Computational-Statistical Tradeoffs in Robust Estimation

Ilias Diakonikolas (USC)

(based on joint work with D. Kane and A. Stewart)

ROBUST HIGH-DIMENSIONAL ESTIMATION

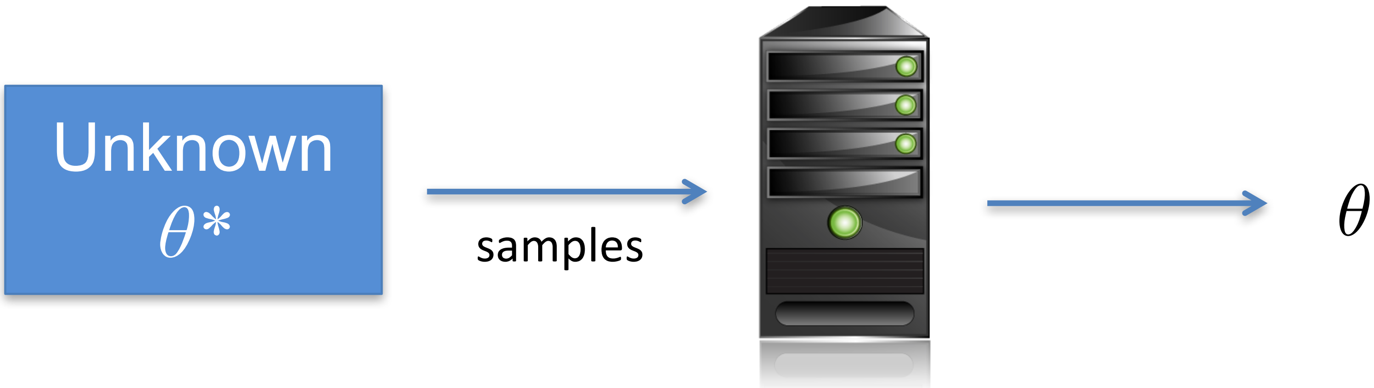
Can we develop learning/estimation algorithms that are ***robust*** to a constant fraction of ***corruptions*** in the data?

Contamination Model:

Let $\mathcal{F}$ be a family of high-dimensional distributions. We say that a distribution F' is ϵ - corrupted with respect to $\mathcal{F}$ if there exists $F \in \mathcal{F}$ such that

$$d_{\mathrm{TV}}(F', F) \leq \epsilon \, .$$

THE UNSUPERVISED LEARNING PROBLEM



- *Input*: sample generated by model with unknown θ^*
- *Goal*: estimate parameters θ so that $\theta \approx \theta^*$

Question 1: Is there an *efficient* learning algorithm?

Main performance criteria:

- Sample size
- Running time
- Robustness

Question 2: Are there *tradeoffs* between these criteria?

ROBUSTLY LEARNING A GAUSSIAN – PRIOR WORK

Basic Problem: Given an ϵ - corrupted version F' of an unknown d-dimensional unknown mean Gaussian

$$\mathcal{N}(\mu, I)$$

efficiently compute a hypothesis distribution H such that

$$d_{\mathrm{TV}}(H, \mathcal{N}(\mu, I)) \leq O(\epsilon) \ .$$

- Extensively studied in robust statistics since the 1960's. Till recently, known efficient estimators get error $\Omega(\epsilon \cdot \sqrt{d})$.
- Recent Algorithmic Progress:

-- **[Lai-Rao-Vempala'16]** $O\left(\epsilon\sqrt{\log(1/\epsilon)} \cdot \sqrt{\log d}\right)$.

-- **[D-Kamath-Kane-Li-Moitra-Stewart'16]** $O\left(\epsilon\sqrt{\log(1/\epsilon)}\right)$.

ROBUSTLY LEARNING A GAUSSIAN

Basic Problem: Given an ϵ - corrupted version F' of an unknown d-dimensional unknown mean Gaussian

$$\mathcal{N}(\mu, I)$$

efficiently compute a hypothesis distribution H such that

$$d_{\mathrm{TV}}(H, \mathcal{N}(\mu, I)) \leq O(\epsilon) \ .$$

$O(\epsilon)$ error is the information-theoretically best possible.

ROBUST LEARNING – OPEN QUESTION

Summary of Prior Work: There is a $\mathrm{poly}(d/\epsilon)$ time algorithm for robustly learning $\mathcal{N}(\mu, I)$ within error $O\left(\epsilon\sqrt{\log(1/\epsilon)}\right)$.

Open Question: Is there a $\mathrm{poly}(d/\epsilon)$ time algorithm for robustly learning $\mathcal{N}(\mu, I)$ within error $o(\epsilon\sqrt{\log(1/\epsilon)})$?
How about $O(\epsilon)$?

OUTLINE

Part I: Introduction

- Unsupervised Learning in High Dimension
- **Statistical Query (SQ) Learning Model**
- Our Results

Part II: Computational SQ Lower Bounds

- Generic SQ Lower Bound Technique
- Two Applications: Learning GMMs, Robustly Learning a Gaussian

