Algorithmic Threshold for Multi-Species Spin Glasses

Mark Sellke

University of Waterloo Statistics and Actuarial Science Seminar

Joint work with Brice Huang (MIT)

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

- $\boldsymbol{x}^{MLE}$ **NP-hard** even to approximate in worst case (Hillar-Lim 13)

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

- $\boldsymbol{x}^{MLE}$ **NP-hard** even to approximate in worst case (Hillar-Lim 13)
- Convex relaxations suboptimal by $N^{(p-2)/4}$ factor (Montanari-Richard 14, Hopkins-Shi-Steurer 15)

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

- $\boldsymbol{x}^{MLE}$ **NP-hard** even to approximate in worst case (Hillar-Lim 13)
- Convex relaxations suboptimal by $N^{(p-2)/4}$ factor (Montanari-Richard 14, Hopkins-Shi-Steurer 15)
- **Existing frameworks leave incomplete understanding of computational limits.**

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

- $\boldsymbol{x}^{MLE}$ **NP-hard** even to approximate in worst case (Hillar-Lim 13)
- Convex relaxations suboptimal by $N^{(p-2)/4}$ factor (Montanari-Richard 14, Hopkins-Shi-Steurer 15)
- **Existing frameworks leave incomplete understanding of computational limits.**

What are the basic computational limits of random optimization problems?

Motivating Example: Tensor PCA

Fix $p \geq 2$. Recover signal $\boldsymbol{x}_0 \in S_N = \sqrt{N}\mathbb{S}^{N-1}$ from noisy tensor observation

$$\boldsymbol{T} = \lambda \boldsymbol{x}_0^{\otimes p} + \boldsymbol{G}^{(p)}, \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ has i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

- E.g. $\boldsymbol{x}_0^{\otimes 2} \in \mathbb{R}^{N \times N}$ is a matrix with (i,j) entry $x_i x_j$.
- Applications to topic modelling (Anandkumar-Ge-Hsu-Kakade-Telgarsky 12), collaborative filtering, hypergraph matching (Duchenne-Bach-Kwon-Ponce 09)
- Max-likelihood estimator is **non-convex, random** optimization problem:

$$\boldsymbol{x}^{MLE} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{T}, \boldsymbol{x}^{\otimes p} \rangle$$

- $\boldsymbol{x}^{MLE}$ **NP-hard** even to approximate in worst case (Hillar-Lim 13)
- Convex relaxations suboptimal by $N^{(p-2)/4}$ factor (Montanari-Richard 14, Hopkins-Shi-Steurer 15)
- **Existing frameworks leave incomplete understanding of computational limits.**

What are the basic computational limits of random optimization problems?

- Null model MLE is precisely optimization of a **spin glass**:

$$\boldsymbol{x}^{null} = \arg\max_{\boldsymbol{x} \in S_N} \langle \boldsymbol{G}^{(p)}, \boldsymbol{x}^{\otimes p} \rangle$$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1, i_2, i_3 = 1}^{N} g_{i_1, i_2, i_3} \cdot \sigma_{i_1} \sigma_{i_2} \sigma_{i_3} \qquad g_{i_1, i_2, i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1,i_2,i_3=1}^{N} g_{i_1,i_2,i_3} \cdot \sigma_{i_1}\sigma_{i_2}\sigma_{i_3} = \frac{1}{N}\langle \boldsymbol{G}^{(3)}, \boldsymbol{\sigma}^{\otimes 3}\rangle \qquad g_{i_1,i_2,i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1,i_2,i_3=1}^{N} g_{i_1,i_2,i_3} \cdot \sigma_{i_1}\sigma_{i_2}\sigma_{i_3} = \frac{1}{N}\langle \boldsymbol{G}^{(3)}, \boldsymbol{\sigma}^{\otimes 3}\rangle \qquad g_{i_1,i_2,i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

More generally, mix different degrees. For $\gamma_2, \gamma_3, \ldots \geq 0$,

$$H_N(\boldsymbol{\sigma}) = \sum_{p=2}^{P} \frac{\gamma_p}{N^{(p-1)/2}} \langle \boldsymbol{G}^{(p)}, \boldsymbol{\sigma}^{\otimes p}\rangle \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ i.i.d. } \mathcal{N}(0,1)\text{s}$$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1,i_2,i_3=1}^{N} g_{i_1,i_2,i_3} \cdot \sigma_{i_1}\sigma_{i_2}\sigma_{i_3} = \frac{1}{N}\langle \boldsymbol{G}^{(3)}, \boldsymbol{\sigma}^{\otimes 3}\rangle \qquad g_{i_1,i_2,i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

More generally, mix different degrees. For $\gamma_2, \gamma_3, \ldots \geq 0$,

$$H_N(\boldsymbol{\sigma}) = \sum_{p=2}^{P} \frac{\gamma_p}{N^{(p-1)/2}} \langle \boldsymbol{G}^{(p)}, \boldsymbol{\sigma}^{\otimes p}\rangle \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ i.i.d. } \mathcal{N}(0,1)\text{s}$$

Gaussian process on $\mathbb{R}^N$ with covariance

$$\mathbb{E}[H_N(\boldsymbol{\sigma})H_N(\boldsymbol{\rho})] = N\xi(\langle \boldsymbol{\sigma}, \boldsymbol{\rho}\rangle/N), \qquad \xi(q) = \sum_{p=2}^{P} \gamma_p^2 q^p$$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1,i_2,i_3=1}^{N} g_{i_1,i_2,i_3} \cdot \sigma_{i_1}\sigma_{i_2}\sigma_{i_3} = \frac{1}{N}\langle \boldsymbol{G}^{(3)}, \boldsymbol{\sigma}^{\otimes 3}\rangle \qquad g_{i_1,i_2,i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

More generally, mix different degrees. For $\gamma_2, \gamma_3, \ldots \geq 0$,

$$H_N(\boldsymbol{\sigma}) = \sum_{p=2}^{P} \frac{\gamma_p}{N^{(p-1)/2}} \langle \boldsymbol{G}^{(p)}, \boldsymbol{\sigma}^{\otimes p}\rangle \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ i.i.d. } \mathcal{N}(0,1)\text{s}$$

Gaussian process on $\mathbb{R}^N$ with covariance

$$\mathbb{E}[H_N(\boldsymbol{\sigma})H_N(\boldsymbol{\rho})] = N\xi(\langle \boldsymbol{\sigma}, \boldsymbol{\rho}\rangle/N), \qquad \xi(q) = \sum_{p=2}^{P} \gamma_p^2 q^p$$

ξ **mixture function**, determines model. Cubic above: $\xi(q) = q^3$

Mean Field Spin Glasses

Polynomials $H_N : \mathbb{R}^N \to \mathbb{R}$ with **random** coefficients, e.g. random cubic

$$H_N(\boldsymbol{\sigma}) = \frac{1}{N} \sum_{i_1,i_2,i_3=1}^{N} g_{i_1,i_2,i_3} \cdot \sigma_{i_1}\sigma_{i_2}\sigma_{i_3} = \frac{1}{N}\langle \boldsymbol{G}^{(3)}, \boldsymbol{\sigma}^{\otimes 3}\rangle \qquad g_{i_1,i_2,i_3} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

More generally, mix different degrees. For $\gamma_2, \gamma_3, \ldots \geq 0$,

$$H_N(\boldsymbol{\sigma}) = \sum_{p=2}^{P} \frac{\gamma_p}{N^{(p-1)/2}} \langle \boldsymbol{G}^{(p)}, \boldsymbol{\sigma}^{\otimes p}\rangle \qquad \boldsymbol{G}^{(p)} \in (\mathbb{R}^N)^{\otimes p} \text{ i.i.d. } \mathcal{N}(0,1)\text{s}$$

Gaussian process on $\mathbb{R}^N$ with covariance

$$\mathbb{E}[H_N(\boldsymbol{\sigma})H_N(\boldsymbol{\rho})] = N\xi(\langle \boldsymbol{\sigma}, \boldsymbol{\rho}\rangle / N), \qquad \xi(q) = \sum_{p=2}^{P} \gamma_p^2 q^p$$

ξ **mixture function**, determines model. Cubic above: $\xi(q) = q^3$

Goal: optimize H_N over sphere $S_N = \sqrt{N}\mathbb{S}^{N-1}$

Motivations and Connections

- Origin: diluted magnetic alloys (Sherrington-Kirkpatrick 75)

Motivations and Connections

- Origin: diluted magnetic alloys (Sherrington-Kirkpatrick 75)
- Natural **high-dimensional, non-convex** random optimization problem

Motivations and Connections

- Origin: diluted magnetic alloys (Sherrington-Kirkpatrick 75)
- Natural **high-dimensional, non-convex** random optimization problem
- MLE for tensor PCA log-likelihood in null model (Ben Arous-Mei-Montanari-Nica 17)

Motivations and Connections

- Origin: diluted magnetic alloys (Sherrington-Kirkpatrick 75)
- Natural **high-dimensional, non-convex** random optimization problem
- MLE for tensor PCA log-likelihood in null model (Ben Arous-Mei-Montanari-Nica 17)
- Random MaxCut and MaxSAT with many constraints (Dembo-Montanari-Sen 17, Panchenko 18)

Motivations and Connections

- Origin: diluted magnetic alloys (Sherrington-Kirkpatrick 75)
- Natural **high-dimensional, non-convex** random optimization problem
- MLE for tensor PCA log-likelihood in null model (Ben Arous-Mei-Montanari-Nica 17)
- Random MaxCut and MaxSAT with many constraints (Dembo-Montanari-Sen 17, Panchenko 18)
- Neural networks, high-dimensional statistics (Hopfield 82, Gardner-Derrida 87/88, Talagrand 00/02, Choromanska-Henaff-Mathieu-Ben Arous-LeCun 15, Ding-Sun 18, Fan-Mei-Montanari 21)

The maximum of H_N

Two basic questions for any random optimization problem:

- OPT: maximum value that **exists**?
- ALG: maximum value found by **efficient algorithm**?

The maximum of H_N

Two basic questions for any random optimization problem:

- OPT: maximum value that **exists**?
- ALG: maximum value found by **efficient algorithm**?

Theorem (Parisi 82, Talagrand 06/10, Panchenko 14, Auffinger-Chen 17)

The limiting maximum value

$$\mathrm{OPT} = \operatorname*{p-lim}_{N\to\infty} \frac{1}{N} \max_{\boldsymbol{\sigma}\in S_N} H_N(\boldsymbol{\sigma})$$

*exists and is given by the **Parisi formula*** $\mathsf{P}(\xi)$.

