The Complexity of Sparse Tensor PCA

Davin Choo [1]   Tommaso D'Orsi [2]

[1]National University of Singapore

[2]ETH Zürich

Motivating example

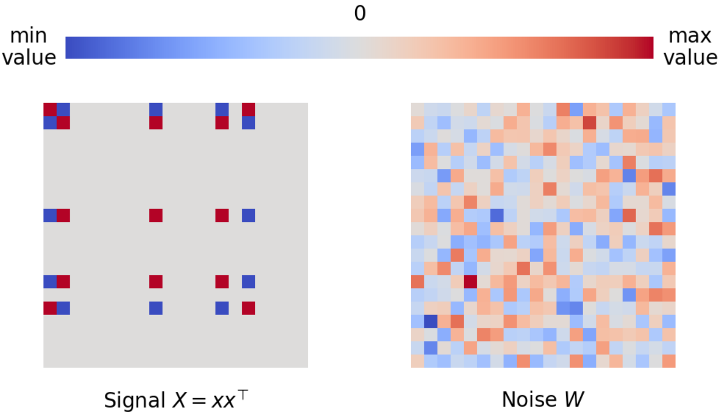


Figure: $x = \left[\frac{1}{\sqrt{5}}, \frac{-1}{\sqrt{5}}, 0, \ldots, 0, \frac{-1}{\sqrt{5}}, 0, \ldots, 0, \frac{-1}{\sqrt{5}}, 0, \frac{1}{\sqrt{5}}, 0, \ldots, 0\right]^\top$. $\mathrm{supp}(x) = \{1, 2, 9, 14, 16\}$. Noise $\boldsymbol{W} \in \mathbb{R}^{20\times 20}$ has i.i.d. $N(0,1)$ entries.

Motivating example

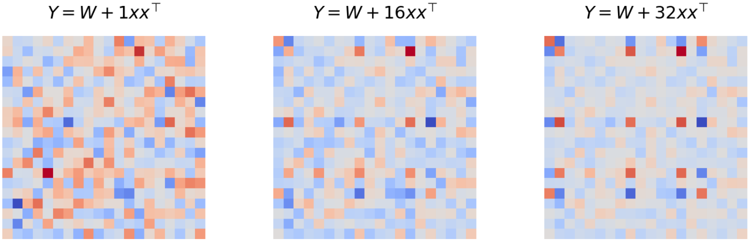


Figure: Observation $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top$ under different signal strengths $\lambda \in \{1, 16, 32\}$. Colours are rescaled to emphasize relative values.

Sparse tensor PCA model (Simplified)

Observe: Tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, for $p \geq 2$

- $\boldsymbol{W}$ is order p tensor with i.i.d. $N(0,1)$ entries
- Signal x is flat, k-sparse, unit length

Approximate recovery: Find $\widehat{x}$ such that $|\langle x, \widehat{x} \rangle| \gg 1 - o(1)$

Sparse tensor PCA model (Simplified)

Observe: Tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, for $p \geq 2$

- $\boldsymbol{W}$ is order p tensor with i.i.d. $N(0, 1)$ entries
- Signal x is flat, k-sparse, unit length

Approximate recovery: Find $\widehat{x}$ such that $|\langle x, \widehat{x} \rangle| \gg 1 - o(1)$

Extensions (briefly discussed later)

- Approximately flat signals
- Multiple spikes
- General tensor spikes

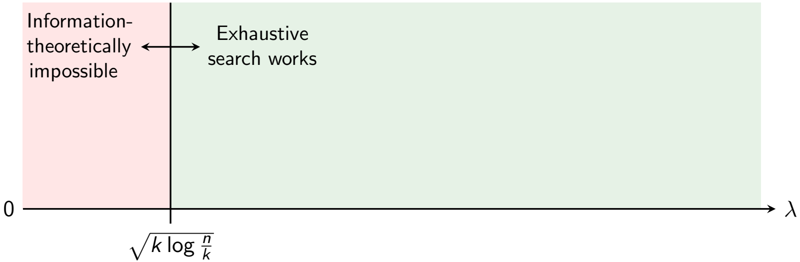
Remark: See paper for references (no citations for cleaner slides)