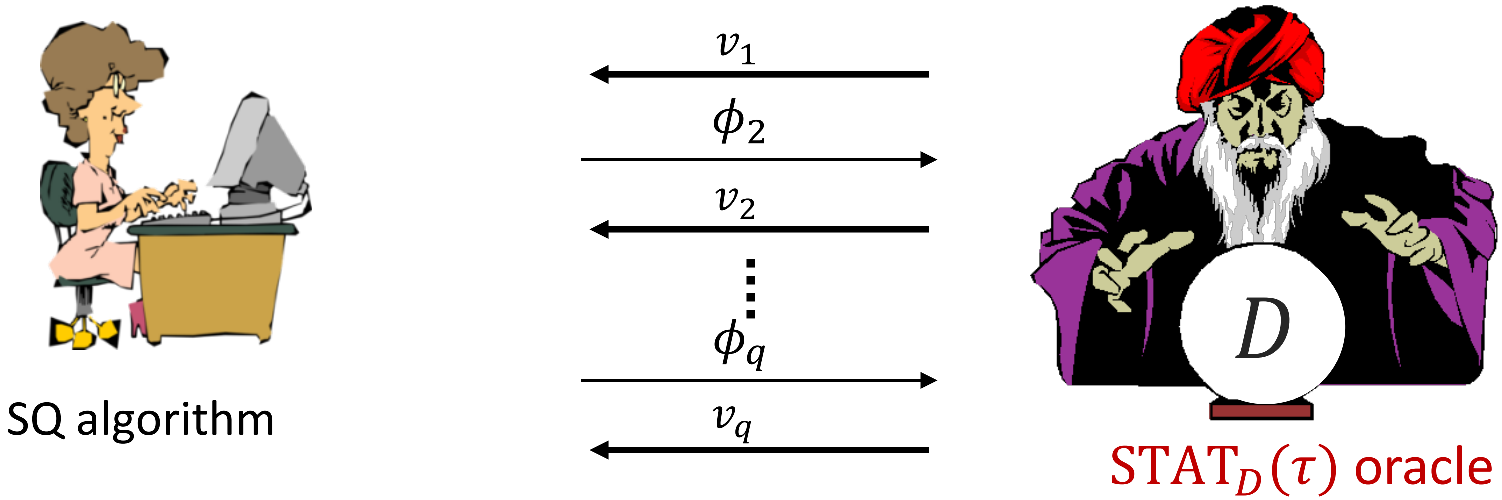
Part III: Extensions

Part IV: Summary and Conclusions

STATISTICAL QUERIES [KEARNS' 93]

$x_1, x_2, \dots, x_m \sim D$ over X

STATISTICAL QUERIES [KEARNS' 93]



$$\phi_1 : X \to [-1,1] \qquad |v_1 - \mathbf{E}_{x\sim D}[\phi_1(x)]| \leq \tau$$

τ is tolerance of the query; $\tau = 1/\sqrt{m}$

Problem $P \in \mathrm{SQCompl}(q, m)$:
If exists a SQ algorithm that solves P using q queries to $\mathrm{STAT}_D(\tau = 1/\sqrt{m})$

POWER OF SQ ALGORITHMS (?)

Restricted Model: Hope to prove unconditional computational lower bounds.

Powerful Model: Wide range of algorithmic techniques in ML are implementable using SQs*:

- PAC Learning: AC^0, decision trees, linear separators, boosting.
- Unsupervised Learning: stochastic convex optimization, moment-based methods, k-means clustering, EM, …
  [Feldman-Grigorescu-Reyzin-Vempala-Xiao/JACM'17]

Only known exception: Gaussian elimination over finite fields (e.g., learning parities).

For all problems in this talk, strongest known algorithms are SQ.

METHODOLOGY FOR SQ LOWER BOUNDS

Statistical Query Dimension:

- Fixed-distribution PAC Learning
  [Blum-Furst-Jackson-Kearns-Mansour-Rudich’95; …]

- General Statistical Problems
  [Feldman-Grigorescu-Reyzin-Vempala-Xiao’13, …, Feldman’16]

Pairwise correlation between D_1 and D_2 with respect to D:

$$\chi_D(D_1, D_2) := \int_{\mathbb{R}^d} D_1(x) D_2(x) / D(x) dx - 1$$

Fact: Suffices to construct a large set of distributions that are *nearly* uncorrelated.

OUTLINE

Part I: Introduction

- Unsupervised Learning in High Dimension
- Statistical Query (SQ) Learning Model
- **Our Results**

Part II: Computational SQ Lower Bounds

- Generic SQ Lower Bound Technique
- Two Applications: Learning GMMs, Robustly Learning a Gaussian

Part III: Summary and Conclusions

STATISTICAL QUERY LOWER BOUND FOR ROBUSTLY LEARNING A GAUSSIAN

Theorem: Suppose $d \geq \mathrm{polylog}(1/\epsilon)$. Any SQ algorithm that learns an ϵ - corrupted Gaussian $\mathcal{N}(\mu, I)$ within statistical distance error

$$O(\epsilon\sqrt{\log(1/\epsilon)}/M)$$

requires either:

- SQ queries of accuracy $d^{-M/6}$

or

- At least

$$d^{\Omega(M^{1/2})}$$

many SQ queries.

Take-away: Any asymptotic improvement in error guarantee over prior work requires super-polynomial time.

GENERAL LOWER BOUND CONSTRUCTION

General Technique for SQ Lower Bounds:
Leads to Tight Lower Bounds
for a range of High-dimensional Estimation Tasks

Concrete Applications of our Technique:

- Robustly Learning the Mean and Covariance
- Learning Gaussian Mixture Models (GMMs)
- Statistical-Computational Tradeoffs
- Robustly Testing a Gaussian

APPLICATIONS: CONCRETE SQ LOWER BOUNDS

Unified technique yielding a range of applications

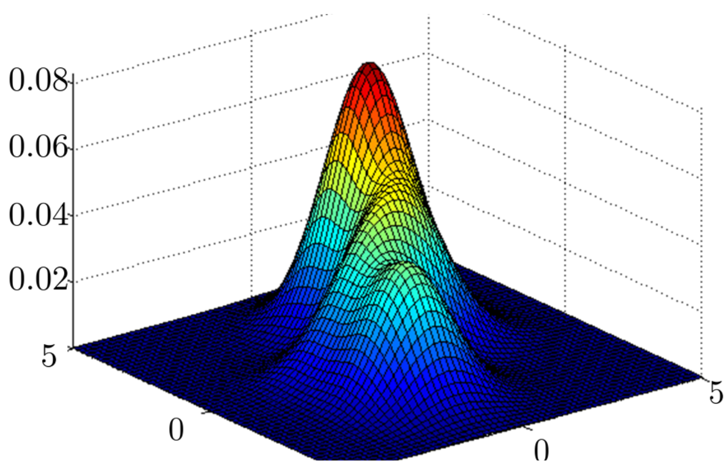
Learning Problem	Upper Bound	SQ Lower Bound
Robust Gaussian Mean Estimation	Error: $O(\epsilon \log^{1/2}(1/\epsilon))$ [DKKLMS'16]	Runtime Lower Bound: $d^{\mathrm{poly}(M)}$ for factor M improvement in error.
Robust Gaussian Covariance Estimation	Error: $O(\epsilon \log(1/\epsilon))$ [DKKLMS'16]	
Learning k-GMMs (without noise)	Runtime: $d^{g(k)}$ [MV'10, BS'10]	Runtime Lower Bound: $d^{\Omega(k)}$
Robust k-Sparse Mean Estimation	Sample size: $\tilde{O}(k^2 \log d)$ [Li'17, DBS'17]	If sample size is $O(k^{1.99})$ runtime lower bound: $d^{k^{\Omega(1)}}$
Robust Covariance Estimation in Spectral Norm	Sample size: $\tilde{O}(d^2)$ [DKKLMS'16]	If sample size is $O(d^{1.99})$ runtime lower bound: $2^{d^{\Omega(1)}}$