Efficient Optimization

- Today's goal: understand power of **efficient** algorithms $\mathcal{A}$ to optimize H_N. For $\boldsymbol{\sigma} = \mathcal{A}(H_N)$, what is max of

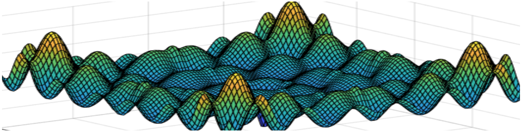
$$E = \frac{1}{N} H_N(\boldsymbol{\sigma}) \ ?$$

Efficient Optimization

- Today's goal: understand power of **efficient** algorithms $\mathcal{A}$ to optimize H_N. For $\boldsymbol{\sigma} = \mathcal{A}(H_N)$, what is max of

$$E = \frac{1}{N} H_N(\boldsymbol{\sigma}) \text{ ?}$$

- Gradient descent, convex optimization don't cut it ☹
  - Rich landscapes, e^{cN} bad local maxima well below OPT (ABAČ 13, Subag 17)

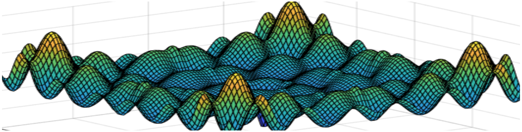


Efficient Optimization

- Today's goal: understand power of **efficient** algorithms $\mathcal{A}$ to optimize H_N. For $\boldsymbol{\sigma} = \mathcal{A}(H_N)$, what is max of

$$E = \frac{1}{N} H_N(\boldsymbol{\sigma}) \ ?$$

- Gradient descent, convex optimization don't cut it ☹
  - Rich landscapes, e^{cN} bad local maxima well below OPT (ABAČ 13, Subag 17)



- Worst-case lower bounds overly pessimistic ☹
  - Adversarial H_N: $(\log^c N)$-approximation NP-hard (ABHKS 05, BBHKSZ 12)

Efficient Optimization: Some Approaches

Can study specific algorithms like **Langevin/Glauber dynamics**

Efficient Optimization: Some Approaches

Can study specific algorithms like **Langevin/Glauber dynamics**

- Rich literature (Cugliandolo-Kurchen 92, Ben Arous-Dembo-Guionnet 01& 06, Ben Arous-Gheissari-Jagannath 20)
- Slow mixing, stuck at threshold energy on short time scales

Efficient Optimization: Some Approaches

Can study specific algorithms like **Langevin/Glauber dynamics**

- Rich literature (Cugliandolo-Kurchen 92, Ben Arous-Dembo-Guionnet 01& 06, Ben Arous-Gheissari-Jagannath 20)
- Slow mixing, stuck at threshold energy on short time scales

Can study **critical points** of H_N

Efficient Optimization: Some Approaches

Can study specific algorithms like **Langevin/Glauber dynamics**

- Rich literature (Cugliandolo-Kurchen 92, Ben Arous-Dembo-Guionnet 01& 06, Ben Arous-Gheissari-Jagannath 20)
- Slow mixing, stuck at threshold energy on short time scales

Can study **critical points** of H_N

- Pure p-spin models ($p \geq 3$): e^{cN} local maxima appear at value $E_\infty < \mathrm{OPT}$ (Auffinger-Ben Arous-Černý 13, Subag 17)
- Conjectured to obstruct e.g. gradient descent
- But no rigorous hardness implications

Informal Result

We determine sharp threshold ALG for a class of **Lipschitz** algorithms

- A Lipschitz algorithm attains ALG
- No Lipschitz algorithm surpasses ALG
- No **known** efficient algorithm surpasses ALG

Informal Result

We determine sharp threshold ALG for a class of **Lipschitz** algorithms

- A Lipschitz algorithm attains ALG
- No Lipschitz algorithm surpasses ALG
- No **known** efficient algorithm surpasses ALG

Result holds for yet more general **multi-species spin glasses**

Overlap Gap Property

💡 solution geometry **clustering** $\Rightarrow$ rigorous hardness for **stable** algorithms

Overlap Gap Property

solution geometry **clustering** $\Rightarrow$ rigorous hardness for **stable** algorithms

- Max independent set in random sparse graphs (Gamarnik-Sudan 14, Rahman-Virág 17, Gamarnik-Jagannath-Wein 20, Wein 20)
- Random (NAE-)k-SAT (Gamarnik-Sudan 17, Bresler-Huang 21)
- Hypergraph maxcut (Chen-Gamarnik-Panchenko-Rahman 19)
- Symmetric binary perceptron (Gamarnik-Kızıldağ-Perkins-Xu 22)
- Mean field spin glass (Gamarnik-Jagannath 19, Gamarnik-Jagannath-Wein 20)

Overlap Gap Property

💡 solution geometry **clustering** $\Rightarrow$ rigorous hardness for **stable** algorithms

- Max independent set in random sparse graphs (Gamarnik-Sudan 14, Rahman-Virág 17, Gamarnik-Jagannath-Wein 20, Wein 20)
- Random (NAE-)k-SAT (Gamarnik-Sudan 17, Bresler-Huang 21)
- Hypergraph maxcut (Chen-Gamarnik-Panchenko-Rahman 19)
- Symmetric binary perceptron (Gamarnik-Kızıldağ-Perkins-Xu 22)
- Mean field spin glass (Gamarnik-Jagannath 19, Gamarnik-Jagannath-Wein 20)

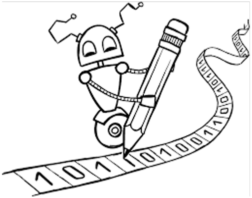
Overlap: $\langle \boldsymbol{\sigma}, \boldsymbol{\rho} \rangle / N \in [-1, 1]$

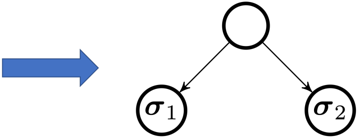
Overlap gap: no high-value $\boldsymbol{\sigma}, \boldsymbol{\rho}$ have **medium** overlap $\in [\nu_1, \nu_2]$

- Means high-value points are either close together or far apart

Classic OGP (Gamarnik-Sudan 14)

1. Stable algorithm $\mathcal{A}$ reaching $E \Rightarrow$ 2 points of value E with **medium overlap**

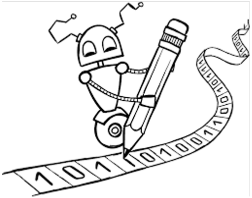


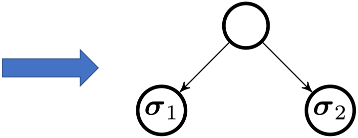


Construct by partially rerandomizing $\mathcal{A}$

Classic OGP (Gamarnik-Sudan 14)

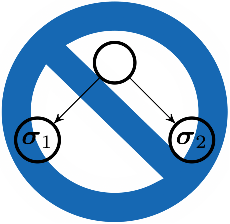
1. Stable algorithm $\mathcal{A}$ reaching $E \Rightarrow$ 2 points of value E with **medium overlap**



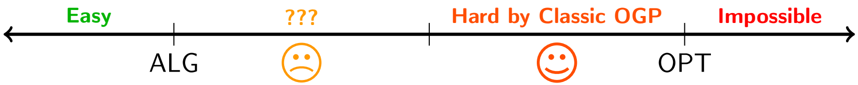


   Construct by partially rerandomizing $\mathcal{A}$

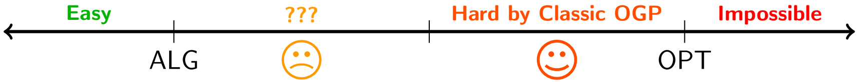
2. **Overlap gap** $\Rightarrow$ this pair does not exist. So $\mathcal{A}$ cannot reach E



Classic OGP to Multi-OGP



Classic OGP to Multi-OGP



Multi-OGP: more complex forbidden structure

Classic OGP to Multi-OGP

Multi-OGP: more complex forbidden structure

Classic OGP to Multi-OGP

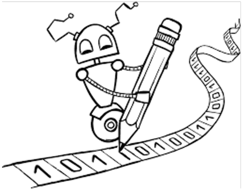
Multi-OGP: more complex forbidden structure

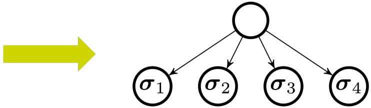
Can we push hardness all the way to ALG?

Star OGP (Rahman-Virág 17)

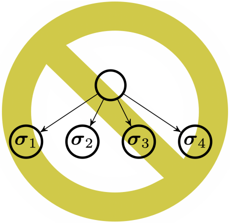
For max independent set

1. Stable algorithm $\mathcal{A}$ reaching $E \Rightarrow$ constellation of points of value E





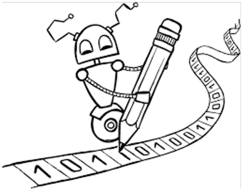
2. Such a constellation does not exist. So $\mathcal{A}$ cannot reach E

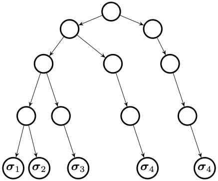


Ladder OGP (Wein 20, Bresler-Huang 21)

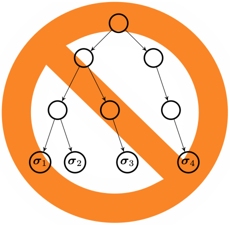
For max independent set, random k-SAT

1. Stable algorithm $\mathcal{A}$ reaching $E \Rightarrow$ constellation of points of value E





2. Such a constellation does not exist. So $\mathcal{A}$ cannot reach E



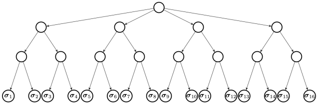
Overview of Main Result (Huang-**S** 21, 23+)

- We show that for spin glasses, **Branching OGP** gives tight hardness
  - Matches value ALG of best algorithm

Overview of Main Result (Huang-**S** 21, 23+)

- We show that for spin glasses, **Branching OGP** gives tight hardness
  - Matches value ALG of best algorithm

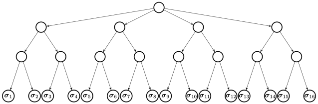
- Forbidden constellation is **densely branching ultrametric tree**
  - Inspired by ultrametricity of Gibbs measures $e^{\beta H_N(\boldsymbol{x})} d\boldsymbol{x}$ (Parisi 82, Panchenko 14, Jagannath 17, Chatterjee-Sloman 21)