Sparse (Wigner) PCA: $p = 2$, $k \leq n$

- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\mathrm{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)

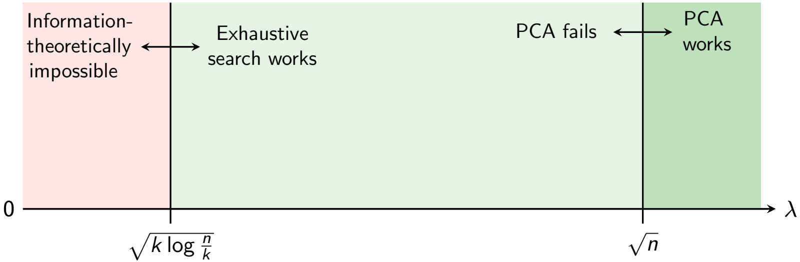
Sparse (Wigner) PCA: $p = 2,\ k \leq n$

- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\mathrm{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)



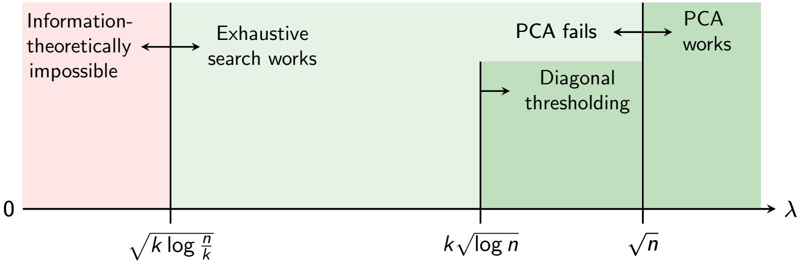
Sparse (Wigner) PCA: $p = 2,\ k \leq n$

- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\mathrm{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)



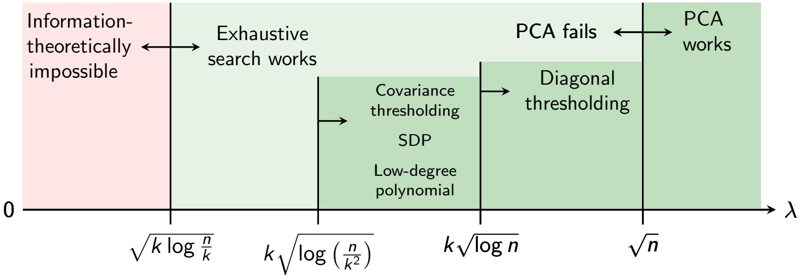
Sparse (Wigner) PCA: $p = 2$, $k \le n$

- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\mathrm{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)



Sparse (Wigner) PCA: $p = 2$, $k \leq n$

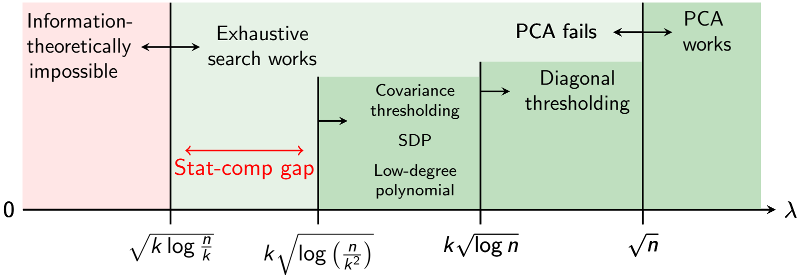
- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\text{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)



- Efficient algorithms when $\sqrt{k} \ll \lambda \ll \min\{k, \sqrt{n}\}$?