GAUSSIAN MIXTURE MODEL (GMM)

- GMM: Distribution on $\mathbb{R}^d$ with probability density function

$$F = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$$

- Extensively studied in statistics and TCS



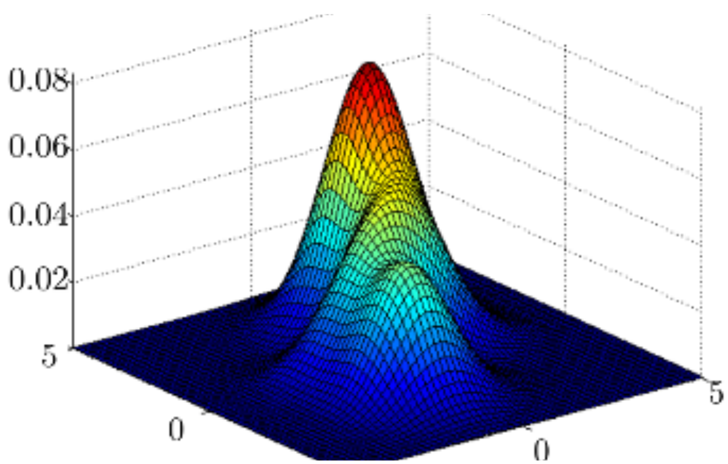
Karl Pearson (1894)

GAUSSIAN MIXTURE MODEL (GMM)

- GMM: Distribution on $\mathbb{R}^d$ with probability density function

$$F = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$$

- Extensively studied in statistics and TCS



Karl Pearson (1894)

LEARNING GMMS - PRIOR WORK (I)

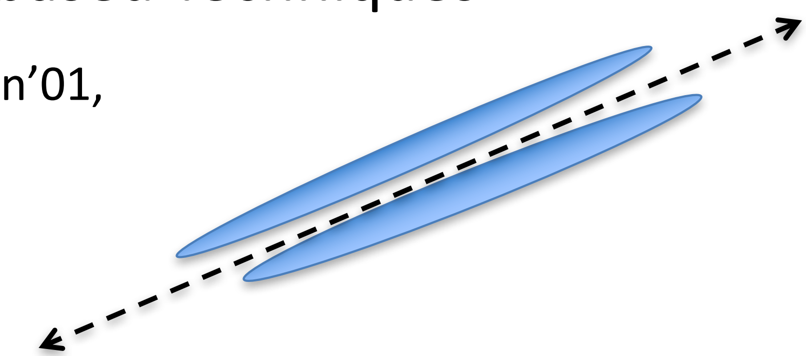
Two Related Learning Problems

Parameter Estimation: Recover model parameters.

- **Separation Assumptions**: Clustering-based Techniques

  [Dasgupta'99, Dasgupta-Schulman'00, Arora-Kanan'01, Vempala-Wang'02, Achlioptas-McSherry'05, **Brubaker-Vempala'08**]

  Sample Complexity: $\mathrm{poly}(d, k)$
  (Best Known) Runtime: $\mathrm{poly}(d, k)$



- **No Separation**: Moment Method

  [Kalai-Moitra-Valiant'10, Moitra-Valiant'10, Belkin-Sinha'10, Hardt-Price'15]

  Sample Complexity: $\mathrm{poly}(d) \cdot (1/\gamma)^{\Theta(k)}$
  (Best Known) Runtime: $(d/\gamma)^{\Omega(k)}$

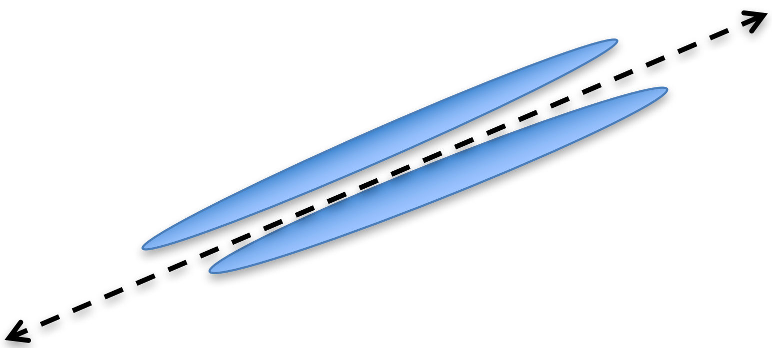
SEPARATION ASSUMPTIONS

- Clustering is possible only when the components have very little overlap.

- Formally, we want the total variation distance between components to be close to 1.

- Algorithms for learning spherical GMMS work under this assumption.

- For non-spherical GMMs, known algorithms require stronger assumptions.



LEARNING GMMS - PRIOR WORK (II)

Density Estimation: Recover underlying distribution (within statistical distance ϵ).

[Feldman-O'Donnell-Servedio'05, Moitra-Valiant'10, Suresh-Orlitsky-Acharya-Jafarpour'14, Hardt-Price'15, Li-Schmidt'15]

Sample Complexity: $\mathrm{poly}(d, k, 1/\epsilon)$

(Best Known) Runtime: $(d/\epsilon)^{\Omega(k)}$

Fact: For separated GMMs, density estimation and parameter estimation are equivalent.

