2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Noisy population recovery in polynomial time

Anindya De
Department of EECS
Northwestern University
Evanston, IL
anindya@eecs.northwestern.edu

Michael Saks and Sijian Tang
Department of Mathematics
Rutgers University
Piscataway, NJ
saks, st509@math.rutgers.edu

***Abstract*—In the noisy population recovery problem of Dvir et al. [6], the goal is to learn an unknown distribution f on binary strings of length n from noisy samples. A noisy sample with parameter $\mu \in [0,1]$ is generated by selecting a sample from f, and independently flipping each coordinate of the sample with probability $(1-\mu)/2$. We assume an upper bound k on the size of the support of the distribution, and the goal is to estimate the probability of any string to within some given error ε. It is known that the algorithmic complexity and sample complexity of this problem are polynomially related to each other.**

We describe an algorithm that for each $\mu > 0$, provides the desired estimate of the distribution in time bounded by a polynomial in k, n and $1/\varepsilon$ improving upon the previous best result of $\mathsf{poly}(k^{\log\log k}, n, 1/\varepsilon)$ due to Lovett and Zhang [9].

Our proof combines ideas from [9] with a *noise attenuated* version of Möbius inversion. The latter crucially uses the *robust local inverse* construction of Moitra and Saks [11].

***Keywords*-Population recovery; Reverse Bonami-Beckner inequality; Fourier transform;**

I. Introduction

A. Background

The population recovery problem is a problem in noisy unsupervised learning which has received recent attention [6], [14], [11], [9]. In this problem, there is an unknown distribution f over binary strings of length n, and a noise parameter $0 < \mu < 1$. A *noisy sample* from f is generated as follows:

- Choose a string x according to f.
- Choose a binary string N according to the distribution η_μ in which each coordinate is independently set to 1 with probability $(1-\mu)/2$.
- Output $x \oplus N$, where $\oplus$ denotes bitwise sum modulo 2.

Given access to these noisy samples and error parameter ε, the learner must output an estimate of the function f (denoted by $\tilde{f}$), which it does by specifying the set S of strings for which the estimate is nonzero, and an estimate $\tilde{f}(x)$ for each $x \in S$. The algorithm is said to succeed provided that $|\tilde{f}(x) - f(x)| \leq \varepsilon$ for all $x \in \{0,1\}^n$. If the algorithm succeeds with probability at least $1-\delta$ we say that it is an (ε, δ)-estimation algorithm for f.

For $\mu = 1$, there is no noise and the problem is easy to solve, whereas for $\mu = 0$, the distribution f cannot be recovered with any number of samples. As μ becomes smaller, the learning problem becomes harder.

There is an alternate (and easier) model called the *lossy model* in which each sample presented to the learner is generated by selecting x from f and then replacing each entry by a '?' independently with probability $1-\mu$. This model is easier since the learner can simulate samples from the noisy model given samples from the lossy model by replacing each '?' by a random bit.

The complexity of an algorithm for this problem depends on four parameters, namely, μ, n, ε, $\frac{1}{\delta}$. As usual, the value of δ is not very significant for the complexity; if we have an algorithm that works for $\delta = 1/4$, we can improve it to an arbitrary δ by repeating the algorithm $\log(1/\delta)$ times and assign to each $x \in \{0,1\}^n$ the median of the estimates of $f(x)$ from the different runs. We generally think of μ and δ as constants and focus on expressing the running time as a function of n and $1/\varepsilon$.

The problem (in both the noisy and lossy versions) was introduced by Dvir et al. [6] who related it to the problem of learning DNF from restrictions. For the lossy model, Dvir et al. [6] gave an algorithm with run time polynomial in n and $1/\varepsilon$ provided that $\mu \gtrsim 0.365$. Their analysis was improved by Batman, et al. [1] who showed that the same algorithm is polynomial time for any $\mu > 1 - 1/\sqrt{2} \approx 0.293$. Subsequently, Moitra and Saks [11] gave a polynomial time algorithm for population recovery in the lossy model for any $\mu > 0$.

For the noisy model, algorithms are known only under the following additional assumption:

> **Bounded Support Assumption** $BSA(k)$: $f(x) \neq 0$ for at most k strings x.

Under $BSA(k)$, k becomes an additional parameter for the problem.

Dvir, et al. [6] showed that noisy population recovery under $BSA(k)$ can be reduced to the following seemingly easier problem of estimating the value of the distribution at the point $\mathbf{0} = 0^n$, when given as input a small subset that contains $\mathrm{supp}(f) \cup \{\mathbf{0}\}$.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.77

IEEE computer society

Noisy Population Point Recovery with Known Support (NPPRKS) We are given as input $X \subseteq [n]$ of size at most k that contains $\mathrm{supp}(f) \cup \{\mathbf{0}\}$, and an error parameter ε. Given access to samples from $T_\mu f$, output an estimate of $f(\mathbf{0})$ that has additive error at most ε.

They show that if NPPRKS can be solved with number of samples S in time T then the original problem can be solved in number of samples at most $S \cdot \mathsf{poly}(kn)$ and time $T \cdot \mathsf{poly}(kn)$.

Wigderson and Yehudayoff [14] developed a framework called "partial identification" and used this to give an algorithm for NPPRKS (and therefore also for NPR under $BSA(k)$) that runs in time $\mathsf{poly}(k^{\log k}, n, 1/\varepsilon)$ for any $\mu > 0$. They also showed that their framework cannot obtain algorithms running in time better than $\mathsf{poly}(k^{\log\log k})$.

Lovett and Zheng [9] gave an algorithm with a better time complexity of $\mathsf{poly}(k^{\log\log k}, n, 1/\varepsilon)$ for any $\mu > 0$. Interestingly, while their algorithm matches the lower bound in [14], their algorithm departs from the framework of [14], and thus is not subject to the same lower bound.

B. Our result

Here we show that for any $\mu > 0$, the time complexity of noisy population recovery problem is at most $\mathsf{poly}(k, n, \frac{1}{\varepsilon}, \log(\frac{1}{\delta}))$

Theorem I.1. *For any $\mu > 0$, there is an algorithm for NPPRKS (and therefore also for noisy population recovery under $BSA(k)$), with running time $n^{O(1)} \cdot \left(\frac{k}{\varepsilon}\right)^{O_\mu(1)} \cdot \log(1/\delta)$. Here $O_\mu(1) = \tilde{O}(1/\mu^4)$.*

Previously no polynomial time algorithm was known[1] for any $\mu < 1$.

C. A reverse Bonami-Beckner inequality

As with past results on the population recovery problem, our result has interesting functional analytic consequences. The process we are observing generates observations that are obtained by taking a sample from $\{0,1\}^n$ according to the probability distribution f and applying noise independently to each coordinate. Thus, the observed samples come from a distribution that is obtained from f by applying a linear operator T_μ, where for each $x \in \{0,1\}^n$:

$$(T_\mu f)(x) = \mathbb{E}_{N \sim \eta_\mu}[f(x \oplus N)].$$

The operator T_μ is usually referred to in the literature as the Bonami-Beckner operator [3], [2], [8], [12]. Intuitively, T_μ "smooths" f by by replacing the value of f at x by a weighted average of values of f near x. One way that this smoothing property is made precise is via *hypercontractive inequalities* [3], [2], [8], which have the following flavor: "(A higher order) norm of $T_\mu f$ can be upper bounded by (a lower order) norm of f", where the bounds are independent of the dimension (number of input variables) of the function.

Given such smoothing theorems, it is natural to try to establish reverse inequalities that assert that some norm of $T_\mu f$ is never too much smaller than (the same or different) norm of f. No such dimension independent inequality can hold for all functions, as is demonstrated by the signed parity function $(-1)^{\sum_i x_i}$, but such reverse inequalities are possible for restricted classes of functions. For example, Borell [4] proved a reverse Bonami-Beckner inequality which roughly states that for positive valued functions $f : \{0,1\}^n \to \mathbb{R}^+$, the norm of $T_\mu f$ can't be too small if the norm of f is large.

Lovett and Zhang [9] observed that the existence of fast algorithms for population recovery of functions satisfying $BSA(k)$ is equivalent to a reverse Bonami-Beckner type inequality for sparse functions. In particular, they showed that for $f : \{0,1\}^n \to \mathbb{R}$, if $\mathrm{supp}(f) = k$, then $\|T_\mu f\|_1 \geq k^{-O_\mu(\log\log k)}\|f\|_1$. The results of the present paper lead to the following improved reverse Bonami-Beckner inequality for sparse functions:

Theorem I.2. *Assume $f : \{0,1\}^n \to \mathbb{R}$ with $|\mathrm{supp}(f)| = k$. Then $\|T_\mu f\|_1 \geq k^{-O_\mu(1)}\|f\|_1$, where $O_\mu(1) = \tilde{O}(1/\mu^4)$.*

1) Related work:: In concurrent and independent work, Lovett and Zhang [10] considered the population recovery problem when the noise for each coordinate is independent but not necessarily identically distributed and further, the flipping probabilities are unknown. For this setting, Lovett and Zhang give an algorithm which outputs a sparse distribution g such that the statistical distance between noisy samples from f and g is guaranteed to be small. The running time of the algorithm is $\mathsf{poly}(n^{\log k}, (1/\varepsilon)^{\log^2 k}, k^{\log^3 k})$. Note that in contrast to the population recovery problem where the distance between f and g is guaranteed to be small, here we only have the weaker conclusion that the distance between the noisy samples from f and g is small. In the parlance of unsupervised learning, Lovett and Zhang do "proper density estimation" whereas the current (and previous) work on the population recovery problem does "parametric estimation". As Lovett and Zhang observe in their paper, such a relaxation is necessary once the flipping probabilities are unknown. This algorithm is obtained by an extension of the Wigderson-Yehudayoff approach [14]. To contrast it with the current paper, while their running time is worse (even compared to [14]) and the guarantee is weaker, their algorithm works in the more general setting when the flipping probabilities are allowed to be distinct and unknown.

