Hardness of Certification for Constrained PCA

Alex Wein
Courant Institute, NYU

Joint work with:

Afonso Bandeira (NYU)

Tim Kunisky (NYU)

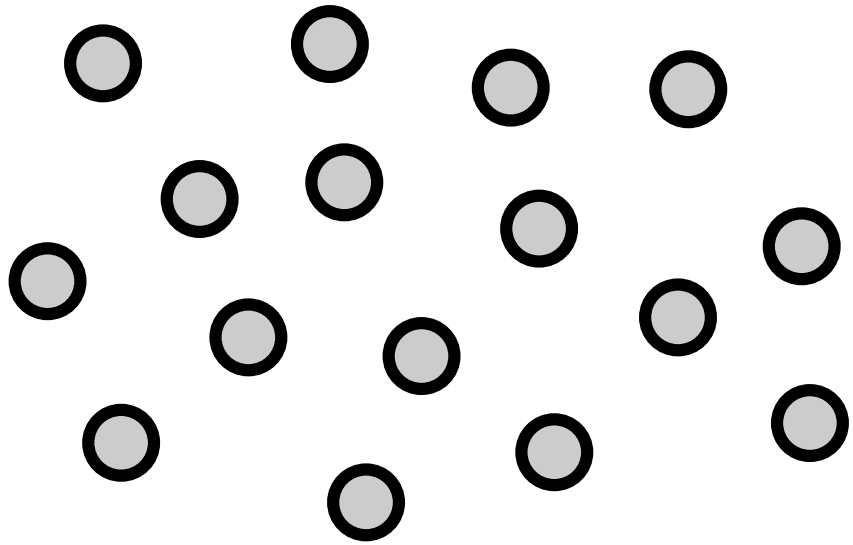
Part I: Statistical-to-Computational Gaps and the "Low-Degree Method"

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$

Statistical-to-Computational Gaps

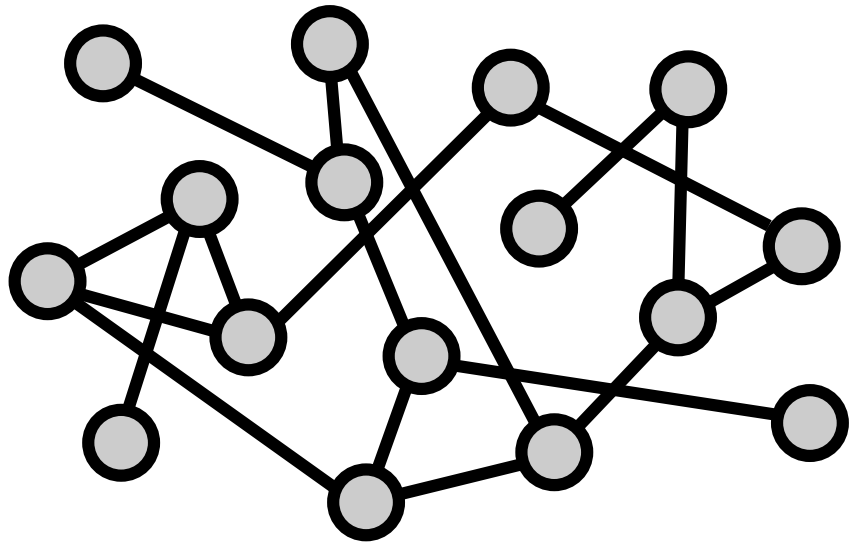
- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$



  - n vertices

Statistical-to-Computational Gaps

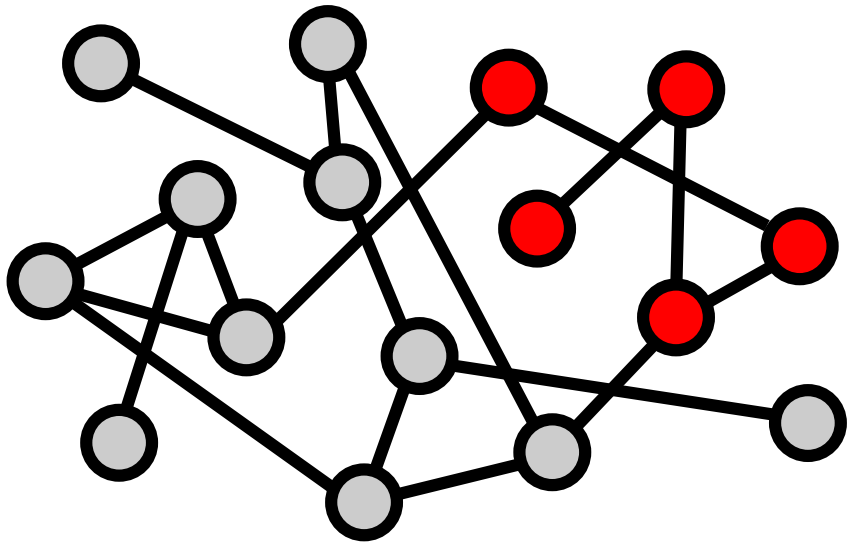
- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$