LEARNING GMMS – OPEN QUESTION

Summary: The sample complexity of density estimation for k-GMMs is $\mathrm{poly}(d, k)$. The sample complexity of parameter estimation for *separated* k-GMMs is $\mathrm{poly}(d, k)$.

Question: Is there a $\mathrm{poly}(d, k)$ ***time*** learning algorithm?

STATISTICAL QUERY LOWER BOUND FOR LEARNING GMMS

Theorem: Suppose that $d \geq \mathrm{poly}(k)$. Any SQ algorithm that learns separated k-GMMs over $\mathbb{R}^d$ to constant error requires either:

- SQ queries of accuracy

$$d^{-k/6}$$

or

- At least

$$2^{\Omega(d^{1/8})} \geq d^{2k}$$

many SQ queries.

Take-away: Computational complexity of learning GMMs is inherently exponential in **number of components**.

OUTLINE

Part I: Introduction

- Unsupervised Learning in High Dimension
- Statistical Query (SQ) Learning Model
- Our Results

Part II: Computational SQ Lower Bounds

- **Generic SQ Lower Bound Technique**
- Two Applications: Learning GMMs,
  Robustly Learning a Gaussian

Part III: Summary and Conclusions

GENERAL RECIPE FOR (SQ) LOWER BOUNDS

Our generic technique for proving SQ Lower Bounds:

- **Step #1:** Construct distribution $\mathbf{P}_v$ that is standard Gaussian in all directions except v.
- **Step #2:** Construct the univariate projection in the v direction so that it matches the first m moments of $\mathcal{N}(0,1)$
- **Step #3:** Consider the family of instances $\mathcal{D} = \{\mathbf{P}_v\}_v$

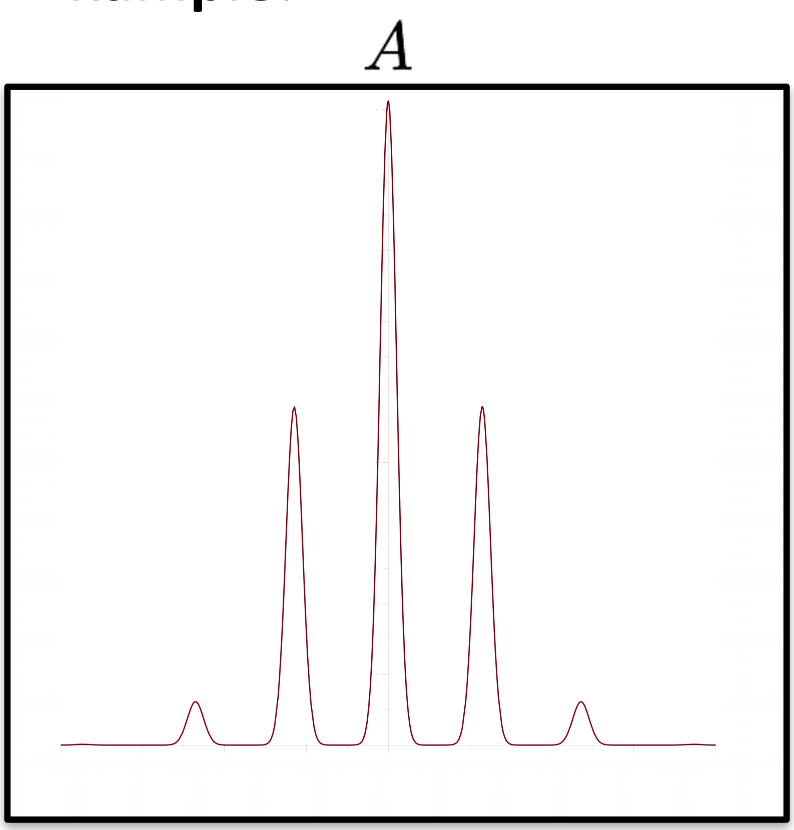
Non-Gaussian Component Analysis [Blanchard et al. 2006]

