)(1-\rho(i)) - \rho(i)(1-\rho(i)-\rho(0)) = \rho(0)\min_i \rho(i)$. □

4.2.2 Sums of Discretized Gaussians

In this section, we will obtain total variation distance bounds on merging the sum of discretized Gaussians. It is well known that the sum of multiple Gaussians has the same distribution as a single Gaussian with parameters equal to the sum of the components' parameters. However, this is not true if we are summing discretized Gaussians – we quantify the amount we lose by replacing the distribution with a single Gaussian, and then discretizing afterwards.

As a tool, we will use the following result from [18]:

Proposition 36 (Proposition B.5 in [18]). *Let $X \sim \mathcal{N}(\mu, \sigma^2)$ and $\lambda \in \mathbb{R}$. Then*

$$d_{\mathrm{TV}}\left(\lfloor X + \lambda \rceil, \lfloor X \rceil + \lfloor \lambda \rceil\right) \le \frac{1}{2\sigma}.$$

From this, we can obtain the following:

Proposition 37. *Let $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$. Then*

$$d_{\mathrm{TV}}\left(\lfloor X_1 + X_2 \rceil, \lfloor X_1 \rceil + \lfloor X_2 \rceil\right) \le \frac{1}{2\sigma},$$

where $\sigma = \max_i \sigma_i$.

Proof. First, suppose without loss of generality that $\sigma_1 \ge \sigma_2$.

$$\begin{aligned}
& d_{\mathrm{TV}}\left(\lfloor X_1 + X_2 \rceil, \lfloor X_1 \rceil + \lfloor X_2 \rceil\right) \\
&= \frac{1}{2} \sum_{i=-\infty}^{\infty} \left|\Pr(\lfloor X_1 + X_2 \rceil = i) - \Pr(\lfloor X_1 \rceil + \lfloor X_2 \rceil = i)\right| \\
&= \frac{1}{2} \sum_{i=-\infty}^{\infty} \left| \int_{-\infty}^{\infty} f_{X_2}(\lambda) \Pr(\lfloor X_1 + \lambda \rceil = i)\, d\lambda - \int_{-\infty}^{\infty} f_{X_2}(\lambda) \Pr(\lfloor X_1 \rceil + \lfloor \lambda \rceil = i)\, d\lambda \right| \\
&= \frac{1}{2} \sum_{i=-\infty}^{\infty} \left| \int_{-\infty}^{\infty} f_{X_2}(\lambda) (\Pr(\lfloor X_1 + \lambda \rceil = i) - \Pr(\lfloor X_1 \rceil + \lfloor \lambda \rceil = i))\, d\lambda \right| \\
&\le \frac{1}{2} \sum_{i=-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X_2}(\lambda) \left|(\Pr(\lfloor X_1 + \lambda \rceil = i) - \Pr(\lfloor X_1 \rceil + \lfloor \lambda \rceil = i))\right| d\lambda \\
&= \frac{1}{2} \int_{-\infty}^{\infty} f_{X_2}(\lambda) \left(\sum_{i=-\infty}^{\infty} \left|(\Pr(\lfloor X_1 + \lambda \rceil = i) - \Pr(\lfloor X_1 \rceil + \lfloor \lambda \rceil = i))\right| \right) d\lambda \\
&\le \int_{-\infty}^{\infty} f_{X_2}(\lambda) \frac{1}{2\sigma_1}\, d\lambda \\
&= \frac{1}{2\sigma}
\end{aligned}$$

The second inequality uses Proposition 36. □

This leads to the following lemma:

Lemma 38. *Let* $X_1 \sim \mathcal{N}(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$ *and* $X_2 \sim \mathcal{N}(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$ *be* k*-dimensional Gaussian random variables, and let* $\sigma = \min_j \max_i \sigma_{i,j}$ *where* $\sigma_{i,j}$ *is the standard deviation of* X_i *in the direction parallel to the* j*th coordinate axis. Then*

$$d_{\mathrm{TV}}\left(\lfloor X_1 + X_2 \rceil, \lfloor X_1 \rceil + \lfloor X_2 \rceil\right) \leq \frac{k}{2\sigma}.$$

Proof. The proof is by induction on k. The base case of $k = 1$ is handled by Proposition 37. For general k, we use a standard hybridization argument. Denote the jth coordinate of X_i as x_{ij}.

$$\begin{aligned}
&d_{\mathrm{TV}}\left(\lfloor X_1 + X_2 \rceil, \lfloor X_1 \rceil + \lfloor X_2 \rceil\right)\\
&= d_{\mathrm{TV}}\left((\lfloor x_{11} + x_{21} \rceil, \ldots, \lfloor x_{1k} + x_{2k} \rceil), (\lfloor x_{11} \rceil + \lfloor x_{21} \rceil, \ldots, \lfloor x_{1k} \rceil + \lfloor x_{2k} \rceil)\right)\\
&\leq d_{\mathrm{TV}}\left((\lfloor x_{11} + x_{21} \rceil, \ldots, \lfloor x_{1k} + x_{2k} \rceil), (\lfloor x_{11} \rceil + \lfloor x_{21} \rceil, \ldots, \lfloor x_{1k} + x_{2k} \rceil)\right)\\
&+ d_{\mathrm{TV}}\left((\lfloor x_{11} \rceil + \lfloor x_{21} \rceil, \ldots, \lfloor x_{1k} + x_{2k} \rceil), (\lfloor x_{11} \rceil + \lfloor x_{21} \rceil, \ldots, \lfloor x_{1k} \rceil + \lfloor x_{2k} \rceil)\right)\\
&\leq d_{\mathrm{TV}}\big((\lfloor x_{11} + x_{21} \rceil, \ldots, \lfloor x_{1(k-1)} + x_{2(k-1)} \rceil), (\lfloor x_{11} \rceil + \lfloor x_{21} \rceil, \ldots, \lfloor x_{1(k-1)} \rceil + \lfloor x_{2(k-1)} \rceil\big)\\
&+ d_{\mathrm{TV}}\left(\lfloor x_{1k} + x_{2k} \rceil, \lfloor x_{1k} \rceil + \lfloor x_{2k} \rceil\right)\\
&\leq \frac{k-1}{2\sigma} + \frac{1}{2\sigma} = \frac{k}{2\sigma}
\end{aligned}$$

The first inequality is the triangle inequality, the second uses Lemma 1, and the third uses the induction hypothesis and Proposition 37. □

4.3 Bounding the Parameters Away from Zero

In this section, our goal is to replace our PMD with one where all non-zero probabilities are sufficiently large, while still being close to the original in total variation distance. This can be summarized in the following theorem:

Theorem 6. *For any $c \leq \frac{1}{2k}$, given access to the parameter matrix ρ for a PMD M^{ρ}, we can efficiently construct another PMD $M^{\hat{\rho}}$, such that $\hat{\rho}(i,j) \notin (0,c)$ and*

$$d_{\mathrm{TV}}\left(M^{\rho}, M^{\hat{\rho}}\right) < O\left(c^{1/2} k^{5/2} \log^{1/2}\left(\frac{1}{ck}\right)\right)$$

To prove this theorem, we will use a stripped-down version of the analysis from [24]. At a high level, we will round the values which are in $(0,c)$ by shifting probability mass either to or from "heaviest" coordinate, while simultaneously (approximately) preserving the mean of the PMD. We relate the two distributions using a careful coupling argument and Poisson approximations to the Binomial distribution.

We will apply a rounding procedure to $O(k^2)$ sets. Fix some coordinate x, and select all CRVs where the parameter in coordinate x is in the range $(0,c)$. Partition this subset into $k-1$ sets, depending on which coordinate $y \neq x$ is the heaviest. We apply a rounding procedure separately to each of these sets. After this procedure, none of the parameters in coordinate x will be in $(0,c)$. We repeat this for all k possible settings of x. From the description below (and the restriction that $c \leq \frac{1}{2k}$), it will be clear that we will not "undo" any of our work and move probabilities back into $(0,c)$, so $O(k^2)$ applications of our rounding procedure will produce the result claimed in the theorem statement.

We fix some x, y in order to describe and analyze the process more formally. Define $\mathcal{I}_y^x = \{i \,|\, 0 < \rho(i,x) < c \wedge y = \arg\max_j \rho(i,j)\}$ (breaking ties lexicographically), and let $M^{\rho_{I_y^x}}$ be the PMD induced by this set. For the remainder of this section, without loss of generality, assume that the indices selected by $\mathcal{I}_y^x$ are 1 through $|\mathcal{I}_y^x|$.