Sparse (Wigner) PCA: $p = 2$, $k \leq n$

- Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x x^\top \in \mathbb{R}^{n \times n}$
- x is k-sparse ($|\mathrm{supp}(x)| = k$) and unit length ($\|x\|_2 = 1$)



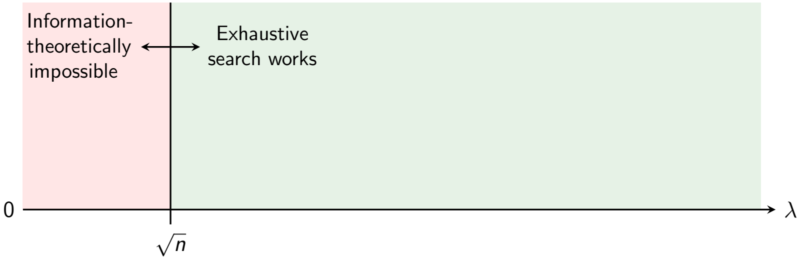
- Efficient algorithms when $\sqrt{k} \ll \lambda \ll \min\{k, \sqrt{n}\}$?
- Unlikely due to Sum-of-Squares (SoS) lower bound

Tensor PCA: $p \geq 2$, $k = n$

- Observe $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p} \in \mathbb{R}^{n^{\otimes p}}$
- Signal x is unit length ($\|x\|_2 = 1$)
- Remark: Computing $\max_{\|x\|=1} \langle \boldsymbol{Y}, x^{\otimes p} \rangle$ is NP-hard for $p \geq 3$

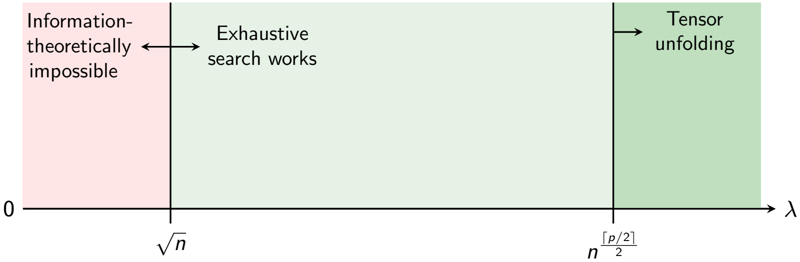
Tensor PCA: $p \geq 2$, $k = n$

- Observe $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p} \in \mathbb{R}^{n^{\otimes p}}$
- Signal x is unit length ($\|x\|_2 = 1$)
- Remark: Computing $\max_{\|x\|=1} \langle \boldsymbol{Y}, x^{\otimes p} \rangle$ is NP-hard for $p \geq 3$



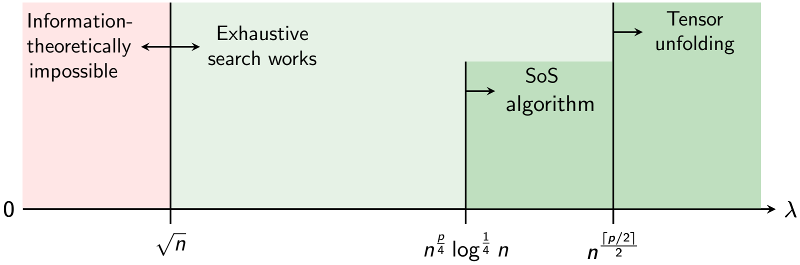
Tensor PCA: $p \geq 2$, $k = n$

- Observe $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p} \in \mathbb{R}^{n^{\otimes p}}$
- Signal x is unit length ($\|x\|_2 = 1$)
- Remark: Computing $\max_{\|x\|=1} \langle \boldsymbol{Y}, x^{\otimes p} \rangle$ is NP-hard for $p \geq 3$