  - n vertices
  - Each of the $\binom{n}{2}$ edges occurs with probability $1/2$

Statistical-to-Computational Gaps

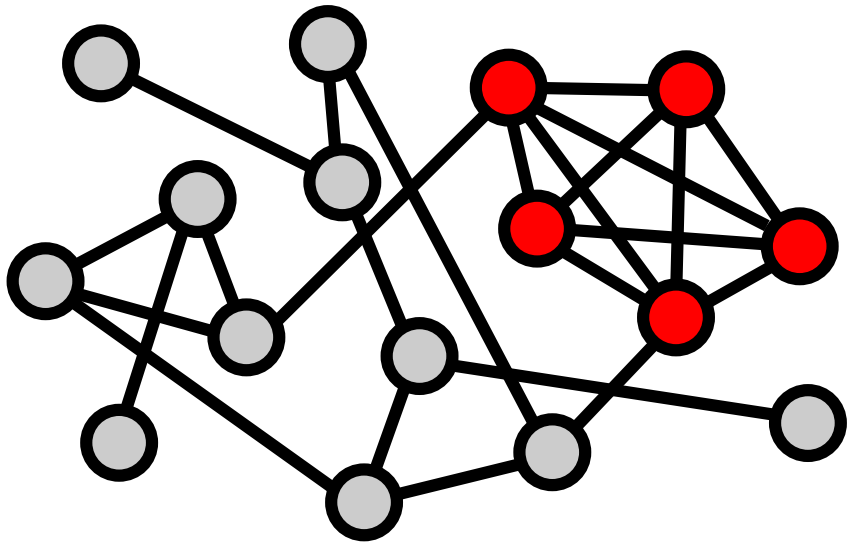
- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$



  - n vertices
  - Each of the $\binom{n}{2}$ edges occurs with probability $1/2$
  - Planted clique on k vertices

Statistical-to-Computational Gaps

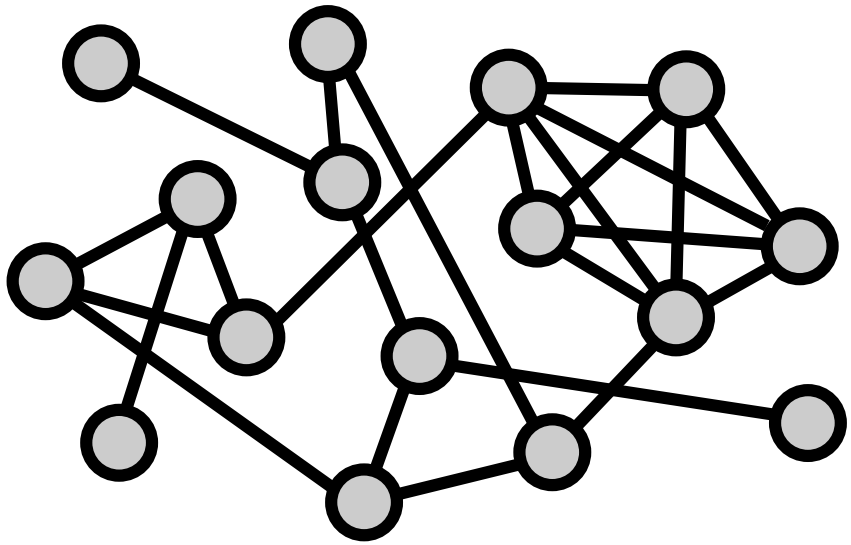
- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$



  - n vertices
  - Each of the $\binom{n}{2}$ edges occurs with probability $1/2$
  - Planted clique on k vertices

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$



  - n vertices
  - Each of the $\binom{n}{2}$ edges occurs with probability $1/2$
  - Planted clique on k vertices
  - Goal: find the clique

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)
- Tensor PCA

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)
- Tensor PCA
- Tensor decomposition

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)
- Tensor PCA
- Tensor decomposition
- Synchronization / orbit recovery

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)
- Tensor PCA
- Tensor decomposition
- Synchronization / orbit recovery

Different from theory of NP-completeness: average-case

Statistical-to-Computational Gaps

- Planted clique: $G(n, 1/2) \cup \{k\text{-clique}\}$
  - Statistically, can find planted clique of size $(2 + \varepsilon) \log n$
  - In poly-time, can only find clique of size $\Omega(\sqrt{n})$

- Sparse PCA
- Stochastic block model (community detection)
- Random constraint satisfaction problems (e.g. 3-SAT)
- Tensor PCA
- Tensor decomposition
- Synchronization / orbit recovery

Different from theory of NP-completeness: average-case

Q: What fundamentally makes a problem easy or hard?