We will apply the following rounding scheme to $\rho_{I_y^x}$ to obtain a new parameter matrix $\hat{\rho}_{I_y^x}$:

$$\hat{\rho}_{I_y^x}(i,j) = \begin{cases} \rho_{I_y^x}(i,j) & \text{if } j \notin \{x,y\} \\ c & \text{if } j = x \wedge i \leq \left\lfloor \frac{\sum_{i' \in I_y^x} \rho_{I_y^x}(i',x)}{c} \right\rfloor \\ 0 & \text{if } j = x \wedge i > \left\lfloor \frac{\sum_{i' \in I_y^x} \rho_{I_y^x}(i',x)}{c} \right\rfloor \\ 1 - \sum_{j' \neq y} \hat{\rho}_{I_y^x}(i,j') & \text{if } j = y \end{cases}$$

For the sake of analysis, we define the process **Fork**, for sampling from a CRV $\rho(i,\cdot)$ in $\mathcal{I}_y^x$:

- Let X_i be an indicator random variable, taking 1 with probability $\frac{1}{k}$ and 0 otherwise.
- If $X_i = 1$, then return e_x with probability $k\rho(i,x)$ and e_y with probability $1 - k\rho(i,x)$.
- If $X_i = 0$, then return e_j with probability 0 if $j = x$, $\frac{k}{k-1}(\rho(i,x) + \rho(i,y) - \frac{1}{k})$ if $j = y$, and $\frac{k}{k-1}\rho(i,j)$ otherwise.

We note that **Fork** is well defined as long as $\rho(i,x) \leq \frac{1}{k}$ and $\rho(i,x) + \rho(i,y) \geq \frac{1}{k}$. The former is true since $c \leq \frac{1}{k}$, and the latter is true since y was chosen to be the heaviest coordinate. Additionally, by calculating the probability of any outcome, we can see that **Fork** is equivalent to the regular sampling process. Define the (random) set $\boldsymbol{X} = \{i \,|\, X_i = 1\}$. We will use $\boldsymbol{\theta}$ to refer to a particular realization of this set. We define **Fork** for sampling from $\hat{\rho}(i,\cdot)$ in the same way, though we will denote the indicator random variables by $\hat{X}_i$ and $\hat{\boldsymbol{X}}$ instead. Note that, if $c \leq \frac{1}{k}$, the process will still be well defined after rounding. This is because $\hat{\rho}(i,x) \leq c \leq \frac{1}{k}$, and $\hat{\rho}(i,x) + \hat{\rho}(i,y) = \rho(i,x) + \rho(i,y) \geq \frac{1}{k}$. For the rest of this section, when we are drawing a sample from a CRV, we draw it via the process **Fork**.

The proof of Theorem 6 follows from the following lemmata:

Lemma 39.

$$\Pr\left(\boldsymbol{\theta} : \left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) - E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2} \right.$$

$$\left. \wedge \left|\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,j) - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right)^{1/2}\right)$$

$$\geq 1 - 4(ck)^{1/6}$$

Lemma 40. *Suppose that, for some $\boldsymbol{\theta}$, the following hold:*

$$\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) - E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2}$$

$$\left|\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,j) - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right)^{1/2}$$

Then, letting Z_i be the Bernoulli random variable with expectation $k\rho_{I_y^x}(i,x)$ (and $\hat{Z}_i$ defined similarly with $k\hat{\rho}_{I_y^x}(i,x)$),

$$\mathrm{d}_{\mathrm{TV}}\left(\sum_{i\in\boldsymbol{\theta}} Z_i, \sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\right) < O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right)$$

Lemma 41. *For any $\mathcal{I}_y^x$,*

$$d_{\mathrm{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}}\right) < O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right)$$

Since our final rounded PMD is generated after applying this rounding procedure $O(k^2)$ times, Theorem 6 follows from our construction and Lemma 41 via the triangle inequality.

Proof of Lemma 39: Note that $\sum_{i \in \boldsymbol{X}} k\rho_{I_y^x}(i,j) = \sum_{i \in I_y^x} \Omega_i$, where

$$\Omega_i = \begin{cases} k\rho_{I_y^x}(i,x) & \text{with probability } \frac{1}{k} \\ 0 & \text{with probability } 1 - \frac{1}{k} \end{cases}$$

We apply Lemma 33 to the rescaled random variables $\Omega_i' = \frac{1}{ck}\Omega_i$, with $\gamma = \sqrt{\frac{\log \frac{1}{ck}}{2E[\sum_{i \in I_y^x} \Omega_i]}}$, giving

$$\Pr\left[\left|\sum_{i \in I_y^x} \Omega_i' - E\Big[\sum_{i \in I_y^x} \Omega_i'\Big]\right| \geq \left(\frac{1}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i \in I_y^x} \Omega_i'\Big]\right)^{1/2}\right] \leq 2(ck)^{1/6}.$$

Unscaling the variables gives

$$\Pr\left[\left|\sum_{i \in \boldsymbol{X}} k\rho_{I_y^x}(i,x) - E\Big[\sum_{i \in \boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right| \geq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i \in \boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right)^{1/2}\right] \leq 2(ck)^{1/6}.$$

Applying the same argument to $\hat{\rho}_{I_y^x}$ gives

$$\Pr\left[\left|\sum_{i \in \hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x) - E\Big[\sum_{i \in \hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big]\right| \geq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i \in \hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big]\right)^{1/2}\right] \leq 2(ck)^{1/6}.$$

Since $\boldsymbol{X} \sim \hat{\boldsymbol{X}}$, by considering the joint probability space where $\boldsymbol{\theta} = \boldsymbol{X} = \hat{\boldsymbol{X}}$ and applying a union bound, we get

$$\Pr\left(\boldsymbol{\theta} : \left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) - E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right| \le \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2}\right.$$
$$\left.\wedge \left|\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,j) - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right| \le \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right)^{1/2}\right)$$
$$\ge 1 - 4(ck)^{1/6}$$

□

Proof of Lemma 40: Fix some $\boldsymbol{\theta} = \boldsymbol{X} = \hat{\boldsymbol{X}}$. Without loss of generality, assume $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] \ge E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]$. There are two cases:

Case 1. $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] \le (ck)^{3/4}$

From the first assumption in the lemma statement,

$$\begin{aligned}\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) &\le E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] + \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2}\\ &\le (ck)^{3/4} + \frac{(ck)^{7/8}}{\sqrt{2}}\log^{1/2}\left(\frac{1}{ck}\right) := g(c,k)\end{aligned}$$

Similarly, by the second assumption in the lemma statement and since $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] \ge E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]$, we also have that $\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,j) \le g(c,k)$.

By Markov's inequality, $\Pr\Big[\sum_{i\in\boldsymbol{\theta}} Z_i \ge 1\Big] \le \sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) \le g(c,k)$, and similarly, $\Pr\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i \ge 1\Big] \le g(c,k)$. This implies that

$$\left|\Pr\Big[\sum_{i\in\boldsymbol{\theta}} Z_i = 0\Big] - \Pr\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i = 0\Big]\right| \le 2g(c,k),$$

and thus by the coupling lemma,

$$d_{\mathrm{TV}}\left(\sum_{i\in\boldsymbol{\theta}} Z_i, \sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\right) \le 4g(c,k) = 4\left((ck)^{3/4} + \frac{(ck)^{7/8}}{\sqrt{2}}\log^{1/2}\left(\frac{1}{ck}\right)\right)$$

Case 2. $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] \geq (ck)^{3/4}$

We use the following claim, which is a combination of a classical result in Poisson approximation [6] and Lemma 3.10 in [23].

Claim 42. *For any set of independent Bernoulli random variables $\{Z_i\}_i$ with expectations $E[Z_i] \leq ck$,*

$$d_{\mathrm{TV}}\left(\sum_i Z_i, Poisson\Big(E\Big[\sum_i Z_i\Big]\Big)\right) \leq ck.$$

Applying this, we see

$$d_{\mathrm{TV}}\left(\sum_{i\in\boldsymbol{\theta}} Z_i, Poisson\Big(E\Big[\sum_{i\in\boldsymbol{\theta}} Z_i\Big]\Big)\right) \leq ck$$

$$d_{\mathrm{TV}}\left(\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i, Poisson\Big(E\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\Big]\Big)\right) \leq ck$$

We must now bound the distance between the two Poisson distributions. We use the following lemma from [24]:

Lemma 43 (Lemma B.2 in [24]). *If $\lambda = \lambda_0 + D$ for some $D > 0, \lambda_0 > 0$,*

$$d_{\mathrm{TV}}\left(Poisson(\lambda), Poisson(\lambda_0)\right) \leq D\sqrt{\frac{2}{\lambda_0}}.$$

Applying this gives that

$$\begin{aligned} & d_{\mathrm{TV}}\left(Poisson\Big(E\Big[\sum_{i\in\boldsymbol{\theta}} Z_i\Big]\Big), Poisson\Big(E\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\Big]\Big)\right) \\ & \leq \left|E\Big[\sum_{i\in\boldsymbol{\theta}} Z_i\Big] - E\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\Big]\right| \sqrt{\frac{2}{\min\Big\{E\Big[\sum_{i\in\boldsymbol{\theta}} Z_i\Big], E\Big[\sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\Big]\Big\}}} \end{aligned}$$

To bound this, we need the following proposition, which we prove below:

Proposition 44.

$$\sqrt{\frac{2\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x) - \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right|^2}{\min\left\{\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x), \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right\}}} \leq \sqrt{28ck\log\left(\frac{1}{ck}\right)}$$

Thus, using the triangle inequality and this proposition, for sufficiently small c, we get

$$d_{\mathrm{TV}}\left(\sum_{i\in\boldsymbol{\theta}} Z_i, \sum_{i\in\boldsymbol{\theta}} \hat{Z}_i\right) \leq 2ck + \sqrt{28ck\log\left(\frac{1}{ck}\right)} = O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right).$$

By comparing Cases 1 and 2, we see that the desired bound holds in both cases.

