Statistical Query Lower Bounds for Robust Statistics Problems

Ilias Diakonikolas [1] **Daniel M. Kane** [2] Alistair Stewart [3]

[1]Department of Computer Science
University of Southern California
diakonik@usc.edu

[2]Departments of CS/Math
University of California, San Diego
dakane
dakane@ucsd.edu

[3]Department of Computer Science
University of Southern California
stewart.al@gmail.com

August 14th, 2018

Outline

- The SQ Model
- Basic SQ Lower Bounds
- The Moment Matching Method
- Applications

Mean Estimation Error

Robustly estimate the mean of an ϵ-corrupted Gaussian:

- Can achieve error $O(\epsilon\sqrt{\log(1/\epsilon)})$ in polynomial time.
- Can achieve error $O(\epsilon)$ information theoretically.

Mean Estimation Error

Robustly estimate the mean of an ϵ-corrupted Gaussian:

- Can achieve error $O(\epsilon\sqrt{\log(1/\epsilon)})$ in polynomial time.
- Can achieve error $O(\epsilon)$ information theoretically.

Question: What is the best error that can be achieve efficiently?

Lower Bounds

The $\epsilon\sqrt{\log(1/\epsilon)}$ seems right, but how do we prove it?

Lower Bounds

The $\epsilon\sqrt{\log(1/\epsilon)}$ seems right, but how do we prove it?

Can't do so unconditionally, since `TukeyMedian` $\in PH$. Even hardness reductions seem difficult since they would need to be very average case.

Lower Bounds

The $\epsilon\sqrt{\log(1/\epsilon)}$ seems right, but how do we prove it?

Can't do so unconditionally, since `TukeyMedian` $\in PH$. Even hardness reductions seem difficult since they would need to be very average case.

So we work in a restricted computational model.

Model

What sorts of things do our algorithms do?

- Approximate moments of distributions.
- Approximate moments after applying filters or weights.

Model

What sorts of things do our algorithms do?

- Approximate moments of distributions.
- Approximate moments after applying filters or weights.
- Generally, approximate expectations of functions of distributions.

Statistical Queries [Kearns '93]

Given i.i.d. samples $X_1, X_2, \ldots, X_N$ can use to approximate expectations of (normalized) functions to error $O(1/\sqrt{N})$.

Statistical Queries [Kearns '93]

Given i.i.d. samples $X_1, X_2, \ldots, X_N$ can use to approximate expectations of (normalized) functions to error $O(1/\sqrt{N})$.

Query: A $\mathtt{Stat}(\tau)$ query takes a function $f : \mathbb{R}^n \to [-1, 1]$ and returns a v so that $|\mathbb{E}[f(X)] - v| \leq \tau$.

Statistical Queries [Kearns '93]

Given i.i.d. samples $X_1, X_2, \ldots, X_N$ can use to approximate expectations of (normalized) functions to error $O(1/\sqrt{N})$.

Query: A $\mathtt{Stat}(\tau)$ query takes a function $f : \mathbb{R}^n \to [-1, 1]$ and returns a v so that $|\mathbb{E}[f(X)] - v| \leq \tau$.

Model: An SQ algorithm can adaptively make statistical queries at accuracy τ.

Statistical Queries [Kearns '93]

Given i.i.d. samples $X_1, X_2, \ldots, X_N$ can use to approximate expectations of (normalized) functions to error $O(1/\sqrt{N})$.

Query: A $\texttt{Stat}(\tau)$ query takes a function $f : \mathbb{R}^n \to [-1, 1]$ and returns a v so that $|\mathbb{E}[f(X)] - v| \leq \tau$.

Model: An SQ algorithm can adaptively make statistical queries at accuracy τ. Morally, this corresponds to an algorithm with Number of Samples $O(\tau^{-2})$ and Runtime equal to number of queries.

Power of SQ

- **Restricted Model:** Hope to prove unconditional lower bounds information-theoretically.