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]
- Failure of local algorithms [Gamarnik, Sudan '13]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]
- Failure of local algorithms [Gamarnik, Sudan '13]
- Statistical physics, BP [Decelle, Krzakala, Moore, Zdeborová '11]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]
- Failure of local algorithms [Gamarnik, Sudan '13]
- Statistical physics, BP [Decelle, Krzakala, Moore, Zdeborová '11]
- Optimization landscape, Kac-Rice [Auffinger, Ben Arous, Cerný '10]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]
- Failure of local algorithms [Gamarnik, Sudan '13]
- Statistical physics, BP [Decelle, Krzakala, Moore, Zdeborová '11]
- Optimization landscape, Kac-Rice [Auffinger, Ben Arous, Cerný '10]
- Sum-of-squares lower bounds [Barak, Hopkins, Kelner, Kothari, Moitra, Potechin '16]

How to Show that a Problem is Hard?

We don't know how to prove that average-case problems are hard, but various forms of evidence:

- Reductions (e.g. from planted clique) [Berthet, Rigollet '13]
- Failure of MCMC [Jerrum '92]
- Shattering of solution space [Achlioptas, Coja-Oghlan '08]
- Failure of local algorithms [Gamarnik, Sudan '13]
- Statistical physics, BP [Decelle, Krzakala, Moore, Zdeborová '11]
- Optimization landscape, Kac-Rice [Auffinger, Ben Arous, Cerný '10]
- Sum-of-squares lower bounds [Barak, Hopkins, Kelner, Kothari, Moitra, Potechin '16]
- This talk: "low-degree method"

  [Barak, Hopkins, Kelner, Kothari, Moitra, Potechin '16; Hopkins, Steurer '17;

  Hopkins, Kothari, Potechin, Raghavendra, Schramm, Steurer '17; Hopkins '18 (PhD thesis)]

The Low-Degree Method

Suppose we want to hypothesis test (with error probability $o(1)$) between two distributions:

The Low-Degree Method

Suppose we want to hypothesis test (with error probability $o(1)$) between two distributions:

- Null model $Y \sim \mathbb{Q}_n$ e.g. $G(n, 1/2)$

The Low-Degree Method

Suppose we want to hypothesis test (with error probability $o(1)$) between two distributions:

- Null model $Y \sim \mathbb{Q}_n$ e.g. $G(n, 1/2)$
- Planted model $Y \sim \mathbb{P}_n$ e.g. $G(n, 1/2) \cup \{k\text{-clique}\}$

The Low-Degree Method

Suppose we want to hypothesis test (with error probability $o(1)$) between two distributions:

- Null model $Y \sim \mathbb{Q}_n$ e.g. $G(n, 1/2)$
- Planted model $Y \sim \mathbb{P}_n$ e.g. $G(n, 1/2) \cup \{k\text{-clique}\}$

Look for a degree-D multivariate polynomial f that distinguishes $\mathbb{P}$ from $\mathbb{Q}$:

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

The Low-Degree Method

Suppose we want to hypothesis test (with error probability $o(1)$) between two distributions:

- Null model $Y \sim \mathbb{Q}_n$ e.g. $G(n, 1/2)$
- Planted model $Y \sim \mathbb{P}_n$ e.g. $G(n, 1/2) \cup \{k\text{-clique}\}$

Look for a degree-D multivariate polynomial f that distinguishes $\mathbb{P}$ from $\mathbb{Q}$:

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

Want $f(Y)$ to be big when $Y \sim \mathbb{P}$ and small when $Y \sim \mathbb{Q}$

The Low-Degree Method

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$\mathbb{R}[Y]_D$: polynomials of degree $\leq D$ (subspace)

The Low-Degree Method

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{Q}}[L(Y) f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$\mathbb{R}[Y]_D$: polynomials of degree $\leq D$ (subspace)

$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y)$

The Low-Degree Method

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{Q}}[L(Y) f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\langle L, f \rangle}{\|f\|}$$

$\mathbb{R}[Y]_D$: polynomials of degree $\leq D$ (subspace)

$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y)$

$\langle f, g \rangle = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y) g(Y)]$

$\|f\| = \sqrt{\langle f, f \rangle}$

The Low-Degree Method

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{Q}}[L(Y) f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\langle L, f \rangle}{\|f\|}$$

$$= \|L^{\leq D}\|$$

$\mathbb{R}[Y]_D$: polynomials of degree $\leq D$ (subspace)

$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y)$

$\langle f, g \rangle = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y) g(Y)]$

$\|f\| = \sqrt{\langle f, f \rangle}$

Maximizer: $f = L^{\leq D} := \mathrm{proj}_{(\mathbb{R}[Y]_D)} L$

The Low-Degree Method

$$\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{Q}}[L(Y) f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}}$$

$$= \max_{f \in \mathbb{R}[Y]_D} \frac{\langle L, f \rangle}{\|f\|}$$

$$= \|L^{\leq D}\|$$

$\mathbb{R}[Y]_D$: polynomials of degree $\leq D$ (subspace)

$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y)$

$\langle f, g \rangle = \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y) g(Y)]$

$\|f\| = \sqrt{\langle f, f \rangle}$

Maximizer: $f = L^{\leq D} := \mathrm{proj}_{(\mathbb{R}[Y]_D)} L$

Norm of low-degree likelihood ratio

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

Degree-$O(\log n)$ polynomials $\Leftrightarrow$ Polynomial-time algorithms

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

Degree-$O(\log n)$ polynomials $\Leftrightarrow$ Polynomial-time algorithms

- Spectral method: distinguish via top eigenvalue of matrix $M = M(Y)$ whose entries are $O(1)$-degree polynomials in Y