Proof of Proposition 44: By the definition of our rounding procedure, we observe that

$$\left|E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big]\right| \leq c$$

By the assumptions of Lemma 40 and the assumption that $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] \geq E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]$,

$$\begin{aligned}\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x) - \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right| &\leq \left|E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big]\right| \\ &\quad + \left(2ck\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right)^{1/2} \\ &\leq c + \left(2ck\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right)^{1/2},\end{aligned}$$

and thus,

$$\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x) - \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right|^2$$
$$\leq c^2 + 2ck\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] + \left(8c^3k\log\left(\frac{1}{ck}\right) E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right)^{1/2} \tag{4.1}$$

From the assumption that $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] \geq (ck)^{3/4}$, for sufficiently small c,

$$\begin{aligned} E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] &\geq (ck)^{3/8}\left(E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big]\right)^{1/2} \\ &\geq \left(2ck\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2} \end{aligned}$$

Combining this with the first assumption of Lemma 40,

$$\begin{aligned} \sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x) &\geq E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] - \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2} \\ &\geq \frac{1}{2}E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] \end{aligned}$$

Similarly, since $E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big] \geq E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big] - c \geq (ck)^{3/4} - c$, for c sufficiently small,

$$\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x) \geq \frac{1}{2}E\Big[\sum_{i\in\boldsymbol{X}} k\hat{\rho}_{I_y^x}(i,x)\Big]$$

It follows that

$$\begin{aligned}
&\min\left\{\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x), \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right\} \\
&\geq \frac{1}{2}\min\left\{E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big], E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big]\right\} \\
&= \frac{1}{2}E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,x)\Big] \\
&\geq \frac{1}{2}\left(E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] - c\right) \\
&\geq \frac{1}{4}E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] \qquad (4.2)
\end{aligned}$$

where the last equality follows for c sufficiently small because $E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,x)\Big] \geq (ck)^{3/4}$.

From (4.1) and (4.2), for c sufficiently small,

$$\frac{2\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x) - \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right|^2}{\min\left\{\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,x), \sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,x)\right\}} \leq 28ck\log\left(\frac{1}{ck}\right),$$

from which the proposition statement follows. □

□

Proof of Lemma 41: Throughout this proof, we will couple the two sampling processes such that $\boldsymbol{\theta} := \boldsymbol{X} = \hat{\boldsymbol{X}}$, which is possible since $\boldsymbol{X} \sim \hat{\boldsymbol{X}}$. Let ϕ be the random event that $\boldsymbol{\theta}$ satisfies the following conditions:

$$\left|\sum_{i\in\boldsymbol{\theta}} k\rho_{I_y^x}(i,j) - E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\boldsymbol{X}} k\rho_{I_y^x}(i,j)\Big]\right)^{1/2}$$

$$\left|\sum_{i\in\boldsymbol{\theta}} k\hat{\rho}_{I_y^x}(i,j) - E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right| \leq \left(\frac{ck}{2}\log\left(\frac{1}{ck}\right)E\Big[\sum_{i\in\hat{\boldsymbol{X}}} k\hat{\rho}_{I_y^x}(i,j)\Big]\right)^{1/2}$$

Suppose that ϕ occurs, and fix a $\boldsymbol{\theta}$ in this probability space. We start by showing that for such a $\boldsymbol{\theta}$,

$$\mathrm{d_{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) < O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right)$$

Let $M^{\rho_{I_y^x}^{\boldsymbol{\theta}}}$ and $M^{\rho_{I_y^x}^{\bar{\boldsymbol{\theta}}}}$ be the PMDs induced by the CRVs in $M^{\rho_{I_y^x}}$ with indices in $\boldsymbol{\theta}$ and not in $\boldsymbol{\theta}$, respectively. Define $M^{\hat{\rho}_{I_y^x}^{\boldsymbol{\theta}}}$ and $M^{\hat{\rho}_{I_y^x}^{\bar{\boldsymbol{\theta}}}}$ similarly. We can see

$$\begin{aligned}
\mathrm{d_{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) &= \mathrm{d_{TV}}\left(M^{\rho_{I_y^x}^{\boldsymbol{\theta}}} + M^{\rho_{I_y^x}^{\bar{\boldsymbol{\theta}}}}, M^{\hat{\rho}_{I_y^x}^{\boldsymbol{\theta}}} + M^{\hat{\rho}_{I_y^x}^{\bar{\boldsymbol{\theta}}}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) \\
&\leq \mathrm{d_{TV}}\left(M^{\rho_{I_y^x}^{\boldsymbol{\theta}}}, M^{\hat{\rho}_{I_y^x}^{\boldsymbol{\theta}}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) \\
&+ \mathrm{d_{TV}}\left(M^{\rho_{I_y^x}^{\bar{\boldsymbol{\theta}}}}, M^{\hat{\rho}_{I_y^x}^{\bar{\boldsymbol{\theta}}}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) \\
&\leq \mathrm{d_{TV}}\left(M^{\rho_{I_y^x}^{\boldsymbol{\theta}}}, M^{\hat{\rho}_{I_y^x}^{\boldsymbol{\theta}}} \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) \\
&= \mathrm{d_{TV}}\left(\sum_{i \in \boldsymbol{\theta}} Z_i, \sum_{i \in \boldsymbol{\theta}} \hat{Z}_i \,\middle|\, \boldsymbol{X} = \hat{\boldsymbol{X}} = \boldsymbol{\theta}\right) \\
&\leq O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right)
\end{aligned}$$

The first inequality is the triangle inequality, the second inequality is because the distributions for CRVs in $\bar{\boldsymbol{\theta}}$ are identical (since we do not change them in our rounding), and the third inequality is Lemma 40.

By the law of total probability for total variation distance,

$$\begin{aligned}
\mathrm{d_{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}}\right) &= \Pr(\phi)\mathrm{d_{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}} \,\middle|\, \phi\right) + \Pr(\bar{\phi})\mathrm{d_{TV}}\left(M^{\rho_{I_y^x}}, M^{\hat{\rho}_{I_y^x}} \,\middle|\, \bar{\phi}\right) \\
&\leq \left(1 - 4(ck)^{1/6}\right) \cdot O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right) + 4(ck)^{1/6} \cdot 1 \\
&= O\left(c^{1/2}k^{1/2}\log^{1/2}\left(\frac{1}{ck}\right)\right)
\end{aligned}$$

where the inequality is obtained by applying Lemma 39 and the bound shown above pointwise for $\boldsymbol{\theta}$ which satisfy ϕ.

□

4.4 From PMDs to GMDs to Gaussians via Valiants' CLT

4.4.1 Overview

At this point we have a "massaged" PMD $M^{\hat{\pi}}$, with no parameters lying in the interval $(0, c)$. We set c to be some polynomial of ε and $\frac{1}{k}$ which will be specified later. In this section, we will show how to relate the massaged PMD to our target distribution: a sum of k detruncated discretized Gaussians, plus a sparse PMD (i.e., one with size only $\mathrm{poly}(k, \frac{1}{\varepsilon})$).

The general roadmap is as follows: We start by partitioning the CRVs into k sets based on which basis vector we are most likely to observe. This allows us to argue that the minimum eigenvalue of the covariance matrix of the PMD defined by each set is sufficiently large. For each of these sets, we apply a central limit theorem by Valiant and Valiant [53], which bounds the total variation distance between a PMD and a discretized Gaussian with the same mean and covariance matrix. We must be careful when applying this result – since their bound depends on the size of the PMD, we must further partition each of these sets, apply the result to each of these subsets, and then "merge" the resulting discretized Gaussians together using Lemma 38. This allows us to replace most of the CRVs with a single discretized Gaussian, leaving us with a PMD of size $\mathrm{poly}(2^k, \frac{1}{\varepsilon})$. We can apply the central limit theorem again to all but $\mathrm{poly}(k, \frac{1}{\varepsilon})$ of these CRVs and obtain another discretized Gaussian, which we merge with the others. Combining the result from each of the sets of the original partition, we obtain the sum of k discretized Gaussians and a PMD of size $\mathrm{poly}(k, \frac{1}{\varepsilon})$, as desired.

