The Kikuchi Hierarchy and Tensor PCA

Alex Wein
Courant Institute, NYU

Joint work with:

Ahmed El Alaoui
Stanford

Cris Moore
Santa Fe Institute
High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...
High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...

Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution
Statistical Physics of Inference

- High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...

- Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution

- Algorithms: belief propagation (BP) [Pearl '86], approximate message passing (AMP) [Donoho-Maleki-Montanari '09]
Statistical Physics of Inference

- High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...

- Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution

- Algorithms: belief propagation (BP) [Pearl '86], approximate message passing (AMP) [Donoho-Maleki-Montanari '09]
  - Known/believed to be optimal in many settings
Statistical Physics of Inference

- High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...

- Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution

- Algorithms: belief propagation (BP) [Pearl '86], approximate message passing (AMP) [Donoho-Maleki-Montanari '09]
  - Known/believed to be optimal in many settings
  - Sharp results: exact MMSE, phase transitions
Statistical Physics of Inference

- High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...

- Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution

- Algorithms: belief propagation (BP) [Pearl ’86], approximate message passing (AMP) [Donoho-Maleki-Montanari ’09]
  - Known/believed to be optimal in many settings
  - Sharp results: exact MMSE, phase transitions

Statistical Physics of Inference

- High-dimensional inference problems: compressed sensing, community detection, spiked Wigner/Wishart, sparse PCA, planted clique, group synchronization, ...
- Connection to statistical physics: posterior distribution is a Gibbs/Boltzmann distribution
- Algorithms: belief propagation (BP) [Pearl '86], approximate message passing (AMP) [Donoho-Maleki-Montanari '09]
  - Known/believed to be optimal in many settings
  - Sharp results: exact MMSE, phase transitions

This theory has been hugely successful at precisely understanding statistical and computational limits of many problems.
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy [Parrilo '00, Lasserre '01]
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy [Parrilo '00, Lasserre '01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy [Parrilo ’00, Lasserre ’01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-\(d\) relaxation can be solved in \(n^{O(d)}\)-time
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy \cite{Parrilo00, Lasserre01}

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-$d$ relaxation can be solved in $n^{O(d)}$-time
- Higher degree gives more powerful algorithms

Meta-question: unify the statistical physics and SoS approaches?

This talk: case study on tensor PCA – a problem where statistical physics and SoS disagree (!!!)
A competing theory: sum-of-squares hierarchy  [Parrilo ’00, Lasserre ’01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-$d$ relaxation can be solved in $n^{O(d)}$-time
- Higher degree gives more powerful algorithms
- State-of-the-art algorithms for many statistical problems:
  tensor decomposition, tensor completion, planted sparse vector, dictionary learning, refuting random CSPs, mixtures of Gaussians, ...
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy [Parrilo '00, Lasserre '01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-$d$ relaxation can be solved in $n^{O(d)}$-time
- Higher degree gives more powerful algorithms
- State-of-the-art algorithms for many statistical problems:
  - tensor decomposition, tensor completion, planted sparse vector,
  - dictionary learning, refuting random CSPs, mixtures of Gaussians, ...
- Evidence for computational hardness: SoS lower bounds
A competing theory: sum-of-squares hierarchy [Parrilo '00, Lasserre '01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-$d$ relaxation can be solved in $n^{O(d)}$-time
- Higher degree gives more powerful algorithms
- State-of-the-art algorithms for many statistical problems: tensor decomposition, tensor completion, planted sparse vector, dictionary learning, refuting random CSPs, mixtures of Gaussians, ...
- Evidence for computational hardness: SoS lower bounds

Meta-question: unify the statistical physics and SoS approaches?
Sum-of-Squares (SoS) Hierarchy

A competing theory: sum-of-squares hierarchy [Parrilo '00, Lasserre '01]

- Systematic way to obtain convex relaxations of polynomial optimization problems
- Degree-$d$ relaxation can be solved in $n^{O(d)}$-time
- Higher degree gives more powerful algorithms
- State-of-the-art algorithms for many statistical problems:
  tensor decomposition, tensor completion, planted sparse vector,
  dictionary learning, refuting random CSPs, mixtures of Gaussians, ...
- Evidence for computational hardness: SoS lower bounds

Meta-question: unify the statistical physics and SoS approaches?

This talk: case study on tensor PCA – a problem where statistical physics and SoS disagree (!!!)
Tensor PCA (Principal Component Analysis)

Definition (Spiked Tensor Model [Richard-Montanari '14])

\[ x \in \{\pm 1\}^n - \text{signal} \]
\[ p \in \{2, 3, 4, \ldots\} - \text{tensor order} \]

For each subset \( U \subseteq [n] \) of size \(|U| = p\), observe

\[ Y_U = \lambda \prod_{i \in U} x_i + \mathcal{N}(0, 1) \]

\( \lambda \geq 0 - \text{signal-to-noise parameter} \)

Goal: given \( \{Y_U\} \), recover \( x \) (with high probability as \( n \to \infty \))

- “For every \( p \) variables, get a noisy observation of their parity”
- In tensor notation: \( Y = \lambda x^{\otimes p} + Z \) where \( Z \) is symmetric noise
- Case \( p = 2 \) is the spiked Wigner matrix model \( Y = \lambda xx^\top + Z \)
Maximum likelihood estimation (MLE):

\[
\Pr[x|Y] \propto \exp \left( \sum_{|U|=p} \lambda Y_U \prod_{i \in U} x_i \right) = \exp \left( \frac{\lambda}{p} \langle Y, x \otimes^p \rangle \right)
\]