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

Degree-$O(\log n)$ polynomials $\Leftrightarrow$ Polynomial-time algorithms

- Spectral method: distinguish via top eigenvalue of matrix $M = M(Y)$ whose entries are $O(1)$-degree polynomials in Y
- Log-degree distinguisher: $f(Y) = \mathrm{Tr}(M^q)$ with $q = \Theta(\log n)$

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

Degree-$O(\log n)$ polynomials $\Leftrightarrow$ Polynomial-time algorithms

- Spectral method: distinguish via top eigenvalue of matrix $M = M(Y)$ whose entries are $O(1)$-degree polynomials in Y
- Log-degree distinguisher: $f(Y) = \mathrm{Tr}(M^q)$ with $q = \Theta(\log n)$
- Spectral methods $\Leftrightarrow$ sum-of-squares [HKPRSS '17]

The Low-Degree Method

Conclusion: $\max_{f \in \mathbb{R}[Y]_D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} = \|L^{\leq D}\|$

Heuristically,

$$\|L^{\leq D}\| = \begin{cases} \omega(1) & \text{degree-}D \text{ polynomial can distinguish } \mathbb{Q}, \mathbb{P} \\ O(1) & \text{degree-}D \text{ polynomials fail} \end{cases}$$

Degree-$O(\log n)$ polynomials $\Leftrightarrow$ Polynomial-time algorithms

- Spectral method: distinguish via top eigenvalue of matrix $M = M(Y)$ whose entries are $O(1)$-degree polynomials in Y
- Log-degree distinguisher: $f(Y) = \mathrm{Tr}(M^q)$ with $q = \Theta(\log n)$
- Spectral methods $\Leftrightarrow$ sum-of-squares [HKPRSS '17]

Conjecture (informal variant of [Hopkins '18])

For "nice" $\mathbb{Q}, \mathbb{P}$, if $\|L^{\leq D}\| = O(1)$ for $D = \log^{1+\Omega(1)}(n)$ then no polynomial-time algorithm can distinguish $\mathbb{Q}, \mathbb{P}$ with success probability $1 - o(1)$.

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems
- And the predictions are correct! (i.e. matching widely-believed conjectures)

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems
- And the predictions are correct! (i.e. matching widely-believed conjectures)
  - Planted clique, sparse PCA, stochastic block model, tensor PCA, ...

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems
- And the predictions are correct! (i.e. matching widely-believed conjectures)
  - Planted clique, sparse PCA, stochastic block model, tensor PCA, ...
- Heuristically, low-degree prediction matches performance of sum-of-squares

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems
- And the predictions are correct! (i.e. matching widely-believed conjectures)
  - Planted clique, sparse PCA, stochastic block model, tensor PCA, ...
- Heuristically, low-degree prediction matches performance of sum-of-squares
  - But low-degree calculation is much easier than proving SOS lower bounds

Advantages of the Low-Degree Method

- Can actually calculate/bound $\|L^{\leq D}\|$ for many problems
- And the predictions are correct! (i.e. matching widely-believed conjectures)
  - Planted clique, sparse PCA, stochastic block model, tensor PCA, ...
- Heuristically, low-degree prediction matches performance of sum-of-squares
  - But low-degree calculation is much easier than proving SOS lower bounds
- By varying degree D, can explore power of subexponential-time algorithms:
  - Degree-n^{δ} polynomials $\Leftrightarrow$ Time-$2^{n^{\delta}}$ algorithms $\delta \in (0, 1)$

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0, 1)$

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0, 1)$

$$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y) = \frac{\mathbb{E}_X \exp(-\frac{1}{2}\|Y - X\|^2)}{\exp(-\frac{1}{2}\|Y\|^2)} = \mathbb{E}_X \exp(\langle Y, X\rangle - \frac{1}{2}\|X\|^2)$$

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0, 1)$

$$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y) = \frac{\mathbb{E}_X \exp(-\frac{1}{2}\|Y - X\|^2)}{\exp(-\frac{1}{2}\|Y\|^2)} = \mathbb{E}_X \exp(\langle Y, X\rangle - \frac{1}{2}\|X\|^2)$$

Write $L = \sum_\alpha c_\alpha h_\alpha$ where $\{h_\alpha\}$ are Hermite polynomials
(orthonormal basis w.r.t. $\mathbb{Q}$)

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0,1)$

$$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y) = \frac{\mathbb{E}_X \exp(-\frac{1}{2}\|Y - X\|^2)}{\exp(-\frac{1}{2}\|Y\|^2)} = \mathbb{E}_X \exp(\langle Y, X\rangle - \frac{1}{2}\|X\|^2)$$

Write $L = \sum_\alpha c_\alpha h_\alpha$ where $\{h_\alpha\}$ are Hermite polynomials
(orthonormal basis w.r.t. $\mathbb{Q}$)