Overview of Main Result (Huang-**S** 21, 23+)

- We show that for spin glasses, **Branching OGP** gives tight hardness
  - Matches value ALG of best algorithm

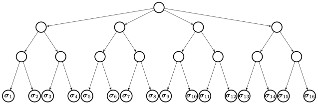
- Forbidden constellation is **densely branching ultrametric tree**
  - Inspired by ultrametricity of Gibbs measures $e^{\beta H_N(\boldsymbol{x})}d\boldsymbol{x}$ (Parisi 82, Panchenko 14, Jagannath 17, Chatterjee-Sloman 21)



- Hardness for $O(1)$-**Lipschitz** algorithms
  - View $\mathcal{A}$ as map from $(g_{1,1},\ldots,g_{N,N},g_{1,1,1},\ldots)$ to $\mathbb{R}^N$ (with L^2 distance)

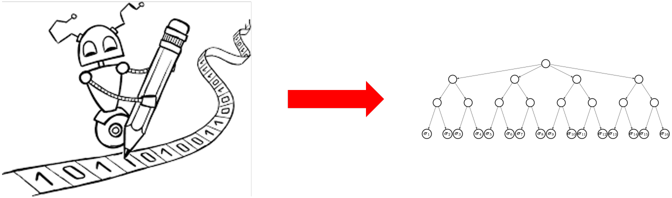
Overview of Main Result (Huang-**S** 21, 23+)

- We show that for spin glasses, **Branching OGP** gives tight hardness
  - Matches value ALG of best algorithm
- Forbidden constellation is **densely branching ultrametric tree**
  - Inspired by ultrametricity of Gibbs measures $e^{\beta H_N(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}$ (Parisi 82, Panchenko 14, Jagannath 17, Chatterjee-Sloman 21)



- Hardness for $O(1)$-**Lipschitz** algorithms
  - View $\mathcal{A}$ as map from $(g_{1,1}, \ldots, g_{N,N}, g_{1,1,1}, \ldots)$ to $\mathbb{R}^N$ (with L^2 distance)
  - Includes:
    - $O(1)$ rounds of gradient descent or any constant order method
    - Langevin dynamics for $e^{\beta H_N}$ for $O(1)$ time
    - The algorithm attaining ALG

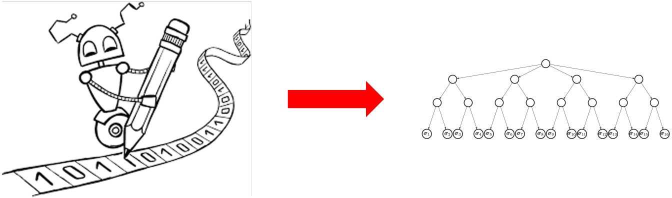
1. $O(1)$-Lipschitz algorithm $\mathcal{A}$ reaching E $\Rightarrow$ ultrametric of points of value E



Construct from correlated Hamiltonian ensemble (more later)

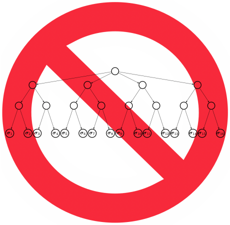
Branching OGP (Huang-**S** 21)

1. $O(1)$-Lipschitz algorithm $\mathcal{A}$ reaching $E \Rightarrow$ ultrametric of points of value E



Construct from correlated Hamiltonian ensemble (more later)

2. Constellation does not exist for $E = \mathsf{ALG} + \varepsilon$. So $\mathcal{A}$ cannot beat ALG



The Algorithmic Threshold

Theorem (Subag 18)

An efficient algorithm finds $\boldsymbol{\sigma}$ such that

$$\frac{1}{N} H_N(\boldsymbol{\sigma}) \geq \mathsf{ALG} \equiv \int_0^1 \xi''(q)^{1/2} \mathrm{d}q.$$

The Algorithmic Threshold

Theorem (Subag 18)

An efficient algorithm finds $\boldsymbol{\sigma}$ such that

$$\frac{1}{N}H_N(\boldsymbol{\sigma}) \geq \mathsf{ALG} \equiv \int_0^1 \xi''(q)^{1/2}\mathrm{d}q.$$

Theorem (Huang-S 21)

*If ξ **even**, no $O(1)$-Lipschitz algorithm beats* ALG *with probability e^{-cN}.*

Tight answer for even models, but brittle proof using **Guerra's interpolation**

The Algorithmic Threshold

Theorem (Subag 18)

An efficient algorithm finds $\boldsymbol{\sigma}$ such that

$$\frac{1}{N} H_N(\boldsymbol{\sigma}) \geq \mathsf{ALG} \equiv \int_0^1 \xi''(q)^{1/2} \mathrm{d}q.$$

Theorem (Huang-S 21)

*If ξ **even**, no $O(1)$-Lipschitz algorithm beats ALG with probability e^{-cN}.*

Tight answer for even models, but brittle proof using **Guerra's interpolation**

Theorem (Huang-S 23+)

*For **all** ξ, no $O(1)$-Lipschitz algorithm beats ALG with probability e^{-cN}.*

- New proof avoids Guerra's interpolation

The Algorithmic Threshold

Theorem (Subag 18)

An efficient algorithm finds $\boldsymbol{\sigma}$ *such that*

$$\frac{1}{N} H_N(\boldsymbol{\sigma}) \geq \mathsf{ALG} \equiv \int_0^1 \xi''(q)^{1/2} \mathrm{d}q.$$

Theorem (Huang-**S** 21)

If ξ **even***, no* $O(1)$*-Lipschitz algorithm beats* ALG *with probability* e^{-cN}*.*

Tight answer for even models, but brittle proof using **Guerra's interpolation**

Theorem (Huang-**S** 23+)

For ***all*** ξ*, no* $O(1)$*-Lipschitz algorithm beats* ALG *with probability* e^{-cN}*.*

- New proof avoids Guerra's interpolation
- Same method works for multi-species spin glasses (described later)
  - In these models, OPT not always known! (Because Guerra's interpolation fails)

Subag's Algorithm (Hessian Ascent)

For $\delta = 1/D$ constant, $\boldsymbol{x}^0 = \mathbf{0} \in \mathbb{R}^N$:

Subag's Algorithm (Hessian Ascent)

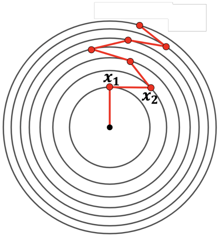
For $\delta = 1/D$ constant, $\boldsymbol{x}^0 = \mathbf{0} \in \mathbb{R}^N$:

1. Take $\boldsymbol{v}^t$ the top eigenvector of tangential Hessian $\nabla^2 H_N(\boldsymbol{x}^t)|_{(\boldsymbol{x}^t)^\perp}$

Subag's Algorithm (Hessian Ascent)

For $\delta = 1/D$ constant, $\boldsymbol{x}^0 = \mathbf{0} \in \mathbb{R}^N$:

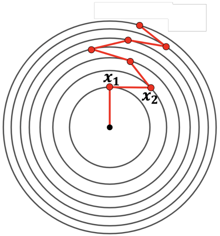
1. Take $\boldsymbol{v}^t$ the top eigenvector of tangential Hessian $\nabla^2 H_N(\boldsymbol{x}^t)|_{(\boldsymbol{x}^t)^\perp}$
2. Explore with small orthogonal steps: $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t \pm \sqrt{\delta N}\boldsymbol{v}^t$.
   (Since $\boldsymbol{v}^t \perp \boldsymbol{x}^t$, we have $\|\boldsymbol{x}^t\|_2^2 = t\delta N$)



Subag's Algorithm (Hessian Ascent)

For $\delta = 1/D$ constant, $\boldsymbol{x}^0 = \mathbf{0} \in \mathbb{R}^N$:

1. Take $\boldsymbol{v}^t$ the top eigenvector of tangential Hessian $\nabla^2 H_N(\boldsymbol{x}^t)|_{(\boldsymbol{x}^t)^\perp}$
2. Explore with small orthogonal steps: $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t \pm \sqrt{\delta N}\boldsymbol{v}^t$.
   (Since $\boldsymbol{v}^t \perp \boldsymbol{x}^t$, we have $\|\boldsymbol{x}^t\|_2^2 = t\delta N$)

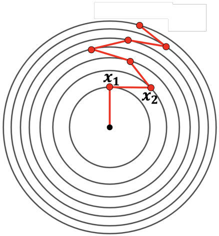


3. Output $\boldsymbol{\sigma} = \boldsymbol{x}^D \in S_N$

Subag's Algorithm (Hessian Ascent)

For $\delta = 1/D$ constant, $\boldsymbol{x}^0 = \boldsymbol{0} \in \mathbb{R}^N$:

1. Take $\boldsymbol{v}^t$ the top eigenvector of tangential Hessian $\nabla^2 H_N(\boldsymbol{x}^t)|_{(\boldsymbol{x}^t)^\perp}$
2. Explore with small orthogonal steps: $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t \pm \sqrt{\delta N}\boldsymbol{v}^t$.
   (Since $\boldsymbol{v}^t \perp \boldsymbol{x}^t$, we have $\|\boldsymbol{x}^t\|_2^2 = t\delta N$)



3. Output $\boldsymbol{\sigma} = \boldsymbol{x}^D \in S_N$

Can be implemented as $O(1)$-Lipschitz algorithm (El Alaoui-Montanari-Sellke 20)

Analysis of Subag's Algorithm

- If $\|\boldsymbol{x}\|_2 = \sqrt{qN}$, tangential Hessian $\nabla^2 H_N(\boldsymbol{x})_{\boldsymbol{x}^\perp}$ has law $\xi''(q)^{1/2} \times GOE_{N-1}$

Analysis of Subag's Algorithm

- If $\|\boldsymbol{x}\|_2 = \sqrt{qN}$, tangential Hessian $\nabla^2 H_N(\boldsymbol{x})_{\boldsymbol{x}^\perp}$ has law $\xi''(q)^{1/2} \times GOE_{N-1}$
- $\lambda_{\max}(GOE) \approx 2$, so step t gains

$$\frac{H_N(\boldsymbol{x}^{t+1}) - H_N(\boldsymbol{x}^t)}{N} \approx \delta \xi''(t\delta)^{1/2}$$