Power of SQ

- **Restricted Model:** Hope to prove unconditional lower bounds information-theoretically.
- **Powerful Model:** Wide range of algorithmic techniques can be formalized in SQ:
  - Filter & Convex Program techniques for robust statistics.
  - PAC learning for AC^0, decision trees, linear separators, boosting.
  - Unsupervised Learning: stochastic convex optimization, moment-based methods, k-means clustering, EM,

    ... [Feldman-Grigorescu-Reyzin-Vempala-Xiao/JACM17]

Power of SQ

- **Restricted Model:** Hope to prove unconditional lower bounds information-theoretically.
- **Powerful Model:** Wide range of algorithmic techniques can be formalized in SQ:
  - Filter & Convex Program techniques for robust statistics.
  - PAC learning for AC^0, decision trees, linear separators, boosting.
  - Unsupervised Learning: stochastic convex optimization, moment-based methods, k-means clustering, EM, . . . [Feldman-Grigorescu-Reyzin-Vempala-Xiao/JACM17]
- **Only Major Exception:** Gaussian elimination over finite fields (for example, for learning parity).

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

Test function f

$$\mathbb{E}[f(X)] = (\hat{f}(1) + \hat{f}(\chi))$$

where X is uniform over the halfspace defined by the character χ.

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

Test function f

$$\mathbb{E}[f(X)] = (\hat{f}(1) + \hat{f}(\chi))$$

where X is uniform over the halfspace defined by the character χ.

Oracle could return $\hat{f}(1)$ unless $|\hat{f}(\chi)| \geq \tau$.

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

Test function f

$$\mathbb{E}[f(X)] = (\hat{f}(1) + \hat{f}(\chi))$$

where X is uniform over the halfspace defined by the character χ.

Oracle could return $\hat{f}(1)$ unless $|\hat{f}(\chi)| \geq \tau$. Plancherel Inequality says

$$\sum_{\chi} |\hat{f}(\chi)|^2 = \mathbb{E}_{y \in_u \mathbb{F}_2^n}[|f(y)|^2] \leq 1.$$

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

Test function f

$$\mathbb{E}[f(X)] = (\hat{f}(1) + \hat{f}(\chi))$$

where X is uniform over the halfspace defined by the character χ.

Oracle could return $\hat{f}(1)$ unless $|\hat{f}(\chi)| \geq \tau$. Plancherel Inequality says

$$\sum_{\chi} |\hat{f}(\chi)|^2 = \mathbb{E}_{y \in_u \mathbb{F}_2^n}[|f(y)|^2] \leq 1.$$

Unless τ is exponentially small, $|\hat{f}(\chi)| > \tau$ with exponentially small probability over choice of X.

SQ Lower Bounds [Kearns]

Example: Learning parities: X is the uniform distribution over a random dimension $n-1$ subspace of $\mathbb{F}_2^n$.

Test function f

$$\mathbb{E}[f(X)] = (\hat{f}(1) + \hat{f}(\chi))$$

where X is uniform over the halfspace defined by the character χ.

Oracle could return $\hat{f}(1)$ unless $|\hat{f}(\chi)| \geq \tau$. Plancherel Inequality says

$$\sum_{\chi} |\hat{f}(\chi)|^2 = \mathbb{E}_{y \in_u \mathbb{F}_2^n}[|f(y)|^2] \leq 1.$$

Unless τ is exponentially small, $|\hat{f}(\chi)| > \tau$ with exponentially small probability over choice of X.

Upshot: Either τ exponentially small, or exponentially many queries required.

General Lower Bound Method

Need:

- Many possible distributions X_i, pretending to be like some distribution D (the uniform distribution in the previous example).
- The differences between X_i and D are nearly orthogonal.

General Lower Bound Method

Need:

- Many possible distributions X_i, pretending to be like some distribution D (the uniform distribution in the previous example).
- The differences between X_i and D are nearly orthogonal.

Use χ^2 inner product: $\chi^2_D(X_1, X_2) := \int_{\mathbb{R}^n} X_1(x)X_2(x)/D(x)dx - 1.$

General Lower Bound Method

Need:

- Many possible distributions X_i, pretending to be like some distribution D (the uniform distribution in the previous example).
- The differences between X_i and D are nearly orthogonal.

Use χ^2 inner product: $\chi^2_D(X_1, X_2) := \int_{\mathbb{R}^n} X_1(x)X_2(x)/D(x)dx - 1.$

Theorem (Feldman-Grigorescu-Reyzin-Vempala-Xiao '13)

Suppose that there are distributions $X_1, X_2, \ldots, X_m$ and D so that for all i, j

$$|\chi^2_D(X_i, X_j)| \leq \begin{cases} \gamma & \text{if } i \neq j \\ \beta & \text{if } i = j \end{cases}$$

General Lower Bound Method

Need:

- Many possible distributions X_i, pretending to be like some distribution D (the uniform distribution in the previous example).
- The differences between X_i and D are nearly orthogonal.