MLE: \( \hat{x} = \arg\max_{ v \in \{\pm 1\}^n} \langle Y, v \otimes^p \rangle \)
Maximum likelihood estimation (MLE):

\[
\Pr[x \mid Y] \propto \exp \left( \sum_{|U| = p} \lambda Y_U \prod_{i \in U} x_i \right) = \exp \left( \frac{\lambda}{p} \langle Y, x \otimes^p \rangle \right)
\]

MLE: \( \hat{x} = \arg\max_{v \in \{\pm 1\}^n} \langle Y, v \otimes^p \rangle \)

- Succeeds when \( \lambda \gtrsim n^{(1-p)/2} \) \[Richard-Montanari '14\]
Algorithms for Tensor PCA

**Maximum likelihood estimation (MLE):**

$$\text{Pr}[x|Y] \propto \exp \left( \sum_{|U|=p} \lambda Y_U \prod_{i \in U} x_i \right) = \exp \left( \frac{\lambda}{p} \langle Y, x \otimes^p \rangle \right)$$

**MLE:**

$$\hat{x} = \arg\max_{v \in \{\pm 1\}^n} \langle Y, v \otimes^p \rangle$$

- Succeeds when $\lambda \gtrsim n^{(1-p)/2}$ [Richard-Montanari '14]
- Statistically optimal (up to constant factors in $\lambda$)
Algorithms for Tensor PCA

Maximum likelihood estimation (MLE):

\[
\Pr[x|Y] \propto \exp \left( \sum_{|U|=p} \lambda Y_U \prod_{i \in U} x_i \right) = \exp \left( \frac{\lambda}{p} \langle Y, x^{\otimes p} \rangle \right)
\]

MLE: \( \hat{x} = \arg\max_{v \in \{\pm 1\}^n} \langle Y, v^{\otimes p} \rangle \)

- Succeeds when \( \lambda \gtrsim n^{(1-p)/2} \) [Richard-Montanari '14]
- Statistically optimal (up to constant factors in \( \lambda \))
- Problem: requires exponential time \( 2^n \)
Algorithms for Tensor PCA

**Local algorithms:** keep track of a “guess” $v \in \mathbb{R}^n$ and locally maximize the log-likelihood $\mathcal{L}(v) = \langle Y, v^\otimes p \rangle$
Algorithms for Tensor PCA

**Local algorithms**: keep track of a “guess” \( v \in \mathbb{R}^n \) and locally maximize the log-likelihood \( \mathcal{L}(v) = \langle Y, v \otimes p \rangle \)

- Gradient descent [Ben Arous-Gheissari-Jagannath '18]
Algorithms for Tensor PCA

**Local algorithms:** keep track of a “guess” $v \in \mathbb{R}^n$ and locally maximize the log-likelihood $\mathcal{L}(v) = \langle Y, v \otimes p \rangle$

- Gradient descent [Ben Arous-Gheissari-Jagannath '18]
- Tensor power iteration [Richard-Montanari '14]
Algorithms for Tensor PCA

**Local algorithms:** keep track of a “guess” \( v \in \mathbb{R}^n \) and locally maximize the log-likelihood \( \mathcal{L}(v) = \langle Y, v^\otimes p \rangle \)

- **Gradient descent** [Ben Arous-Gheissari-Jagannath '18]

- **Tensor power iteration** [Richard-Montanari '14]

- **Langevin dynamics** [Ben Arous-Gheissari-Jagannath '18]
Local algorithms: keep track of a “guess” $v \in \mathbb{R}^n$ and locally maximize the log-likelihood $\mathcal{L}(v) = \langle Y, v^\otimes p \rangle$

- Gradient descent [Ben Arous-Gheissari-Jagannath '18]
- Tensor power iteration [Richard-Montanari '14]
- Langevin dynamics [Ben Arous-Gheissari-Jagannath '18]
- Approximate message passing (AMP) [Richard-Montanari '14]
Local algorithms: keep track of a “guess” $v \in \mathbb{R}^n$ and locally maximize the log-likelihood $\mathcal{L}(v) = \langle Y, v \otimes p \rangle$

- Gradient descent [Ben Arous-Gheissari-Jagannath '18]
- Tensor power iteration [Richard-Montanari '14]
- Langevin dynamics [Ben Arous-Gheissari-Jagannath '18]
- Approximate message passing (AMP) [Richard-Montanari '14]

These only succeed when $\lambda \gg n^{-1/2}$

- Recall: MLE works for $\lambda \sim n^{(1-p)/2}$
Algorithms for Tensor PCA

Sum-of-squares (SoS) and spectral methods:

\[ \lambda \gg n - p/4 \]

SoS lower bounds suggest no poly-time algorithm when
\[ \lambda \ll n - p/4 \]

\[ \lambda \text{impossible} \]
Algorithms for Tensor PCA

Sum-of-squares (SoS) and spectral methods:

- SoS semidefinite program [Hopkins-Shi-Steurer '15]
Algorithms for Tensor PCA

**Sum-of-squares (SoS) and spectral methods:**

- SoS semidefinite program [Hopkins-Shi-Steurer '15]
- Spectral SoS [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15]
Algorithms for Tensor PCA

**Sum-of-squares (SoS) and spectral methods:**

- **SoS semidefinite program** [Hopkins-Shi-Steurer '15]
- **Spectral SoS** [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15]
- **Tensor unfolding** [Richard-Montanari '14, Hopkins-Shi-Steurer '15]
Algorithms for Tensor PCA

Sum-of-squares (SoS) and spectral methods:

- SoS semidefinite program [Hopkins-Shi-Steurer '15]
- Spectral SoS [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15]
- Tensor unfolding [Richard-Montanari '14, Hopkins-Shi-Steurer '15]

These are poly-time and succeed when $\lambda \gg n^{-p/4}$
Algorithms for Tensor PCA

Sum-of-squares (SoS) and spectral methods:

- SoS semidefinite program [Hopkins-Shi-Steurer '15]
- Spectral SoS [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15]
- Tensor unfolding [Richard-Montanari '14, Hopkins-Shi-Steurer '15]

These are poly-time and succeed when $\lambda \gg n^{-p/4}$

SoS lower bounds suggest no poly-time algorithm when $\lambda \ll n^{-p/4}$

[Hopkins-Shi-Steurer '15, Hopkins-Kothari-Potechin-Raghavendra-Schramm-Steurer '17]
Algorithms for Tensor PCA

Sum-of-squares (SoS) and spectral methods:

- SoS semidefinite program [Hopkins-Shi-Steurer '15]
- Spectral SoS [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15]
- Tensor unfolding [Richard-Montanari '14, Hopkins-Shi-Steurer '15]

These are poly-time and succeed when $\lambda \gg n^{-p/4}$

SoS lower bounds suggest no poly-time algorithm when $\lambda \ll n^{-p/4}$

[Hopkins-Shi-Steurer '15, Hopkins-Kothari-Potechin-Raghavendra-Schramm-Steurer '17]

Local algorithms (gradient descent, AMP, ...) are suboptimal when $p \geq 3$
Subexponential-Time Algorithms

Subexponential-time: $2^{n^\delta}$ for $\delta \in (0, 1)$
Subexponential-Time Algorithms

Subexponential-time: $2^{n^\delta}$ for $\delta \in (0, 1)$

Tensor PCA has a smooth tradeoff between runtime and statistical power: for $\delta \in (0, 1)$,

$$\text{there is a } 2^{n^\delta} \text{-time algorithm for } \lambda \sim n^{-p/4+\delta(1/2-p/4)}$$

[Raghavendra-Rao-Schramm ’16, Bhattiprolu-Guruswami-Lee ’16]
Subexponential-Time Algorithms

Subexponential-time: $2^{n^{\delta}}$ for $\delta \in (0, 1)$

Tensor PCA has a smooth tradeoff between runtime and statistical power: for $\delta \in (0, 1)$,

there is a $2^{n^{\delta}}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

[Raghavendra-Rao-Schramm ’16, Bhattiprolu-Guruswami-Lee ’16]

Interpolates between SoS and MLE:

- $\delta = 0 \Rightarrow$ poly-time algorithm for $\lambda \sim n^{-p/4}$
- $\delta = 1 \Rightarrow$ $2^n$-time algorithm for $\lambda \sim n^{(1-p)/2}$
Subexponential-Time Algorithms

Subexponential-time: $2^{n^\delta}$ for $\delta \in (0, 1)$

Tensor PCA has a smooth tradeoff between runtime and statistical power: for $\delta \in (0, 1)$,

there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

[Raghavendra-Rao-Schramm ’16, Bhattiprolu-Guruswami-Lee ’16]

Interpolates between SoS and MLE:

- $\delta = 0 \implies$ poly-time algorithm for $\lambda \sim n^{-p/4}$
- $\delta = 1 \implies 2^n$-time algorithm for $\lambda \sim n^{(1-p)/2}$

In contrast, some problems have a sharp threshold

- E.g., $\lambda > 1$ is nearly-linear time; $\lambda < 1$ needs time $2^n$

For “soft” thresholds (like tensor PCA): BP/AMP can’t be optimal
Subexponential-Time Algorithms

Subexponential-time: $2^{n^\delta}$ for $\delta \in (0, 1)$

Tensor PCA has a smooth tradeoff between runtime and statistical power: for $\delta \in (0, 1)$,

there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4 + \delta(1/2 - p/4)}$

[Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Interpolates between SoS and MLE:

- $\delta = 0 \implies$ poly-time algorithm for $\lambda \sim n^{-p/4}$
- $\delta = 1 \implies 2^n$-time algorithm for $\lambda \sim n^{(1-p)/2}$

In contrast, some problems have a sharp threshold

- E.g., $\lambda > 1$ is nearly-linear time; $\lambda < 1$ needs time $2^n$
Subexponential-Time Algorithms

Subexponential-time: $2^{n^\delta}$ for $\delta \in (0, 1)$

Tensor PCA has a smooth tradeoff between runtime and statistical power: for $\delta \in (0, 1)$,

there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4 + \delta(1/2 - p/4)}$

[Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Interpolates between SoS and MLE:

- $\delta = 0 \Rightarrow$ poly-time algorithm for $\lambda \sim n^{-p/4}$
- $\delta = 1 \Rightarrow$ $2^n$-time algorithm for $\lambda \sim n^{(1-p)/2}$

In contrast, some problems have a sharp threshold

- E.g., $\lambda > 1$ is nearly-linear time; $\lambda < 1$ needs time $2^n$

For “soft” thresholds (like tensor PCA): BP/AMP can’t be optimal
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4 + \delta(1/2 - p/4)}$. 

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Arose from the study of SoS lower bounds, pseudo-calibration
- Idea: look for a low-degree polynomial (of $Y$) that distinguishes $P$ (spiked tensor) and $Q$ (pure noise)

$$\max_{\text{degree} \leq D} \frac{\mathbb{E}_{Y \sim P}[f(Y)]}{\mathbb{E}_{Y \sim Q}[f(Y)]^2} = \begin{cases} O(1) & \Rightarrow \text{"hard"} \\ \omega(1) & \Rightarrow \text{"easy"} \end{cases}$$

- Take deg-$D$ polynomials as a proxy for $n^{\tilde{\Theta}(D)}$-time algorithms

For more, see the survey Kunisky-W.-Bandeira, "Notes on Computational Hardness of Hypothesis Testing: Predictions using the Low-Degree Likelihood Ratio", arXiv:1907.11636
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4 + \delta(1/2 - p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Arose from the study of SoS lower bounds, pseudo-calibration

Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Arose from the study of SoS lower bounds, pseudo-calibration
- Idea: look for a low-degree polynomial (of $Y$) that distinguishes $P$ (spiked tensor) and $Q$ (pure noise)
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Idea: look for a low-degree polynomial (of $Y$) that distinguishes $\mathbb{P}$ (spiked tensor) and $\mathbb{Q}$ (pure noise)

$$\max_{f \text{ degree } \leq D} \frac{\mathbb{E}_{Y \sim \mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)^2]}} \overset{?}{=} \begin{cases} O(1) & \Rightarrow \text{“hard”} \\ \omega(1) & \Rightarrow \text{“easy”} \end{cases}$$
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Arose from the study of SoS lower bounds, pseudo-calibration
- Idea: look for a low-degree polynomial (of $Y$) that distinguishes $P$ (spiked tensor) and $Q$ (pure noise)

\[
\max_{f \text{ degree } \leq D} \frac{E_{Y \sim P}[f(Y)]}{\sqrt{E_{Y \sim Q}[f(Y)^2]}} \overset{?}{=} \begin{cases} O(1) & \Rightarrow \text{“hard”} \\ \omega(1) & \Rightarrow \text{“easy”} \end{cases}
\]

- Take deg-$D$ polynomials as a proxy for $n^{\tilde{O}(D)}$-time algorithms
Aside: Low-Degree Likelihood Ratio

Recall: there is a $2^{n^\delta}$-time algorithm for $\lambda \sim n^{-p/4+\delta(1/2-p/4)}$

Evidence that this tradeoff is optimal: low-degree likelihood ratio

- A relatively simple calculation that predicts the computational complexity of high-dimensional inference problems
- Arose from the study of SoS lower bounds, pseudo-calibration
  \cite{Barak-Hopkins-Kelner-Kothari-Moitra-Potechin '16, Hopkins-Steurer '17, Hopkins-Kothari-Potechin-Raghavendra-Schramm-Steurer '17, Hopkins PhD thesis '18}
- Idea: look for a low-degree polynomial (of $Y$) that distinguishes $P$ (spiked tensor) and $Q$ (pure noise)

\[
\max_{f \text{ degree } \leq D} \frac{\mathbb{E}_{Y \sim P}[f(Y)]}{\mathbb{E}_{Y \sim Q}[f(Y)^2]} \approx \begin{cases} O(1) \Rightarrow \text{“hard”} \\ \omega(1) \Rightarrow \text{“easy”} \end{cases}
\]

- Take deg-$D$ polynomials as a proxy for $n^{\tilde{O}(D)}$-time algorithms

For more, see the survey Kunisky-W.-Bandeira, “Notes on Computational Hardness of Hypothesis Testing: Predictions using the Low-Degree Likelihood Ratio”, arXiv:1907.11636
Our Contributions

We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time.

Analogous to SoS hierarchy.

We prove that these algorithms match the performance of SoS.

Both for poly-time and for subexponential-time tradeoff.

This refines and "redeems" the statistical physics approach to algorithm design.

Our algorithms and analysis are simpler than prior work.

This talk: even-order tensors only.

Similar results for refuting random XOR formulas.
Our Contributions

- We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  - Analogous to SoS hierarchy
- This refines and “redeems” the statistical physics approach to algorithm design
- Our algorithms and analysis are simpler than prior work
- This talk: even-order tensors only
- Similar results for refuting random XOR formulas
Our Contributions

- We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  - Analogous to SoS hierarchy
- We prove that these algorithms match the performance of SoS
  - Both for poly-time and for subexponential-time tradeoff
Our Contributions

- We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  - Analogous to SoS hierarchy

- We prove that these algorithms match the performance of SoS
  - Both for poly-time and for subexponential-time tradeoff

- This refines and “redeems” the statistical physics approach to algorithm design
Our Contributions

- We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  - Analogous to SoS hierarchy

- We prove that these algorithms match the performance of SoS
  - Both for poly-time and for subexponential-time tradeoff

- This refines and “redeems” the statistical physics approach to algorithm design

- Our algorithms and analysis are simpler than prior work
Our Contributions

▶ We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  ▶ Analogous to SoS hierarchy

▶ We prove that these algorithms match the performance of SoS
  ▶ Both for poly-time and for subexponential-time tradeoff

▶ This refines and “redeems” the statistical physics approach to algorithm design

▶ Our algorithms and analysis are simpler than prior work

▶ This talk: even-order tensors only
Our Contributions

- We give a hierarchy of increasingly powerful BP/AMP-type algorithms: level $\ell$ requires $n^{O(\ell)}$ time
  - Analogous to SoS hierarchy

- We prove that these algorithms match the performance of SoS
  - Both for poly-time and for subexponential-time tradeoff

- This refines and “redeems” the statistical physics approach to algorithm design

- Our algorithms and analysis are simpler than prior work

- This talk: even-order tensors only

- Similar results for refuting random XOR formulas
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{\pm 1\}^n$, observed data $Y$.

Want to understand posterior $\Pr[x|Y]$.

Find distribution $\mu$ over $\{\pm 1\}^n$ minimizing free energy $F(\mu) = E(\mu) - S(\mu)$.

▶ “Energy” and “entropy” terms

The unique minimizer is $\Pr[x|Y]$.

Problem: need exponentially-many parameters to describe $\mu$.

BP/AMP: just keep track of marginals $m_i = E[x_i]$ and minimize a proxy, Bethe free energy $B(m)$.