4.4.2 Getting Our Hands Dirty

We now want to apply a result by Valiant and Valiant [53].

Theorem 7 (Theorem 4 from [53])**.** *Given a generalized multinomial distribution G^ρ, with k dimensions and n rows, let μ denote its mean and Σ denote its covariance* *matrix, then*

$$d_{\mathrm{TV}}\left(G^{\rho}, \lfloor \mathcal{N}(\mu,\sigma) \rceil\right) \leq \frac{k^{4/3}}{\sigma^{1/3}} \cdot 2.2 \cdot (3.1 + 0.83 \log n)^{2/3}$$

where σ^2 is the minimum eigenvalue of Σ.

As we can see from this inequality, there are two issues that may arise and lead to a bad approximation:

- The GMD has small variance in some direction (cf. Proposition 35)
- The GMD has a large size parameter

We must avoid both of these issues simultaneously – we will apply this result to several carefully chosen sets, and then merge the resulting Gaussians into one using Lemma 38.

The first step is to convert our PMD into several GMDs. We start by partitioning the CRVs into k sets $S_1, \ldots, S_k$, where $S_{j'} = \{i \,|\, j' = \arg\max_j \hat{\pi}(i,j)\}$ and ties are broken by lexicographic ordering. This defines $S_{j'}$ to be the set of indices of CRVs in which j' is the heaviest coordinate. Let $M^{\hat{\pi}_{j'}}$ be the PMD induced by taking the CRVs in $S_{j'}$. For the remainder of this section, we will focus on S_k, the other cases follow symmetrically.

We convert each CRV in S_k into a truncated CRV by omitting the kth coordinate, giving us a GMD $G^{\hat{\rho}_k}$. Since the kth coordinate was the heaviest, we can make the following observation:

Observation 45. *$\hat{\rho}_k(i,0) \geq \frac{1}{k}$ for all $i \in S_k$.*

If we tried to apply Theorem 7 to $G^{\hat{\rho}_k}$, we would obtain a vacuous result. For instance, if there exists a j such that $\hat{\rho}_k(i,j) = 0$ for all i, the variance in this direction would be 0 and the CLT would give us a trivial result. Therefore, we further partition S_k into 2^{k-1} sets indexed by $2^{[k-1]}$, where each set contains the elements of S_k which are non-zero on its indexing set and zero otherwise. More formally, $S_k^{\mathcal{I}} = \{i \,|\, (i \in S_k) \wedge (\hat{\rho}(i,j) \geq c \;\forall j \in \mathcal{I}) \wedge (\hat{\rho}(i,j) = 0 \;\forall j \notin \mathcal{I})\}$. For each of these sets, due to our rounding procedure, we know that the variance is non-negligible in each of the non-zero directions. The issue is that the size of each set might still be large compared to this variance. As one last step before applying the CLT, we group the sets $S_k^{\mathcal{I}}$ into buckets.

Define γ as a constant, and t as a polynomial of k and $\frac{1}{\varepsilon}$, both of which will be specified later. Define $B^l = \bigcup_{\mathcal{I} \in Q_l} S_k^{\mathcal{I}}$, where $Q_l = \{\mathcal{I} \,|\, |S_k^{\mathcal{I}}| \in [l^\gamma t, (l+1)^\gamma t)\}$. In other words, bin l will contain a collection of truncated CRVs, defined by the union of the previously defined sets which have a size falling in a particular interval.

At this point, we are ready to apply the central limit theorem:

Lemma 46. *Let $G^{\hat{\rho}_k^l}$ be the GMD induced by the truncated CRVs in B^l, and μ_k^l and Σ_k^l be its mean and covariance matrix. Then*

$$d_{\mathrm{TV}}\left(G^{\hat{\rho}_k^l}, \lfloor \mathcal{N}(\mu_k^l, \Sigma_k^l) \rceil\right) \leq \frac{8.646 k^{3/2} \log^{2/3}(2^k (l+1)^\gamma t)}{l^{\gamma/6} t^{1/6} c^{1/6}}$$

Proof. This follows from Theorem 7, it suffices to bound the values of "n" and "σ^2" which appear in the theorem statement.

B^l is the union of at most 2^k sets, each of size at most $(l+1)^\gamma t$, which gives us the upper bound of $2^k (l+1)^\gamma t$ as the size of induced GMD.

We must be more careful when reasoning about the minimum eigenvalue of Σ_k^l – indeed, it may be 0 if there exists a j' such that for all i, $\hat{\rho}_k^l(i, j') = 0$. Therefore, we apply the CLT on the GMD defined by removing all zero-columns from ρ_k^l, taking us down to a dimension $k' \leq k$. Afterwards, we lift the related discretized Gaussian up to k dimensions by inserting 0 for the means and covariances involving any of the $k - k'$ dimensions we removed. This operation will not increase the total variation distance, by Lemma 1. From this point, we assume that all columns of $\hat{\rho}_k^l$ are non-zero.

Consider an arbitrary $S_k^{\mathcal{I}}$ which is included in B^l. Let $\mathcal{E}_{\mathcal{I}} = \text{span}\{e_i \,|\, i \in \mathcal{I}\}$. Applying Proposition 35, Observation 45, and the properties necessary for inclusion in $S_k^{\mathcal{I}}$, we can see that a CRV in $S_k^{\mathcal{I}}$ has variance at least $\frac{c}{k}$ within $\mathcal{E}_{\mathcal{I}}$. Since inclusion in B^l means that $|S_k^{\mathcal{I}}| \geq l^\gamma t$, and variance is additive for independent random variables, the GMD induced by $S_k^{\mathcal{I}}$ has variance at least $l^\gamma t \frac{c}{k}$ within $\mathcal{E}_{\mathcal{I}}$. To conclude, we note that if a column in $\hat{\rho}_k^l$ is non-zero, there must be some $\mathcal{I}^* \in Q_l$ which intersects the corresponding dimension. Since $S_k^{\mathcal{I}^*}$ causes the variance in this direction to be at least $l^\gamma t \frac{c}{k}$, we see that the variance in every direction must be this large.

By substituting these values into Theorem 7, we obtain the claimed bound. □

We note that this gives us a vacuous bound for B^0, which we must deal with separately. The issue with this bin is that the variance in some directions might be small compared to the size of the GMD induced by the bin. The intuition is that we can remove the truncated CRVs which are non-zero in these low-variance dimensions, and the remaining truncated CRVs can be combined into another GMD.

Lemma 47. *Let $G^{\hat{\rho}_k^0}$ be the GMD induced by the truncated CRVs in B^0. Given $\hat{\rho}_k^0$, we can efficiently compute a partition of B^0 into S and $\bar{S}$, where $|\bar{S}| \le kt$. Letting μ_S and Σ_S be the mean and covariance matrix of the GMD induced by S, and $G^{\hat{\rho}_k^{\bar{S}}}$ be the GMD induced by $\bar{S}$,*

$$d_{\mathrm{TV}}\left(G^{\hat{\rho}_k^0}, \lfloor \mathcal{N}(\mu_S, \Sigma_S) \rceil + G^{\hat{\rho}_k^{\bar{S}}}\right) \le \frac{8.646 k^{3/2} \log^{2/3}(2^k t)}{t^{1/6} c^{1/6}}$$

Proof. The algorithm iteratively eliminates columns which have fewer than t non-zero entries. For each such column j, add all truncated CRVs which have non-zero entries in column j to $\bar{S}$. Since there are only k columns, we add at most kt truncated CRVs to $\bar{S}$.

Now, we apply Theorem 7 to the truncated CRVs in S. The analysis of this is similar to the proof of Lemma 46. As argued before, we can drop the dimensions which have 0 variance. This time, the size of the GMD is at most $2^k t$, which follows from the definition of B^0. Recall that the minimum variance of a single truncated CRV in S is at least $\frac{c}{k}$ in any direction in the span of its non-zero columns. After removing the CRVs in $\bar{S}$, every dimension with non-zero variance must have at least t truncated CRVs which are non-zero in that dimension, giving a variance of at least $\frac{tc}{k}$. Substituting these parameters into Theorem 7 gives the claimed bound. □

We put these two lemmata together to obtain the following result:

Lemma 48. *Let $G^{\hat{\rho}_k}$ be a GMD with $\hat{\rho}_k(i,j) \notin (0,c)$ and $\sum_j \rho_k(i,j) \leq 1 - \frac{1}{k}$ for all i, and let S_k be its set of component truncated CRVs. There exists an efficiently computable partition of S_k into S and $\bar{S}$, where $|\bar{S}| \leq kt$. Furthermore, letting μ_S and Σ_S be the mean and covariance matrix of the GMD induced by S, and $G^{\hat{\rho}_k^{\bar{S}}}$ be the GMD induced by $\bar{S}$,*

$$d_{\mathrm{TV}}\left(G^{\hat{\rho}_k}, \lfloor \mathcal{N}(\mu_S, \Sigma_S) \rceil + G^{\hat{\rho}_k^{\bar{S}}}\right) \leq O\left(\frac{k^{13/6}\log^{2/3} t}{c^{1/6}t^{1/6}} + \frac{k^{3/2}}{c^{1/2}t^{1/2}}\right)$$

Proof. This is a combination of Lemmas 46 and 47, with the results merged using Lemma 38.