Use χ^2 inner product: $\chi^2_D(X_1, X_2) := \int_{\mathbb{R}^n} X_1(x)X_2(x)/D(x)dx - 1.$

Theorem (Feldman-Grigorescu-Reyzin-Vempala-Xiao '13)

Suppose that there are distributions $X_1, X_2, \ldots, X_m$ and D so that for all i, j

$$|\chi^2_D(X_i, X_j)| \leq \begin{cases} \gamma & \text{if } i \neq j \\ \beta & \text{if } i = j \end{cases}$$

Then any statistical query algorithm for learning which of the X_i a distribution is must use either queries of accuracy $O(\sqrt{\gamma})$ or a number of queries $\Omega(m\gamma/\beta)$.

Lower Bound for Robust Mean

- Take $D = N(0, I)$.

Lower Bound for Robust Mean

- Take $D = N(0, I)$.
- **Need:** distributions $X_1, X_2, \ldots, X_m$ so that:
  - $d_{TV}(X_i, N(\mu_i, I)) \leq \epsilon$.
  - $|\mu_i - \mu_j|$ large for all $i \neq j$.
  - $|\chi^2_D(X_i, X_j)|$ small for all i, j.
  - m is large.

Lower Bound for Robust Mean

- Take $D = N(0, I)$.
- **Need:** distributions $X_1, X_2, \ldots, X_m$ so that:
  - $d_{TV}(X_i, N(\mu_i, I)) \leq \epsilon$.
  - $|\mu_i - \mu_j|$ large for all $i \neq j$.
  - $|\chi^2_D(X_i, X_j)|$ small for all i, j.
  - m is large.
- SQ algorithms *can* detect moments. Try to make low degree moments of X_i agree with D.

Moment Matching

Consider $1D$ problem:

- $D = N(0,1)$.
- A is ϵ-close to $N(\mu, 1)$.
- First d moments of A and D agree.

Moment Matching

Consider $1D$ problem:

- $D = N(0, 1)$.
- A is ϵ-close to $N(\mu, 1)$.
- First d moments of A and D agree.

$$A(x) = G(x - \mu) + E(x)$$

Moment Matching

$$A(x) = G(x - \mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

Moment Matching

$$A(x) = G(x - \mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

1. $|E(x)| \leq G(x - \mu)$ pointwise.

Moment Matching

$$A(x) = G(x - \mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

1. $|E(x)| \leq G(x - \mu)$ pointwise.
2. $|E|_1 \leq \epsilon$.

Moment Matching

$$A(x) = G(x - \mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

1. $|E(x)| \leq G(x - \mu)$ pointwise.
2. $|E|_1 \leq \epsilon$.
3. $E(x)$ matches first d moments with $G(x) - G(x - \mu)$. For $k \leq d$

$$\int E(x) x^k dx = \mathbb{E}[G^k - (G + \mu)^k] = O_k(\mu)$$

Moment Matching

$$A(x) = G(x-\mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

1. $|E(x)| \leq G(x-\mu)$ pointwise.
2. $|E|_1 \leq \epsilon$.
3. $E(x)$ matches first d moments with $G(x) - G(x-\mu)$. For $k \leq d$

$$\int E(x) x^k dx = \mathbb{E}[G^k - (G+\mu)^k] = O_k(\mu)$$

Idea: $E(x) = p(x)\mathbf{1}(|x| < \sqrt{\log(1/\epsilon)}/2)$.

- p is the *unique* degree-d polynomial so that (3) holds.
  - $|p|_\infty$ has size $O_d(\mu/\log(1/\epsilon))$.

Moment Matching

$$A(x) = G(x - \mu) + E(x)$$

where $G(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ is the Gaussian pdf and:

1. $|E(x)| \leq G(x - \mu)$ pointwise.
2. $|E|_1 \leq \epsilon$.
3. $E(x)$ matches first d moments with $G(x) - G(x - \mu)$. For $k \leq d$

$$\int E(x) x^k dx = \mathbb{E}[G^k - (G + \mu)^k] = O_k(\mu)$$

Idea: $E(x) = p(x)\mathbf{1}(|x| < \sqrt{\log(1/\epsilon)}/2)$.