Analysis of Subag's Algorithm

- If $\|\boldsymbol{x}\|_2 = \sqrt{qN}$, tangential Hessian $\nabla^2 H_N(\boldsymbol{x})_{\boldsymbol{x}^\perp}$ has law $\xi''(q)^{1/2} \times GOE_{N-1}$
- $\lambda_{\max}(GOE) \approx 2$, so step t gains

$$\frac{H_N(\boldsymbol{x}^{t+1}) - H_N(\boldsymbol{x}^t)}{N} \approx \delta \xi''(t\delta)^{1/2}$$

- Summing over $t = 1, \dots, D$ and taking $\delta \to 0$,

$$\frac{1}{N} H_N(\boldsymbol{x}^D) \approx \int_0^1 \xi''(q)^{1/2} \mathrm{d}q = \mathrm{ALG}$$

Analysis of Subag's Algorithm

- If $\|\boldsymbol{x}\|_2 = \sqrt{qN}$, tangential Hessian $\nabla^2 H_N(\boldsymbol{x})_{\boldsymbol{x}^\perp}$ has law $\xi''(q)^{1/2} \times GOE_{N-1}$
- $\lambda_{\max}(GOE) \approx 2$, so step t gains

$$\frac{H_N(\boldsymbol{x}^{t+1}) - H_N(\boldsymbol{x}^t)}{N} \approx \delta\xi''(t\delta)^{1/2}$$

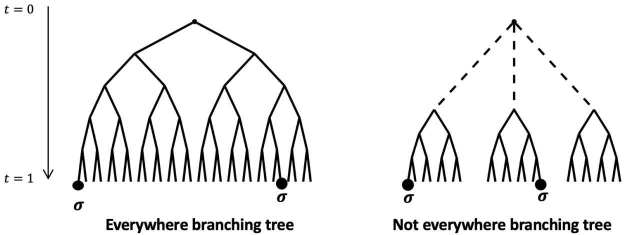
- Summing over $t = 1, \ldots, D$ and taking $\delta \to 0$,

$$\frac{1}{N} H_N(\boldsymbol{x}^D) \approx \int_0^1 \xi''(q)^{1/2} \mathrm{d}q = \mathrm{ALG}$$

- Although $\boldsymbol{x}^t$ depends on H_N, ok by **uniform** lower bound on $\lambda_{\max}(H_N(\boldsymbol{x})_{\boldsymbol{x}^\perp})$ for all $\|\boldsymbol{x}\|_2 = \sqrt{qN}$

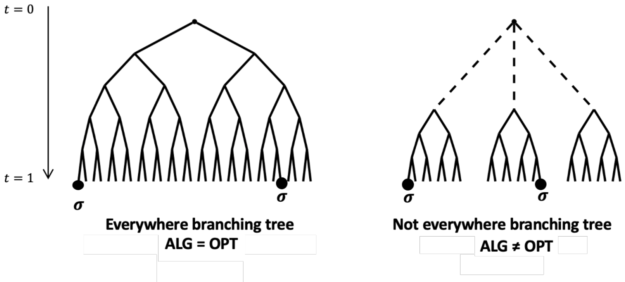
Connection to Physics Theory

- Approximate maxima of H_N are **ultrametric**, i.e. isometric to a tree



Connection to Physics Theory

- Approximate maxima of H_N are **ultrametric**, i.e. isometric to a tree

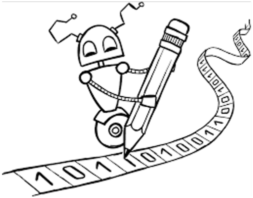


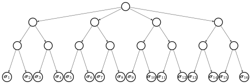
Subag's algorithm attains OPT iff branching occurs at all depths

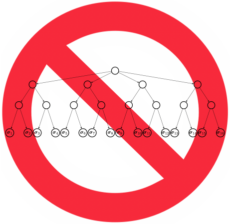
- Intuition: algorithm traces root-to-leaf path of tree

Branching OGP

Subag's algorithm reaches ALG. We next see how to show hardness beyond ALG

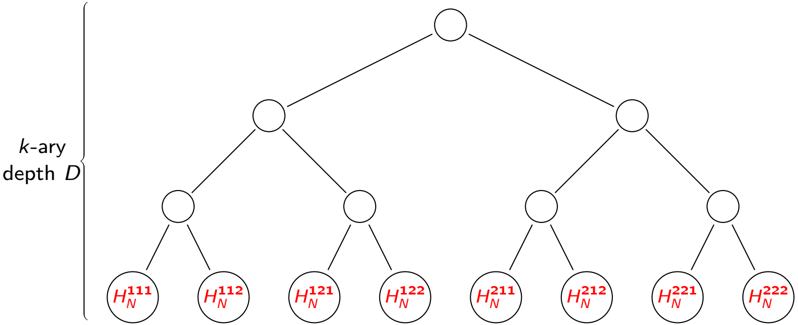






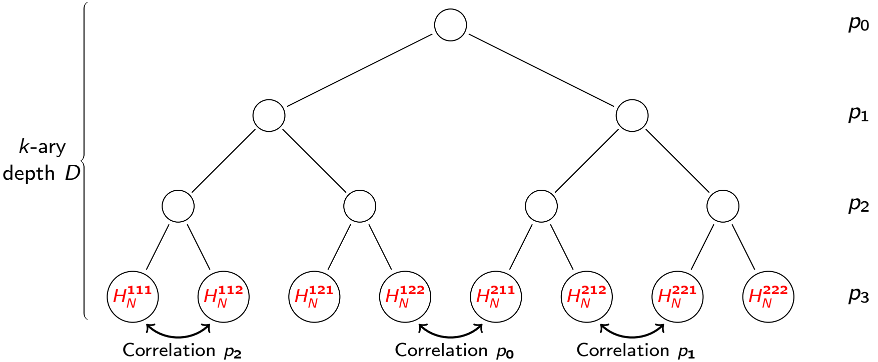
Hierarchically Correlated Hamiltonians

Generate tree of Hamiltonians $(H_N^u)_{u\in[k]^D}$



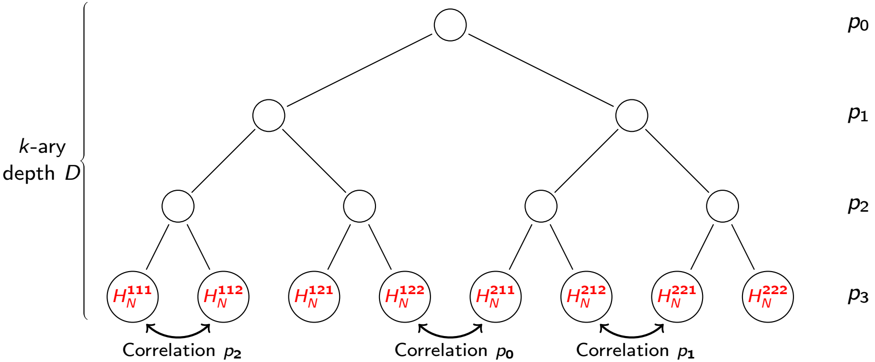
Hierarchically Correlated Hamiltonians

Generate tree of Hamiltonians $(H_N^u)_{u \in [k]^D}$



Hierarchically Correlated Hamiltonians

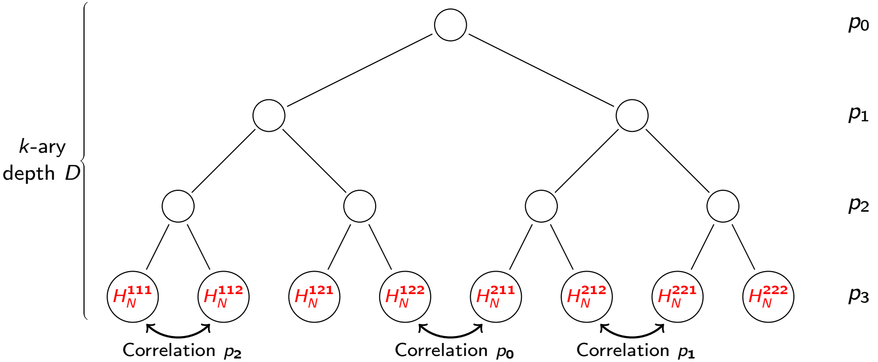
Generate tree of Hamiltonians $(H_N^u)_{u \in [k]^D}$



$$k, D \in \mathbb{N} \text{ large, } 0 \le p_0 < p_1 < \cdots < p_D = 1$$

Hierarchically Correlated Hamiltonians

Generate tree of Hamiltonians $(H_N^u)_{u\in[k]^D}$

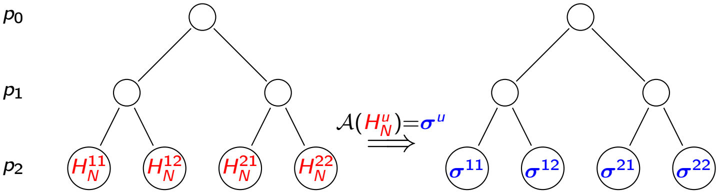


$$k, D \in \mathbb{N} \text{ large, } 0 \le p_0 < p_1 < \cdots < p_D = 1$$

Vocab: "$(H_N^u)_{u\in[k]^D}$ has correlation $\vec{p} = (p_0, \dots, p_D)$"

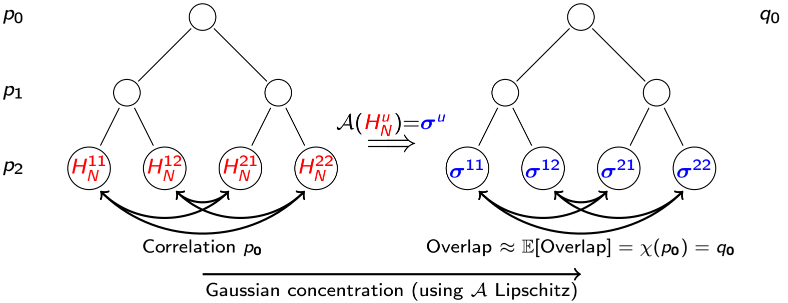
Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