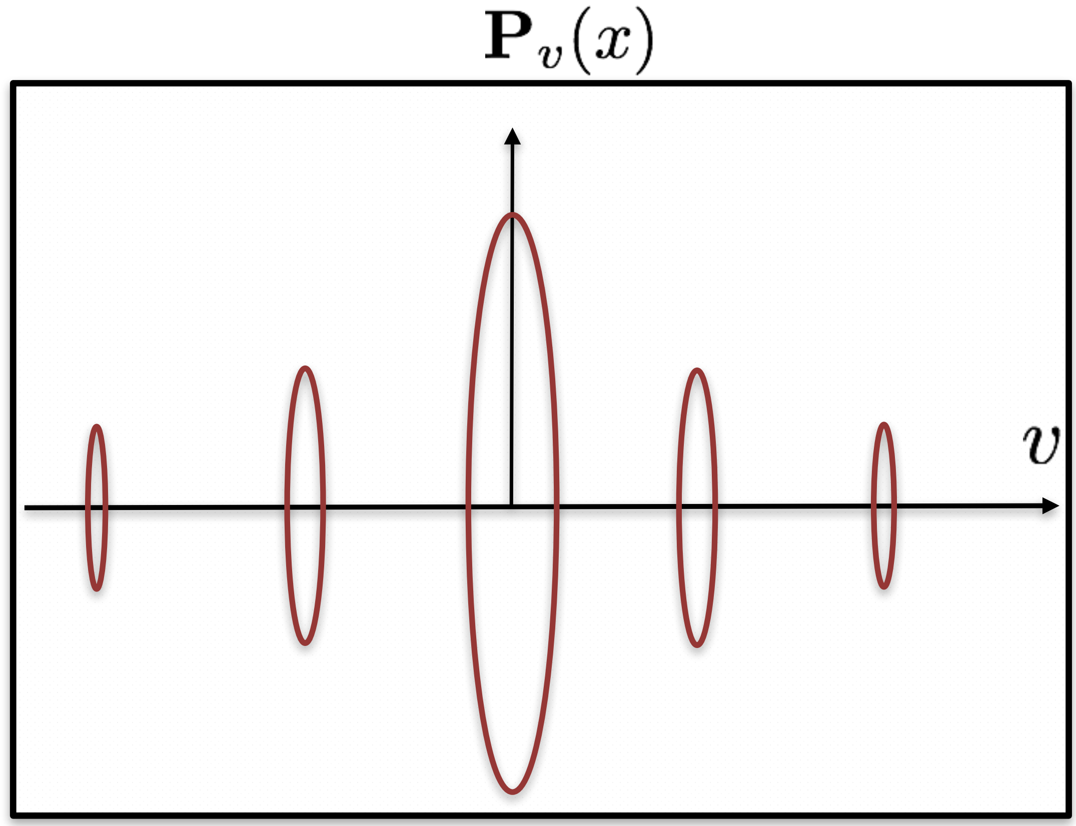
HIDDEN DIRECTION DISTRIBUTION

Definition: For a unit vector v and a univariate distribution with density A, consider the high-dimensional distribution

$$\mathbf{P}_v(x) = A(v \cdot x) \exp\left(-\|x - (v \cdot x)v\|_2^2/2\right) / (2\pi)^{(d-1)/2}.$$

Example:





GENERIC SQ LOWER BOUND

Definition: For a unit vector v and a univariate distribution with density A, consider the high-dimensional distribution

$$\mathbf{P}_v(x) = A(v \cdot x) \exp\left(-\|x - (v \cdot x)v\|_2^2/2\right) / (2\pi)^{(d-1)/2}.$$

Proposition: Suppose that:

- A matches the first m moments of $\mathcal{N}(0,1)$
- We have $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) > 2\delta$ as long as v, v' are *nearly* orthogonal.

Then any SQ algorithm that learns an unknown $\mathbf{P}_v$ within error δ requires either queries of accuracy d^{-m} or $2^{d^{\Omega(1)}}$ many queries.

WHY IS FINDING A HIDDEN DIRECTION HARD?

Observation: Low-Degree Moments do not help.

- A matches the first m moments of $\mathcal{N}(0,1)$
- The first m moments of $\mathbf{P}_v$ are identical to those of $\mathcal{N}(0,I)$
- Degree-(m+1) moment tensor has $\Omega(d^m)$ entries.

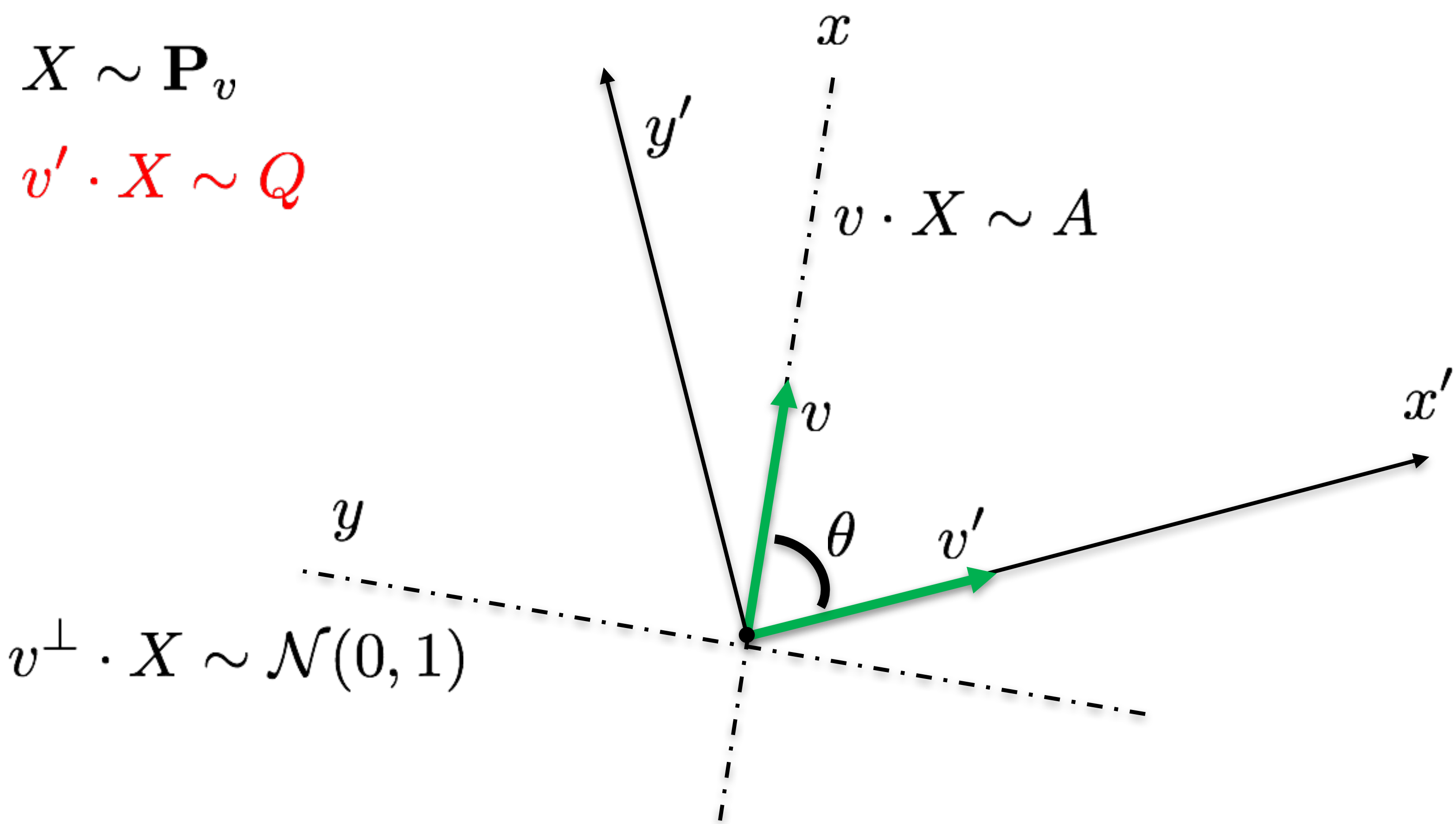
Claim: Random projections do not help.