- p is the *unique* degree-d polynomial so that (3) holds.
  - $|p|_\infty$ has size $O_d(\mu/\log(1/\epsilon))$.
- (1) holds since $G(x - \mu) = \Omega(\sqrt{\epsilon})$ on the support of E.
- $|E|_1 = O_d(\mu/\sqrt{\log(1/\epsilon)})$, so (2) holds if $\mu \ll_d \epsilon\sqrt{\log(1/\epsilon)}$.

Higher Dimensions

How do we make this n-dimensional?

Higher Dimensions

How do we make this n-dimensional?

Idea: Have a copy of A in one direction, and standard Gaussian in orthogonal directions.

Higher Dimensions

How do we make this *n*-dimensional?

Idea: Have a copy of A in one direction, and standard Gaussian in orthogonal directions.

For unit vectors v,

$$P_v(x) = A(v \cdot x)(2\pi)^{-(n-1)/2} e^{-(|x|^2 - (x \cdot v)^2)/2}.$$

Higher Dimensions

How do we make this *n*-dimensional?

Idea: Have a copy of A in one direction, and standard Gaussian in orthogonal directions.

For unit vectors v,

$$P_v(x) = A(v \cdot x)(2\pi)^{-(n-1)/2} e^{-(|x|^2 - (x \cdot v)^2)/2}.$$

If u and v are orthogonal, $\chi^2_D(P_u, P_v) = 0$. Can only fit n mutually orthogonal vectors, so what happens if u, v are *nearly* orthogonal?

Computation

Want to evaluate:

$$\int_{\mathbb{R}^n} P_u(x) P_v(x) / G(x) dx.$$

Computation

Want to evaluate:

$$\int_{\mathbb{R}^n} P_u(x)P_v(x)/G(x)dx.$$

In directions orthogonal to u and v, get standard Gaussian and integrate out to 1.

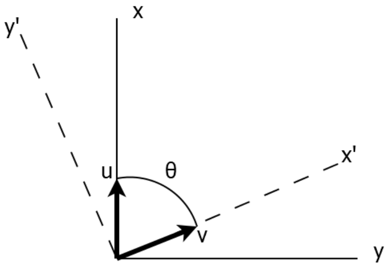
Computation

Want to evaluate:

$$\int_{\mathbb{R}^n} P_u(x)P_v(x)/G(x)dx.$$

In directions orthogonal to u and v, get standard Gaussian and integrate out to 1. Get

$$\int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy.$$



Computation

$$\int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy.$$

Integrate out over y:

$$Q(x) = \int A(x')G(y')dy$$

Computation

$$\int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy.$$

Integrate out over y:

$$\begin{aligned} Q(x) &= \int A(x')G(y')dy \\ &= \int A(x\cos(\theta) + y\sin(\theta))G(x\sin(\theta) - y\cos(\theta))dy \end{aligned}$$

Computation

$$\int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy.$$

Integrate out over y:

$$\begin{aligned} Q(x) &= \int A(x')G(y')dy \\ &= \int A(x\cos(\theta) + y\sin(\theta))G(x\sin(\theta) - y\cos(\theta))dy \\ &= U_\theta A(x) \end{aligned}$$

where U_θ is the Ornstein-Uhlenbeck operator on functions $f : \mathbb{R} \to \mathbb{R}$.

Eigenfunctions of the Ornstein-Uhlenbeck Operator

Linear operator U_θ on functions $f : \mathbb{R} \to \mathbb{R}$:

$$U_\theta f(x) = \int f(x\cos(\theta) + y\sin(\theta))G(x\sin(\theta) - y\cos(\theta))dy$$

Eigenfunctions of the Ornstein-Uhlenbeck Operator

Linear operator U_θ on functions $f : \mathbb{R} \to \mathbb{R}$:

$$U_\theta f(x) = \int f(x\cos(\theta) + y\sin(\theta))G(x\sin(\theta) - y\cos(\theta))dy$$