As described above, we will group the truncated CRVs into several bins. We first apply Lemma 46 to each of the non-empty bins B_l for $l > 0$. This will give us a sum of many discretized Gaussians. If applicable, we apply Lemma 47 to B_0 to obtain another discretized Gaussian and a set $\bar{S}$ of $\leq kt$ truncated CRVs. By applying Lemma 38, we can "merge" the sum of many discretized Gaussians into a single discretized Gaussian. By triangle inequality, the error occured in the theorem statement is the sum of all of these approximations.

We start by analyzing the cost of applying Lemma 46. Fix $\gamma = 6 + \delta$ for some constant $\delta > 0$. Let the set of N non-empty bins be $\mathcal{X}$. Then the sum of the errors incurred by all N applications of Lemma 46 is at most

$$\begin{aligned}
\sum_{l \in \mathcal{X}} O\left(\frac{k^{3/2}\log^{2/3}(2^k(l+1)^{(6+\delta)}t)}{l^{(6+\delta)/6}t^{1/6}c^{1/6}}\right) &\leq \sum_{l=1}^{\infty} O\left(\frac{k^{3/2}\log^{2/3}(2^k(l+1)^{(6+\delta)}t)}{l^{(6+\delta)/6}t^{1/6}c^{1/6}}\right) \\
&\leq \sum_{l=1}^{\infty} O\left(\frac{k^{13/6}\log^{2/3} l \log^{2/3} t}{l^{(6+\delta)/6}t^{1/6}c^{1/6}}\right) \\
&\leq \frac{k^{13/6}\log^{2/3} t}{c^{1/6}t^{1/6}} \sum_{l=1}^{\infty} O\left(\frac{\log^{2/3} l}{l^{(6+\delta)/6}}\right) \\
&\leq \frac{k^{13/6}\log^{2/3} t}{c^{1/6}t^{1/6}} \sum_{l=1}^{\infty} O\left(\frac{1}{l^{(6+\delta')/6}}\right) \\
&\leq O\left(\frac{k^{13/6}\log^{2/3} t}{c^{1/6}t^{1/6}}\right)
\end{aligned}$$

for any constant $0 < \delta' < \delta$. The final inequality is because the series $\sum_{n=1}^{\infty} n^{-c}$ converges for any $c > 1$.

The cost of applying Lemma 47 is analyzed similarly,

$$\frac{8.646k^{3/2}\log^{2/3}(2^k t)}{t^{1/6}c^{1/6}} \leq O\left(\frac{k^{13/6}\log^{2/3} t}{c^{1/6}t^{1/6}}\right)$$

Finally, we analyze the cost of merging the $N+1$ Gaussians into one. We will analyze this by considering the following process: we maintain a discretized Gaussian, which we will name the candidate. The candidate is initialized to be the Gaussian generated from the highest numbered non-empty bucket. At every time step, we update the candidate to be the result of merging itself with the Gaussian from the highest numbered non-empty bucket which has not yet been merged. We continue until the Gaussian from every non-empty bucket has been merged with the candidate.

By Lemma 38, the cost of merging two Gaussians is at most $O\left(\frac{k}{\sigma}\right)$, where σ^2 is the minimum variance of either Gaussian in any direction where either has a non-zero variance. Recall from the proof of Lemma 46 that the variance of the Gaussian from B_l is at least $l^\gamma t \frac{c}{k}$ in every direction of non-zero variance. Since we are considering the bins in decreasing order and merging two Gaussians only increases the variance, when merging the candidate with bucket l, the maximum cost we can incur is $\left(\frac{k^{3/2}}{l^{\gamma/2}c^{1/2}t^{1/2}}\right)$. Summing over all bins in $\mathcal{X}$,

$$\begin{aligned}\sum_{l\in\mathcal{X}} O\left(\frac{k^{3/2}}{l^{\gamma/2}c^{1/2}t^{1/2}}\right) &\leq \frac{k^{3/2}}{c^{1/2}t^{1/2}}\sum_{l=1}^{\infty} O\left(\frac{1}{l^{(6+\delta)/2}}\right)\\ &\leq O\left(\frac{k^{3/2}}{c^{1/2}t^{1/2}}\right)\end{aligned}$$

where the second inequality is because the series $\sum_{n=1}^{\infty} n^{-c}$ converges for any $c > 1$. We note that the cost incurred by merging the Gaussian obtained from B_0 is upper bounded by the term in this sum corresponding to $l = 1$, so it does not affect our bound.

By adding the error terms obtained from each of the approximations, we obtain the claimed bound. □

We make the following observation, which allows us to convert back from relating GMDs and discretized Gaussians to relating PMDs and detruncated discretized Gaussians:

Observation 49. *Let G^ρ be a GMD of size n and dimension k, and M^π be the PMD of size n and dimension k such that the submatrix of π which excludes the ith column is ρ. Then*

$$d_{\mathrm{TV}}\left(G^\rho, \lfloor \mathcal{N}(\mu, \Sigma) \rceil\right) = d_{\mathrm{TV}}\left(M^\pi, \lfloor \mathcal{N}(\mu, \Sigma, n, i) \rceil\right)$$

Finally, we conclude with our main theorem of the section by combining our results for the sets S_1 through S_k.

Theorem 8. *Every Poisson Multinomial distribution of size n and dimension k is ε-close to a sum of k detruncated discretized Gaussians and a Poisson Multinomial distribution of size $\leq k^2 t$ and dimension k.*

Proof. First, we justify the structure of the approximation, and then show that it can be ε-close by carefully choosing the parameters c and t. We start by applying Theorem 6 to obtain a PMD $M^{\hat{\pi}}$ such that $\hat{\pi}(i,j) \notin (0, c)$ for all i, j. Partition the component CRVs into k sets $S_1, \ldots, S_k$, where the ith CRV is placed in the lth set if $l = \arg\max_j \hat{\pi}(i,j)$ (with ties broken lexicographically). Since index l is the heaviest, every CRV i in S_l has $\hat{\rho}(i,l) \geq \frac{1}{k}$. We convert the PMD induced by each S_l to a GMD by dropping the lth column. Applying Lemma 48 to each set, applying Observation 49, and summing the results from all sets gives the claimed structure.

Now, we must choose the parameters c and t in order to make the resulting distribution be ε-close to the original. Applying Theorem 6 introduces a cost of $O\left(c^{1/2} k^{5/2} \log^{1/2}\left(\frac{1}{ck}\right)\right)$ in our approximation. Choosing $c = \left(\frac{\varepsilon^2}{k^5}\right)^{1+\delta_c}$ for any constant $\delta_c > 0$ makes this cost become $O(\varepsilon)$, for ε sufficiently small. We apply Lemma 48 k times (once to each set S_l), so the total cost introduced here is $O\left(\frac{k^{19/6}\log^{2/3} t}{c^{1/6} t^{1/6}} + \frac{k^{5/2}}{c^{1/2} t^{1/2}}\right)$. Choosing $t = \left(\frac{k^{19}}{c\varepsilon^6}\right)^{1+\delta_t}$ makes this cost $O(\varepsilon)$ as well, for ε sufficiently small. By triangle inequality, this makes the resulting approximation $O(\varepsilon)$ close to the original distribution. □

4.5 A Sparse Cover for PMDs

Proof of Theorem 5: Our strategy will be as follows: Theorem 8 implies that the original distribution is $O(\varepsilon)$ close to a particular class of distributions. We generate an $O(\varepsilon)$-cover for this generated class. By triangle inequality, this is an $O(\varepsilon)$-cover for PMDs. In order to generate a cover, we will use a technique known as "gridding" (see Section 1.2.1 for more details). We will generate a set of values for each parameter, and take the Cartesian product of these sets. Our guarantee is that the resulting set will contain at least one set of parameters defining a distribution which is $O(\varepsilon)$-close to the PMD. We will assume c, t are such that the approximation guaranteed by Theorem 8 is $O(\varepsilon)$, and we will substitute in the values from the proof of Theorem 8 at the end.