▶ Locally minimize $B(m)$ via iterative update.
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal \( x \in \{\pm 1\}^n \), observed data \( Y \)
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{\pm 1\}^n$, observed data $Y$

Want to understand posterior $\Pr[x|Y]$
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{-1\}^n$, observed data $Y$

Want to understand posterior $\Pr[x|Y]$

Find distribution $\mu$ over $\{-1\}^n$ minimizing free energy
$\mathcal{F}(\mu) = \mathcal{E}(\mu) - S(\mu)$

▶ “Energy” and “entropy” terms
▶ The unique minimizer is $\Pr[x|Y]$
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{\pm 1\}^n$, observed data $Y$

Want to understand posterior $\Pr[x|Y]$

Find distribution $\mu$ over $\{\pm 1\}^n$ minimizing free energy $F(\mu) = E(\mu) - S(\mu)$

- “Energy” and “entropy” terms
- The unique minimizer is $\Pr[x|Y]$

Problem: need exponentially-many parameters to describe $\mu$
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{\pm 1\}^n$, observed data $Y$

Want to understand posterior $\Pr[x|Y]$

Find distribution $\mu$ over $\{\pm 1\}^n$ minimizing free energy $\mathcal{F}(\mu) = \mathcal{E}(\mu) - S(\mu)$

- “Energy” and “entropy” terms
- The unique minimizer is $\Pr[x|Y]$

Problem: need exponentially-many parameters to describe $\mu$

BP/AMP: just keep track of marginals $m_i = \mathbb{E}[x_i]$ and minimize a proxy, Bethe free energy $\mathcal{B}(m)$
Motivating the Algorithm: Belief Propagation / AMP

General setup: unknown signal $x \in \{\pm 1\}^n$, observed data $Y$

Want to understand posterior $\Pr[x|Y]$

Find distribution $\mu$ over $\{\pm 1\}^n$ minimizing free energy $\mathcal{F}(\mu) = \mathcal{E}(\mu) - S(\mu)$

- “Energy” and “entropy” terms
- The unique minimizer is $\Pr[x|Y]$

Problem: need exponentially-many parameters to describe $\mu$

BP/AMP: just keep track of marginals $m_i = \mathbb{E}[x_i]$ and minimize a proxy, Bethe free energy $\mathcal{B}(m)$

- Locally minimize $\mathcal{B}(m)$ via iterative update
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $B(m)$

Natural higher-order variant:
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals \( m_i = \mathbb{E}[x_i] \) and minimizes Bethe free energy \( \mathcal{B}(m) \)

Natural higher-order variant:

- Keep track of \( m_i = \mathbb{E}[x_i] \), \( m_{ij} = \mathbb{E}[x_i x_j] \), \ldots \) (up to degree \( \ell \))
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$.

Natural higher-order variant:

- Keep track of $m_i = \mathbb{E}[x_i]$, $m_{ij} = \mathbb{E}[x_ix_j], \ldots$ (up to degree $\ell$)
- Minimize Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51]
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$

Natural higher-order variant:

- Keep track of $m_i = \mathbb{E}[x_i]$, $m_{ij} = \mathbb{E}[x_i x_j]$, ... (up to degree $\ell$)
- Minimize Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51]

Various ways to locally minimize Kikuchi free energy
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$

Natural higher-order variant:

- Keep track of $m_i = \mathbb{E}[x_i], m_{ij} = \mathbb{E}[x_ix_j], \ldots$ (up to degree $\ell$)
- Minimize Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51]

Various ways to locally minimize Kikuchi free energy

- Gradient descent
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$

Natural higher-order variant:

- Keep track of $m_i = \mathbb{E}[x_i]$, $m_{ij} = \mathbb{E}[x_ix_j]$, \ldots (up to degree $\ell$)
- Minimize Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51]

Various ways to locally minimize Kikuchi free energy

- Gradient descent
- Generalized belief propagation (GBP) [Yedidia-Freeman-Weiss '03]
Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[x_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$

Natural higher-order variant:

- Keep track of $m_i = \mathbb{E}[x_i]$, $m_{ij} = \mathbb{E}[x_i x_j]$, ... (up to degree $\ell$)
- Minimize Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51]

Various ways to locally minimize Kikuchi free energy

- Gradient descent
- Generalized belief propagation (GBP) [Yedidia-Freeman-Weiss '03]
- We will use a spectral method based on the Kikuchi Hessian
The Kikuchi Hessian
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

- Recall: want to minimize $\mathcal{B}(m)$ with respect to $m = \{m_i\}$
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

- Recall: want to minimize $B(m)$ with respect to $m = \{m_i\}$
- Trivial “uninformative” stationary point $m^*$ where $\nabla B(m) = 0$
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

► Recall: want to minimize $B(m)$ with respect to $m = \{m_i\}$
► Trivial “uninformative” stationary point $m^*$ where $\nabla B(m) = 0$
► Bethe Hessian matrix $H_{ij} = \left. \frac{\partial^2 B}{\partial m_i \partial m_j} \right|_{m=m^*}$
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

▶ Recall: want to minimize $\mathcal{B}(m)$ with respect to $m = \{m_i\}$
▶ Trivial “uninformative” stationary point $m^*$ where $\nabla \mathcal{B}(m) = 0$
▶ Bethe Hessian matrix $H_{ij} = \frac{\partial^2 \mathcal{B}}{\partial m_i \partial m_j} \big|_{m=m^*}$
▶ Algorithm: compute bottom eigenvector of $H$
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

- Recall: want to minimize \( B(m) \) with respect to \( m = \{m_i\} \)
- Trivial “uninformative” stationary point \( m^* \) where \( \nabla B(m) = 0 \)
- Bethe Hessian matrix \( H_{ij} = \frac{\partial^2 B}{\partial m_i \partial m_j} |_{m=m^*} \)
- Algorithm: compute bottom eigenvector of \( H \)
- Why: best direction of local improvement
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