Fact (Mehler '66)

$$U_\theta(H_k G) = \cos^k(\theta) H_k G.$$

Where H_k is the degree-k Hermite polynomial. They form an orthonormal basis for the inner product

$$\langle f, g \rangle = \int f(x)g(x)G(x)dx.$$

Computation

$$\chi_D^2(P_u, P_v) + 1 = \int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy$$
$$= \int_{\mathbb{R}} A(x)(U_\theta A)(x)/G(x)dx.$$

Computation

$$\chi_D^2(P_u, P_v) + 1 = \int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy$$
$$= \int_{\mathbb{R}} A(x)(U_\theta A)(x)/G(x)dx.$$

Write

$$A(x) = \sum_{k=0}^{\infty} a_k H_k(x) G(x).$$

Computation

$$\chi^2_D(P_u, P_v) + 1 = \int_{\mathbb{R}^2} A(x)G(y)A(x')G(y')/G(x)G(y)dxdy$$
$$= \int_{\mathbb{R}} A(x)(U_\theta A)(x)/G(x)dx.$$

Write

$$A(x) = \sum_{k=0}^{\infty} a_k H_k(x) G(x).$$

$$\chi^2_D(P_u, P_v) + 1 = \int_{\mathbb{R}} \left(\sum_k \cos^k(\theta) a_k H_k(x) \right) \left(\sum_{k'} a_{k'} H_{k'}(x) \right) G(x)dx$$
$$= \sum_{k=0}^{\infty} \cos^k(\theta) a_k^2.$$

Coefficients

Note that

$$a_k = \int_{\mathbb{R}} H_k(x)A(x)dx.$$

Coefficients

Note that

$$a_k = \int_{\mathbb{R}} H_k(x)A(x)dx.$$

If $A(x)$ and $G(x)$ match first d moments, then for $k \leq d$,

$$a_k = \int_{\mathbb{R}} H_k(x)A(x)dx = \int_{\mathbb{R}} H_k(x)G(x)dx = \begin{cases} 1 & k = 0 \\ 0 & k > 0 \end{cases}$$

Coefficients

Note that

$$a_k = \int_{\mathbb{R}} H_k(x)A(x)dx.$$

If $A(x)$ and $G(x)$ match first d moments, then for $k \leq d$,

$$a_k = \int_{\mathbb{R}} H_k(x)A(x)dx = \int_{\mathbb{R}} H_k(x)G(x)dx = \begin{cases} 1 & k = 0 \\ 0 & k > 0 \end{cases}$$

Also

$$\chi^2_G(A, A) + 1 = \int_{\mathbb{R}} A(x)^2/G(x)dx = \int_{\mathbb{R}} \sum_{k,k'} a_k a_{k'} H_k(x) H_{k'}(x) G(x)dx$$

$$= \sum_{k=0}^{\infty} a_k^2.$$

Key Lemma

So

$$\chi^2_D(P_u, P_v) = \sum_{k=0}^{\infty} \cos^k(\theta) a_k^2 - 1$$
$$= \sum_{k>d}^{\infty} \cos^k(\theta) a_k^2$$
$$\leq \cos^{d+1}(\theta) \chi^2_G(A, A).$$

Key Lemma

So

$$\chi^2_D(P_u, P_v) = \sum_{k=0}^{\infty} \cos^k(\theta) a_k^2 - 1$$
$$= \sum_{k>d}^{\infty} \cos^k(\theta) a_k^2$$
$$\leq \cos^{d+1}(\theta) \chi^2_G(A, A).$$

Lemma

If $A(x)$ is a one dimensional distribution whose first d moments agree with $G(x)$, then for vectors u and v,

$$|\chi^2_D(P_u, P_v)| \leq |u \cdot v|^{d+1} \chi^2_G(A, A).$$

Packing

Have $\chi^2_D(P_u, P_v)$ small if $u \cdot v$ is. For lower bound need many distributions that are pairwise nearly othrogonal.

Lemma

For $1/2 > c > 0$, there exists a collection of $2^{\Omega(n^{1-2c})}$ unit vectors whose pairwise dot products are at most n^{-c}.

Putting it Together

- Can find A, ϵ-close to $N(\mu, 1)$ matching first d moments with G with $\mu \geq \epsilon\sqrt{\log(1/\epsilon)}/\mathrm{poly}(d)$.