$\|L^{\leq D}\|^2 = \sum_{|\alpha| \leq D} c_\alpha^2$ where $c_\alpha = \langle L, h_\alpha \rangle = \mathbb{E}_{Y \sim \mathbb{Q}}[L(Y) h_\alpha(Y)]$

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0, 1)$

$$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y) = \frac{\mathbb{E}_X \exp(-\frac{1}{2}\|Y - X\|^2)}{\exp(-\frac{1}{2}\|Y\|^2)} = \mathbb{E}_X \exp(\langle Y, X\rangle - \frac{1}{2}\|X\|^2)$$

Write $L = \sum_\alpha c_\alpha h_\alpha$ where $\{h_\alpha\}$ are Hermite polynomials
(orthonormal basis w.r.t. $\mathbb{Q}$)

$\|L^{\leq D}\|^2 = \sum_{|\alpha|\leq D} c_\alpha^2$ where $c_\alpha = \langle L, h_\alpha\rangle = \mathbb{E}_{Y\sim\mathbb{Q}}[L(Y)h_\alpha(Y)]$

...

How to Compute $\|L^{\leq D}\|$

Additive Gaussian noise: $\mathbb{P} : Y = X + Z$ vs $\mathbb{Q} : Y = Z$
where $X \sim \mathcal{P}$, any distribution over $\mathbb{R}^N$
and Z is i.i.d. $\mathcal{N}(0,1)$

$$L(Y) = \frac{d\mathbb{P}}{d\mathbb{Q}}(Y) = \frac{\mathbb{E}_X \exp(-\frac{1}{2}\|Y - X\|^2)}{\exp(-\frac{1}{2}\|Y\|^2)} = \mathbb{E}_X \exp(\langle Y, X\rangle - \frac{1}{2}\|X\|^2)$$

Write $L = \sum_\alpha c_\alpha h_\alpha$ where $\{h_\alpha\}$ are Hermite polynomials
(orthonormal basis w.r.t. $\mathbb{Q}$)

$\|L^{\leq D}\|^2 = \sum_{|\alpha|\leq D} c_\alpha^2$ where $c_\alpha = \langle L, h_\alpha\rangle = \mathbb{E}_{Y\sim\mathbb{Q}}[L(Y)h_\alpha(Y)]$

...

Result: $\|L^{\leq D}\|^2 = \sum_{d=0}^{D} \frac{1}{d!}\mathbb{E}_{X,X'}[\langle X, X'\rangle^d]$

Part II: Hardness of Certification for Constrained PCA Problems

Constrained PCA

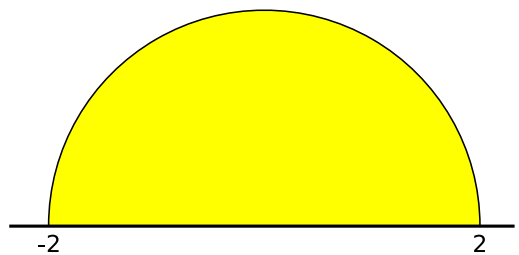
Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$

Constrained PCA

Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

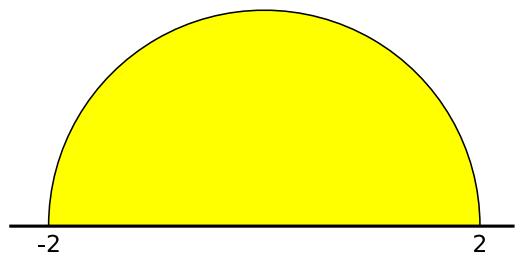
- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$
- Eigenvalues follow semicircle law on $[-2, 2]$



Constrained PCA

Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$
- Eigenvalues follow semicircle law on $[-2, 2]$

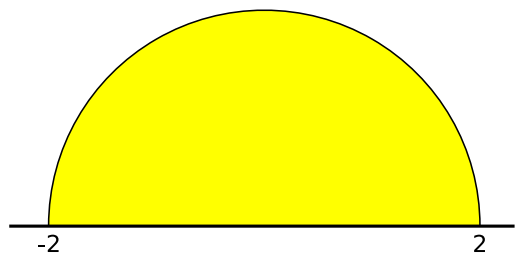


PCA: $\max_{\|x\|=1} x^\top W x$

Constrained PCA

Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$
- Eigenvalues follow semicircle law on $[-2, 2]$

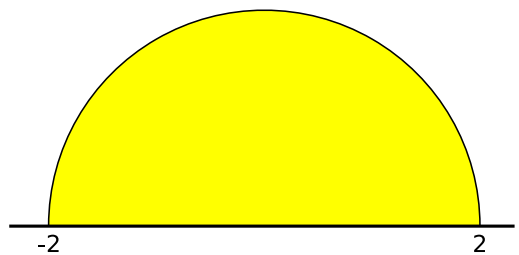


PCA: $\max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2 \quad \text{as } n \to \infty$

Constrained PCA

Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$
- Eigenvalues follow semicircle law on $[-2, 2]$