- To distinguish between $\mathbf{P}_v$ and $\mathcal{N}(0,I)$, would need exponentially many random projections.

ONE-DIMENSIONAL PROJECTIONS ARE ALMOST GAUSSIAN

Key Lemma: Let Q be the distribution of $v' \cdot X$, where $X \sim \mathbf{P}_v$. Then, we have that:

$$\chi^2(Q, \mathcal{N}(0,1)) \leq (v \cdot v')^{2(m+1)} \chi^2\left(A, \mathcal{N}(0,1)\right)$$

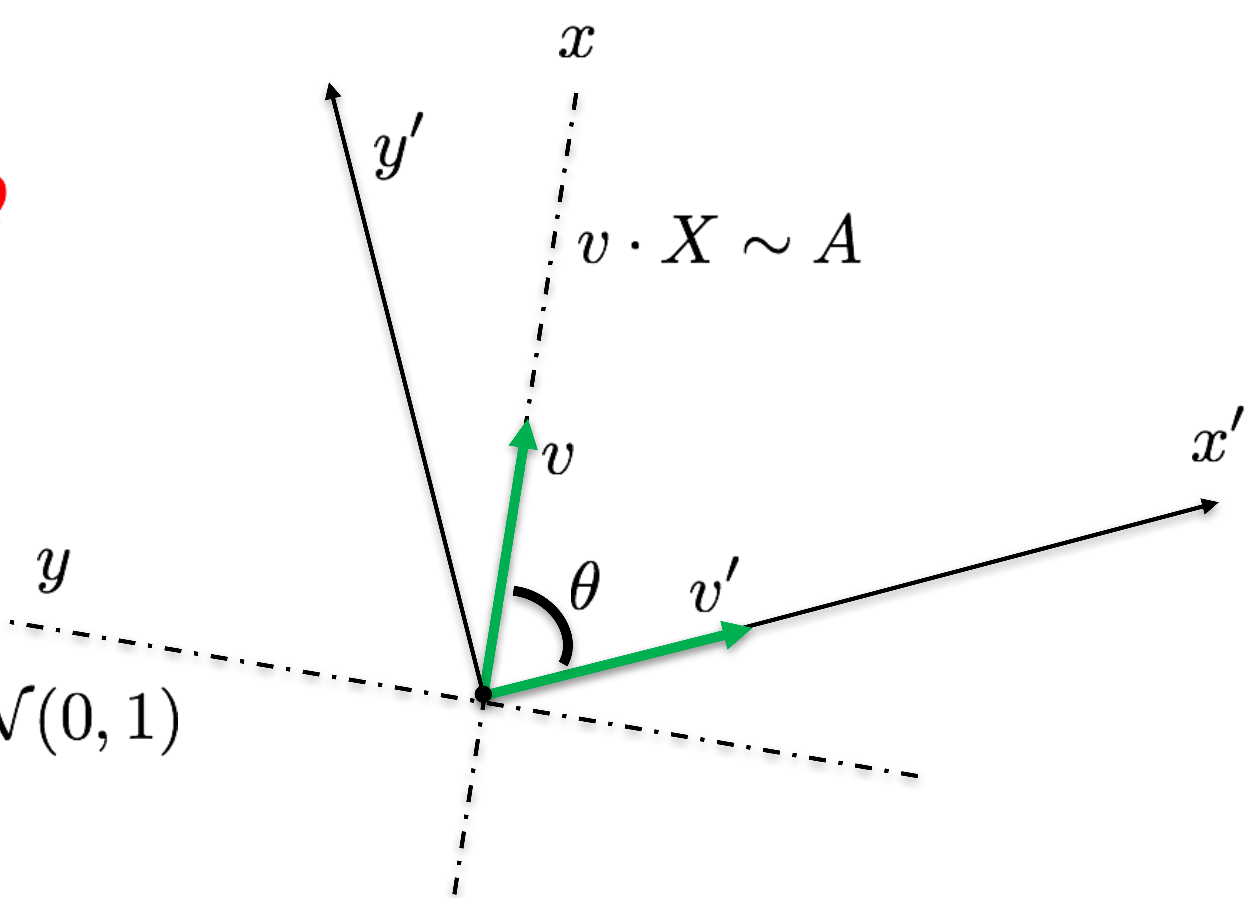


PROOF OF KEY LEMMA (I)