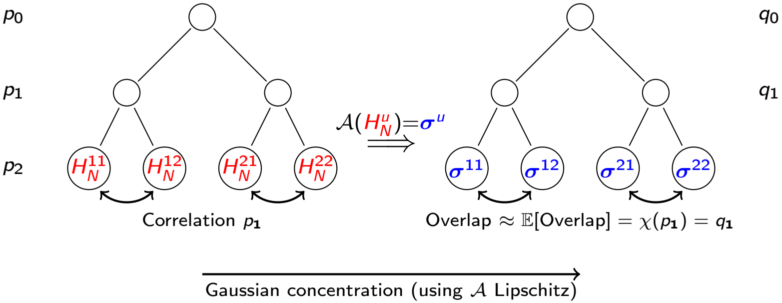
Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



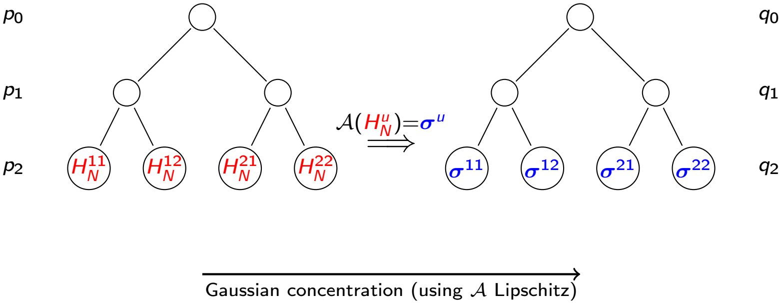
Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



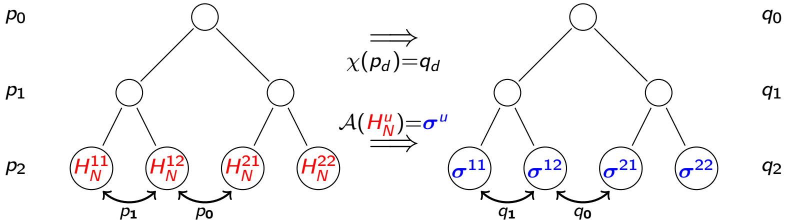
Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz

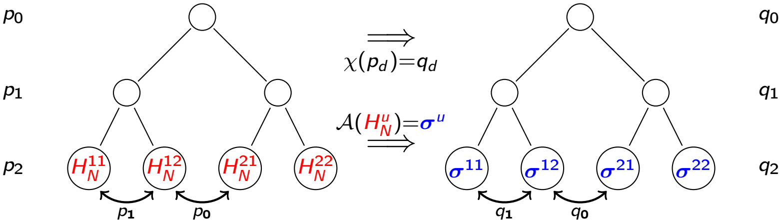


$\longrightarrow$
Gaussian concentration (using $\mathcal{A}$ Lipschitz)

$(\boldsymbol{\sigma}^u)_{u\in[k]^D}$ is approximately ultrametric

Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



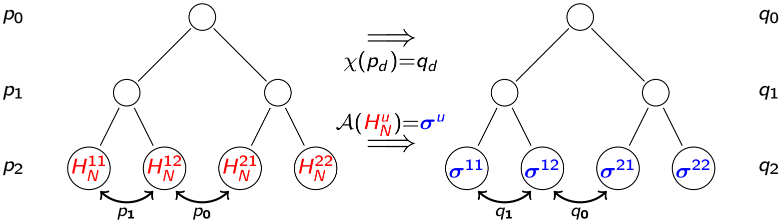
$\xrightarrow{\hspace{8cm}}$
Gaussian concentration (using $\mathcal{A}$ Lipschitz)

$(\boldsymbol{\sigma}^u)_{u\in[k]^D}$ is approximately ultrametric

Vocab: "$(\boldsymbol{\sigma}^u)_{u\in[k]^D}$ has geometry $\vec{q} = (q_0, \ldots, q_D)$"

Lipschitz Algorithms to Ultrametric Trees

Let $\mathcal{A}$ be $O(1)$-Lipschitz



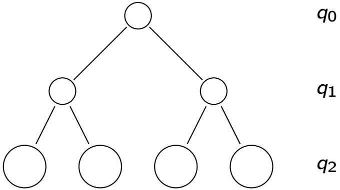
$\xrightarrow{\hspace{6cm}}$
Gaussian concentration (using $\mathcal{A}$ Lipschitz)

$(\boldsymbol{\sigma}^u)_{u\in[k]^D}$ is approximately ultrametric

Vocab: "$(\boldsymbol{\sigma}^u)_{u\in[k]^D}$ has geometry $\vec{q} = (q_0, \dots, q_D)$"

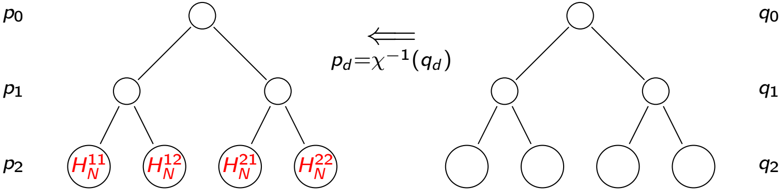
χ continuous. Can choose $\vec{p}$ to achieve **any** $0 \le q_0 < \cdots < q_D = 1$

Lipschitz Algorithms to Ultrametric Trees



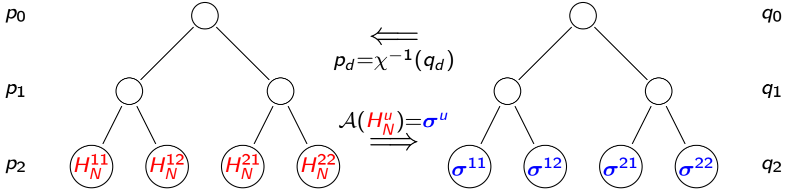
- Suppose Lipschitz $\mathcal{A}$ reaches E. Then, for any target $\vec{q}$,

Lipschitz Algorithms to Ultrametric Trees



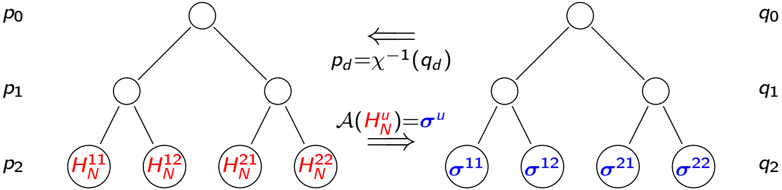
- Suppose Lipschitz $\mathcal{A}$ reaches E. Then, for any target $\vec{q}$,
- Exists $\vec{p}$

Lipschitz Algorithms to Ultrametric Trees



- Suppose Lipschitz $\mathcal{A}$ reaches E. Then, for any target $\vec{q}$,
- Exists $\vec{p}$ and $(\boldsymbol{\sigma}^u)_{u \in [k]^D}$ with geometry $\vec{q}$, so that

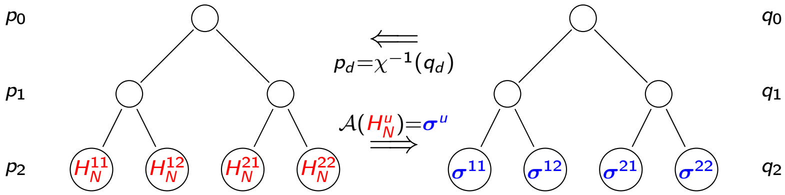
Lipschitz Algorithms to Ultrametric Trees



- Suppose Lipschitz $\mathcal{A}$ reaches E. Then, for any target $\vec{q}$,
- Exists $\vec{p}$ and $(\boldsymbol{\sigma}^u)_{u \in [k]^D}$ with geometry $\vec{q}$, so that

$$\frac{1}{N} H_N^u(\boldsymbol{\sigma}^u) \geq E \qquad \text{for all } u \in [k]^D$$

Lipschitz Algorithms to Ultrametric Trees

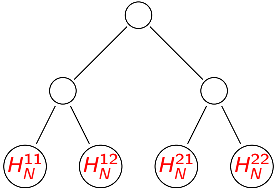


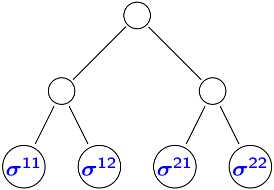
- Suppose Lipschitz $\mathcal{A}$ reaches E. Then, for any target $\vec{q}$,
- Exists $\vec{p}$ and $(\boldsymbol{\sigma}^u)_{u \in [k]^D}$ with geometry $\vec{q}$, so that

$$\frac{1}{N} H_N^u(\boldsymbol{\sigma}^u) \geq E \qquad \text{for all } u \in [k]^D$$

- *For* ***some*** $\vec{p}$, *there is a tree constellation with value* E *and geometry* $\vec{q}$

The value BOGP



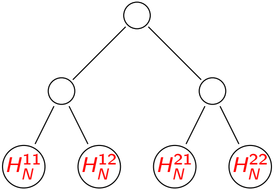


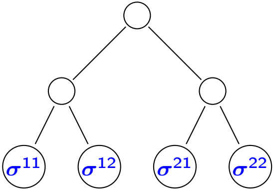
Correlations $\vec{p} = (p_0, \ldots, p_D)$

Geometry $\vec{q} = (q_0, \ldots, q_D) = (0, \delta, \ldots, 1)$

$\delta = 1/D$

The value BOGP





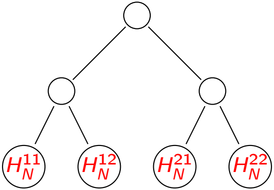
Correlations $\vec{p} = (p_0, \dots, p_D)$

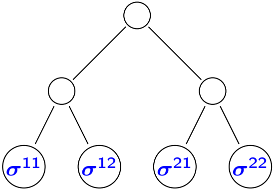
Geometry $\vec{q} = (q_0, \dots, q_D) = (0, \delta, \dots, 1)$

$\delta = 1/D$

$$\text{TreeValue}(\vec{p}) = \operatorname*{p-lim}_{N \to \infty} \max_{\substack{(\boldsymbol{\sigma}^u)_{u \in [k]^D} \\ \text{geometry } \vec{q}}} \frac{1}{k^D} \sum_{u \in [k]^D} \frac{1}{N} H_N^u(\boldsymbol{\sigma}^u)$$

The value BOGP





Correlations $\vec{p} = (p_0, \ldots, p_D)$

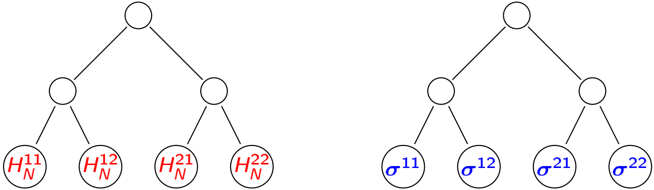
Geometry $\vec{q} = (q_0, \ldots, q_D) = (0, \delta, \ldots, 1)$