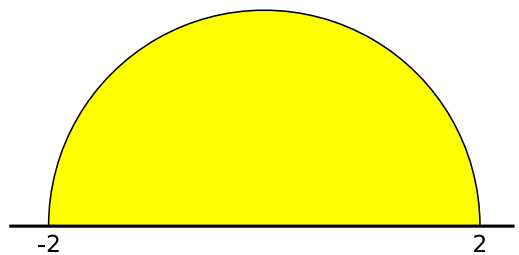
PCA: $\max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2 \quad \text{as } n \to \infty$

Constrained PCA: $\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$

Constrained PCA

Let $W \sim \mathrm{GOE}(n)$ "Gaussian orthogonal ensemble"

- $n \times n$ random symmetric matrix:
  $W_{ij} = W_{ji} \sim \mathcal{N}(0, 1/n), \quad W_{ii} \sim \mathcal{N}(0, 2/n)$
- Eigenvalues follow semicircle law on $[-2, 2]$



PCA: $\max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2 \quad \text{as } n \to \infty$

Constrained PCA: $\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$

Statistical physics: "Sherrington–Kirkpatrick spin glass model"

- $\phi(W) \to 2\mathrm{P}_* \approx 1.5264$ as $n \to \infty$ [Parisi '80; Talagrand '06]

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \leq B$ for some bound B

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \leq B$ for some bound B
  - Formally: algorithm $\{f_n\}$ outputs $f_n(W) \in \mathbb{R}$ such that:

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \leq B$ for some bound B
  - Formally: algorithm $\{f_n\}$ outputs $f_n(W) \in \mathbb{R}$ such that:
    (i) $\phi(W) \leq f_n(W) \quad \forall W \in \mathbb{R}^{n \times n}$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \le B$ for some bound B
  - Formally: algorithm $\{f_n\}$ outputs $f_n(W) \in \mathbb{R}$ such that:
    (i) $\phi(W) \le f_n(W) \ \ \forall W \in \mathbb{R}^{n \times n}$
    (ii) if $W \sim GOE(n)$, $f_n(W) \le B + o(1)$ w.p. $1 - o(1)$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

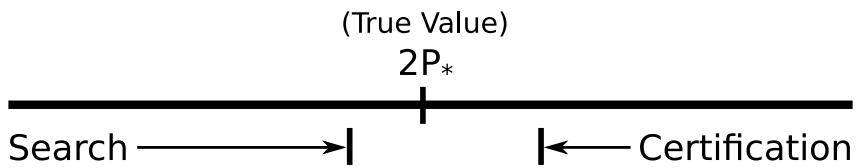
- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \le B$ for some bound B
  - Formally: algorithm $\{f_n\}$ outputs $f_n(W) \in \mathbb{R}$ such that:
    (i) $\phi(W) \le f_n(W) \ \ \forall W \in \mathbb{R}^{n \times n}$
    (ii) if $W \sim GOE(n)$, $f_n(W) \le B + o(1)$ w.p. $1 - o(1)$
  - Note: cannot just output $f_n(W) = 2\mathrm{P}_* + \varepsilon$

Search vs Certification

$$\phi(W) := \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x, \quad W \sim GOE(n)$$

Two computational problems:

- Search: given W, find $x \in \{\pm 1/\sqrt{n}\}^n$ with large $x^\top W x$
  - Proves a lower bound on $\phi(W)$
- Certification: given W, prove $\phi(W) \le B$ for some bound B
  - Formally: algorithm $\{f_n\}$ outputs $f_n(W) \in \mathbb{R}$ such that:
    (i) $\phi(W) \le f_n(W) \quad \forall W \in \mathbb{R}^{n \times n}$
    (ii) if $W \sim GOE(n)$, $f_n(W) \le B + o(1)$ w.p. $1 - o(1)$
  - Note: cannot just output $f_n(W) = 2\mathrm{P}_* + \varepsilon$



Search vs Certification: Prior Work

Search vs Certification: Prior Work

Perfect search is possible in poly time

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

- Convex relaxation?

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

- Convex relaxation?
- Sum-of-squares?

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

- Convex relaxation?
- Sum-of-squares?

Answer: no!

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

- Convex relaxation?
- Sum-of-squares?

Answer: no!

- In particular, any convex relaxation fails

Search vs Certification: Prior Work

Perfect search is possible in poly time

- Can find $x \in \{\pm 1/\sqrt{n}\}^n$ such that $x^\top W x \geq 2\mathrm{P}* - \varepsilon$ [Montanari '18]
- Optimization of full-RSB models [Subag '18]

Trivial spectral certification:

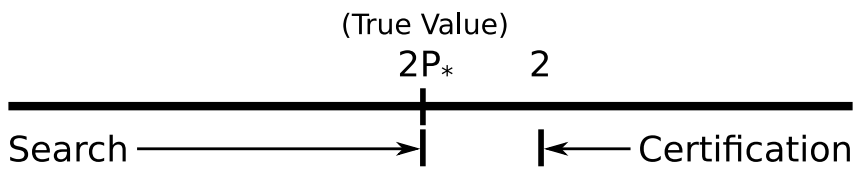
$$\phi(W) \leq \max_{\|x\|=1} x^\top W x = \lambda_{\max}(W) \to 2$$

Can we do better (in poly time)?