- Recall: want to minimize $B(m)$ with respect to $m = \{m_i\}$
- Trivial “uninformative” stationary point $m^*$ where $\nabla B(m) = 0$
- Bethe Hessian matrix $H_{ij} = \frac{\partial^2 B}{\partial m_i \partial m_j}|_{m=m^*}$
- Algorithm: compute bottom eigenvector of $H$
- Why: best direction of local improvement
- Spectral method with performance essentially as good as BP for community detection
The Kikuchi Hessian

Bethe Hessian approach [Saade-Krzakala-Zdeborová ’14]

- Recall: want to minimize $B(m)$ with respect to $m = \{m_i\}$
- Trivial “uninformative” stationary point $m^*$ where $\nabla B(m) = 0$
- Bethe Hessian matrix $H_{ij} = \frac{\partial^2 B}{\partial m_i \partial m_j} |_{m=m^*}$
- Algorithm: compute bottom eigenvector of $H$
- Why: best direction of local improvement
- Spectral method with performance essentially as good as BP for community detection

Our approach: Kikuchi Hessian

- Bottom eigenvector of Hessian of $K(m)$ with respect to moments $m = \{m_i, m_{ij}, \ldots\}$
The Algorithm

Definition (Symmetric Difference Matrix)

Input: an order-$p$ tensor $Y = (Y_U)_{|U| = p}$ (with $p$ even) and an integer $\ell$ in the range $p/2 \leq \ell \leq n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by $\ell$-subsets of $[n]$)

$$M_{S,T} = \begin{cases} Y_{S\triangle T} & \text{if } |S \triangle T| = p, \\ 0 & \text{otherwise.} \end{cases}$$
Definition (Symmetric Difference Matrix)

Input: an order-$p$ tensor $Y = (Y_{|U| = p})$ (with $p$ even) and an integer $\ell$ in the range $p/2 \leq \ell \leq n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by $\ell$-subsets of $[n]$)

$$M_{S,T} = \begin{cases} 
Y_{S \triangle T} & \text{if } |S \triangle T| = p, \\
0 & \text{otherwise.}
\end{cases}$$

- This is (approximately) a submatrix of the Kikuchi Hessian
The Algorithm

Definition (Symmetric Difference Matrix)

Input: an order-$p$ tensor $Y = (Y_U)_{|U|=p}$ (with $p$ even) and an integer $\ell$ in the range $p/2 \leq \ell \leq n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by $\ell$-subsets of $[n]$)

$$M_{S,T} = \begin{cases} 
    Y_{S \triangle T} & \text{if } |S \triangle T| = p, \\
    0 & \text{otherwise.}
\end{cases}$$

- This is (approximately) a submatrix of the Kikuchi Hessian
- Algorithm: compute leading eigenvalue/eigenvector of $M$
The Algorithm

Definition (Symmetric Difference Matrix)

Input: an order-$p$ tensor $Y = (Y_U)_{|U|=p}$ (with $p$ even) and an integer $\ell$ in the range $p/2 \leq \ell \leq n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by $\ell$-subsets of $[n]$)

$$M_{S,T} = \begin{cases} Y_{S \triangle T} & \text{if } |S \triangle T| = p, \\ 0 & \text{otherwise.} \end{cases}$$

- This is (approximately) a submatrix of the Kikuchi Hessian
- Algorithm: compute leading eigenvalue/eigenvector of $M$
- Runtime: $n^{O(\ell)}$
The Algorithm

Definition (Symmetric Difference Matrix)

Input: an order-\(p\) tensor \(Y = (Y_U)_{|U|=p}\) (with \(p\) even) and an integer \(\ell\) in the range \(p/2 \leq \ell \leq n - p/2\). Define the \(\binom{n}{\ell} \times \binom{n}{\ell}\) matrix (indexed by \(\ell\)-subsets of \([n]\))

\[
M_{S,T} = \begin{cases} 
Y_{S \triangle T} & \text{if } |S \triangle T| = p, \\
0 & \text{otherwise}.
\end{cases}
\]

- This is (approximately) a submatrix of the Kikuchi Hessian
- Algorithm: compute leading eigenvalue/eigenvector of \(M\)
- Runtime: \(n^{O(\ell)}\)
- The case \(\ell = p/2\) is “tensor unfolding,” which is poly-time and succeeds up to the SoS threshold
The Algorithm

Definition (Symmetric Difference Matrix)

Input: an order-$p$ tensor $Y = (Y_U)_{|U|=p}$ (with $p$ even) and an integer $\ell$ in the range $p/2 \leq \ell \leq n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by $\ell$-subsets of $[n]$)

$$M_{S,T} = \begin{cases} Y_{S \triangle T} & \text{if } |S \triangle T| = p, \\
0 & \text{otherwise.} \end{cases}$$

- This is (approximately) a submatrix of the Kikuchi Hessian
- Algorithm: compute leading eigenvalue/eigenvector of $M$
- Runtime: $n^{O(\ell)}$
- The case $\ell = p/2$ is “tensor unfolding,” which is poly-time and succeeds up to the SoS threshold
- $\ell = n^\delta$ gives an algorithm of runtime $n^{O(n^\ell)} = 2^{n^{\delta + o(1)}}$
Intuition for Symmetric Difference Matrix

Recall: \( M_{S,T} = \mathbb{1}_{|S \Delta T| = p} Y_{S \Delta T} \) where \( |S| = |T| = \ell \)
Intuition for Symmetric Difference Matrix

Recall: $M_{S,T} = \mathbb{1}_{|S\triangle T|} = p \ Y_{S\triangle T}$ where $|S| = |T| = \ell$

Compute top eigenvector via power iteration: $v \leftarrow Mv$

- $v \in \mathbb{R}^{\binom{n}{\ell}}$ where $v_s$ is an estimate of $x^S := \prod_{i \in S} x_i$
Intuition for Symmetric Difference Matrix

Recall: \( M_{S,T} = \mathbb{1}_{|S \triangle T| = p} Y_{S \triangle T} \) where \( |S| = |T| = \ell \)