Tensor PCA: $p \geq 2$, $k = n$

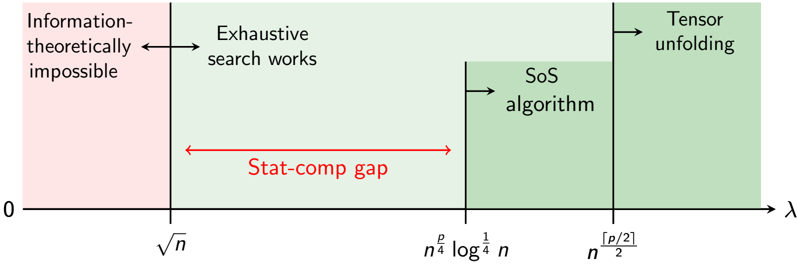
- Observe $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p} \in \mathbb{R}^{n^{\otimes p}}$
- Signal x is unit length ($\|x\|_2 = 1$)
- Remark: Computing $\max_{\|x\|=1} \langle \boldsymbol{Y}, x^{\otimes p} \rangle$ is NP-hard for $p \geq 3$



- Efficient algorithms when $\sqrt{n} \ll \lambda \ll n^{\frac{p}{4}}$?

Tensor PCA: $p \geq 2$, $k = n$

- Observe $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p} \in \mathbb{R}^{n^{\otimes p}}$
- Signal x is unit length ($\|x\|_2 = 1$)
- Remark: Computing $\max_{\|x\|=1} \langle \boldsymbol{Y}, x^{\otimes p} \rangle$ is NP-hard for $p \geq 3$



- Efficient algorithms when $\sqrt{n} \ll \lambda \ll n^{\frac{p}{4}}$?
- Unlikely due to Sum-of-Squares (SoS) lower bound

Setup

Sparse tensor PCA (single spike)

Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$

- $\boldsymbol{W}$ is order p tensor with i.i.d. $N(0,1)$ entries
- Signal x is flat, k-sparse, unit length

Goal: Find $\widehat{x}$ such that $|\langle x, \widehat{x} \rangle| \gg 1 - o(1)$

Setup

Sparse tensor PCA (single spike)

Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$

- $\boldsymbol{W}$ is order p tensor with i.i.d. $N(0,1)$ entries
- Signal x is flat, k-sparse, unit length

Goal: Find $\widehat{x}$ such that $|\langle x, \widehat{x} \rangle| \gg 1 - o(1)$

Simplifying assumptions

- We will ignore some technical preprocessing steps
- We will briefly discuss how we handle some extensions at the end such as the case where there are multiple planted signals

A parametric recovery algorithm

Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$; x is k-sparse, flat and unit length

Recovery algorithm

Let $1 \leq t \leq k$ be a computational parameter. Suppose

$$\lambda \gtrsim \sqrt{t\left(\frac{k}{t}\right)^p \log n}\,.$$

Then, there exists an algorithm that runs in $\mathcal{O}(pn^{p+t})$ time and, with probability 0.99, outputs the support of x.

A parametric recovery algorithm

Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$; x is k-sparse, flat and unit length

Recovery algorithm

Let $1 \leq t \leq k$ be a computational parameter. Suppose

$$\lambda \gtrsim \sqrt{t \left(\frac{k}{t}\right)^p \log n} \, .$$

Then, there exists an algorithm that runs in $\mathcal{O}(pn^{p+t})$ time and, with probability 0.99, outputs the support of x.

Extreme values of t

- $t = 1 \Rightarrow$ Polynomial running time, $\lambda \gtrsim \sqrt{k^p \log n}$
- $t = k \Rightarrow$ Exponential running time, $\lambda \gtrsim \sqrt{k \log n}$

A parametric recovery algorithm

Observe tensor $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$; x is k-sparse, flat and unit length

Recovery algorithm

Let $1 \leq t \leq k$ be a computational parameter. Suppose

$$\lambda \gtrsim \sqrt{t \left(\frac{k}{t}\right)^p \log n}\,.$$

Then, there exists an algorithm that runs in $\mathcal{O}(pn^{p+t})$ time and, with probability 0.99, outputs the support of x.