- Convex relaxation?
- Sum-of-squares?

Answer: no!

- In particular, any convex relaxation fails



Main Result

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

- In fact, need essentially exponential time: $2^{n^{1-o(1)}}$

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

- In fact, need essentially exponential time: $2^{n^{1-o(1)}}$
- Also for constraint sets other than $\{\pm 1/\sqrt{n}\}^n$

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

- In fact, need essentially exponential time: $2^{n^{1-o(1)}}$
- Also for constraint sets other than $\{\pm 1/\sqrt{n}\}^n$

Proof outline:

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

- In fact, need essentially exponential time: $2^{n^{1-o(1)}}$
- Also for constraint sets other than $\{\pm 1/\sqrt{n}\}^n$

Proof outline:

(i) Reduction from a hypothesis testing problem (negatively-spiked Wishart) to certification problem

Main Result

Theorem (informal)

Conditional on the low-degree method, for any $\varepsilon > 0$, no polynomial-time algorithm can certify an upper bound of $2 - \varepsilon$ on $\phi(W)$.

- In fact, need essentially exponential time: $2^{n^{1-o(1)}}$
- Also for constraint sets other than $\{\pm 1/\sqrt{n}\}^n$

Proof outline:

(i) Reduction from a hypothesis testing problem (negatively-spiked Wishart) to certification problem

(ii) Use low-degree method to show that the hypothesis testing problem is hard

Spiked Wishart Model

Spiked Wishart Model

$\mathbb{Q}$: Observe N independent samples $y_1, \ldots, y_N$ where $y_i \sim \mathcal{N}(0, I_n)$

Spiked Wishart Model

$\mathbb{Q}$: Observe N independent samples $y_1, \ldots, y_N$ where $y_i \sim \mathcal{N}(0, I_n)$

$\mathbb{P}$: Planted vector $x \sim \mathrm{Unif}(\{\pm 1/\sqrt{n}\}^n)$

Observe $y_1, \ldots, y_N$ with $y_i \sim \mathcal{N}(0, I_n + \beta x x^\top)$

Parameters: $n/N \to \gamma, \quad \beta \in [-1, \infty)$

Spiked Wishart Model

$\mathbb{Q}$: Observe N independent samples $y_1, \ldots, y_N$ where $y_i \sim \mathcal{N}(0, I_n)$

$\mathbb{P}$: Planted vector $x \sim \mathrm{Unif}(\{\pm 1/\sqrt{n}\}^n)$

Observe $y_1, \ldots, y_N$ with $y_i \sim \mathcal{N}(0, I_n + \beta x x^\top)$

Parameters: $n/N \to \gamma, \quad \beta \in [-1, \infty)$

Spectral threshold: if $\beta^2 > \gamma$, can distinguish $\mathbb{Q}, \mathbb{P}$ using top/bottom eigenvalue of sample covariance matrix $Y = \frac{1}{N} \sum_i y_i y_i^\top$ [Baik, Ben Arous, Péché '05]

Spiked Wishart Model

$\mathbb{Q}$: Observe N independent samples $y_1, \ldots, y_N$ where $y_i \sim \mathcal{N}(0, I_n)$

$\mathbb{P}$: Planted vector $x \sim \mathrm{Unif}(\{\pm 1/\sqrt{n}\}^n)$

Observe $y_1, \ldots, y_N$ with $y_i \sim \mathcal{N}(0, I_n + \beta x x^\top)$

Parameters: $n/N \to \gamma, \quad \beta \in [-1, \infty)$

Spectral threshold: if $\beta^2 > \gamma$, can distinguish $\mathbb{Q}, \mathbb{P}$ using top/bottom eigenvalue of sample covariance matrix $Y = \frac{1}{N} \sum_i y_i y_i^\top$ [Baik, Ben Arous, Péché '05]

Using low-degree method, we show: if $\beta^2 < \gamma$, cannot distinguish $\mathbb{Q}, \mathbb{P}$ (unless given exponential time)

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

$\mathbb{Q}$: observe N random vectors in $\mathbb{R}^n$

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

$\mathbb{Q}$: observe N random vectors in $\mathbb{R}^n$

$\mathbb{P}$: observe N random vectors that are all orthogonal to a planted hypercube vector $x \in \{\pm 1/\sqrt{n}\}^n$

- $y_i \sim \mathcal{N}(0, I_n - xx^\top)$

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

$\mathbb{Q}$: observe N random vectors in $\mathbb{R}^n$

$\mathbb{P}$: observe N random vectors that are all orthogonal to a planted hypercube vector $x \in \{\pm 1/\sqrt{n}\}^n$

- $y_i \sim \mathcal{N}(0, I_n - xx^\top)$

Spectral threshold: if $N \geq n$, can distinguish using $\mathrm{rank}(y_1, \ldots, y_N)$

- $\mathbb{Q}$: rank n
- $\mathbb{P}$: rank $n - 1$

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

$\mathbb{Q}$: observe N random vectors in $\mathbb{R}^n$

$\mathbb{P}$: observe N random vectors that are all orthogonal to a planted hypercube vector $x \in \{\pm 1/\sqrt{n}\}^n$