Putting it Together

- Can find A, ϵ-close to $N(\mu, 1)$ matching first d moments with G with $\mu \geq \epsilon\sqrt{\log(1/\epsilon)}/\mathrm{poly}(d)$.
- Have $2^{\Omega(n^{1/3})}$ vectors v_i with $|v_i \cdot v_j| \leq n^{-1/3}$.

Putting it Together

- Can find A, ϵ-close to $N(\mu, 1)$ matching first d moments with G with $\mu \geq \epsilon\sqrt{\log(1/\epsilon)}/\mathrm{poly}(d)$.
- Have $2^{\Omega(n^{1/3})}$ vectors v_i with $|v_i \cdot v_j| \leq n^{-1/3}$.
- $|\chi^2_D(P_{v_i}, P_{v_j})| = O(n^{-d/3})$.

Putting it Together

- Can find A, ϵ-close to $N(\mu, 1)$ matching first d moments with G with $\mu \geq \epsilon\sqrt{\log(1/\epsilon)}/\mathrm{poly}(d)$.
- Have $2^{\Omega(n^{1/3})}$ vectors v_i with $|v_i \cdot v_j| \leq n^{-1/3}$.
- $|\chi^2_D(P_{v_i}, P_{v_j})| = O(n^{-d/3})$.
- The P_{v_i} are ϵ-corrupted Gaussians with means differing by at least μ.

Putting it Together

- Can find A, ϵ-close to $N(\mu, 1)$ matching first d moments with G with $\mu \geq \epsilon\sqrt{\log(1/\epsilon)}/\mathrm{poly}(d)$.
- Have $2^{\Omega(n^{1/3})}$ vectors v_i with $|v_i \cdot v_j| \leq n^{-1/3}$.
- $|\chi^2_D(P_{v_i}, P_{v_j})| = O(n^{-d/3})$.
- The P_{v_i} are ϵ-corrupted Gaussians with means differing by at least μ.

Theorem

Any SQ algorithm that learns the mean of an ϵ-corrupted Gaussian to error better than $\epsilon\sqrt{\log(1/\epsilon)}/M$ must either make queries with accuracy $n^{-poly(M)}$ or a number of queries $2^{n^{1/3}}$.

Notes

Remark

The lower bound requires both additive and subtractive error. In the Huber model, can achieve $O(\epsilon)$ error in polynomial time.

Notes

Remark

The lower bound requires both additive and subtractive error. In the Huber model, can achieve $O(\epsilon)$ error in polynomial time.

Remark

Improvement is tight up to the polynomial in M. There is an algorithm achieving error $O(\epsilon\sqrt{\log(1/\epsilon)}/M + \epsilon)$ error in $(n/\epsilon)^{poly(M)}$ time.

Algorithm

1. Obtain a rough approximation $\hat{\mu}$ to μ.
2. Approximate the higher moment tensors of X.
3. If for any k the k^{th} moments differ too much from those of $N(\hat{\mu}, I)$, create a filter.
4. Otherwise, only a few directions in which higher moments are non-trivial. μ is close to sample mean in the trivial directions.
5. Brute force the mean in the non-trivial directions.

Other Applications

This technique is very general and has a number of other applications for proving SQ lower bounds in a number of Gaussian-like problems.

Robust Covariance

We show that it's hard to robustly learn the covariance to error $o(\epsilon \log(1/\epsilon))$.

Robust Covariance

We show that it's hard to robustly learn the covariance to error $o(\epsilon \log(1/\epsilon))$.

- Need one dimensional distribution:

$$A(x) = G(x/\sigma) + E(x)$$

Robust Covariance

We show that it's hard to robustly learn the covariance to error $o(\epsilon \log(1/\epsilon))$.

- Need one dimensional distribution:

$$A(x) = G(x/\sigma) + E(x)$$

- Once again $E(x)$:
  - Needs to fix first d moments
  - Is supported on an interval of length $O(\sqrt{\log(1/\epsilon)})$
  - Has L^1 norm $O(\epsilon)$.

Robust Covariance

We show that it's hard to robustly learn the covariance to error $o(\epsilon \log(1/\epsilon))$.

- Need one dimensional distribution:

$$A(x) = G(x/\sigma) + E(x)$$

- Once again $E(x)$:
  - Needs to fix first d moments
  - Is supported on an interval of length $O(\sqrt{\log(1/\epsilon)})$
  - Has L^1 norm $O(\epsilon)$.
- Needs to fix moments by $O(\sigma - 1)$, but only needs to fix second on higher degree moments.
- Can do for $\sigma = 1 + \Omega_d(\epsilon \log(1/\epsilon))$.