[1] An earlier version of this paper [5] gave a polynomial time algorithm for $\mu > 0.555$; the theorem here holds for all $\mu > 0$.

II. Preliminaries

A. Fourier analysis of Boolean Functions

We begin with some definitions. We write $\mathbf{0}$ for the point 0^n in $\{0,1\}^n$. For $x \in \{0,1\}^n$, $|x|$ is the Hamming weight of x, which is equal to the number of 1's. For binary strings $x, y \in \{0,1\}^n$, $x \oplus y$ denotes the bitwise sum mod 2, and $d_H(x,y) = |x \oplus y|$ is the Hamming distance between x and y, which is the number of positions where x and y differ.

For a set S, 2^S denotes the set of subsets of S, $\binom{S}{r}$ denotes the set of subsets of size r and $\binom{S}{\leq r}$ denotes the set of subsets of size at most r. For sets S, T, $S \triangle T$ denotes their symmetric difference $(S - T) \cup (T - S)$.

We define the following sets of functions:

- $\mathcal{F} = \mathcal{F}_n$ is the space of real-valued functions on $\{0,1\}^n$
- $\mathcal{D} = \mathcal{D}_n$ is the set of nonnegative-valued $f \in \mathcal{F}$ satisfying $\sum_{x \in \{0,1\}^n} f(x) = 1$.
- For $X \subseteq \{0,1\}^n$ and $\eta \geq 0$, $\mathcal{G}_\eta(X)$ is the set of $f \in \mathcal{F}$ such that $f(\mathbf{0}) = 1$ and $|f(x)| \leq \eta$ for $x \in X - \{\mathbf{0}\}$.

We view $\mathcal{F}$ as an inner product space with inner product $\langle f, g \rangle = 2^{-n} \sum_{x \in \{0,1\}^n} f(x) g(x)$.

For $x \in \{0,1\}^n$, the function $\mathbf{1}_x$ maps x to 1 and all other points to 0, and for $P \subseteq \{0,1\}^n$, $\mathbf{1}_P = \sum_{x \in P} \mathbf{1}_x$.

Functions in $\mathcal{D}$ can be viewed as probability measures on $\{0,1\}^n$. For $f \in \mathcal{D}$ we write $x \sim f$ to mean that x is a random string sampled according to f. The set $\mathcal{D}$ is a compact subset of $\mathcal{F}$ whose extreme points are the functions $\{\mathbf{1}_x : x \in \{0,1\}^n\}$.

For $S \subseteq [n]$, the character $\chi_S \in \mathcal{F}$ is defined by $\chi_S(x) = \prod_{i \in S} (-1)^{x_i}$. The functions $\{\chi_S : S \subseteq [n]\}$ form an orthonormal basis for $\mathcal{F}$. Thus every $f \in \mathcal{F}$ can be written as a linear combination of characters: $f = \sum_{S \subseteq [n]} \langle f, \chi_S \rangle \chi_S$. The *Fourier coefficient* of function f at $S \subseteq [n]$ is defined[2] by $\widehat{f}(S) = 2^n \langle f, \chi_S \rangle = \sum_{x \in \{0,1\}^n} f(x) \chi_S(x)$. For $f \in \mathcal{D}$ we have:

$$\hat{f}(S) = \mathop{\mathbf{E}}_{x \sim f}[\chi_S(x)]. \tag{1}$$

The following equation, known as Plancherel's theorem expresses the inner product of f and g in terms of their Fourier coefficients.

$$\langle f, g \rangle = \sum_{S \subseteq [n]} \hat{f}(S) \hat{g}(S). \tag{2}$$

We define:

- The *support of* f, $\text{supp}(f) = \{x \in \{0,1\}^n : f(x) \neq 0\}$.
- The *Fourier support of* f, $\text{supp}(\hat{f}) = \{S \subseteq [n] : \hat{f}(S) \neq 0\}$
- $\|f\|_1 = \sum_{x \in \{0,1\}^n} |f(x)|$
- $\|\hat{f}\|_{L_1} = 2^{-n} \sum_{S \subseteq [n]} |\hat{f}(S)|$

$\mathcal{F}$ has two natural products. For $f, g \in F$, the *pointwise product* fg is given by $fg(x) = f(x)g(x)$ for all x and the *convolution product* $f * g$ is given by $f * g(x) = \sum_y f(y) g(x \oplus y)$.

If $f \in \mathcal{D}$ then $f * g(x) = \mathbb{E}_{z \sim f}[g(x \oplus z)]$. If f and g are both in $\mathcal{D}$ then $f * g \in \mathcal{D}$ and a sample from $f * g$ can be obtained by taking $x \oplus z$ where z is sampled according to f and x is sampled according to g.

For $S \subseteq [n]$, we have:

$$\begin{aligned} \widehat{fg}(S) &= 2^{-n} \sum_{T \subseteq [n]} \hat{f}(T) \hat{g}(T \triangle S) \\ \widehat{f * g}(S) &= 2^{-n} \hat{f}(S) \hat{g}(S). \end{aligned}$$

For a linear operator L on $\mathcal{F}$, and norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$, the $\alpha \to \beta$ norm of L, denoted by $\|L\|_{\alpha \to \beta}$ is defined to be the supremum of $\frac{\|Lv\|_\beta}{\|v\|_\alpha}$ over all $v \in V$.

For $S \subseteq [n]$, the operator $X_S : \mathcal{F} \to \mathcal{F}$ is defined as $X_S : f \mapsto \chi_S \cdot f$.

The Bonami-Beckner noise operator T_μ, defined for any real number μ, is most easily defined by its action on the character basis:

$$T_\mu \chi_S = \mu^{|S|} \chi_S.$$

More generally for $U \subseteq [n]$, the operator $T_{\mu,U}$ is defined by:

$$T_{\mu,U} \chi_S = \mu^{|S \cap U|} \chi_S.$$

Thus $T_\mu = T_{\mu,[n]}$. Using linearity, we can extend the action of T_μ to the space of all functions $\mathcal{F}$. For $i \in [n]$ we will adopt the shorthand $T_{\mu,i}$ for $T_{\mu,\{i\}}$.

It is easy to see that for any $\mu \neq 0$, $T_{\mu,U}$ is an invertible operator with its inverse being $T_{1/\mu,U}$. Likewise, for any U, U', the operators $T_{\mu,U}$ and $T_{\mu,U'}$ commute. In fact, if U and U' are disjoint, then $T_{\mu,U} \circ T_{\mu,U'} = T_{\mu,U \cup U'}$. Given the definition of $T_{\mu,U}$, it is straightforward to verify that for $x \in \{0,1\}^n$,

$$T_{\mu,U} f(x) = \sum_{z \in \{0,1\}^n : z_i = 0 \text{ for } i \notin U} f(x \oplus z) \prod_{i \in U} \frac{1}{2}(1 + (-1)^{z_i} \mu).$$

When $\mu \in [-1,1]$, $T_{\mu,U}$ has a nice probabilistic description. Recall from the introduction that for $\mu \in [-1,1]$, ν_μ is the probability distribution on $\{0,1\}^n$ obtained by setting each bit to 1 independently with probability $(1-\mu)/2$. More generally, for $U \subseteq [n]$ $\nu_{\mu,U}$ denotes the probability distribution on $\{0,1\}^n$ obtained by setting each of the bits indexed by U independently to 1 with probability $(1-\mu)/2$ and setting all the bits indexed by $[n] \backslash U$ to 0. We then have for $\mu \in [-1,1]$, that

$$T_{\mu,U} f = \nu_{\mu,U} * f,$$

[2] The Fourier coefficient is often defined without the normalizing factor of 2^n; this factor is included here to make (1) true.

and thus for $x \in \{0,1\}^n$

$$(T_{\mu,U}f)(x) \quad = \quad \mathbb{E}_{z \sim \nu_{\mu,U}}[f(x \oplus z)].$$

It is easy to verify that if $f \in \mathcal{D}$ and $\mu \in [-1,1]$ then $T_{\mu,U}f \in \mathcal{D}$. A sample from $T_{\mu,U}f$ is generated by taking $x \oplus z$ where $x \sim f$ and $z \sim \nu_{\mu,U}$. A *μ-noisy sample from f* is a sample from $T_\mu f$.