First, we note that we can naively grid over the set of PMDs of size k^2t and dimension k. We note that if two CRVs have parameters which are within $\pm\frac{\varepsilon}{k}$ of each other, then their total variation distance is at most ε. Similarly, by triangle inequality, two PMDs of size k^2t and dimension k with parameters within $\pm\frac{\varepsilon}{k^3t}$ of each other have a total variation distance at most ε. By taking an additive grid of granularity $\frac{\varepsilon}{k^3t}$ over all k^2t parameters, we can generate an $O(\varepsilon)$-cover for PMDs of size k^2t and dimension k with $O\left(\frac{k^3t}{\varepsilon}\right)^{k^2t}$ candidates.

Next, we wish to cover the detruncated discretized Gaussians. There are k Gaussians, and each has a mean with $k-1$ parameters, a covariance matrix with $(k-1)^2$ parameters, and a single size parameter. We describe how to generate a $O\left(\frac{\varepsilon}{k}\right)$-cover for a single one of these Gaussians. By taking the Cartesian product of the cover for each of the Gaussians and applying the triangle inequality, we generate a $O(\varepsilon)$-cover for the collection of Gaussians at the cost of a factor of k in the exponent of the cover size.

First, we examine the size parameter. Since the size parameter is an integer between 0 and n, we can simply try them all, giving us a factor of $O(n)$ in the size of our cover.

Covering the mean and covariance matrix takes a bit more care, and we use the following Proposition from [53]:

Proposition 50 (Proposition 32 in [53]). *Given two k-dimensional Gaussians $\mathcal{N}_1 = \mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}_2 = \mathcal{N}(\mu_2, \Sigma_2)$ such that for all $i, j \in [k]$, $|\Sigma_1(i,j) - \Sigma_2(i,j)| \leq \alpha$, and the minimum eigenvalue of Σ_1 is at least σ^2,*

$$d_{\mathrm{TV}}\left(\mathcal{N}_1, \mathcal{N}_2\right) \leq \frac{\|\mu_1 - \mu_2\|_2}{\sqrt{2\pi\sigma^2}} + \frac{k\alpha}{\sqrt{2\pi e}(\sigma^2 - \alpha)}.$$

For the remainder of this proof, we let $\mathcal{N}_1$ be the Gaussian produced via Lemma 48, and we will construct a $\mathcal{N}_2$ which is close to it. We note that, by the construction described in Lemma 48 (which is used in Theorem 8), after dropping all zero rows and columns from the covariance matrix for one of our Gaussians, the minimum eigenvalue of the resulting matrix will be $\geq \frac{ct}{k}$. Since our gridding strategy will try setting every possible subset of the parameters to 0, there will exist an element in our cover which matches these 0's exactly, and we can focus on the sub-problem which excludes these coordinates which are constantly 0. Thus, we assume that $\sigma^2 \geq \frac{ct}{k}$.

We examine the first term of the bound in Proposition 50. If $\|\mu_1 - \mu_2\|_2 \leq \frac{\varepsilon\sqrt{ct}}{k^{3/2}} := \beta$, then this term is $O\left(\frac{\varepsilon}{k}\right)$. Note that $\mu_1 \in [0,n]^{k-1}$, since each coordinate is the sum of at most n parameters which are at most 1. We can create a β-cover of size $O\left(\frac{n\sqrt{k}}{\beta}\right)^{O(k)}$ for this space, with respect to the ℓ_2 distance. To see this, consider covering the space with $(k-1)$-cubes of side length $\frac{\beta}{\sqrt{k-1}}$. Any two points within the same cube are at ℓ_2 distance at most β. Our cover is defined as taking the same vertex from each of these cubes. The volume of $[0,n]^{k-1}$ is n^{k-1}, and the volume of each cube is $\left(\frac{\beta}{\sqrt{k-1}}\right)^{k-1}$, so the total number of points needed is $O\left(\frac{n\sqrt{k}}{\beta}\right)^{O(k)}$. Substituting in the value of β shows that a set of size $O\left(\frac{nk^2}{\varepsilon\sqrt{ct}}\right)^{O(k)}$ suffices to cover the mean of the Gaussian to a sufficient accuracy.

Next, we examine the second term in Proposition 50. Taking $\alpha \leq \frac{\varepsilon ct}{k(k^2+\varepsilon)} = O\left(\frac{\varepsilon ct}{k^3}\right)$ sets this term to be $O\left(\frac{\varepsilon}{k}\right)$. Since there are $O(k^2)$ parameters in the covariance matrix, and each is in $[0,n]$, gridding at this granularity generates a set of size $O\left(\frac{nk^3}{\varepsilon ct}\right)^{O(k^2)}$. Combining with the gridding for the mean and size parameter, a $O\left(\frac{\varepsilon}{k}\right)$-cover for one of the Gaussians is of size

$$O\left(n\left(\frac{nk^2}{\varepsilon\sqrt{ct}}\right)^{O(k)}\left(\frac{nk^3}{\varepsilon ct}\right)^{O(k^2)}\right)=\left(\frac{nk}{\varepsilon ct}\right)^{O(k^2)}.$$

As mentioned before, this implies a $O(\varepsilon)$ cover for the set of k Gaussians of size

$$\left(\frac{nk}{\varepsilon ct}\right)^{O(k^3)}.$$

Combining the cover for the Gaussian component and the sparse PMD gives us a cover of size

$$O\left(\left(\frac{nk}{\varepsilon ct}\right)^{O(k^3)}\left(\frac{k^3t}{\varepsilon}\right)^{k^2t}\right).$$

Substituting in the values of c and t provided in the proof of Theorem 8 gives us a cover of size

$$n^{O(k^3)}\left(\frac{k}{\varepsilon}\right)^{O\left(\frac{k^{26+\delta_1}}{\varepsilon^{8+\delta_2}}\right)},$$

for any constants $\delta_1, \delta_2 > 0$, which satisfies the statement of the theorem. □

4.6 Open Problems

The results presented here are just the beginning – we intend to further investigate properties of Poisson Multinomial distributions. Here are some potential directions for study:

- **Can we reduce the size of the cover?** The current cover size is $n^{O(k^3)}\left(\frac{k}{\varepsilon}\right)^{\text{poly}\left(k,\frac{1}{\varepsilon}\right)}$. We would ideally like to reduce the size to a form which is similar to [25], $n^k\left(\frac{k}{\varepsilon}\right)^{\text{poly}\left(k,\log\frac{1}{\varepsilon}\right)}$. Some preliminary investigation has shown that it may be possible to reduce the poly$(1/\varepsilon)$ to poly$(\log(1/\varepsilon))$, but at the price of an exponential in k blowup.

- **Can we improve the structure of the cover?** The structure of the cover is currently the sum of k discretized detruncated Gaussians and a sparse PMD, but we believe it is possible to merge the k Gaussians into a single Gaussian.

While we can merge the Gaussians using the same method as before, it is not clear how to detruncate the Gaussian afterward, since we must now detruncate in k dimensions instead of just one.

- **Can we efficiently learn Poisson Multinomial Distributions?** In the same vein as [20], we would like to obtain efficient learning algorithms. While we provide an algorithm with sample complexity poly $\left(\log n, k, \frac{1}{\varepsilon}\right)$, the results in [20] suggest that it may be possible to obtain an algorithm with poly $\left(k, \frac{1}{\varepsilon}\right)$ sample complexity and poly $\left(\log n, k, \frac{1}{\varepsilon}\right)$ time complexity.

Appendix A

Robust Estimation of Scale from a Mixture of Gaussians

In this chapter, we examine the following statistic:

Given some point $x \in \mathbb{R}$ and n IID random variables $X_1, \ldots, X_n$, what is the minimum distance between x and any X_i?

We give an interval in which this statistic is likely to fall (Proposition 51), and examine its robustness when sampling from distributions which are statistically close to the distribution under consideration (Proposition 53). We then apply these results to mixtures of Gaussians (Proposition 54 and Lemma 28).

Proposition 51. *Suppose we have n IID random variables $X_1, \ldots, X_n \sim X$, some $x \in \mathbb{R}$, and $y = F_X(x)$. Let I_N be the interval $[F_X^{-1}(y - \frac{c_1}{n}), F_X^{-1}(y + \frac{c_1}{n})]$ and I_F be the interval $[F_X^{-1}(y - \frac{c_2}{n}), F_X^{-1}(y + \frac{c_2}{n})]$ for some constants $0 < c_1 < c_2 \leq n$, and $I = I_F \backslash I_N$. Let $j = \arg\min_i |X_i - x|$. Then $\Pr[X_j \in I] \geq \frac{9}{10}$ for all $n > 0$.*

Proof. We show that $\Pr[X_j \notin I] \leq \frac{1}{10}$ by showing that the following two bad events are unlikely:

1. We have a sample which is too close to x

2. All our samples are too far from x

Showing these events occur with low probability and combining with the union bound gives the desired result.