$\delta = 1/D$

$$\mathrm{TreeValue}(\vec{p}) = \operatorname*{p-lim}_{N\to\infty} \max_{\substack{(\boldsymbol{\sigma}^u)_{u\in[k]^D} \\ \text{geometry } \vec{q}}} \frac{1}{k^D} \sum_{u\in[k]^D} \frac{1}{N} H_N^u(\boldsymbol{\sigma}^u)$$

$$\mathrm{BOGP} = \max_{\vec{p}} \mathrm{TreeValue}(\vec{p})$$

The value BOGP



Correlations $\vec{p} = (p_0, \dots, p_D)$ Geometry $\vec{q} = (q_0, \dots, q_D) = (0, \delta, \dots, 1)$

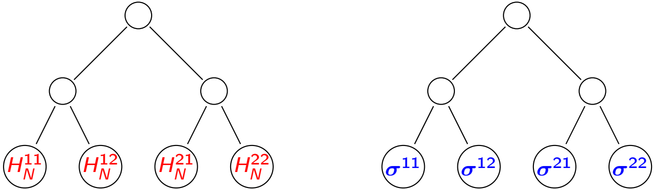
$\delta = 1/D$

$$\mathrm{TreeValue}(\vec{p}) = \operatorname*{p-lim}_{N \to \infty} \max_{\substack{(\boldsymbol{\sigma}^u)_{u \in [k]^D} \\ \text{geometry } \vec{q}}} \frac{1}{k^D} \sum_{u \in [k]^D} \frac{1}{N} H_N^u(\boldsymbol{\sigma}^u)$$

$$\mathrm{BOGP} = \max_{\vec{p}} \mathrm{TreeValue}(\vec{p})$$

- *For **any** $\vec{p}$, there is **no** tree constellation with value* $\mathrm{BOGP} + \varepsilon$ *and geometry* $\vec{q}$

The value BOGP



Correlations $\vec{p} = (p_0, \dots, p_D)$ Geometry $\vec{q} = (q_0, \dots, q_D) = (0, \delta, \dots, 1)$

$\delta = 1/D$

$$\text{TreeValue}(\vec{p}) = \operatorname*{p-lim}_{N\to\infty} \max_{\substack{(\boldsymbol{\sigma}^u)_{u\in[k]^D} \\ \text{geometry } \vec{q}}} \frac{1}{k^D} \sum_{u\in[k]^D} \frac{1}{N} H_N^u(\boldsymbol{\sigma}^u)$$

$$\text{BOGP} = \max_{\vec{p}} \text{TreeValue}(\vec{p})$$

- *For **any** $\vec{p}$, there is **no** tree constellation with value* $\text{BOGP} + \varepsilon$ *and geometry* $\vec{q}$
- $\Rightarrow$ No $O(1)$-Lipschitz algorithm attains $\text{BOGP} + \varepsilon$

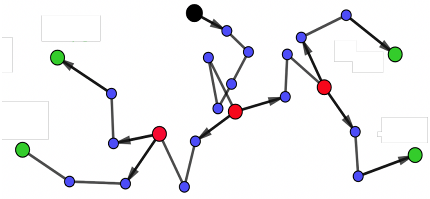
New Proof Idea: Greedy is Best

Remains to upper bound BOGP (by ALG)

New Proof Idea: Greedy is Best

Remains to upper bound BOGP (by ALG)

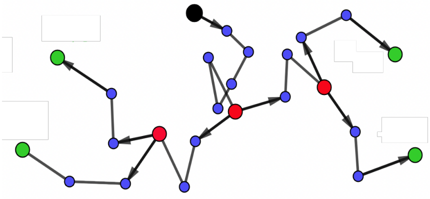
- Can **branch** Subag's algorithm by taking top k eigenvectors
- This is a multi-valued algorithm. All outputs $\approx$ ALG by same analysis



New Proof Idea: Greedy is Best

Remains to upper bound BOGP (by ALG)

- Can **branch** Subag's algorithm by taking top k eigenvectors
- This is a multi-valued algorithm. All outputs $\approx$ ALG by same analysis

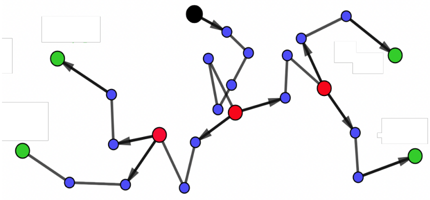


- This tree is built in a greedy way

New Proof Idea: Greedy is Best

Remains to upper bound BOGP (by ALG)

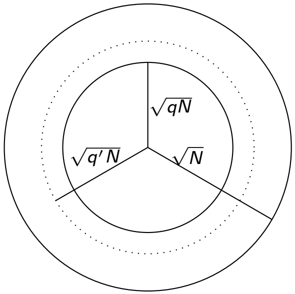
- Can **branch** Subag's algorithm by taking top k eigenvectors
- This is a multi-valued algorithm. All outputs $\approx$ ALG by same analysis



- This tree is built in a greedy way
- Main claim: best way to construct tree is greedy
    - "Can't plan ahead so that my gain at 20th level is unusually big"
    - Proved by **uniform concentration**

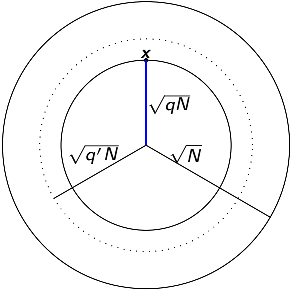
Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \dots, \boldsymbol{x}^k$:



Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k$:

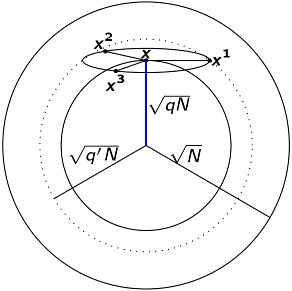


Radius:

$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$

Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k$:

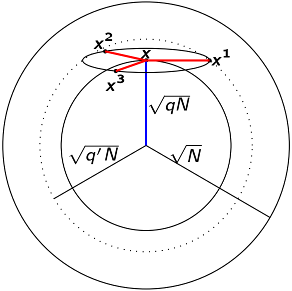


Radius:

$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$
$$\|\boldsymbol{x}^i\|_2 = \sqrt{q'N}$$

Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \dots, \boldsymbol{x}^k$:



Radius:

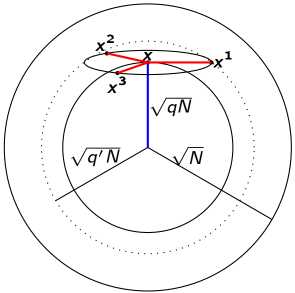
$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$
$$\|\boldsymbol{x}^i\|_2 = \sqrt{q'N}$$

Increment orthogonality:

$$\boldsymbol{x}^i - \boldsymbol{x} \perp \boldsymbol{x}^j - \boldsymbol{x} \perp \boldsymbol{x}$$

Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \dots, \boldsymbol{x}^k$:



$$F(\boldsymbol{x}) = \max_{\boldsymbol{x}^1,\dots,\boldsymbol{x}^k} \frac{1}{kN} \sum_{i=1}^{k} (H_N(\boldsymbol{x}^i) - H_N(\boldsymbol{x}))$$

“Improvement in H_N from $\boldsymbol{x}$ to its children"

Radius:

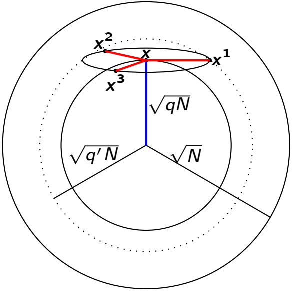
$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$
$$\|\boldsymbol{x}^i\|_2 = \sqrt{q'N}$$

Increment orthogonality:

$$\boldsymbol{x}^i - \boldsymbol{x} \perp \boldsymbol{x}^j - \boldsymbol{x} \perp \boldsymbol{x}$$

Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k$:



Radius:

$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$
$$\|\boldsymbol{x}^i\|_2 = \sqrt{q'N}$$

Increment orthogonality:

$$\boldsymbol{x}^i - \boldsymbol{x} \perp \boldsymbol{x}^j - \boldsymbol{x} \perp \boldsymbol{x}$$

$$F(\boldsymbol{x}) = \max_{\boldsymbol{x}^1,\ldots,\boldsymbol{x}^k} \frac{1}{kN} \sum_{i=1}^{k} (H_N(\boldsymbol{x}^i) - H_N(\boldsymbol{x}))$$

"Improvement in H_N from $\boldsymbol{x}$ to its children"

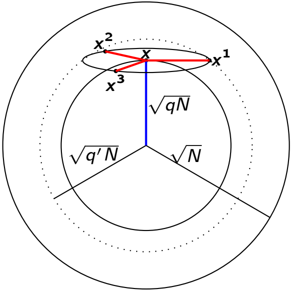
Lemma (Uniform Concentration, cf. Subag 18)

For any $\eta > 0$, for sufficiently large $k \geq k_0(\eta)$,

$$\mathbb{P}\left[|F(\boldsymbol{x}) - \mathbb{E}\, F(\boldsymbol{x})| \leq \eta \ \forall \|\boldsymbol{x}\|_2 = \sqrt{qN}\right] \geq 1 - e^{-cN}$$

Uniform Concentration

Configuration $\boldsymbol{x}, \boldsymbol{x}^1, \dots, \boldsymbol{x}^k$:



$$F(\boldsymbol{x}) = \max_{\boldsymbol{x}^1,\dots,\boldsymbol{x}^k} \frac{1}{kN} \sum_{i=1}^{k} (H_N(\boldsymbol{x}^i) - H_N(\boldsymbol{x}))$$

"Improvement in H_N from $\boldsymbol{x}$ to its children"

Radius:

$$\|\boldsymbol{x}\|_2 = \sqrt{qN}$$

$$\|\boldsymbol{x}^i\|_2 = \sqrt{q'N}$$

Increment orthogonality:

$$\boldsymbol{x}^i - \boldsymbol{x} \perp \boldsymbol{x}^j - \boldsymbol{x} \perp \boldsymbol{x}$$

Lemma (Uniform Concentration, cf. Subag 18)

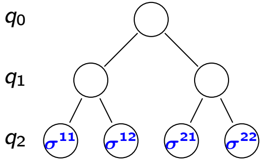
For any $\eta > 0$, for sufficiently large $k \geq k_0(\eta)$,

$$\mathbb{P}\left[|F(\boldsymbol{x}) - \mathbb{E}\, F(\boldsymbol{x})| \leq \eta \ \forall \|\boldsymbol{x}\|_2 = \sqrt{qN} \right] \geq 1 - e^{-cN}$$

No $\|\boldsymbol{x}\|_2 = \sqrt{qN}$ is unusually good for building a tree, so might as well be greedy.