B. Parameter estimation

We consider the general problems of estimating a real-valued parameter $P = P(g)$ of an unknown probability distribution $g \in \mathcal{D}_n$. An estimator P_{est} is a random variable that is a function of a collection of independent samples.

- The *bias of* P_{est} (as an estimator of P) is $|\mathbf{E}[P_{\text{est}} - P]|$.
- The *range of* P_{est} is the maximum of $|P_{\text{est}}|$.
- P_{est} is an (ε, κ)*-estimator* of P provided that $\Pr[|P_{\text{est}} - P| > \varepsilon] < \kappa$.

It is well known that one can build (ε, κ)-estimators from independent copies of estimators wth fairly weak estimation properties. For an estimator P_{est} and positive integer k, let $A_k(P_{\text{est}})$ denote the average of k independent copies of P_{est}.

Proposition II.1. *For any $\varepsilon, \delta \in (0,1)$, if the estimator P_{est} of P has bias at most $\frac{\varepsilon}{2}$, and range at most M, then the estimator $A_k(P_{\text{est}})$, with $k > 8\frac{M^2}{\varepsilon^2}\ln(\frac{1}{\kappa})$ is a (ε, δ)-estimator.*

Proof: Obviously $\mathbf{E}[A_k(P_{\text{est}})] = \mathbf{E}[P_{\text{est}}]$. By the Chernoff-Hoeffding bound [7],

$$\begin{aligned}\Pr[|A_k(P_{\text{est}}) - P| \geq \varepsilon] &\leq \Pr[|A_k(P_{\text{est}}) - \mathbf{E}[P_{\text{est}}]|] \geq \varepsilon/2] \\ &\leq e^{-\varepsilon^2 k/8M^2} \leq \delta.\end{aligned}$$

This concludes the proof. ∎

1) Möbius transforms: Let $(P, \preceq)$ be a poset. Define function $\zeta_P : P \times P \to \mathbb{R}$ as $\zeta(x,y) = 1$ if and only if $x \preceq y$ and 0 otherwise. Also define $\mu_P : P \times P \to \mathbb{R}$ recursively as follows:

$$\text{For } x \in P, \ \mu_P(x,x) = 1.$$

$$\text{For } x,y \in P, \ \mu_P(x,y) = \mathbf{1}_{x \preceq y} \cdot \left(\sum_{x \preceq z \prec y} -\mu_P(x,z) \right).$$

Let $\mathcal{F}_P$ be the space of real-valued functions on P. We define operators $\boldsymbol{\zeta}_P : \mathcal{F}_P \to \mathcal{F}_P$ and $\boldsymbol{\mu}_P : \mathcal{F}_P \to \mathcal{F}_P$ by:

$$(\boldsymbol{\zeta}_P f)(x) = \sum_{x \in P} \zeta(x,y) f(y) = \sum_{x \preceq y} f(y),$$

$$(\boldsymbol{\mu}_P f)(x) = \sum_{x \preceq y} \mu_P(x,y) \cdot f(y).$$

It is well known (see [13]) that the transforms $\boldsymbol{\zeta}_P$ and $\boldsymbol{\mu}_P$ are inverses of each other. $\boldsymbol{\mu}_P$ is usually referred to as the Möbius transform of the poset P. The above notions can be extended to the more general setting of functions from P to a fixed vector space.

Proposition II.2. *Let P be a poset and V be an arbitrary vector space over $\mathbb{R}$. Suppose $(f_x : x \in P)$ and $(g_x : x \in P)$ are families of vectors in V satisfying $f_x = \sum_{x \preceq y} g_y$. Then*

$$g_x = \sum_{x \preceq y} \mu(x,y) \cdot f_y$$

Definition 1. *For $x \in P$, define $x^\downarrow = \{y : y \preceq x\}$. For $C \subseteq P$, define $C^\downarrow = \cup_{x \in C} x^\downarrow$. A subset D of P such that $D^\downarrow = D$ is a "downset". It is easy to see that $C^\downarrow$ is the unique minimal subset of P that is a downset and contains C, and is referred to as the "downset generated by C".*

If $C \subseteq P$, then we can view C as a poset, which has its own Möbius function μ_C. In general it is not true that for all $x,y \in C$, $\mu_C(x,y) = \mu_P(x,y)$ but it is true if C is a downset.

Proposition II.3. *If $D \subseteq P$ is a downset, then for all $x,y \in D$, $\mu_D(x,y) = \mu_P(x,y)$.*

This is easily verified by induction using the above inductive definition of μ_C and μ_P.

We denote by $\mathcal{P}([n])$ the poset on $2^{[n]}$ ordered by set inclusion. It is well known that in this poset, for $x \preceq y$, $\mu_{\mathcal{P}([n])}(x,y) = (-1)^{|y \setminus x|}$. Combining with Proposition II.3 we have:

Corollary II.4. *If D is a downset of $\mathcal{P}([n])$ then for $x \preceq y \in D$ we have $\mu_D(x,y) = (-1)^{|y \setminus x|}$.*

C. Technical computational considerations

We now mention a few technical considerations concerning the cost of computation. In some cases, we will have known functions $b, \ell \in \mathcal{F}_n$, given by an $n^{O(1)}$-time algorithm that on input $S \subseteq [n]$ evaluates $\hat{\ell}(S)$ and $\hat{b}(S)$, and we will want to evaluate a function of the form $\sum_{S \subseteq [n]} \hat{\ell}(S) b(S)$. The cost of the trivial summation algorithm is $2^n n^{O(1)}$, but if $\text{supp}(\hat{\ell})$ is small compared to 2^n we can hope to speed this up by enumerating only over sets in $\text{supp}(\hat{\ell})$. However, even if we can evaluate $\hat{\ell}(S)$ for any given S, this does not mean that we can enumerate over sets in the support without looking at all sets. Technically what we want is a family of subsets $\mathcal{H}$ that contains $\text{supp}(\hat{\ell})$ together with an *efficient listing algorithm* for $\mathcal{H}$ which is an algorithm that lists all members of $\mathcal{H}$ in time $|\mathcal{H}| n^{O(1)}$. We will say that $\mathcal{H}$ is an *listable support* for $\hat{\ell}$.

Further, for the sake of clarity of exposition, throughout the paper, we will assume that we are able to do basic arithmetic operations on real numbers with infinite precision. In an actual implementation, we will only be working with finite precision approximations of these numbers. The next simple proposition (stated without a proof) asserts that basic arithmetic operations on real numbers can be done efficiently to any finite precision.

Proposition II.5. *A sum of the form $\sum_{i=1}^{m} A_i$ where each A_i is a product of $O(1)$ numbers can be approximated to within additive error δ in time $O(m(\log(\frac{1}{\delta}\sum_i(1+|A_i|))^{O(1)})$.*

III. Proof of Theorem I.1

We have an unknown probability distribution f on $\{0,1\}^n$ together with a subset X that contains $\mathrm{supp}(f)$. We have access to samples from the distribution $T_\mu f$. Our goal is to give a good estimate for $f(0^n)$ in time $\mathsf{poly}(n, |X|, \frac{1}{\varepsilon}, \log(\frac{1}{\delta}))$. Our algorithm is based on the approach of [9] (which built on ideas from [14]). We present a framework that abstracts this approach, and identify a critical improvement. The key ingredient to our algorithm is a function u that satisfies the conclusions of the following lemma.

Lemma III.1. *Given X and ε, there is a function $u \in \mathcal{F}_n$ such that for all f with $\mathrm{supp}(f) \subseteq X$:*

1) *There is a real valued function α defined on $\{0,1\}^n$ computable in time $(k/\varepsilon)^{\tilde{O}(1/\mu^4)} n^{O(1)}$ such that (a) For all $x \in \{0,1\}^n$, $|\alpha(x)| \le (k/\varepsilon)^{\tilde{O}(1/\mu^4)}$, and (b) for $z \sim T_\mu f$, $\alpha(z)$ is an unbiased estimator for $\langle u, f\rangle$.*
2) $|\langle u, f\rangle - u(\mathbf{0})f(\mathbf{0})| \le \varepsilon/10$.
3) $u(\mathbf{0}) \in [1/2, 1]$ *and there is an algorithm that estimates $u(\mathbf{0})$ to within an additive $\varepsilon/9$ and runs in time* $\mathsf{poly}\big(nk\frac{1}{\varepsilon}\log\frac{1}{\kappa}\big)$.

Theorem I.1 follows easily from this lemma.

Proof of Theorem I.1: Let $R = (k/\varepsilon)^{\tilde{O}(1/\mu^4)}$ be the range of the estimator for $\langle u, f\rangle$. Applying Proposition II.2, the average of $m = \mathsf{poly}(R/\epsilon) = (k/\varepsilon)^{\tilde{O}(1/\mu^4)}$ independent copies of this estimator yields an estimate A that is within $\varepsilon/10$ of $\langle u, f\rangle$ with probability at least 7/8. Also, let B be the estimate of $u(\mathbf{0})$ given by the third part of the lemma that is within $\varepsilon/10$ with probability at least 7/8. Our algorithm outputs A/B (or, more precisely, a floating point approximation C to A/B that is within $\varepsilon/10$ of A/B) as the estimate of $f(\mathbf{0})$. The bound on the running time of the algorithm follows easily from the bounds on the running time of the estimator for $\langle u, f\rangle$ and computation of $u(\mathbf{0})$.