$$Q(x') = \int_{\mathbb{R}} A(x)G(y)dy'$$

$X \sim \mathbf{P}_v$

$v' \cdot X \sim Q$



$v^{\perp} \cdot X \sim \mathcal{N}(0,1)$

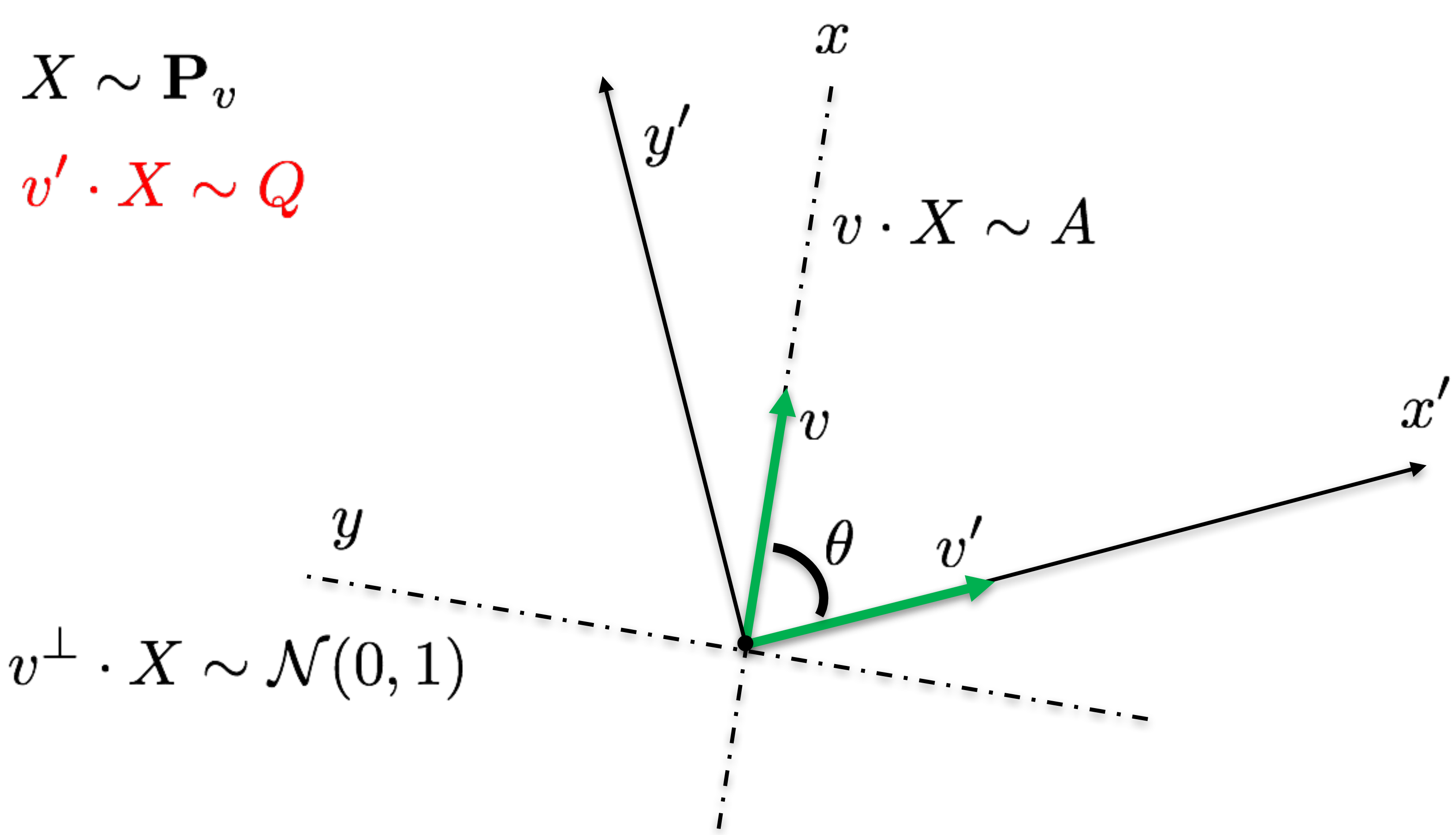
PROOF OF KEY LEMMA (I)

$$Q(x') = \int_{\mathbb{R}} A(x)G(y)dy'$$

$$= \int_{\mathbb{R}} A(x'\cos\theta + y'\sin\theta)G(x'\sin\theta - y'\cos\theta)dy'$$

$X \sim \mathbf{P}_v$

$v' \cdot X \sim Q$



PROOF OF KEY LEMMA (II)

$$Q(x') = \int_{\mathbb{R}} A(x' \cos\theta + y' \sin\theta) G(x' \sin\theta - y' \cos\theta) dy'$$

$$= (U_\theta A)(x')$$

where U_θ is the operator over $f : \mathbb{R} \to \mathbb{R}$

$$U_\theta f(x) := \int_{y \in \mathbb{R}} f(x \cos\theta + y \sin\theta) G(x \sin\theta - y \cos\theta) dy$$

Gaussian Noise (Ornstein-Uhlenbeck) Operator

EIGENFUNCTIONS OF ORNSTEIN-UHLENBECK OPERATOR

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

$$U_\theta f(x) := \int_{y\in\mathbb{R}} f(x\cos\theta + y\sin\theta)G(x\sin\theta - y\cos\theta)dy$$

Fact (Mehler'66): $U_\theta(He_i G)(x) = \cos^i(\theta)He_i(x)G(x)$

- $He_i(x)$ denotes the degree-i Hermite polynomial.
- Note that $\{He_i(x)G(x)/\sqrt{i!}\}_{i\geq 0}$ are orthonormal with respect to the inner product

$$\langle f, g\rangle = \int_{\mathbb{R}} f(x)g(x)/G(x)dx$$

GENERIC SQ LOWER BOUND

Definition: For a unit vector v and a univariate distribution with density A, consider the high-dimensional distribution

$$\mathbf{P}_v(x) = A(v \cdot x) \exp\left(-\|x - (v \cdot x)v\|_2^2/2\right) / (2\pi)^{(d-1)/2}.$$

Proposition: Suppose that:

- A matches the first m moments of $\mathcal{N}(0,1)$
- We have $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) > 2\delta$ as long as v, v' are *nearly* orthogonal.

Then any SQ algorithm that learns an unknown $\mathbf{P}_v$ within error δ requires either queries of accuracy d^{-m} or $2^{d^{\Omega(1)}}$ many queries.

OUTLINE

Part I: Introduction

- Unsupervised Learning in High Dimension
- Statistical Query (SQ) Learning Model
- Our Results

Part II: Computational SQ Lower Bounds

- Generic SQ Lower Bound Technique
- **Application: Learning GMMs**

Part III: Summary and Conclusions

APPLICATION: SQ LOWER BOUND FOR GMMS (I)

Want to show:

Theorem: Any SQ algorithm that learns separated k-GMMs over $\mathbb{R}^d$ to constant error requires either SQ queries of accuracy $d^{-k/6}$ or at least $2^{\Omega(d^{1/8})} \geq d^{2k}$ many SQ queries.

by using our generic proposition:

Proposition: Suppose that:

- A matches the first m moments of $\mathcal{N}(0,1)$
- We have $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) > 2\delta$ as long as v, v' are *nearly* orthogonal.

Then any SQ algorithm that learns an unknown $\mathbf{P}_v$ within error δ requires either queries of accuracy d^{-m} or $2^{d^{\Omega(1)}}$ many queries.