Result

Theorem

Any SQ algorithm that learns the covariance of an ϵ-corrupted Gaussian to error better than $\epsilon \log(1/\epsilon)/M$ must either make queries with accuracy $n^{-poly(M)}$ or a number of queries $2^{n^{1/3}}$.

Covariance Sample Complexity

To learn the covariance (even in operator norm) robustly, all known algorithms require $\Omega(n^2)$ samples, however information-theoretically, only $O(n)$ are required.

Covariance Sample Complexity

To learn the covariance (even in operator norm) robustly, all known algorithms require $\Omega(n^2)$ samples, however information-theoretically, only $O(n)$ are required.

Can prove SQ lower bound.

Covariance Sample Complexity

One dimensional version:

$$A(x) = (1-\epsilon)N(0,1/2) + (\epsilon/2)N(\sqrt{2/\epsilon},1/2) + (\epsilon/2)N(-\sqrt{2/\epsilon},1/2)$$

- Matches 3 moments with $N(0,1)$.

Covariance Sample Complexity

One dimensional version:

$$A(x) = (1-\epsilon)N(0,1/2) + (\epsilon/2)N(\sqrt{2/\epsilon},1/2) + (\epsilon/2)N(-\sqrt{2/\epsilon},1/2)$$

- Matches 3 moments with $N(0,1)$.

Have $2^{n^{\Omega(1)}}$ vectors v_i with pairwise dot products $n^{-0.499}$.

Covariance Sample Complexity

One dimensional version:

$$A(x) = (1-\epsilon)N(0,1/2) + (\epsilon/2)N(\sqrt{2/\epsilon},1/2) + (\epsilon/2)N(-\sqrt{2/\epsilon},1/2)$$

- Matches 3 moments with $N(0,1)$.

Have $2^{n^{\Omega(1)}}$ vectors v_i with pairwise dot products $n^{-0.499}$. Gives many P_{v_i} with χ^2 at most $n^{-1.99}$.

Covariance Sample Complexity

One dimensional version:

$$A(x) = (1-\epsilon)N(0,1/2) + (\epsilon/2)N(\sqrt{2/\epsilon},1/2) + (\epsilon/2)N(-\sqrt{2/\epsilon},1/2)$$

- Matches 3 moments with $N(0,1)$.

Have $2^{n^{\Omega(1)}}$ vectors v_i with pairwise dot products $n^{-0.499}$. Gives many P_{v_i} with χ^2 at most $n^{-1.99}$.

Theorem

For ϵ sufficiently large (a careful analysis allows anything subpolynomial), any SQ algorithm that learns the covariance of an ϵ-corrupted Gaussian to constant error needs either queries of accuracy $n^{-0.99}$, or $2^{n^{\Omega(1)}}$ queries.

Covariance Sample Complexity

One dimensional version:

$$A(x) = (1-\epsilon)N(0,1/2) + (\epsilon/2)N(\sqrt{2/\epsilon},1/2) + (\epsilon/2)N(-\sqrt{2/\epsilon},1/2)$$

- Matches 3 moments with $N(0,1)$.

Have $2^{n^{\Omega(1)}}$ vectors v_i with pairwise dot products $n^{-0.499}$. Gives many P_{v_i} with χ^2 at most $n^{-1.99}$.

Theorem

For ϵ sufficiently large (a careful analysis allows anything subpolynomial), any SQ algorithm that learns the covariance of an ϵ-corrupted Gaussian to constant error needs either queries of accuracy $n^{-0.99}$, or $2^{n^{\Omega(1)}}$ queries.

Morally, this means we need either $n^{1.99}$ samples, or exponential time.

Robust Mean Testing

Problem: Given a distribution X that is either:

1. $N(0, I)$
2. An ϵ-corrupted version of $N(\mu, I)$ for some $|\mu| > \delta$

Determine with $2/3$ probability which case we are in.

Robust Mean Testing

Problem: Given a distribution X that is either:

1. $N(0, I)$
2. An ϵ-corrupted version of $N(\mu, I)$ for some $|\mu| > \delta$

Determine with $2/3$ probability which case we are in.