Next, we claim that with probability at least 3/4, the output A/B is within ε of $f(\mathbf{0})$. Note that with probability at least 3/4, $|A - \langle u, f\rangle| \le \varepsilon/10$ and $|B - u(\mathbf{0})| \le \varepsilon/10$. Assuming this is the case, we also have $B \ge 1/3$ since $u(\mathbf{0}) \ge 1/2$ and we can assume that $\varepsilon < 1$. So with probability at least 3/4, we have

$$\begin{aligned}|f(\mathbf{0}) - C| &\le \frac{\varepsilon}{10} + \left|f(\mathbf{0}) - \frac{A}{B}\right| \\ &= \frac{\varepsilon}{10} + \frac{1}{B}\cdot|Bf(\mathbf{0}) - A| \\ &\le \frac{\varepsilon}{10} + 3|Bf(\mathbf{0}) - A|\end{aligned}$$

Using triangle inequality, we have

$$\begin{aligned}|Bf(\mathbf{0}) - A| &\le |Bf(\mathbf{0}) - u(\mathbf{0})f(\mathbf{0})| + |u(\mathbf{0})f(\mathbf{0}) - \langle u, f\rangle| \\ &+ |\langle u, f\rangle - A|.\end{aligned}$$

All three quantities on the right hand side are bounded by $\epsilon/10$. Thus,

$$|f(\mathbf{0}) - C| \le \frac{\varepsilon}{10} + 3\left(\frac{\varepsilon}{10} + \frac{\varepsilon}{10} + \frac{\varepsilon}{10}\right) \le \varepsilon.$$

■

So the main part of the proof of the theorem is the construction of the function u and the proof of the associated Lemma III.1. It turns out that u is best described as the pointwise product of two functions ℓ and q, and in the next section we motivate their construction and state the essential properties of the functions q and ℓ (see Lemmas IV.2 and IV.1). These properties immediately give Lemma III.1. In Section V we construct ℓ and show that it satisfies Lemma IV.2 and in Appendix A we construct q and show that it satisfies Lemma IV.1.

IV. Constructing the function u

A. Estimating $f(\mathbf{0})$ via estimates of Fourier coefficients

We have access to samples from $T_\mu f$ and we want to estimate $f(\mathbf{0})$. Suppose $\ell \in \mathcal{F}$ satisfies

$$f(\mathbf{0}) = \langle \ell, f\rangle. \tag{3}$$

By (2), this equals $\sum_{S\subseteq[n]} \hat{\ell}(S)\hat{f}(S)$, which suggests estimating $\langle \ell, f\rangle$ by using samples from $T_\mu f$ to construct estimators for $\hat{f}(S)$ and replacing $\hat{f}(S)$ with its estimate in the above sum.

There is a natural estimator for $\hat{f}(S)$ given samples from $T_\mu f$. To see this, note that for any $d \in \mathcal{D}_n$, if z is a sample from $d \in \mathcal{D}_n$, then by (1), $\chi_S(z)$ is an unbiased estimator for $\hat{d}(S)$. In particular if $z \sim T_\mu f$ then $\chi_S(z)$ is an unbiased estimator of $\widehat{T_\mu f}(S) = \mu^{|S|}\hat{f}(S)$. Therefore $(\frac{1}{\mu})^{|S|}\chi_S(z)$ is an unbiased estimator for $\hat{f}(S)$. Thus for ℓ satisfying (3),

$$W_\ell(z) = \sum_{S\subseteq[n]} \hat{\ell}(S)\left(\frac{1}{\mu}\right)^{|S|}\chi_S(z),$$

is an unbiased estimator of $f(\mathbf{0})$.

An obvious choice for ℓ satisfying (3) is $\mathbf{1_0}$, in which case $\hat{\ell}(S) = 1$ for all S, so the resulting estimator of $f(\mathbf{0})$ is $\sum_{S\subseteq[n]}(\frac{1}{\mu})^{|S|}\chi_S(z)$. Unfortunately, the quality of the resulting estimator is not very good. To see this, note that the sum $W_\ell(z)$ simplifies to $(1-\frac{1}{\mu})^{|z|}(1+\frac{1}{\mu})^{n-|z|}$. Thus, the range of this estimator (and in fact, the variance) is exponentially large in n. As a result,the estimator obtained using Proposition II.1 has sample complexity exponentially large in n.

So we look for an alternative ℓ satisfying (3) for which both the cost of evaluating $W_\ell(z)$, and the range of $W_\ell(z)$ are "small". Since we know that $\mathrm{supp}(f) \subseteq X$, it suffices to choose a $\ell \in \mathcal{G}_0(X)$ (recall, $\mathcal{G}_0(X)$ is the set of functions that map $\mathbf{0}$ to 1 and all $x \in X \setminus \{\mathbf{0}\}$ to 0). To bound the cost of the induced (ε, δ)-estimator we need to bound both the cost of computing $W_\ell(z)$ and its range.

To compute $W_\ell(z)$ we need to sum $\hat{\ell}(S)(\frac{1}{\mu})^{|S|}\chi_S(z)$ over $S \in \mathrm{supp}(\hat{\ell})$. As discussed in Section II-C, to evaluate this sum quickly it is not enough to know that $|\mathrm{supp}(\hat{\ell})|$ is small; we also need a listable support $\mathcal{H}$ for $\hat{\ell}$. With this, $W_\ell(z)$ can be evaluated in time $|\mathcal{H}|(T + n^{O(1)})$ where T is an upper bound on the time needed to evaluate $\hat{\ell}(S)$ on input $S \in \mathcal{H}$. To upper bound the range of $W_\ell(z)$, note that every term in the sum is bounded (in absolute value) by $\hat{\ell}(S)(\frac{1}{\mu})^{m(\mathcal{H})}$ where $m(\mathcal{H})$ is an upper bound on size of the largest set in $\mathcal{H}$. Thus, the range R of this estimator is bounded by $|\mathcal{H}| \cdot \|\hat{\ell}\|_\infty(\frac{1}{\mu})^{m(\mathcal{H})}$. Hence, the running time of the estimator is $\mathsf{poly}(|\mathcal{H}|, R, \frac{1}{\varepsilon}, \log(1/\delta))$.

The algorithm of Wigderson and Yehudayoff [14] can be formulated in this framework: They (implicitly) show how to (efficiently) construct a function $\ell_{WY} \in \mathcal{G}_0(X)$, and listable support $\mathcal{H}$ for $\hat{\ell}$ so that

- All sets in $\mathcal{H}$ have size at most $O(\log|X|)$.
- $|\mathcal{H}| \leq |X|^{\log|X|}$.
- $\|\widehat{\ell_{\mathsf{WY}}}\|_{L_1} = O(|X|^{\log|X|})$.

Thus the running time of the induced estimator for $f(\mathbf{0})$ is $\mathsf{poly}(|X|^{\log|X|}, \frac{1}{\varepsilon}, \log(1/\delta))$.

B. The Lovett-Zhang approach

The improved running time of Lovett and Zhang [9] involves two steps: (i) Constructing a function ℓ_{LZ} that gives a faster estimator in the case that all of the points in X have small Hamming weight, i.e., $O(\log|X|)$. (ii) A reduction from the case of general X to the small Hamming weight case.

For $Y \subseteq \{0,1\}^n$, let $w(Y)$ be the maximum Hamming weight of any string in Y. Lovett and Zhang showed how to construct, for any set Y, a function $\ell_{\mathsf{LZ}} \in \mathcal{G}_0(Y)$ and a listable support $\mathcal{H}$ for $\hat{\ell}_{\mathsf{LZ}}$ such that

- $m(\mathcal{H})$, the size of the largest set in $\mathcal{H}$, is at most $w(Y)$.
- $|\mathcal{H}| \leq |Y|2^{w(Y)}$.
- $\|\widehat{\ell_{\mathsf{LZ}}}\|_{L_1} = |Y|^{O(\log w(Y))}$.

(This result is implied by Proposition 3.6 in their paper.) Applying this construction with $Y = X$ yields an estimator for $f(\mathbf{0})$. Unlike the WY estimator, the running time of this estimator deteriorates as $w(X)$ increases. For e.g., for $w(X) = O(\log|X|)$ the derived estimator has running time is $|X|^{O(\log\log|X|)}$.

Lovett and Zhang present a kind of a reduction of the general case ($w(X) \leq n$) to the case that $w(X) = O(\log|X|)$. This reduction combined with the application of ℓ_{LZ} yields their $O(|X|^{\log\log|X|})$ algorithm for the general case [3].