Extreme values of t

- $t = 1 \Rightarrow$ Polynomial running time, $\lambda \gtrsim \sqrt{k^p \log n}$
- $t = k \Rightarrow$ Exponential running time, $\lambda \gtrsim \sqrt{k \log n}$

Remark: To obtain $\hat{x}$, run an existing tensor PCA algorithm on the smaller subtensor defined by the support of x.

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\text{supp}(u)| = t \right\}$

Limited brute force

- Define $U_t = \left\{u \in \left\{-\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}}\right\} : |\mathrm{supp}(u)| = t\right\}$
- For any $u \in U_t$,
  - Since $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, $\langle \boldsymbol{Y}, u^{\otimes p} \rangle = \langle \boldsymbol{W}, u^{\otimes p} \rangle + \lambda \langle u, x \rangle^p$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For any $u \in U_t$,
  - Since $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, $\langle \boldsymbol{Y}, u^{\otimes p} \rangle = \langle \boldsymbol{W}, u^{\otimes p} \rangle + \lambda \langle u, x \rangle^p$
  - Since $\boldsymbol{W}$ is Gaussian tensor, $|\langle \boldsymbol{W}, u^{\otimes p} \rangle| \leq \mathcal{O}(\sqrt{t \log n})$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\text{supp}(u)| = t \right\}$
- For any $u \in U_t$,
  - Since $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, $\langle \boldsymbol{Y}, u^{\otimes p} \rangle = \langle \boldsymbol{W}, u^{\otimes p} \rangle + \lambda \langle u, x \rangle^p$
  - Since $\boldsymbol{W}$ is Gaussian tensor, $|\langle \boldsymbol{W}, u^{\otimes p} \rangle| \leq \mathcal{O}(\sqrt{t \log n})$
  - If $|\text{supp}(u) \cap \text{supp}(x)| = t$, then

$$\langle \boldsymbol{Y}, u^{\otimes p} \rangle \geq \lambda \cdot \left(\frac{t}{k} \right)^{\frac{p}{2}} - \mathcal{O}(\sqrt{t \log n})$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For any $u \in U_t$,
  - Since $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, $\langle \boldsymbol{Y}, u^{\otimes p} \rangle = \langle \boldsymbol{W}, u^{\otimes p} \rangle + \lambda \langle u, x \rangle^p$
  - Since $\boldsymbol{W}$ is Gaussian tensor, $|\langle \boldsymbol{W}, u^{\otimes p} \rangle| \leq \mathcal{O}(\sqrt{t \log n})$
  - If $|\mathrm{supp}(u) \cap \mathrm{supp}(x)| = t$, then
  $$\langle \boldsymbol{Y}, u^{\otimes p} \rangle \geq \lambda \cdot \left(\frac{t}{k} \right)^{\frac{p}{2}} - \mathcal{O}(\sqrt{t \log n})$$
  - If $|\mathrm{supp}(u) \cap \mathrm{supp}(x)| < (1 - \epsilon) \cdot t$, then
  $$\langle \boldsymbol{Y}, u^{\otimes p} \rangle \leq \lambda \cdot (1 - \epsilon)^p \cdot \left(\frac{t}{k} \right)^{\frac{p}{2}} + \mathcal{O}(\sqrt{t \log n})$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\text{supp}(u)| = t \right\}$
- For any $u \in U_t$,
  - Since $\boldsymbol{Y} = \boldsymbol{W} + \lambda x^{\otimes p}$, $\langle \boldsymbol{Y}, u^{\otimes p} \rangle = \langle \boldsymbol{W}, u^{\otimes p} \rangle + \lambda \langle u, x \rangle^p$
  - Since $\boldsymbol{W}$ is Gaussian tensor, $|\langle \boldsymbol{W}, u^{\otimes p} \rangle| \leq \mathcal{O}(\sqrt{t \log n})$
  - If $|\text{supp}(u) \cap \text{supp}(x)| = t$, then

$$\langle \boldsymbol{Y}, u^{\otimes p} \rangle \geq \lambda \cdot \left(\frac{t}{k} \right)^{\frac{p}{2}} - \mathcal{O}(\sqrt{t \log n})$$