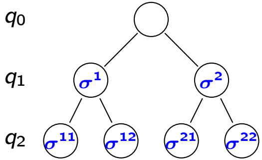
Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



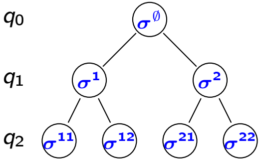
Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



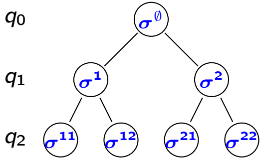
Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



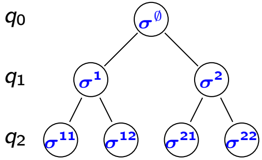
Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters: $\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^k \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

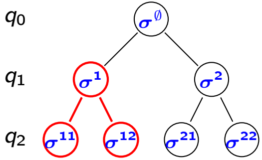
$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. ($\vec{p} = \vec{1}$)
Want to upper bound tree value:

$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. $(\vec{p} = \vec{1})$
Want to upper bound tree value:

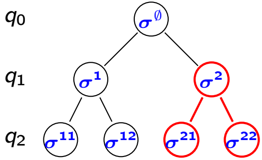
$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^{k}\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big)$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^k \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. ($\vec{p} = \vec{1}$)
Want to upper bound tree value:

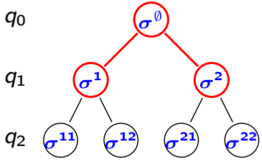
$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^{k}\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big)$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. $(\vec{p} = \vec{1})$
Want to upper bound tree value:

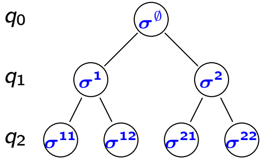
$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^{k}\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big)$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^k \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. $(\vec{p} = \vec{1})$
Want to upper bound tree value:

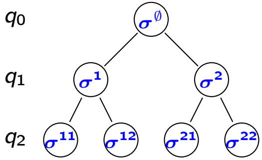
$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^k\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big) \le F(\boldsymbol{\sigma}^u)$$

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^k \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. ($\vec{p} = \vec{1}$)
Want to upper bound tree value:

$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

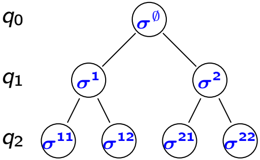
Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^k\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big) \le F(\boldsymbol{\sigma}^u)$$

$F(\boldsymbol{\sigma}^u) \approx \mathbb{E}F(\boldsymbol{\sigma}^u)$ by uniform concentration!

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. ($\vec{p} = \vec{1}$)
Want to upper bound tree value:

$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

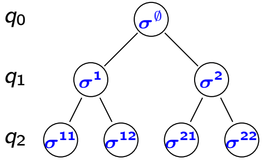
$$\frac{1}{kN}\sum_{i=1}^{k}\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big) \le F(\boldsymbol{\sigma}^u)$$

$F(\boldsymbol{\sigma}^u) \approx \mathbb{E}F(\boldsymbol{\sigma}^u)$ by uniform concentration!

Level-d increments match Subag's algorithm

Upper Bounding the Tree Value

Let interior $\boldsymbol{\sigma}^u$ be recursive barycenters:
$\boldsymbol{\sigma}^u = \frac{1}{k}\sum_{i=1}^k \boldsymbol{\sigma}^{ui}$



Satisfy orthogonality relations approximately if k large:

$$\|\boldsymbol{\sigma}^u\|_2 \approx \sqrt{q_{|u|}N}$$

$$\boldsymbol{\sigma}^{ui} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^{uj} - \boldsymbol{\sigma}^u \perp \boldsymbol{\sigma}^u$$

Suppose first all H_N^u identical. $(\vec{p} = \vec{1})$
Want to upper bound tree value:

$$\frac{1}{k^D}\sum_{u\in[k]^D}\frac{1}{N}H_N(\boldsymbol{\sigma}^u)$$

Write as sum of **claw increments**

$$\frac{1}{kN}\sum_{i=1}^k\Big(H_N(\boldsymbol{\sigma}^{ui}) - H_N(\boldsymbol{\sigma}^u)\Big) \le F(\boldsymbol{\sigma}^u)$$

$F(\boldsymbol{\sigma}^u) \approx \mathbb{E}F(\boldsymbol{\sigma}^u)$ by uniform concentration!

Level-d increments match Subag's algorithm

General $\vec{p}$: similarly bound

$$\frac{1}{kN}\sum_{i=1}^k(H_N^{ui}(\boldsymbol{\sigma}^{ui}) - H_N^u(\boldsymbol{\sigma}^u))$$

Branching OGP is Necessary for Tight Hardness

Branching OGP is Necessary for Tight Hardness

Theorem (Huang-S 21)

If an ultrametric constellation is forbidden at value $\mathrm{ALG} + \varepsilon$*, it must contain a complete binary subtree of diverging depth as* $\varepsilon \to 0$*.*

Multi-Species Spin Glasses

- Up to now: polynomials in variables $x_1, \ldots, x_N$ that **all look alike**

Multi-Species Spin Glasses

- Up to now: polynomials in variables $x_1, \ldots, x_N$ that **all look alike**
- Multi-species models: multiple "variable types" $x_i, y_i, z_i, \ldots$
  - Coefficients of $x_i x_j$, $x_i y_j$, $x_i y_j z_k$ have different variances

Multi-Species Spin Glasses

- Up to now: polynomials in variables $x_1, \ldots, x_N$ that **all look alike**
- Multi-species models: multiple "variable types" $x_i, y_i, z_i, \ldots$
  - Coefficients of $x_i x_j$, $x_i y_j$, $x_i y_j z_k$ have different variances

- Example: **bipartite SK model**

$$H_N(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\sqrt{N}} \langle \boldsymbol{G}\boldsymbol{x}, \boldsymbol{y} \rangle, \qquad \boldsymbol{G} \in \mathbb{R}^{N \times N} \text{ i.i.d. } \mathcal{N}(0,1) \text{ entries}$$

or higher-order polynomials

$$H_N(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N} \langle \boldsymbol{G}, \boldsymbol{x}^{\otimes 3} \rangle + \frac{1}{N} \langle \boldsymbol{G}', \boldsymbol{x} \otimes \boldsymbol{y}^{\otimes 2} \rangle, \qquad \boldsymbol{G}, \boldsymbol{G}' \in (\mathbb{R}^N)^{\otimes 3}$$

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \ldots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \ldots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

- Interaction weights $\gamma_2, \gamma_3, \ldots$ now $(\gamma_{s_1,s_2})_{s_1,s_2 \in \mathscr{S}}, (\gamma_{s_1,s_2,s_3})_{s_1,s_2,s_3 \in \mathscr{S}}, \ldots$

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \ldots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

- Interaction weights $\gamma_2, \gamma_3, \ldots$ now $(\gamma_{s_1,s_2})_{s_1,s_2 \in \mathscr{S}}, (\gamma_{s_1,s_2,s_3})_{s_1,s_2,s_3 \in \mathscr{S}}, \ldots$
- ξ now **multivariate** polynomial in $(q_1, \ldots, q_r)$

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \dots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

- Interaction weights $\gamma_2, \gamma_3, \dots$ now $(\gamma_{s_1,s_2})_{s_1,s_2 \in \mathscr{S}}, (\gamma_{s_1,s_2,s_3})_{s_1,s_2,s_3 \in \mathscr{S}}, \dots$
- ξ now **multivariate** polynomial in $(q_1, \dots, q_r)$

- Goal: optimize H_N over **product of spheres**

$$\mathbb{T}_N = \left\{ \boldsymbol{\sigma} \in \mathbb{R}^N : \|\boldsymbol{\sigma}_{|\mathcal{I}_s}\|_2^2 = \lambda_s N \quad \forall s \in \mathscr{S} \right\}$$

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \dots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

- Interaction weights $\gamma_2, \gamma_3, \dots$ now $(\gamma_{s_1,s_2})_{s_1,s_2 \in \mathscr{S}}, (\gamma_{s_1,s_2,s_3})_{s_1,s_2,s_3 \in \mathscr{S}}, \dots$
- ξ now **multivariate** polynomial in $(q_1, \dots, q_r)$
- Goal: optimize H_N over **product of spheres**

$$\mathbb{T}_N = \left\{ \boldsymbol{\sigma} \in \mathbb{R}^N : \|\boldsymbol{\sigma}_{|\mathcal{I}_s}\|_2^2 = \lambda_s N \quad \forall s \in \mathscr{S} \right\}$$

- OPT known for **convex** or **pure** ξ (Panchenko 15, Subag 21, Bates-Sohn 22)

Multi-Species Spin Glasses

- Formally, each coordinate part of a **species** $s \in \mathscr{S} = \{1, \ldots, r\}$

$$[N] = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_r, \qquad |\mathcal{I}_s| = \lambda_s N$$

- Interaction weights $\gamma_2, \gamma_3, \ldots$ now $(\gamma_{s_1,s_2})_{s_1,s_2 \in \mathscr{S}}, (\gamma_{s_1,s_2,s_3})_{s_1,s_2,s_3 \in \mathscr{S}}, \ldots$
- ξ now **multivariate** polynomial in $(q_1, \ldots, q_r)$
- Goal: optimize H_N over **product of spheres**

$$\mathbb{T}_N = \left\{ \boldsymbol{\sigma} \in \mathbb{R}^N : \|\boldsymbol{\sigma}_{|\mathcal{I}_s}\|_2^2 = \lambda_s N \quad \forall s \in \mathscr{S} \right\}$$

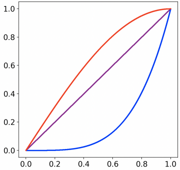
- OPT known for **convex** or **pure** ξ (Panchenko 15, Subag 21, Bates-Sohn 22)
- ALG has richer behavior than in one species