We now elaborate on this. For some threshold r (which we eventually set to $O_\varepsilon(\log|X|)$), let $\text{NEAR} = \text{NEAR}_r(X) = \{x \in X : |x| \leq r\}$ and $\text{FAR} = \text{FAR}_r(X) = X - \text{NEAR}$. Consider the construction of the function ℓ_{LZ} with $Y = \text{NEAR}$ instead of $Y = X$, Then we have:

$$f(\mathbf{0}) = \langle \ell_{LZ}, f\rangle - \sum_{x \in \text{FAR}} \ell_{LZ}(x)f(x). \tag{4}$$

If the sum (error term) being subtracted off is small, then we can still estimate $f(\mathbf{0})$ by estimating $\langle \ell_{LZ}, f\rangle$. It turns out that $\ell_{LZ}(x) \in [0,1]$ for all x and so the error is bounded by $|X|\max_{x\in\text{FAR}} f(x)$. Unfortunately, this might be quite large.

To get around this, Lovett and Zhang effectively replaced f by another function g for which $\max_{x\in\text{FAR}} g(x)$ is very small. To do this, they constructed an explicit function q (depending on X but otherwise not on f) and set $g = q \cdot f$. We have $f(\mathbf{0}) = g(\mathbf{0})/q(\mathbf{0})$ so it suffices to approximate $g(\mathbf{0})$. We no longer have $g(\mathbf{0}) = \sum_{S\subseteq[n]} \hat{\ell}_{\mathsf{LZ}}(S)\hat{g}(S)$, since $\ell_{\mathsf{LZ}}(x)$ need not be 0 for $x \in \text{FAR}$. But we can bound the difference between these quantities as follows:

$$\begin{aligned} |g(\mathbf{0}) - \sum_{S\subseteq[n]} \hat{\ell}_{\mathsf{LZ}}(S)\hat{g}(S)| &\leq \sum_{x\in\text{FAR}} \ell_{LZ}(x)g(x) \\ &\leq \sum_{x\in\text{FAR}} \ell_{LZ}(x)q(x) \\ &\leq \max_{x\in\text{FAR}} q(x) \sum_{s\in\text{FAR}} \ell_{LZ}(x) \end{aligned} \tag{5}$$

The function q is chosen so that $q(x)$ (and therefore $g(x)$) is very small for all $x \in \text{FAR}$, so the contribution of the second sum can be ignored. Additionally, to estimate the first sum, we need to efficiently estimate $\hat{g}(S)$ from samples from $T_\mu f$, which imposes additional constraints on the function q. The precise properties of the function q are given by the following lemma.

Lemma IV.1. *For any X and $r \geq (1/\mu^2) \cdot \log|X|$, there is a function q having the following properties:*

- *For all $x \in \text{FAR}$, $q(x) \leq e^{-\frac{1}{2}\mu^2 r}$.*
- *$q(\mathbf{0}) \in [1/2, 1]$*
- *$q(\mathbf{0})$ can be (ε, κ) approximated in time $\mathsf{poly}(n|X|\frac{1}{\varepsilon}\log\frac{1}{\kappa})$.*
- *For every S, there is a function $\alpha_S(z)$ for $z \in \{0,1\}^n$ such that for $z \sim T_\mu f$, $\alpha_S(z)$ is an unbiased estimator $\widehat{q \cdot f}(S)$ with range at most $\left(\frac{1}{\mu}\right)^{|S|}$ and is computable in time $2^{|S|}n^{O(1)}$.*

Lemma IV.1 is implicit in [9]; we prove it in Appendix A. Using this lemma, Lovett and Zhang estimate $g(\mathbf{0})$ by estimating $\sum_S \hat{\ell}_{\mathsf{LZ}}(S)\hat{g}(S)$ as outlined above.

C. Improving ℓ

We follow the approach outlined above, but replace ℓ_{LZ} by a better function. Our first attempt uses the Möbius function (Section II-B1), to construct a function $\ell_0 = \ell_{0,X}$ with listable support $\mathcal{H}_0$ such that:

- $|\mathcal{H}_0| \leq |X|2^{w(X)}$,
- $\|\hat{\ell}_0\|_{L_1} = |X|2^{w(X)}$.

[3] Actually, the Lovett-Zhang algorithm doesn't actually follow this scheme, because the function ℓ_{LZ} is not efficiently computable, but they use its existence to argue that the maximum likelihood estimator is a good estimator.

Using this choice in the basic approach outlined in Section IV-A gives a polynomial time estimator in the case $w(X) = O(\log n)$ since both $|\mathcal{H}_0|$ and $\|\widehat{\ell_0}\|_{L_1}$ are polynomial in $|X|$.

Using ℓ_0 with the modified approach of Lovett and Zhang, we fix a parameter $r = \theta_\varepsilon(\log |X|)$ and construct ℓ_1 satisfying the above, but using the set $\textsc{near}$ in place of X. We can then bound the error term in (5) using the above bound on $\|\widehat{\ell_0}\|_{L_1}$ and the bound on $q(x)$ for $x \in \textsc{far}$ in Lemma IV.1 to bound the error term in (5) from above by $|X|2^r e^{-\mu^2 r/2}$.

Unfortunately, even when $\mu = 1$, 2^r overwhelms $e^{-\mu^2 r/2}$ and the term is large. In an earlier version of this paper, we showed how to modify q to get improved bounds on $q(x)$ for $x \in \textsc{far}$ of the form $2^{-\beta(\mu)r}$, where $\beta(\mu) > 1$ for $\mu > .555$. Thus, for such values of μ the error term can be made arbitrarily small, thereby getting a polynomial time estimation algorithm for this value of μ. While one might hope to prove this for even smaller values of μ by improving q further, this approach seems to be incapable of working for arbitrary $\mu > 0$ since the functions $\beta(\mu)$ that are obtained in this way tend to 0 as μ tends to 0.

So instead of changing q, we modify the function ℓ to reduce $\|\widehat{\ell}\|_{L_1}$ from $2^r\mathsf{poly}(|X|)$ to $(1+\delta)^r\mathsf{poly}(|X|)$ for an arbitrary $\delta > 0$. By choosing $r = O_\delta(\log |X|)$ appropriately, the error term in (5) can be made arbitrarily small. In order for us to accomplish this, we will relax the condition $\ell \in \mathcal{G}_0(\textsc{near})$ to the condition that $\ell \in \mathcal{G}_\eta(\textsc{near})$ for a suitably small η. (Recall that $\mathcal{G}_\eta(Y)$ is the set of functions ℓ such that $\ell(\mathbf{0}) = 1$ and $|\ell(x)| \le \eta$ for all $x \in Y - \{\mathbf{0}\}$.) The next lemma states several properties that are achieved by our construction of ℓ.

Lemma IV.2. *Let $C \subseteq \{0,1\}^n$, $\delta > 0$ and $\eta > 0$. Let r be an upper bound on $w(C)$. There is a function $\ell = \ell_{C,\delta,\eta} : \{0,1\}^n \to \mathbb{R}$*

- $\ell \in \mathcal{G}_\eta(C^\downarrow)$,
- $\|\widehat{\ell}\|_{L_1} \le |C|^2 \cdot (1+2\delta)^r \cdot (2/\eta)^{\delta^{-1}\cdot\log(2\delta^{-1})}$,
- $\mathrm{supp}(\widehat{\ell}) \subseteq C^\downarrow$,
- *For any $S \subseteq [n]$, the Fourier coefficient $\hat{\ell}(S)$ can be computed in time $\mathsf{poly}(|C^\downarrow|, n)$.*

D. Proof of Lemma III.1

With the aid of Lemmas IV.1 and IV.2, we will now prove Lemma III.1. To do this, apply Lemma IV.1 with $r = (100/\mu^4)\cdot\log(1/\mu)\cdot\log(k/\varepsilon)$ to get function q. We then apply Lemma IV.2 with $C = X \cap B(0,r)$, $\eta = \frac{\varepsilon}{20}$ and $\delta = \frac{\mu^2}{16}$ to get the resulting function ℓ. Define $u = \ell \cdot q$. We will show that this u satisfies all the properties we need in Lemma III.1. We begin by noting that the third item (i.e. $u(0)$ can be efficiently approximated and lies in $[1/2, 1]$) follows by combining that $\ell(0) = 1$ and Lemma IV.1. Next, we give an unbiased estimator for $\langle u, f\rangle$. We know that:

$$\begin{aligned}\langle u, f\rangle &= \langle \ell\cdot q, f\rangle = \langle \ell, q\cdot f\rangle\\ &= \textstyle\sum_S \widehat{\ell}(S)\cdot\widehat{qf}(S)\\ &= \textstyle\sum_{S\subseteq C^\downarrow} \widehat{\ell}(S)\cdot\widehat{qf}(S)\end{aligned}$$

Lemma IV.1 shows that for any S, there exist an unbiased estimator $\alpha_S(z)$ for $\widehat{qf}(S)$, with range at most $\left(\frac{1}{\mu}\right)^{|S|}$ that is computable in time $2^{|S|}n^{O(1)}$. It then follows that $\sum_{S\subseteq C^\downarrow}\widehat{\ell}(S)\cdot\alpha_S(z)$ is an unbiased estimator with range at most $\|\widehat{\ell}\|_{L_1}\cdot\left(\frac{1}{\mu}\right)^r \le (k/\varepsilon)^{\tilde{O}(1/\mu^4)}$ and it can be computed in time $|C^\downarrow|\cdot 2^r n^{O(1)} = (k/\varepsilon)^{\tilde{O}(1/\mu^4)}n^{O(1)}$. All that remains is to bound $|\langle u, f\rangle - u(\mathbf{0})f(\mathbf{0})|$.