  - If $|\text{supp}(u) \cap \text{supp}(x)| < (1 - \epsilon) \cdot t$, then

$$\langle \boldsymbol{Y}, u^{\otimes p} \rangle \leq \lambda \cdot (1 - \epsilon)^p \cdot \left(\frac{t}{k} \right)^{\frac{p}{2}} + \mathcal{O}(\sqrt{t \log n})$$

- For $\boldsymbol{v}_* = \text{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\text{supp}(\boldsymbol{v}_*) \cap \text{supp}(x)| \geq (1 - \epsilon) \cdot t$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1 - \epsilon) \cdot t$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1-\epsilon) \cdot t$$

- Define $\boldsymbol{\alpha} \in \mathbb{R}^n$ with entries $\boldsymbol{\alpha}_\ell := \langle \boldsymbol{Y}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle$, $\forall \ell \in [n]$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\text{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \text{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\text{supp}(\boldsymbol{v}_*) \cap \text{supp}(x)| \geq (1 - \epsilon) \cdot t$$

- Define $\boldsymbol{\alpha} \in \mathbb{R}^n$ with entries $\boldsymbol{\alpha}_\ell := \langle \boldsymbol{Y}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle$, $\forall \ell \in [n]$
- For all $\ell \in \text{supp}(x)$,

$$\boldsymbol{\alpha}_\ell = \lambda \cdot x_\ell \cdot \langle x, \boldsymbol{v}_* \rangle^{p-1} + \langle \boldsymbol{W}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle \gtrsim \sqrt{\log n}$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1 - \epsilon) \cdot t$$

- Define $\boldsymbol{\alpha} \in \mathbb{R}^n$ with entries $\boldsymbol{\alpha}_\ell := \langle \boldsymbol{Y}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle$, $\forall \ell \in [n]$
- For all $\ell \in \mathrm{supp}(x)$,

$$\boldsymbol{\alpha}_\ell = \lambda \cdot x_\ell \cdot \langle x, \boldsymbol{v}_* \rangle^{p-1} + \langle \boldsymbol{W}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle \gtrsim \sqrt{\log n}$$

- For all $\ell \notin \mathrm{supp}(x)$ (i.e. $x_\ell = 0$),

$$\boldsymbol{\alpha}_\ell = \langle \boldsymbol{W}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle \lesssim \sqrt{\log n}$$

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1 - \epsilon) \cdot t$$

- Define $\boldsymbol{\alpha} \in \mathbb{R}^n$ with entries $\boldsymbol{\alpha}_\ell := \langle \boldsymbol{Y}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle$, $\forall \ell \in [n]$
- For all $\ell \in \mathrm{supp}(x)$,

$$\boldsymbol{\alpha}_\ell = \lambda \cdot x_\ell \cdot \langle x, \boldsymbol{v}_* \rangle^{p-1} + \langle \boldsymbol{W}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle \gtrsim \sqrt{\log n}$$

- For all $\ell \notin \mathrm{supp}(x)$ (i.e. $x_\ell = 0$),

$$\boldsymbol{\alpha}_\ell = \langle \boldsymbol{W}, \boldsymbol{v}_*^{\otimes p-1} \otimes e_\ell \rangle \lesssim \sqrt{\log n}$$

- Using $\boldsymbol{v}_*$, the vector $\boldsymbol{\alpha}$ acts as indicator of the support of x

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1 - \epsilon) \cdot t$$

- Using $\boldsymbol{v}_*$, the vector $\boldsymbol{\alpha}$ acts as indicator of the support of x