Let Y be the number of samples at distance $< \frac{c_1}{n}$ in distance in the CDF, i.e., $Y = |\{i \mid |F_X^{-1}(X_i) - y| < \frac{c_1}{n}\}|$. By linearity of expectation, $E[Y] = 2c_1$. By Markov's inequality, $\Pr(Y > 0) < 2c_1$. This allows us to upper bound the probability that one of our samples is too close to x.

Let Z be the number of samples at distance $< \frac{c_2}{n}$ in distance in the CDF, i.e., $Z = |\{i \mid |F_X^{-1}(X_i) - y| < \frac{c_2}{n}\}|$, and let Z_i be an indicator random variable which indicates this property for X_i. We use the second moment principle,

$$\Pr(Z > 0) \geq \frac{E[Z]^2}{E[Z^2]}$$

By linearity of expectation, $E[Z]^2 = 4c_2^2$.

$$\begin{aligned} E[Z^2] &= \sum_i E[Z_i^2] + \sum_i \sum_{j \neq i} E[Z_i Z_j] \\ &= 2c_2 + n(n-1)\left(\frac{4c_2^2}{n^2}\right) \\ &\geq 2c_2 + 4c_2^2 \end{aligned}$$

And thus, $\Pr(Z = 0) \leq \frac{1}{2c_2+1}$. This allows us to upper bound the probability that all of our samples are too far from x.

Setting $c_1 = \frac{1}{40}$ and $c_2 = \frac{19}{2}$ gives probability $< \frac{1}{20}$ for each of the bad events, resulting in a probability $< \frac{1}{10}$ of either bad event by the union bound, and thus the desired result. □

We will need the following property of Kolmogorov distance, which states that probability mass within every interval is approximately preserved:

Proposition 52. *If $d_{\mathrm{K}}(f_X, f_Y) \leq \varepsilon$, then for all intervals $I \subseteq \mathbb{R}$, $|f_X(I) - f_Y(I)| \leq 2\varepsilon$.*

Proof. For an interval $I = [a, b]$, we can rewrite the property as

$$\begin{aligned} |f_X(I) - f_Y(I)| &= |(F_X(b) - F_X(a)) - (F_Y(b) - F_Y(a))| \\ &\leq |F_X(b) - F_Y(b)| + |F_X(a) - F_Y(a)| \\ &\leq 2\varepsilon \end{aligned}$$

as desired, where the first inequality is the triangle inequality and the second inequality is due to the bound on Kolmogorov distance. □

The next proposition says that if we instead draw samples from a distribution which is close in total variation distance, the same property approximately holds with respect to the original distribution.

Proposition 53. *Suppose we have n IID random variables $\hat{X}_1, \ldots, \hat{X}_n \sim \hat{X}$ where $d_{\mathrm{K}}(f_X, f_{\hat{X}}) \leq \delta$, some $x \in \mathbb{R}$, and $y = F_X(x)$. Let I_N be the interval $[F_X^{-1}(y - \frac{c_1}{n} + \delta), F_X^{-1}(y + \frac{c_1}{n} - \delta)]$ and I_F be the interval $[F_X^{-1}(y - \frac{c_2}{n} - \delta), F_X^{-1}(y + \frac{c_2}{n} + \delta)]$ for some constants $0 < c_1 < c_2 \leq n$, and $I = I_F \backslash I_N$. Let $j = \arg\min_i |X_i - a|$. Then $\Pr[X_j \in I] \geq \frac{9}{10}$ for all $n > 0$.*

Proof. First, examine interval I_N. This interval contains $\frac{2c_1}{n} - 2\delta$ probability measure of the distribution F_X. By Proposition 52, $|F_X(I_N) - F_{\hat{X}}(I_N)| \leq 2\delta$, so $F_{\hat{X}}(I_N) \leq \frac{2c_1}{n}$. One can repeat this argument to show that the amount of measure contained by $F_{\hat{X}}$ in $[F_X^{-1}(y - \frac{c_2}{n} - \delta), F_X^{-1}(y + \frac{c_2}{n} + \delta)]$ is $\geq \frac{2c_2}{n}$.

As established through the proof of Proposition 51, with probability $\geq \frac{9}{10}$, there will be no samples in a window containing probability measure $\frac{2c_1}{n}$, but there will be at least one sample in a window containing probability measure $\frac{2c_2}{n}$. Applying the same argument to these intervals, we can arrive at the desired result. □

We examine this statistic for some mixture of k Gaussians with PDF f around the point corresponding to the mean of the component with the minimum value of $\frac{\sigma_i}{w_i}$. Initially, we assume that we know this location exactly and that we are taking samples according to f exactly.

Proposition 54. *Consider a mixture of k Gaussians with PDF f, components $\mathcal{N}(\mu_1, \sigma_1^2), \ldots, \mathcal{N}(\mu_k, \sigma_k^2)$ and weights $w_1, \ldots, w_k$. Let $j = \arg\min_i \frac{\sigma_i}{w_i}$. If we take n samples $X_1, \ldots, X_n$ from the mixture (where $n > \frac{3\sqrt{\pi}c_2}{2w_j}$), then $\min_i |X_i - \mu_j| \in \left[\frac{\sqrt{2\pi}c_1\sigma_j}{kw_jn}, \frac{3\sqrt{2\pi}c_2\sigma_j}{2w_jn}\right]$ with probability $\geq \frac{9}{10}$, where c_1 and c_2 are as defined in Proposition 51.*

Proof. We examine the CDF of the mixture around μ_i. Using Proposition 1 (and symmetry of a Gaussian about its mean), it is sufficient to show that

$$\left[\mu_i + \frac{\sqrt{2\pi}c_1\sigma_i}{kw_in}, \mu_i + \frac{3\sqrt{2\pi}c_2\sigma_i}{2w_in}\right] \supseteq \left[F^{-1}(F(\mu_i) + \frac{c_1}{n}), F^{-1}(F(\mu_i) + \frac{c_2}{n})\right],$$

where F is the CDF of the mixture. We show that each endpoint of the latter interval bounds the corresponding endpoint of the former interval.

First, we show $\frac{c_1}{n} \geq F\left(\mu_i + \frac{\sqrt{2\pi}c_1\sigma_i}{kw_in}\right) - F(\mu_i)$. Let $I = \left[\mu_i, \mu_i + \frac{\sqrt{2\pi}c_1\sigma_i}{kw_in}\right]$, f be the PDF of the mixture, and f_i be the PDF of component i of the mixture. The right-hand side of the inequality we wish to prove is equal to

$$\begin{aligned}\int_I f(x)\,\mathrm{d}x &= \int_I \sum_{j=1}^{k} w_j f_j(x)\,\mathrm{d}x \\ &\leq \int_I \sum_{j=1}^{k} w_j \frac{1}{\sigma_j\sqrt{2\pi}}\,\mathrm{d}x \\ &\leq \int_I \frac{kw_i}{\sigma_i\sqrt{2\pi}}\,\mathrm{d}x \\ &= \frac{c_1}{n}\end{aligned}$$

where the first inequality is since the maximum of the PDF of a Gaussian is $\frac{1}{\sigma\sqrt{2\pi}}$, and the second is since $\frac{\sigma_j}{w_j} \leq \frac{\sigma_i}{w_i}$ for all j.

Next, we show $\frac{c_2}{n} \leq F\left(\mu_i + \frac{3\sqrt{2\pi}c_2\sigma_i}{2w_in}\right) - F(\mu_i)$. We note that the right-hand side is the probability mass contained in the interval - a lower bound for this quantity is the probability mass contributed by the particular Guassian we are examining, which is $\frac{w_i}{2}\text{erf}\Big(\frac{3\sqrt{\pi}c_2}{2w_in}\Big)$. Taking the Taylor expansion of the error function gives

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\left(x - \frac{x^3}{3} + O(x^5)\right) \geq \frac{2}{\sqrt{\pi}}\left(\frac{2}{3}x\right)$$

if $x < 1$. Applying this here, we can lower bound the contributed probability mass by $\frac{w_i}{2}\frac{2}{\sqrt{\pi}}\frac{2}{3}\frac{3\sqrt{\pi}c_2}{2w_in} = \frac{c_2}{n}$, as desired. □

Finally, we deal with uncertainties in parameters and apply the robustness properties of Proposition 53 in the following lemma:

Proof of Lemma 28: We analyze the effect of each uncertainty:

- First, we consider the effect of sampling from $\hat{f}$, which is δ-close to f. By using Proposition 53, we know that the nearest sample to μ_j will be at CDF distance between $\frac{c_1}{n} - \delta \geq \frac{c_1}{2n}$ and $\frac{c_2}{n} + \delta \leq \frac{3c_2}{2n}$. We can then repeat the proof of Proposition 54 with c_1 replaced by $\frac{c_1}{2}$ and c_2 replaced by $\frac{3c_2}{2}$. This gives us that $\min_i |X_i - \mu_j| \in \left[\frac{\sqrt{\pi}c_1}{\sqrt{2}kw_jn}\sigma_j, \frac{9\sqrt{\pi}c_2}{2\sqrt{2}w_jn}\sigma_j\right]$ (where $n \geq \frac{9\sqrt{\pi}c_2}{4w_j}$) with probability $\geq \frac{9}{10}$.