$$\begin{aligned}|\langle u, f\rangle - u(\mathbf{0})f(\mathbf{0})| &= |\sum_{x\in X\setminus\{\mathbf{0}\}} \ell(x)q(x)f(x)|\\ \le\ & |\sum_{x\in\textsc{near}\setminus\{\mathbf{0}\}} \ell(x)q(x)f(x)| + |\sum_{x\in\textsc{far}} \ell(x)q(x)f(x)|\\ \le\ & \eta\cdot|\sum_{x\in\textsc{near}\setminus\{\mathbf{0}\}} f(x)| + \|\ell\|_\infty\cdot e^{-\frac{1}{2}\mu^2 r}|\sum_{x\in\textsc{far}} f(x)|\\ \le\ & \eta + k^2\cdot(1+2\delta)^r\cdot(2/\eta)^{\delta^{-1}\cdot\log(2\delta^{-1})}\cdot e^{-\frac{1}{2}\mu^2 r}\end{aligned}$$

By plugging the values of r, η and δ, we have $|\langle u, f\rangle - u(\mathbf{0})f(\mathbf{0})| \le \varepsilon/10$.

V. Proof of Lemma IV.2

In this section, we prove Lemma IV.2 which given $C \subseteq \{0,1\}^n$ and $\delta, \eta > 0$ constructs a suitable function ℓ. As a warmup, we construct the function ℓ_0 mentioned earlier. The function ℓ_0 is specified by the set $X \subseteq \{0,1\}^n$, which we change to C to match the notation of Lemma IV.2. We are given $C \subseteq \{0,1\}^n$ and want to construct a function $\ell_0 \in \mathcal{G}_0(C)$ with a listable support $\mathcal{H}_0$ for $\hat{\ell}_0$ such that:

- $|\mathcal{H}_0| \le |C|2^{w(C)}$.
- $\|\widehat{\ell_0}\|_{L_1} = |C|2^{w(C)}$

The function we construct will satisfy the stronger condition that $\ell_0 \in \mathcal{G}_0(C^\downarrow)$, which means that it is 1 at $\mathbf{0}$ and 0 on every other point of $C^\downarrow$.

We introduce some notation to represent the natural correspondence between strings in $\{0,1\}^n$ and subsets of $[n]$. For $z \in \{0,1\}^n$, define $\textsc{ones}(z) = \{i \in [n] : z_i = 1\}$. For $A \subseteq \{0,1\}^n$, let $\mathcal{H}(A)$ be the collection of subsets $\{\textsc{ones}(z) : z \in A\}$. We let $\mathcal{H}_0$ be the same as $\mathcal{H}(C^\downarrow)$. Observe that given C, we can efficiently list all the sets of $\mathcal{H}_0$ and $|\mathcal{H}_0| \le |C|2^{w(C)}$.

Note that the requirement of $\widehat{\ell_0}$ being supported on $\mathcal{H}_0 = \mathcal{H}(C^\downarrow)$ is the same as requiring the function ℓ_0 to be of the form $\ell_0 = \sum_{S\in C^\downarrow}\beta_S\cdot\chi_S$. In order to find the coefficients $\{\beta_S\}_{S\in C^\downarrow}$, we start by defining the family of functions $\{\mathbf{1}_{\succeq z}\}_{z\in\{0,1\}^n}$ as follows:

$$\mathbf{1}_{\succeq z}(x) = \mathbf{1}_{x\succeq z}$$

It is easy to verify:

$$\begin{aligned}\mathbf{1}_{\succeq z}(x) &= \prod_{i:z_i=1} x_i = \prod_{i:z_i=1} \frac{1-\chi_i(x)}{2} \\ &= \frac{1}{2^{|z|}} \sum_{S\subseteq \textsc{ones}(z)} (-1)^{|S|}\chi_S(x)\end{aligned}$$

This implies that $\|\widehat{\mathbf{1}_{\succeq z}}\|_{L_1} = 1$ and $\mathrm{supp}(\widehat{\mathbf{1}_{\succeq z}}) \subseteq \mathcal{H}(z^{\downarrow})$. Thus, a linear combination of functions $(\mathbf{1}_{\succeq z})_{z\in C^{\downarrow}}$ will have Fourier support in $\mathcal{H}(C^{\downarrow})$. We will construct ℓ_0 as a linear combination of $(\mathbf{1}_{\succeq z})_{z\in C^{\downarrow}}$. By considering the restriction of the function ℓ_0 to $C^{\downarrow}$ we can use the Möbius transform to find the linear combination.

For a function $f \in \mathcal{F}$, let f^R denote the function obtained by restricting the domain to $C^{\downarrow}$. The condition $\ell_0 \in \mathcal{G}_0(C^{\downarrow})$ is the same as $\ell_0^R = \mathbf{1}_{\mathbf{0}}^R$. Observe that in the poset $C^{\downarrow}$ we have $\mathbf{1}_{\succeq z}^R = \sum_{y\succeq z} \mathbf{1}_y^R$ for all $z \in C^{\downarrow}$. By Proposition II.2 and Corollary II.4 we have:

$$\mathbf{1}_y^R = \sum_{y\preceq z\in C^{\downarrow}} (-1)^{|z\setminus y|}\mathbf{1}_{\succeq z}^R \quad \text{for all } z\in C^{\downarrow}$$

This result can also be verified directly without Proposition II.2 and Corollary II.4.

For $y \in C^{\downarrow}$, define the function $\ell_y = \sum_{y\preceq z\in C^{\downarrow}}(-1)^{|z\setminus y|}\mathbf{1}_{\succeq z}$. We claim that the function $\ell_0 = \ell_{\mathbf{0}}$ satisfies the requirements. To see this, note that $\ell_y^R = \mathbf{1}_y^R$ (but in general ℓ_y may disagree with $\mathbf{1}_y$ out of $C^{\downarrow}$). Thus, $\ell_0 \in \mathcal{G}_0(C^{\downarrow})$. Further,

$$\|\widehat{\ell_0}\|_{L_1} \le \sum_{z\in C^{\downarrow}} \|\widehat{\mathbf{1}_{\succeq z}}\|_{L_1} \le \sum_{z\in C^{\downarrow}} 1 \le |C^{\downarrow}| \le |C|\cdot 2^{w(C)}.$$

We now turn to the proof of Lemma IV.2. We are given $C \subseteq \{0,1\}^n$ and $\delta, \eta > 0$, and an upper bound r on $w(C)$. We want to construct a function ℓ satisfying the conclusions of the lemma.

As mentioned in Section IV-C, the reason why ℓ_0 is not good enough for us is because the Fourier L_1 norm grows too fast. To circumvent this, we start with a modified family of functions $\ell_{\delta,y} = \sum_{y\preceq z\in C^{\downarrow}}(-1)^{|z\setminus y|}\cdot\delta^{|z|}\cdot\mathbf{1}_{\succeq z}$. Note that $\ell_{\delta,y}$ generalizes the function ℓ_y (which is obtained by setting $\delta = 1$). We will construct ℓ as a linear combination of $\{\ell_y\}_{y\in C^{\downarrow}}$. First we prove some properties of $(\ell_{\delta,y} : y \in C^{\downarrow})$.

Proposition V.1. *For any $\delta > 0, y \in C^{\downarrow}$, the function $\ell_{\delta,y} = \sum_{y\preceq z\in C^{\downarrow}}(-1)^{|z\setminus y|}\cdot\delta^{|z|}\cdot\mathbf{1}_{\succeq z}$ satisfies the following properties:*

- *For $x \in C^{\downarrow}$, $\ell_{\delta,y}(x) = \mathbf{1}_{x\succeq y}\cdot(1-\delta)^{|x|-|y|}\cdot\delta^{|y|}$.*
- $\mathrm{supp}(\widehat{\ell_{\delta,y}}) \subseteq C^{\downarrow}$.
- $\|\widehat{\ell_{\delta,y}}\|_{L_1} \le |C|\cdot(1+\delta)^{w(C)-|y|}\cdot\delta^{|y|}$

Proof: First we can rewrite $\ell_{\delta,y}$ as $\ell_{\delta,y} = \delta^{|y|}\sum_{y\preceq z\in C^{\downarrow}}(-\delta)^{|z\setminus y|}\cdot\mathbf{1}_{\succeq z}$. For any $x \in C^{\downarrow}$,

$$\begin{aligned}\ell_{\delta,y}(x) &= \delta^{|y|}\sum_{y\preceq z\in C^{\downarrow}}(-\delta)^{|z\setminus y|}\cdot\mathbf{1}_{\succeq z}(x) \\ &= \delta^{|y|}\sum_{y\preceq z\preceq x}(-\delta)^{|z\setminus y|} = \mathbf{1}_{x\succeq y}\cdot(1-\delta)^{|x|-|y|}\cdot\delta^{|y|}\end{aligned}$$