- $y_i \sim \mathcal{N}(0, I_n - xx^\top)$

Spectral threshold: if $N \geq n$, can distinguish using $\mathrm{rank}(y_1, \ldots, y_N)$

- $\mathbb{Q}$: rank n
- $\mathbb{P}$: rank $n - 1$

Low-degree method: if $N < n$, cannot distinguish (unless given exponential time)

Negatively-Spiked Wishart Model

Our case of interest: $\beta = -1$ (technically $\beta > -1, \beta \approx -1$)

$\mathbb{Q}$: observe N random vectors in $\mathbb{R}^n$

$\mathbb{P}$: observe N random vectors that are all orthogonal to a planted hypercube vector $x \in \{\pm 1/\sqrt{n}\}^n$

- $y_i \sim \mathcal{N}(0, I_n - xx^\top)$

Spectral threshold: if $N \geq n$, can distinguish using $\mathrm{rank}(y_1, \ldots, y_N)$

- $\mathbb{Q}$: rank n
- $\mathbb{P}$: rank $n - 1$

Low-degree method: if $N < n$, cannot distinguish (unless given exponential time)

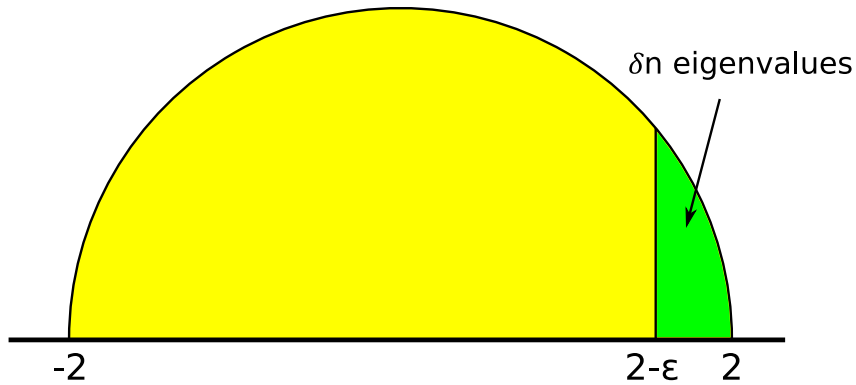
- But statistically possible

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector



  - If hypercube vector x is a linear combination of the top δn eigenvectors, it would satisfy $x^\top W x \geq 2 - \varepsilon$

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector
- So you can certify that a random δn-dimensional subspace does not contain a hypercube vector

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector
- So you can certify that a random δn-dimensional subspace does not contain a hypercube vector
- So you can distinguish between a random δn-dimensional subspace and a δn-dimensional subspace containing a hypercube vector

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector
- So you can certify that a random δn-dimensional subspace does not contain a hypercube vector
- So you can distinguish between a random δn-dimensional subspace and a δn-dimensional subspace containing a hypercube vector
- So you can distinguish between a random $(1-\delta)n$-dimensional subspace and a $(1-\delta)n$-dimensional subspace that is orthogonal to a hypercube vector

Reduction from Wishart to Certification

- Suppose you can certify $\phi(W) \leq 2 - \varepsilon$ when $W \sim \mathrm{GOE}(n)$
  - Recall $\phi(W) = \max_{x \in \{\pm 1/\sqrt{n}\}^n} x^\top W x$
- Then you can certify that the top δn-dimensional eigenspace of W does not contain a hypercube vector
- So you can certify that a random δn-dimensional subspace does not contain a hypercube vector
- So you can distinguish between a random δn-dimensional subspace and a δn-dimensional subspace containing a hypercube vector
- So you can distinguish between a random $(1-\delta)n$-dimensional subspace and a $(1-\delta)n$-dimensional subspace that is orthogonal to a hypercube vector
- But this is exactly the Wishart problem with $\beta = -1$ and $N = (1-\delta)n$, which is hard $\Rightarrow$ contradiction

Summary

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)
  - Sampling

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)
  - Sampling
  - Counting solutions

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)
  - Sampling
  - Counting solutions
- For constrained PCA, we gave low-degree evidence that certification is hard by reduction from a hypothesis testing problem (negatively-spiked Wishart)

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)
  - Sampling
  - Counting solutions
- For constrained PCA, we gave low-degree evidence that certification is hard by reduction from a hypothesis testing problem (negatively-spiked Wishart)
- Future direction: how to systematically predict hardness for other types of certification/search/etc problems?

Summary

- Low-degree method: systematic way to predict when hypothesis testing problems are computationally easy/hard
- But what about other types of average-case problems?
  - Search
  - Certification
  - Recovery (e.g. tensor decomposition)
  - Sampling
  - Counting solutions
- For constrained PCA, we gave low-degree evidence that certification is hard by reduction from a hypothesis testing problem (negatively-spiked Wishart)
- Future direction: how to systematically predict hardness for other types of certification/search/etc problems?

Thanks!