- Next, substituting in the bounds $\frac{1}{2}\hat{w}_j \leq w_j \leq 2\hat{w}_j$, we get $\min_i |X_i - \mu_j| \in \left[\frac{\sqrt{\pi}c_1}{2\sqrt{2}k\hat{w}_jn}\sigma_j, \frac{9\sqrt{\pi}c_2}{\sqrt{2}\hat{w}_jn}\sigma_j\right]$ (where $n \geq \frac{9\sqrt{\pi}c_2}{2\hat{w}_j}$) with probability $\geq \frac{9}{10}$.

- We use $n = \frac{9\sqrt{\pi}c_2}{2\hat{w}_j}$ samples to obtain: $\min_i |X_i - \mu_j| \in \left[\frac{c_3}{k}\sigma_j, \sqrt{2}\sigma_j\right]$ with probability $\geq \frac{9}{10}$.

- Finally, applying $|\hat{\mu}_j - \mu_j| \leq \frac{c_3}{2k}\sigma_j$ gives the lemma statement.

□

Bibliography

[1] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theerta Suresh. Near-optimal-sample estimators for spherical Gaussian mixtures. *Online manuscript*, 2014.

[2] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Sorting with adversarial comparators and application to density estimation. In *Proceedings of the 2014 IEEE International Symposium on Information Theory*, ISIT '14, Washington, DC, USA, 2014. IEEE Computer Society.

[3] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, COLT '05, pages 458–469. Springer, 2005.

[4] Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James R. Voss. The more, the merrier: the blessing of dimensionality for learning large Gaussian mixtures. In *Proceedings of the 27th Annual Conference on Learning Theory*, COLT '14, pages 1135–1164, 2014.

[5] Sanjeev Arora and Ravi Kannan. Learning mixtures of arbitrary Gaussians. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, STOC '01, pages 247–257, New York, NY, USA, 2001. ACM.

[6] Andrew D. Barbour, Lars Holst, and Svante Janson. *Poisson Approximation.* Oxford University Press, New York, 1992.

[7] Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 103–112, Washington, DC, USA, 2010. IEEE Computer Society.

[8] Lucien Birgé. Estimating a density under order restrictions: Nonasymptotic minimax risk. *The Annals of Statistics*, 15(3):995–1012, September 1987.

[9] G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, June 1958.

[10] Spencer Charles Brubaker and Santosh Vempala. Isotropic PCA and affine-invariant clustering. In *Proceedings of the 49th Annual IEEE Symposium on*

Foundations of Computer Science, FOCS '08, pages 551–560, Washington, DC, USA, 2008. IEEE Computer Society.

[11] Clément Canonne, Dana Ron, and Rocco A Servedio. Testing probability distributions using conditional samples. *Online manuscript*, 2012.

[12] Clément L Canonne, Dana Ron, and Rocco A Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1174–1192, Philadelphia, PA, USA, 2014. SIAM.

[13] Clément L. Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *Proceedings of the 41st international conference on Automata, Languages, and Programming-Volume Part I*, pages 283–295, 2014.

[14] Siu On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Learning mixtures of structured distributions over discrete domains. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1380–1394, Philadelphia, PA, USA, 2013. SIAM.

[15] Siu On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14, New York, NY, USA, 2014. ACM.

[16] Sanjoy Dasgupta. Learning mixtures of Gaussians. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '99, pages 634–644, Washington, DC, USA, 1999. IEEE Computer Society.

[17] Constantinos Daskalakis. An efficient PTAS for two-strategy anonymous games. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, WINE '08, pages 186–197, Berlin, Heidelberg, 2008. Springer-Verlag.

[18] Constantinos Daskalakis, Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Li Yang Tan. Learning sums of independent integer random variables. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '13, pages 217–226, Washington, DC, USA, 2013. IEEE Computer Society.

[19] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A Servedio. Learning k-modal distributions via testing. In *Proceedings of the 23th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1371–1385, Philadelphia, PA, USA, 2012. SIAM.

[20] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. Learning Poisson binomial distributions. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing*, STOC '12, pages 709–728, New York, NY, USA, 2012. ACM.

[21] Constantinos Daskalakis, Ilias Diakonikolas, Rocco A Servedio, Gregory Valiant, and Paul Valiant. Testing k-modal distributions: Optimal algorithms via reductions. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1833–1852, Philadelphia, PA, USA, 2013. SIAM.

[22] Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of Gaussians. In *Proceedings of the 27th Annual Conference on Learning Theory*, COLT '14, pages 1183–1213, 2014.

[23] Constantinos Daskalakis and Christos Papadimitriou. Computing equilibria in anonymous games. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 83–93, Washington, DC, USA, 2007. IEEE Computer Society.

[24] Constantinos Daskalakis and Christos Papadimitriou. Discretized multinomial distributions and Nash equilibria in anonymous games. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 25–34, Washington, DC, USA, 2008. IEEE Computer Society.

[25] Constantinos Daskalakis and Christos Papadimitriou. Sparse covers for sums of indicators. *Online manuscript*, 2013.

[26] Constantinos Daskalakis and Christos H. Papadimitriou. On oblivious PTAS's for Nash equilibrium. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing*, STOC '09, pages 75–84, New York, NY, USA, 2009. ACM.

[27] Constantinos Daskalakis and Christos H. Papadimitriou. Approximate Nash equilibria in anonymous games. *Journal of Economic Theory*, 2014.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[29] Luc Devroye and Gábor Lugosi. A universally acceptable smoothing factor for kernel density estimation. *The Annals of Statistics*, 24:2499–2512, 1996.

[30] Luc Devroye and Gábor Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and yatracos classes. *The Annals of Statistics*, 25:2626–2637, 1997.

[31] Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation.* Springer, 2001.

[32] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, 09 1956.

[33] Jon Feldman, Ryan O'Donnell, and Rocco A. Servedio. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *Proceedings of the 19th Annual Conference on Learning Theory*, COLT '06, pages 20–34, Berlin, Heidelberg, 2006. Springer-Verlag.

[34] Jon Feldman, Ryan O'Donnell, and Rocco A. Servedio. Learning mixtures of product distributions over discrete domains. *SIAM Journal on Computing*, 37(5):1536–1564, 2008.

[35] Ronald Aylmer Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pages 309–368, 1922.

[36] S. Gershgorin. Über die abgrenzung der Eigenwerte einer matrix. *Izv. Akad. Nauk. SSSR Ser. Mat.*, 1:749–754, 1931.

[37] Alison L. Gibbs and Francis E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, December 2002.

[38] Moritz Hardt and Eric Price. Sharp bounds for learning a mixture of two Gaussians. *Online manuscript*, 2014.

[39] Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 11–20, New York, NY, USA, 2013. ACM.

[40] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two Gaussians. In *Proceedings of the 42nd Annual ACM Symposium on the Theory of Computing*, STOC '10, pages 553–562, New York, NY, USA, 2010. ACM.

[41] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, STOC '94, pages 273–282, New York, NY, USA, 1994. ACM.

[42] AN Kolmogorov. Certain asymptotic characteristics of completely bounded metric spaces. *Doklady Akademii Nauk SSSR*, 108(3):385–388, 1956.

[43] Andrei Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

[44] P. Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 07 1990.

[45] Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 93–102, Washington, DC, USA, 2010. IEEE Computer Society.

[46] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, pages 71–110, 1894.

[47] Karl Pearson. Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. A*, pages 343–414, 1895.

[48] Leo Reyzin. Extractors and the leftover hash lemma. `http://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-1.pdf`, March 2011. Lecture notes.

[49] B. Roos. Metric multivariate Poisson approximation of the generalized multinomial distribution. *Teor. Veroyatnost. i Primenen.*, 43(2):404–413, 1998.

[50] B. Roos. Multinomial and Krawtchouk approximations to the generalized multinomial distribution. *Theory of Probability & Its Applications*, 46(1):103–117, 2002.

[51] Bero Roos. Poisson approximation of multivariate Poisson mixtures. *Journal of Applied Probability*, 40(2):376–390, 06 2003.

[52] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 09 1956.

[53] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:179, 2010.

[54] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixtures of distributions. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '02, pages 113–123, Washington, DC, USA, 2002. IEEE Computer Society.

[55] Yannis G. Yatracos. Rates of convergence of minimum distance estimators and kolmogorov's entropy. *The Annals of Statistics*, 13(2):768–774, 1985.