Since $\mathrm{supp}(\widehat{\mathbf{1}_{\succeq z}}) \subseteq C^{\downarrow}$, we deduce that $\mathrm{supp}(\widehat{\ell_{\delta,y}}) \subseteq C^{\downarrow}$. For the last requirement,

$$\begin{aligned}\|\widehat{\ell_{\delta,y}}\|_{L_1} &\le \delta^{|y|}\sum_{y\preceq z\in C^{\downarrow}}|(-\delta)^{|z\setminus y|}|\cdot\|\mathbf{1}_{\succeq z}\|_{L_1} \\ &\le \delta^{|y|}\sum_{y\preceq z\in C^{\downarrow}}\delta^{|z\setminus y|} \\ &\le \delta^{|y|}\sum_{t\in C}\sum_{y\preceq z\preceq t}\delta^{|z\setminus y|} \\ &\le |C|\cdot\delta^{|y|}\cdot(1+\delta)^{w(C)-|y|}\end{aligned}$$

■

Note that we have relaxed the requirement on ℓ, namely $\ell \in \mathcal{G}_\eta(C)$ for some appropriately small η as opposed to ℓ_0 which was in $\mathcal{G}_0(C)$. Recall that we will construct ℓ as a linear combination of form $\ell_{\delta,y}$ for $y \in C^{\downarrow}$. We now impose the additional requirement that the coefficient of $\ell_{\delta,y}$ depends only on $|y|$. This will help us in search of the said coefficients. With this, let $\ell = \sum_{y\in C^{\downarrow}} v_{|y|}\cdot\ell_{\delta,y}$, where $v = (v_0, ..., v_{w(C)})$ is the vector of coefficients. By Proposition V.1 for any $x \in C^{\downarrow}$:

$$\ell(x) = \sum_{y\preceq x} v_{|y|}\cdot\delta^{|y|}(1-\delta)^{|x|-|y|} = \sum_{t=0}^{|x|} v_t\cdot\binom{|x|}{t}\delta^t(1-\delta)^{|x|-t}$$

Since the value of ℓ only depends on the weight of x, we can define a function $\tilde{\ell}$ on nonnegative integers so that $\ell(x) = \tilde{\ell}(|x|)$. Now we have $\tilde{\ell}(m) = \sum_{t=0}^{m} v_t\cdot\binom{m}{t}\cdot\delta^t\cdot(1-\delta)^{m-t}$ for $0 \le m \le w(C)$, and the condition $\ell \in \mathcal{G}_\eta(C^{\downarrow})$ is thus equivalent to $\tilde{\ell}(0) = 1$ and $|\tilde{\ell}(i)| \le \eta$ for $i > 0$. Note that these are linear constraints on the entries of the vector v.

Also, applying Proposition V.1, the Fourier L_1 norm can be bounded by:

$$\begin{aligned}\|\widehat{\ell}\|_{L_1} &\le \sum_{y\in C^{\downarrow}}|v_{|y|}|\cdot\|\widehat{\ell_{\delta,y}}\|_{L_1} \\ &\le \|v\|_\infty\sum_{y\in C^{\downarrow}}|C|\cdot\delta^{|y|}\cdot(1+\delta)^{w(C)-|y|} \\ &\le \|v\|_\infty|C|\sum_{j=0}^{w(C)}\delta^j(1+\delta)^{w(C)-j}|\{y\in C^{\downarrow} : |y| = j\}| \\ &\le \|v\|_\infty|C|\sum_{j=0}^{w(C)}\delta^j(1+\delta)^{w(C)-j}\cdot|C|\binom{w(C)}{j} \\ &= \|v\|_\infty|C|^2(1+2\delta)^{w(C)}.\end{aligned}$$

Thus, we seek to find a vector $v = (v_0, \ldots, v_{w(C)})$ such that $\|v\|_\infty$ is as small as possible while satisfying the linear constraints dictated by the requirement $\tilde{\ell}(0) = 1$ and $|\tilde{\ell}(i)| \leq \eta$ for $i > 0$. To do this, recall that $w(C) \leq r$ and define the matrix $A_{\delta,r} \in \mathbb{R}^{(r+1)\times(r+1)}$ as

$$A_{\delta,r}(i,j) = \binom{i}{j} \cdot \delta^j \cdot (1-\delta)^{i-j}.$$

Then we have $\tilde{\ell}(m) = (A_{\delta,r} \cdot v^T)_m$. Now the task of constructing ℓ is equivalent to finding a vector v with L_∞ norm as small as possible such that $(A_{\delta,r}(i,j) \cdot v^T)_0 = 1$ and $|(A_{\delta,r}(i,j) \cdot v^T)_m| \leq \eta$ for $m > 0$.

We note that this problem is equivalent to problem of finding a "robust local inverse" for the matrix $A_{\delta,r}$, which has been studied in [6], [11]. The following theorem is an easy corollary of the main result of [11]. We provide the reduction in the full version.

Theorem V.1. (Moitra-Saks [11]) *For any $\eta > 0$, there exists $v \in \mathbb{R}^{r+1}$ such that $\|A_{\delta,r} \cdot v - e_0\|_\infty \leq \eta$, $\|v\|_\infty \leq (2/\eta)^{(1/\delta)\cdot\log(2/\delta)}$ and the zeroth coordinate of $A_{\delta,r} \cdot v$ is 1. Here $e_0 \in \mathbb{R}^{r+1}$ denotes the unit vector with 1 at the zeroth coordinate. Further, v can be computed in time* $\mathsf{poly}(r)$.

Applying this theorem directly, we have $\ell \in \mathcal{G}_\eta(C^{\downarrow})$ and $\|\widehat{\ell}\|_{L_1} \leq |C|^2 \cdot (1+2\delta)^r \cdot (2/\eta)^{\delta^{-1}\cdot\log(2\delta^{-1})}$. That finishes the proof.

VI. Proof of Theorem I.2

Without loss of generality, assume $\|f\|_1 = 1$. We may further assume $f(\mathbf{0}) > 0$ and it maximizes $|f(x)|$, thus $f(0) > 1/k$. Define $f^+ = f \cdot \mathbf{1}_{>0}$ and $f^- = -f \cdot \mathbf{1}_{<0}$, thus $f = f^+ - f^-$. Normalizing these two terms we have,

$$f = \|f^+\|_1 \cdot \frac{f^+}{\|f^+\|_1} - \|f^-\|_1 \cdot \frac{f^-}{\|f^-\|_1}.$$

If $\|f^-\|_1 = 0$ then we just omit the second term.
Here $g^+ = \frac{f^+}{\|f^+\|_1}$ and $g^- = \frac{f^-}{\|f^-\|_1}$ can be viewed as distributions supported on $\mathrm{supp}(f)$. Applying Lemma III.1 with parameter $\varepsilon = 1/k$ and $X = \mathrm{supp}(f)$, we get functions u and $\alpha : \{0,1\}^n \to \mathbb{R}$ satisfying $u(0) \in [1/2, 1]$ and $|\alpha(z)| \leq k^{\tilde{O}(1/\mu^4)}$, such that

$$\begin{aligned} |\langle u, g^+\rangle - u(0)g^+(0)| \leq{}& \frac{1}{10k}, \\ |\langle u, g^-\rangle - u(0)g^-(0)| \leq{}& \frac{1}{10k} \end{aligned} \tag{6}$$

$$\begin{aligned} \langle u, g^+\rangle &= \mathbf{E}_{z\sim T_\mu g^+}[\alpha(z)], \\ \langle u, g^-\rangle &= \mathbf{E}_{z\sim T_\mu g^-}[\alpha(z)]. \end{aligned} \tag{7}$$

We will show that

$$1/2k \leq \langle u, f\rangle \leq k^{\tilde{O}(1/\mu^4)} \cdot \|T_\mu f\|_1. \tag{8}$$

For the first part of equation (8), since $f = \|f^+\|_1 \cdot g^+ - \|f^-\|_1 \cdot g^-$, the two inequalities in (6) directly imply

$$|\langle u, f\rangle - u(0)f(0)| \leq \frac{1}{10k}(\|f^+\|_1 + \|f^-\|_1) = \frac{1}{10k}$$

Thus $\langle u, f\rangle \geq u(0)f(0) - \frac{1}{10k} \geq \frac{1}{2k}$. For the second part of equation (8), the two equations in (7) imply

$$\begin{aligned} \langle u, f\rangle &= \|f^+\|_1 \cdot \mathbf{E}_{z\sim T_\mu g^+}\alpha(z) - \|f^-\|_1 \cdot \mathbf{E}_{z\sim T_\mu g^-}\alpha(z) \\ &= \sum_{z\in\{0,1\}^n} \alpha(z) \cdot \left(\|f^+\|_1 \cdot T_\mu\left(\frac{f^+}{\|f^+\|_1}\right)(z)\right) \\ &\quad - \sum_{z\in\{0,1\}^n} \alpha(z) \cdot \left(\|f^-\|_1 \cdot T_\mu\left(\frac{f^-}{\|f^-\|_1}\right)(z)\right). \end{aligned}$$

Using the fact that

$$T_\mu f = \|f^+\|_1 \cdot T_\mu\left(\frac{f^+}{\|f^+\|_1}\right) - \|f^-\|_1 \cdot T_\mu\left(\frac{f^-}{\|f^-\|_1}\right),$$

we have,

$$\begin{aligned} \langle u, f\rangle &= \sum_{z\in\{0,1\}^n} \alpha(z) \cdot T_\mu f(z) \\ &\leq \|T_\mu f\|_1 \cdot \max_{z\in\{0,1\}^n} \alpha(z) \\ &\leq k^{\tilde{O}(1/\mu^4)} \cdot \|T_\mu f\|_1 \end{aligned}$$

These two results imply $\|T_\mu f\|_1 \geq k^{-\tilde{O}(1/\mu^4)}$, which finishes the proof.