Limited brute force

- Define $U_t = \left\{ u \in \left\{ -\frac{1}{\sqrt{t}}, 0, \frac{1}{\sqrt{t}} \right\} : |\mathrm{supp}(u)| = t \right\}$
- For $\boldsymbol{v}_* = \mathrm{argmax}_{u \in U_t} \langle \boldsymbol{Y}, u^{\otimes p} \rangle$,

$$|\mathrm{supp}(\boldsymbol{v}_*) \cap \mathrm{supp}(x)| \geq (1 - \epsilon) \cdot t$$

- Using $\boldsymbol{v}_*$, the vector $\boldsymbol{\alpha}$ acts as indicator of the support of x
- Relationship with known algorithms
  - When $t = 1$, this recovers the idea of diagonal thresholding (Pick out largest coordinate, one at a time)
  - When $t = k$, this is literally brute force (MLE)

Algorithmic extensions

Multiple spikes

- $\boldsymbol{Y} = \boldsymbol{W} + \sum_{q=1}^{r} \lambda_q x_{(q)}^{\otimes p}$
- Disjoint support assumption: $\text{supp}(x_i) \cap \text{supp}(x_j) = \emptyset$

Algorithmic extensions

Multiple spikes

- $\boldsymbol{Y} = \boldsymbol{W} + \sum_{q=1}^{r} \lambda_q x_{(q)}^{\otimes p}$
- Disjoint support assumption: $\mathrm{supp}(x_i) \cap \mathrm{supp}(x_j) = \emptyset$

Approximately flat k-sparse unit length signals

Fix a constant $A \geq 1$. For k-sparse signal x,

$$\frac{1}{A\sqrt{k}} \leq |x_\ell| \leq \frac{A}{\sqrt{k}} \quad \text{for } \ell \in \mathrm{supp}(x)$$

Algorithmic extensions

Multiple spikes

- $\boldsymbol{Y} = \boldsymbol{W} + \sum_{q=1}^{r} \lambda_q x_{(q)}^{\otimes p}$
- Disjoint support assumption: $\mathrm{supp}(x_i) \cap \mathrm{supp}(x_j) = \emptyset$

Approximately flat k-sparse unit length signals

Fix a constant $A \geq 1$. For k-sparse signal x,

$$\frac{1}{A\sqrt{k}} \leq |x_\ell| \leq \frac{A}{\sqrt{k}} \quad \text{for } \ell \in \mathrm{supp}(x)$$

General tensor spike

Instead of just $x^{\otimes p}$, we can allow the tensor signal to be $x_{(1)} \otimes \ldots \otimes x_{(p)}$ involving $1 \leq \ell \leq p$ distinct k-sparse vectors

Lower bound summary

- Given a recovery algorithm, one can distinguish whether $\lambda = 0$

Lower bound summary

- Given a recovery algorithm, one can distinguish whether $\lambda = 0$
- Computational lower bound
  - Premise: Low-deg polynomials capture efficient functions
  - Likelihood ratio test, restricted to low-degree polynomials
  - Special cases of our bounds match known sparse PCA and tensor PCA low-degree bounds

Lower bound summary

- Given a recovery algorithm, one can distinguish whether $\lambda = 0$
- Computational lower bound
  - Premise: Low-deg polynomials capture efficient functions
  - Likelihood ratio test, restricted to low-degree polynomials
  - Special cases of our bounds match known sparse PCA and tensor PCA low-degree bounds
- Information-theoretic lower bound
  - Fano's inequality on ϵ-packing of flat k-sparse unit vectors U_k
  - Our bound is equivalent (up to constants) with recent works that study phase transition for weak recovery

Key contributions

1. A parametric multi-spike recovery algorithm for sparse tensor PCA that trades off running time with signal strength requirements
   - Given exponential time, our algorithm can recover the signal at the best known information-theoretic threshold
   - If we insist on polynomial time, our algorithm recovers the signal at the best known computational threshold
2. A computational lower bound based on low-degree polynomials and the low-degree likelihood method
3. An information-theoretic lower bound for approximate recovery