APPLICATION: SQ LOWER BOUND FOR GMMS (II)

Proposition: Suppose that:

- A matches the first m moments of $\mathcal{N}(0,1)$
- We have $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) > 2\delta$ as long as v, v' are *nearly* orthogonal.

Then any SQ algorithm that learns an unknown $\mathbf{P}_v$ within error δ requires either queries of accuracy d^{-m} or $2^{d^{\Omega(1)}}$ many queries.

Lemma: There exists a univariate distribution A that is a k-GMM with components A_i such that:

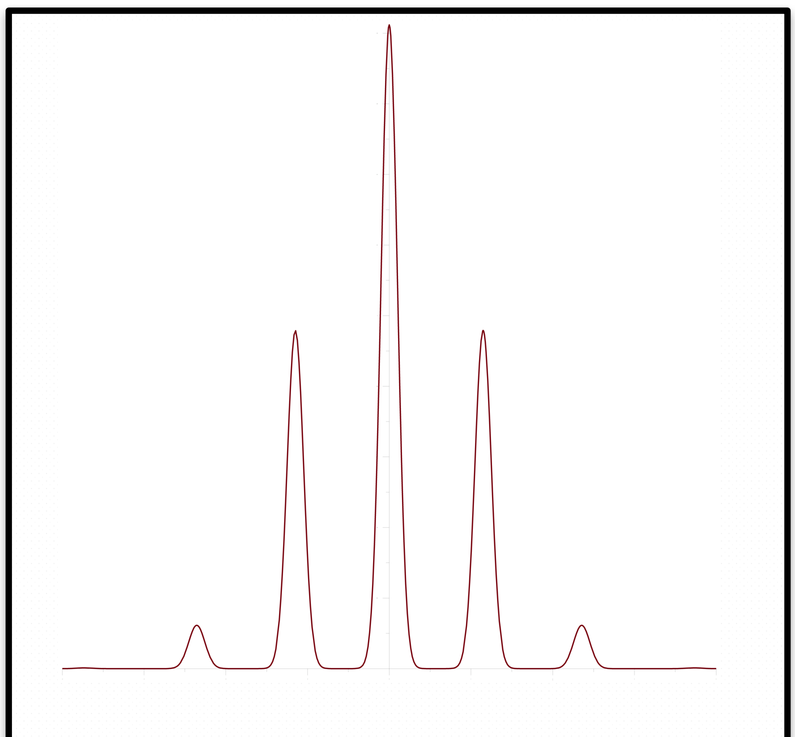
- A agrees with $\mathcal{N}(0,1)$ on the first $2k$-1 moments.
- Each pair of components are separated.
- Whenever v and v' are nearly orthogonal $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) \geq 1/2$.

APPLICATION: SQ LOWER BOUND FOR GMMS (III)

Lemma: There exists a univariate distribution A that is a k-GMM with components A_i such that:

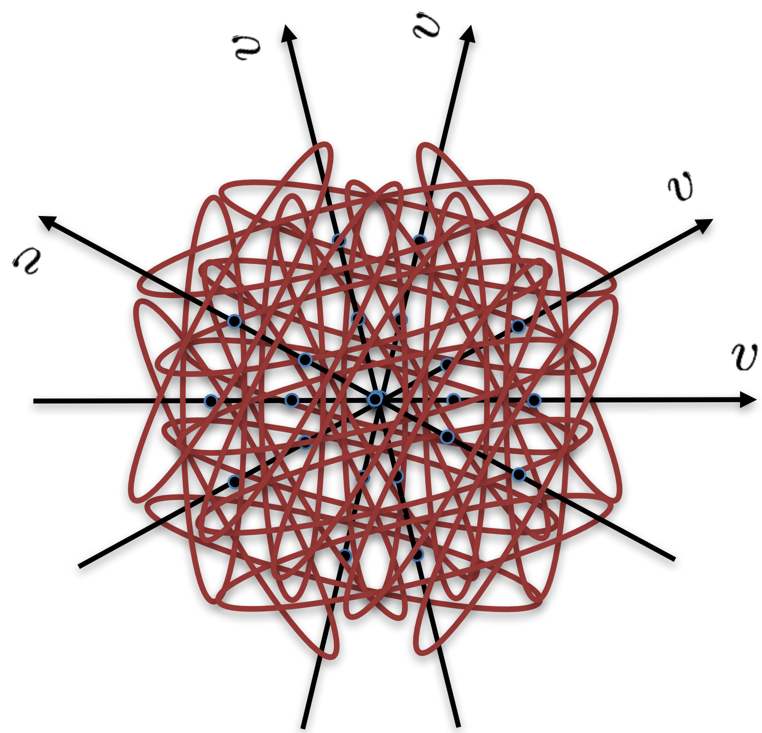
- A agrees with $\mathcal{N}(0,1)$ on the first $2k$-1 moments.
- Each pair of components are separated.
- Whenever v and v' are nearly orthogonal $d_{\mathrm{TV}}(\mathbf{P}_v, \mathbf{P}_{v'}) \geq 1/2$.

A



APPLICATION: SQ LOWER BOUND FOR GMMS (III)

High-Dimensional Distributions $\mathbf{P}_v$ look like "parallel pancakes":



Efficiently learnable for $k=2$. [Brubaker-Vempala'08]

OUTLINE

Part I: Introduction

- Unsupervised Learning in High Dimension
- Statistical Query (SQ) Learning Model
- Our Results

Part II: Computational SQ Lower Bounds

- Generic SQ Lower Bound Technique
- Two Applications: Learning GMMs, Robustly Learning a Gaussian

Part III: Summary and Conclusions

SUMMARY AND FUTURE DIRECTIONS

- General Technique to Prove SQ Lower Bounds
- Robustness can make high-dimensional estimation harder computationally and information-theoretically.

Future Directions:

- Further Applications of our Framework

  List-Decodable Mean Estimation [D-Kane-Stewart'18]

  Discrete Product Distributions [D-Kane-Stewart'18]

  Robust Regression [D-Kong-Stewart'18]

  Adversarial Examples [Bubeck-Price- Razenshteyn'18]

- Alternative Evidence of Computational Hardness?

Thanks! Any Questions?