Remark

In the noiseless case, this requires only $O(\sqrt{n}/\delta^2)$ samples, which is much better than the complexity of $O(n/\delta^2)$ required for learning.

Lower Bound Framework

If $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$, construct moment matching A's and P_v as before.

Lower Bound Framework

If $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$, construct moment matching A's and P_v as before.

- Prove *information-theoretic* lower bounds.

Lower Bound Framework

If $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$, construct moment matching A's and P_v as before.

- Prove *information-theoretic* lower bounds.
- Find collection of v_i, make X either $N(0, I)$ or a random P_{v_i}.

Lower Bound Framework

If $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$, construct moment matching A's and P_v as before.

- Prove *information-theoretic* lower bounds.
- Find collection of v_i, make X either $N(0, I)$ or a random P_{v_i}.
- Either see sample from G^N or from $P_{v_*}^N$ (pick random v_i and return N i.i.d samples from P_{v_i}).
  - Can you distinguish G^N from $P_{v_*}^N$?

Lower Bound Framework

If $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$, construct moment matching A's and P_v as before.

- Prove *information-theoretic* lower bounds.
- Find collection of v_i, make X either $N(0, I)$ or a random P_{v_i}.
- Either see sample from G^N or from $P_{v_*}^N$ (pick random v_i and return N i.i.d samples from P_{v_i}).
  - Can you distinguish G^N from $P_{v_*}^N$?
- Enough to show $\chi^2_{G^N}(P_{v_*}^N, P_{v_*}^N)$ is small.

Calculation

$$\begin{aligned}\chi^2_{G^N}(P^N_{v_*}, P^N_{v_*}) + 1 &= \mathbb{E}_{i,j}[\chi^2_{G^N}(P^N_{v_i}, P^N_{v_j}) + 1] \\ &= \mathbb{E}_{i,j}[(\chi^2_G(P_{v_i}, P_{v_j}) + 1)^N] \\ &= \mathbb{E}_{i,j}[(1 + O(v_i \cdot v_j)^d)^N].\end{aligned}$$

Calculation

$$\begin{aligned}\chi^2_{G^N}(P^N_{v_*}, P^N_{v_*}) + 1 &= \mathbb{E}_{i,j}[\chi^2_{G^N}(P^N_{v_i}, P^N_{v_j}) + 1] \\ &= \mathbb{E}_{i,j}[(\chi^2_G(P_{v_i}, P_{v_j}) + 1)^N] \\ &= \mathbb{E}_{i,j}[(1 + O(v_i \cdot v_j)^d)^N].\end{aligned}$$

There's a $1/m$ probability that $i = j$ and then have $2^{O(N)}$. Otherwise, have $\exp(O((v_i \cdot v_j)^d N))$.

Calculation

$$\begin{aligned}\chi^2_{G^N}(P^N_{v_*}, P^N_{v_*}) + 1 &= \mathbb{E}_{i,j}[\chi^2_{G^N}(P^N_{v_i}, P^N_{v_j}) + 1] \\ &= \mathbb{E}_{i,j}[(\chi^2_G(P_{v_i}, P_{v_j}) + 1)^N] \\ &= \mathbb{E}_{i,j}[(1 + O(v_i \cdot v_j)^d)^N].\end{aligned}$$

There's a $1/m$ probability that $i = j$ and then have $2^{O(N)}$. Otherwise, have $\exp(O((v_i \cdot v_j)^d N))$.

Small if:

1. $2^{O(N)} \ll m$.
2. $(v_i \cdot v_j)^d \ll 1/N$ for all $i \neq j$.

Result

Can pick $m = 2^{n^{0.999}}$ vectors with $|v_i \cdot v_j| < n^{-\Omega(1)}$. Taking d large enough, $N < |v_i \cdot v_j|^{-d}$.

Result

Can pick $m = 2^{n^{0.999}}$ vectors with $|v_i \cdot v_j| < n^{-\Omega(1)}$. Taking d large enough, $N < |v_i \cdot v_j|^{-d}$.

Theorem

Any algorithm to robustly test the mean of a Gaussian for $\delta = o(\epsilon\sqrt{\log(1/\epsilon)})$ requires at least $n^{0.99}$ samples.

Conclusions

We have a general framework for proving computational lower bounds for Gaussian-ish learning problems that yields near-optimal bounds in a number of cases.