Compute top eigenvector via power iteration: \( v \leftarrow Mv \)

\( v \in \mathbb{R}^{\binom{a}{\ell}} \) where \( v_S \) is an estimate of \( x^S := \prod_{i \in S} x_i \)

Expand formula \( v \leftarrow Mv \):

\[
V_S \leftarrow \sum_{T:|S \triangle T| = p} Y_{S \triangle T} v_T
\]

- Recall: \( Y_{S \triangle T} \) is a noisy measurement of \( x^{S \triangle T} \)
- So \( Y_{S \triangle T} v_T \) is \( T \)'s opinion about \( x^S \)
Intuition for Symmetric Difference Matrix

Recall: \( M_{S,T} = 1_{|S \triangle T| = p} Y_{S \triangle T} \) where \(|S| = |T| = \ell\)

Compute top eigenvector via power iteration: \( \nu \leftarrow M \nu \)

\( \nu \in \mathbb{R}^{\binom{n}{\ell}} \) where \( \nu_S \) is an estimate of \( x^S := \prod_{i \in S} x_i \)

Expand formula \( \nu \leftarrow M \nu \):

\[
\nu_S \leftarrow \sum_{T : |S \triangle T| = p} Y_{S \triangle T} \nu_T
\]

Recall: \( Y_{S \triangle T} \) is a noisy measurement of \( x^{S \triangle T} \)

So \( Y_{S \triangle T} \nu_T \) is \( T \)'s opinion about \( x^S \)

This is a message-passing algorithm among sets of size \( \ell \)
Analysis

Simplest statistical task: detection

- Distinguish between $\lambda = \bar{\lambda}$ (spiked tensor) and $\lambda = 0$ (noise)
Analysis

Simplest statistical task: detection

- Distinguish between \( \lambda = \bar{\lambda} \) (spiked tensor) and \( \lambda = 0 \) (noise)

**Algorithm:** given \( Y \), build matrix \( M_{S,T} = \mathbb{1}_{|S \triangle T| = p} Y_{S \triangle T} \), threshold maximum eigenvalue
Analysis

Simplest statistical task: detection

- Distinguish between $\lambda = \bar{\lambda}$ (spiked tensor) and $\lambda = 0$ (noise)

**Algorithm:** given $Y$, build matrix $M_{S,T} = \mathbb{1}_{|S \triangle T| = p} Y_{S \triangle T}$, threshold maximum eigenvalue

**Key step:** bound spectral norm $\|M\|$ when $Y \sim$ i.i.d. $\mathcal{N}(0, 1)$
Analysis

Simplest statistical task: detection

- Distinguish between $\lambda = \bar{\lambda}$ (spiked tensor) and $\lambda = 0$ (noise)

**Algorithm**: given $Y$, build matrix $M_{S,T} = \mathbb{1}_{|S \triangle T| = p} Y_{S \triangle T}$, threshold maximum eigenvalue

**Key step**: bound spectral norm $\|M\|$ when $Y \sim \text{i.i.d. } \mathcal{N}(0, 1)$

**Theorem (Matrix Chernoff Bound [Oliveira ’10, Tropp ’10])**

Let $M = \sum_i z_i A_i$ where $z_i \sim \mathcal{N}(0, 1)$ independently and $\{A_i\}$ is a finite sequence of fixed symmetric $d \times d$ matrices. Then, for all $t \geq 0$,

$$\mathbb{P}(\|M\| \geq t) \leq 2de^{-t^2/2\sigma^2} \text{ where } \sigma^2 = \left\| \sum_i (A_i)^2 \right\|.$$

Analysis

Simplest statistical task: detection

- Distinguish between $\lambda = \bar{\lambda}$ (spiked tensor) and $\lambda = 0$ (noise)

**Algorithm:** given $Y$, build matrix $M_{S,T} = \mathbb{1}_{S \triangle T = p} Y_{S \triangle T}$, threshold maximum eigenvalue

**Key step:** bound spectral norm $\|M\|$ when $Y \sim \text{i.i.d. } \mathcal{N}(0, 1)$

**Theorem (Matrix Chernoff Bound [Oliveira '10, Tropp '10])**

Let $M = \sum_i z_i A_i$ where $z_i \sim \mathcal{N}(0, 1)$ independently and $\{A_i\}$ is a finite sequence of fixed symmetric $d \times d$ matrices. Then, for all $t \geq 0$,

$$
\mathbb{P}(\|M\| \geq t) \leq 2de^{-t^2/2\sigma^2} \quad \text{where} \quad \sigma^2 = \left\| \sum_i (A_i)^2 \right\|.
$$

In our case, $\sum_i (A_i)^2$ is a multiple of the identity
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle|$$
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle|$$

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

$$(x^\otimes \ell)^\top M(x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

Each entry of $M$ is a degree-2 $\ell/p$ polynomial in $Y$

Analysis: trace moment method (complicated)

[Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Our method: instead find $M$ (symm. diff. matrix) such that

$$(x^\otimes \ell)^\top M(x^\otimes \ell) = \langle Y, x^\otimes p \rangle \|x\|^{2\ell} - \|p\| \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

Each entry of $M$ is a degree-1 polynomial in $Y$

Analysis: matrix Chernoff bound (much simpler)
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

\[ \|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle| \]

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

\[ (x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p} \]

and so \[ \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell} \]

- Each entry of $M$ is a degree-$2\ell/p$ polynomial in $Y$

- Analysis: matrix Chernoff bound (much simpler)
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle|$$

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

$$(x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

▶ Each entry of $M$ is a degree-$2\ell/p$ polynomial in $Y$
▶ Analysis: trace moment method (complicated)

[Raghavendra-Rao-Schramm ’16, Bhattiprolu-Guruswami-Lee ’16]
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^p \rangle|$$