Acknowledgments

A. D. is grateful to Rocco Servedio for many illuminating conversations about this problem.

Part of the work was done when A. D. was a postdoc at DIMACS, Rutgers. A. D. is supported by a start-up grant from Northwestern University. M. S. and S. T. are supported by NSF CCF-1218711 and by Simons Foundation award 332622.

References

[1] Lucia Batman, Russell Impagliazzo, Cody Murray, and Ramamohan Paturi. Finding heavy hitters from lossy or noisy data. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 347–362. Springer, 2013.

[2] William Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, pages 159–182, 1975.

[3] Aline Bonami. Étude des coefficients de fourier des fonctions de $l^p(g)$. In *Annales de l'institut Fourier*, volume 20, pages 335–402, 1970.

[4] Christer Borell. Positivity improving operators and hypercontractivity. *Mathematische Zeitschrift*, 180(3):225–234, 1982.

[5] Anindya De, Michael Saks, and Sijian Tang. Noisy population recovery in polynomial time (if the noise is not too high), 2015.

[6] Zeev Dvir, Anup Rao, Avi Wigderson, and Amir Yehudayoff. Restriction access. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 19–33. ACM, 2012.

[7] W. Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, 1968.

[8] Leonard Gross. Logarithmic sobolev inequalities. *American Journal of Mathematics*, 97(4):1061–1083, 1975.

[9] Shachar Lovett and Jiapeng Zhang. Improved noisy population recovery, and reverse bonami-beckner inequality for sparse functions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 137–142. ACM, 2015.

[10] Shachar Lovett and Jiapeng Zhang. Noisy population recovery from unknown noise. Technical report, Electronic Colloquium on Computational Complexity, 2016.

[11] Ankur Moitra and Michael Saks. A polynomial time algorithm for lossy population recovery. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 110–116. IEEE, 2013.

[12] Ryan O'Donnell. *Analysis of Boolean functions*. Cambridge University Press, 2014.

[13] Richard Stanley. *Enumerative Combinatorics*. Cambridge University Press, 1997.

[14] Avi Wigderson and Amir Yehudayoff. Population recovery and partial identification. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 390–399. IEEE, 2012.

Appendix

A. Proof of Lemma IV.1

Recall that $\text{FAR} = \text{FAR}_r = \{x \in X : |x| > r\}$. Define the set $E = \{y \in \{0,1\}^n : d_H(\mathbf{0}, y) \le d_H(x_i, y)$ for all $x_i \in \text{FAR}\}$ and $q = T_\mu \mathbf{1}_E$. Next, we show that q satisfies the requirements. First we state the following lemma, which is essentially identical to Lemma 3.2 in [9], that proves the first three properties we need. The proof is deferred to the full version.

Lemma A.1. *For any X and $r \ge (1/\mu^2) \cdot \log|X|$, define set* FAR *and E as above, we have:*

- $(T_\mu \mathbf{1}_E)(\mathbf{0}) \ge 1/2$.
- *For $x_i \in \mathsf{Far}$, $(T_\mu \mathbf{1}_E)(x_i) \le e^{-\frac{1}{2}\cdot\mu^2\cdot|x_i|}$.*

Clearly, the function $\mathbf{1}_E(\cdot)$ can be computed in time $\mathsf{poly}(n, |X|)$. Further, $(T_\mu \mathbf{1}_E)(0)$ can be computed to additive error ε in time $\mathsf{poly}(n, |X|, 1/\varepsilon) \cdot \log(1/\kappa)$.

For the last requirement of Lemma IV.1, we fist show how to build an unbiased estimator for $\widehat{q \cdot f}(S)$ using random sample $z \sim T_\mu f$. Since $(q \cdot f)(x) = f(x) \cdot (T_\mu \mathbf{1}_E)(x)$, we get that for any $S \subseteq \{0,1\}^n$.

$$\widehat{q \cdot f}(S) = \langle (X_S f), (T_\mu \mathbf{1}_E) \rangle = \langle (T_\mu X_S f), \mathbf{1}_E \rangle.$$

We now make two observations. The first is that for any $S \subseteq [n]$, $T_{\mu,S}$ is a self-adjoint operator. The second is that if $S, S' \subseteq [n]$ are disjoint sets, then the operators $X_{S'}$ and $T_{\mu,S}$ commute. Decomposing $T_\mu = T_{\mu,S} T_{\mu,\overline{S}}$, we have

$$T_\mu X_S f = T_{\mu,S} T_{\mu,\overline{S}} X_S f = T_{\mu,S} X_S T_{\mu,\overline{S}} f = T_{\mu,S} X_S T_{\mu,S}^{-1} T_\mu f.$$

Thus, we get

$$\begin{aligned} \widehat{q \cdot f}(S) &= \langle T_{\mu,S} X_S T_{\mu,S}^{-1} T_\mu f, \mathbf{1}_E \rangle \\ &= \mathbf{E}_{z \sim T_\mu f} \langle T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z, \mathbf{1}_E \rangle \qquad (9) \end{aligned}$$

Defining $\alpha_S(z) = \langle T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z, \mathbf{1}_E \rangle$, we can see that $\alpha_S(z)$ is an unbiased estimator for $\widehat{q \cdot f}(S)$. Now we are going to show that $\alpha_S(z)$ has the properties we need.

Lemma A.2. *For any $S \subseteq \{0,1\}^n$, $\alpha_S(z)$ can be computed in time $2^{|S|} n^{O(1)}$.*

Proof: To see this, define $A_{z,S} = \{y : y_{\overline{S}} = z_{\overline{S}}\}$. Observe that

$$\mathsf{supp}(T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z) \subseteq A_{z,S} \text{ and } |A_{z,S}| = 2^{|S|}.$$

Further, $T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z$ can be computed on any point in $A_{z,S}$ in time $2^{O(|S|)}$. Using the fact that $\mathbf{1}_E(\cdot)$ can be efficiently evaluated, we conclude that $\langle T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z, \mathbf{1}_E \rangle$ can be evaluated in time $2^{|S|} n^{O(1)}$. ∎

Lemma A.3. *For any $S \subseteq \{0,1\}^n$, $|\alpha_S(z)| \le (1/\mu)^{|S|}$.*

Proof: First we recall the following facts from [9] (Claim 3.5 in [9]).

Claim. $\|T_{\mu,i}\|_{1\to1} = 1$ *and* $\|T_{\mu,i}^{-1}\|_{1\to1} = 1/\mu$.

Proof of the Claim: The bound $\|T_{\mu,i} f\|_1 \le \|f\|_1$ is immediate, and is tight for $f = 1$. To derive the bound on $T_{\mu,i}^{-1}$, let x_0, x_1 be such that $(x_0)_i = 0$, $(x_1)_i = 1$ and $(x_0)_j = (x_1)_j$ for all $j \ne i$. If $(f(x_0), f(x_1)) = (a, b)$ then $T_{\mu,1}^{-1} f = (1/2\mu) \cdot ((1+\mu)a - (1-\mu)b, -(1-\mu)a + (1+\mu)b)$. Then $|(T_{\mu,i}^{-1} f)(x_0)| + |(T_{\mu,i}^{-1} f)(x_1)| \le (1/\mu)(|f(x_0)| + |f(x_1)|)$. The claim follows by summing over all choices for x_0, x_1, and noting that the bound is tight for $f(x) = (-1)^{x_i}$. ∎

The above immediately implies

$$\|T_{\mu,S}\|_{1\to1} \le 1, \qquad \|T_{\mu,S}^{-1}\|_{1\to1} \le (1/\mu)^{|S|}. \qquad (10)$$

Using $\|X_S\|_{1\to1} \le 1$, we know $\|T_{\mu,S} X_S T_{\mu,S}^{-1}\|_{1\to1} \le (1/\mu)^{|S|}$. This implies:

$$\begin{aligned} |\alpha_S(z)| &= |\langle T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z, \mathbf{1}_E \rangle| \le \|T_{\mu,S} X_S T_{\mu,S}^{-1} \mathbf{1}_z\|_1 \\ &\le \|T_{\mu,S} X_S T_{\mu,S}^{-1}\|_{1\to1} \le (1/\mu)^{|S|}. \end{aligned}$$

∎

Combining Lemma A.1, Lemma A.2 and Lemma A.3 we get the result.