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

$$(x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p}$$

and so

$$\|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

- Each entry of $M$ is a degree-$2\ell/p$ polynomial in $Y$
- Analysis: trace moment method (complicated)

[Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Our method: instead find $M$ (symm. diff. matrix) such that

$$(x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle \|x\|^{2\ell-p}$$

and so

$$\|Y\|_{\text{inj}} \leq \|M\|$$
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle|$$

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

$$(x^\otimes \ell)^\top M(x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

- Each entry of $M$ is a degree-$2\ell/p$ polynomial in $Y$
- Analysis: trace moment method (complicated)
  
  [Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Our method: instead find $M$ (symm. diff. matrix) such that

$$(x^\otimes \ell)^\top M(x^\otimes \ell) = \langle Y, x^\otimes p \rangle \|x\|^{2\ell-p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|$$

- Each entry of $M$ is a degree-1 polynomial in $Y$
Comparison to Prior Work

SoS approach: given noise tensor $Y$, want to certify (prove) an upper bound on tensor injective norm

$$\|Y\|_{\text{inj}} := \max_{\|x\|=1} |\langle Y, x^\otimes p \rangle|$$

Spectral certification: find an $n^\ell \times n^\ell$ matrix $M$ such that

$$(x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle^{2\ell/p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|^{p/2\ell}$$

- Each entry of $M$ is a degree-$2\ell/p$ polynomial in $Y$
- Analysis: trace moment method (complicated)

[Raghavendra-Rao-Schramm '16, Bhattiprolu-Guruswami-Lee '16]

Our method: instead find $M$ (symm. diff. matrix) such that

$$(x^\otimes \ell)^\top M (x^\otimes \ell) = \langle Y, x^\otimes p \rangle \|x\|^{2\ell-p} \quad \text{and so} \quad \|Y\|_{\text{inj}} \leq \|M\|$$

- Each entry of $M$ is a degree-1 polynomial in $Y$
- Analysis: matrix Chernoff bound (much simpler)
Related Work

[Hastings '19, "Classical and Quantum Algorithms for Tensor PCA"]

Similar construction (symmetric difference matrix) with different motivation: quantum Hamiltonian of system of bosons

[Biroli, Cammarota, Ricci-Tersenghi '19, "How to iron out rough landscapes and get optimal performances"]

A different form of "redemption" for local algorithms

Replicated gradient descent
Related Work

- [Hastings ’19, “Classical and Quantum Algorithms for Tensor PCA”]
  - Similar construction (symmetric difference matrix) with different motivation: quantum
  - Hamiltonian of system of bosons
Related Work

- [Hastings '19, “Classical and Quantum Algorithms for Tensor PCA”]
  - Similar construction (symmetric difference matrix) with different motivation: quantum
  - Hamiltonian of system of bosons

- [Biroli, Cammarota, Ricci-Tersenghi '19, “How to iron out rough landscapes and get optimal performances”]
  - A different form of “redemption” for local algorithms
  - Replicated gradient descent
Summary

Local algorithms are suboptimal for tensor PCA

- E.g. gradient descent, AMP
- Keep track of an \( n \)-dimensional state
- Nearly-linear runtime

Why suboptimal?

- Soft threshold: optimal algorithm cannot be nearly-linear time
- For \( p \)-way data, need \( p \)-way algorithm?

"Redemption" for local algorithms and AMP

- Hierarchy of message-passing algorithms: symm. diff. matrices
- Keep track of beliefs about higher-order correlations
- Minimize Kikuchi free energy
- Matches SoS (conjectured optimal)
- Proof is much simpler than prior work

Future directions

- Unify statistical physics and SoS?
- Systematically obtain optimal spectral methods in general?

Thanks!
Summary

- Local algorithms are suboptimal for tensor PCA
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime

- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime

- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?

- “Redemption” for local algorithms and AMP
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
Local algorithms are suboptimal for tensor PCA
- E.g. gradient descent, AMP
- Keep track of an $n$-dimensional state
- Nearly-linear runtime

Why suboptimal?
- Soft threshold: optimal algorithm cannot be nearly-linear time
- For $p$-way data, need $p$-way algorithm?

“Redemption” for local algorithms and AMP
- Hierarchy of message-passing algorithms: symm. diff. matrices
- Keep track of beliefs about higher-order correlations
- Minimize Kikuchi free energy
- Matches SoS (conjectured optimal)
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
- "Redemption" for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
  - Matches SoS (conjectured optimal)
  - Proof is much simpler than prior work
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime

- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?

- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
  - Matches SoS (conjectured optimal)
  - Proof is much simpler than prior work

- Future directions
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime

- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?

- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
  - Matches SoS (conjectured optimal)
  - Proof is much simpler than prior work

- Future directions
  - Unify statistical physics and SoS?
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime
- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?
- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
  - Matches SoS (conjectured optimal)
  - Proof is much simpler than prior work
- Future directions
  - Unify statistical physics and SoS?
  - Systematically obtain optimal spectral methods in general?
Summary

- Local algorithms are suboptimal for tensor PCA
  - E.g. gradient descent, AMP
  - Keep track of an $n$-dimensional state
  - Nearly-linear runtime

- Why suboptimal?
  - Soft threshold: optimal algorithm cannot be nearly-linear time
  - For $p$-way data, need $p$-way algorithm?

- “Redemption” for local algorithms and AMP
  - Hierarchy of message-passing algorithms: symm. diff. matrices
  - Keep track of beliefs about higher-order correlations
  - Minimize Kikuchi free energy
  - Matches SoS (conjectured optimal)
  - Proof is much simpler than prior work

- Future directions
  - Unify statistical physics and SoS?
  - Systematically obtain optimal spectral methods in general?

Thanks!