Multi-Species Algorithms

- Optimizing on product of spheres $\Rightarrow$ track radius for each species

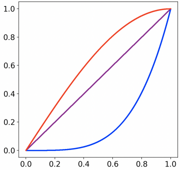
Multi-Species Algorithms

- Optimizing on product of spheres $\Rightarrow$ track radius for each species
  - 2 species: **radius schedule** is up-right path from $(0,0)$ to $(1,1)$



Multi-Species Algorithms

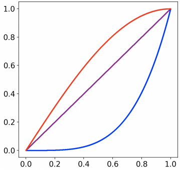
- Optimizing on product of spheres $\Rightarrow$ track radius for each species
  - 2 species: **radius schedule** is up-right path from $(0,0)$ to $(1,1)$



- In general, radius schedule is coordinate-increasing $\Phi : [0,1] \to [0,1]^{\mathscr{S}}$

Multi-Species Algorithms

- Optimizing on product of spheres $\Rightarrow$ track radius for each species
  - 2 species: **radius schedule** is up-right path from $(0,0)$ to $(1,1)$



- In general, radius schedule is coordinate-increasing $\Phi : [0,1] \to [0,1]^{\mathscr{S}}$

- Each Φ gives algorithm taking small orthogonal steps **in each species**
- Algorithm value

$$\mathbb{A}(\Phi) \equiv \sum_{s \in \mathscr{S}} \lambda_s \int_0^1 \sqrt{(\partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\, \mathrm{d}q$$

Multi-Species Algorithmic Threshold

Theorem (Huang-**S** 23+)

Define

$$\mathrm{ALG} = \sup_{\substack{\Phi:[0,1]\to[0,1]^{\mathscr{S}} \\ \textit{increasing, differentiable}}} \sum_{s\in\mathscr{S}} \lambda_s \int_0^1 \sqrt{(\partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\,\mathrm{d}q$$

Multi-Species Algorithmic Threshold

Theorem (Huang-**S** 23+)

Define

$$\mathrm{ALG} = \sup_{\substack{\Phi:[0,1]\to[0,1]^{\mathscr{S}} \\ \textit{increasing, differentiable}}} \sum_{s\in\mathscr{S}} \lambda_s \int_0^1 \sqrt{(\partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\, \mathrm{d}q$$

- *An explicit $O(1)$-Lipschitz algorithm achieves* ALG *w.h.p.*
- *No $O(1)$-Lipschitz algorithm beats* ALG *with probability e^{-cN}.*

Multi-Species Algorithmic Threshold

Theorem (Huang-**S** 23+)

Define

$$\mathrm{ALG} = \sup_{\substack{\Phi:[0,1]\to[0,1]^{\mathscr{S}} \\ \textit{increasing, differentiable}}} \sum_{s\in\mathscr{S}} \lambda_s \int_0^1 \sqrt{(\partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\, \mathrm{d}q$$

- *An explicit $O(1)$-Lipschitz algorithm achieves* ALG *w.h.p.*
- *No $O(1)$-Lipschitz algorithm beats* ALG *with probability e^{-cN}.*

(More general threshold with external fields too)

Multi-Species Algorithmic Threshold

Theorem (Huang-**S** 23+)

Define

$$\mathrm{ALG} = \sup_{\substack{\Phi:[0,1]\to[0,1]^{\mathscr{S}} \\ \textit{increasing, differentiable}}} \sum_{s\in\mathscr{S}} \lambda_s \int_0^1 \sqrt{(\partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\,\mathrm{d}q$$

- *An explicit $O(1)$-Lipschitz algorithm achieves* ALG *w.h.p.*
- *No $O(1)$-Lipschitz algorithm beats* ALG *with probability e^{-cN}.*

(More general threshold with external fields too)

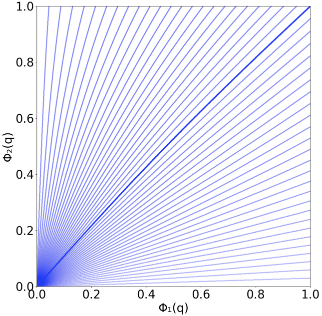
Theorem (Huang-**S** 23+)

The variational formula has a maximizer Φ, which solves an explicit ODE.

Variational Problem Example

Consider $(\lambda_1, \lambda_2) = (1/3, 2/3)$ and

$$\xi(q_1, q_2) = (\lambda_1 q_1)^2 + (\lambda_1 q_1)(\lambda_2 q_1) + (\lambda_2 q_1)^2 + (\lambda_1 q_1)^4 + (\lambda_1 q_1)(\lambda_2 q_2)^3$$

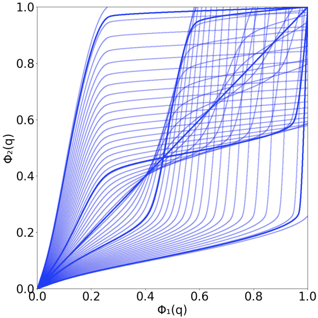


Some ODE solutions. Optimal $\Phi : [0, 1] \to [0, 1]^2$ in bold

Algorithmic Symmetry Breaking

Optimal Φ may be asymmetric, even when model is symmetric!

$$\lambda_1 = \lambda_2 = \frac{1}{2}, \quad \xi(q_1, q_2) = (3q_1)^2 + (3q_1)(3q_2) + (3q_2)^2 + (3q_1)^4 + (3q_2)^4$$



The plot thickens...

Pure Multi-Species Models

- Models where
$$\xi(q_1, \dots, q_r) = q_1^{a_1} q_2^{a_2} \cdots q_r^{a_r}$$
- Example: bipartite SK $\xi(q_1, q_2) = q_1 q_2$

Pure Multi-Species Models

- Models where

$$\xi(q_1, \dots, q_r) = q_1^{a_1} q_2^{a_2} \cdots q_r^{a_r}$$

- Example: bipartite SK $\xi(q_1, q_2) = q_1 q_2$

- Optimal Φ is **polynomial**

$$\Phi(q) = (q^{b_1}, \dots, q^{b_r})$$

- In this case, $\text{ALG} = E_\infty$ has explicit non-variational formula.
- Langevin dynamics is believed to reach the same threshold!

Summary

- We determine algorithmic threshold of $O(1)$-Lipschitz algorithms for optimizing multi-species spherical spin glasses

Summary

- We determine algorithmic threshold of $O(1)$-Lipschitz algorithms for optimizing multi-species spherical spin glasses
- Branching OGP matches Subag algorithm for generic reason

Summary

- We determine algorithmic threshold of $O(1)$-Lipschitz algorithms for optimizing multi-species spherical spin glasses
- Branching OGP matches Subag algorithm for generic reason
- Geometric description of ALG: largest value whose super-level set contains densely-branching ultrametric tree
  - Optimal algorithms climb this tree
  - Absence of this tree implies hardness by BOGP

Summary

- We determine algorithmic threshold of $O(1)$-Lipschitz algorithms for optimizing multi-species spherical spin glasses
- Branching OGP matches Subag algorithm for generic reason
- Geometric description of ALG: largest value whose super-level set contains densely-branching ultrametric tree
  - Optimal algorithms climb this tree
  - Absence of this tree implies hardness by BOGP
  - Comparison with OPT ultrametricity: ALG trees must branch continuously, OPT trees may not

Summary

- We determine algorithmic threshold of $O(1)$-Lipschitz algorithms for optimizing multi-species spherical spin glasses
- Branching OGP matches Subag algorithm for generic reason
- Geometric description of ALG: largest value whose super-level set contains densely-branching ultrametric tree
  - Optimal algorithms climb this tree
  - Absence of this tree implies hardness by BOGP
  - Comparison with OPT ultrametricity: ALG trees must branch continuously, OPT trees may not

Thank you!

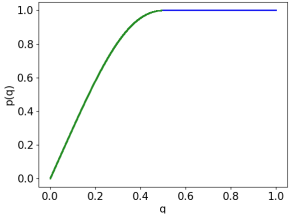
Models with Linear Terms

Suppose model has 1-spin interaction (external field)

$$H_N(\boldsymbol{\sigma}) = \sum_{p=\color{red}{1}}^{P} \frac{\gamma_p}{N^{(p-1)/2}} \langle \boldsymbol{G}^{(p)}, \boldsymbol{\sigma}^{\otimes p} \rangle \qquad \xi(q) = \sum_{p=\color{red}{1}}^{P} \gamma_p^2 q^p$$

Then

$$\mathrm{ALG} = \mathrm{BOGP} = \sup_{\substack{p:[0,1]\to[0,1] \\ \text{increasing, differentiable}}} \int_0^1 \sqrt{(p\xi')'(q)}\,\mathrm{d}q$$



Optimal p for $\xi(q) = q^4 + q$

Multi-Species Algorithmic Threshold with Linear Terms

Theorem (Huang-S 23+)

Define

$$\mathsf{ALG} = \sup_{\substack{p:[0,1]\to[0,1] \\ \Phi:[0,1]\to[0,1]^{\mathscr{S}} \\ \textit{increasing, differentiable}}} \sum_{s\in\mathscr{S}} \lambda_s \int_0^1 \sqrt{(p \times \partial_{q_s}\xi \circ \Phi)'(q)\Phi_s'(q)}\, \mathrm{d}q$$

- *An explicit $O(1)$-Lipschitz algorithm achieves* ALG *w.h.p.*
- *No $O(1)$-Lipschitz algorithm beats* ALG *with probability e^{-cN}*

Theorem (Huang-S 23+)

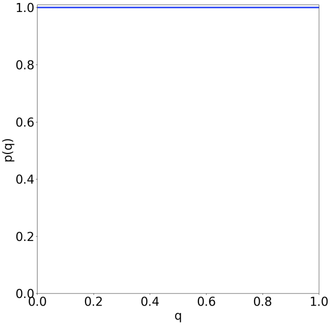
This variational problem has a maximizer (p, Φ).

- *The maximizer solves an explicit ODE.*
- *If ξ has no 1-spin interactions, then $p \equiv 1$.*

Variational Problem Example: No Linear Term

Consider $(\lambda_1, \lambda_2) = (1/3, 2/3)$

$$\xi(q_1, q_2) = (\lambda_1 q_1)^2 + (\lambda_1 q_1)(\lambda_2 q_1) + (\lambda_2 q_1)^2 + (\lambda_1 q_1)^4 + (\lambda_1 q_1)(\lambda_2 q_2)^3$$



Optimal $p : [0,1] \to [0,1]$

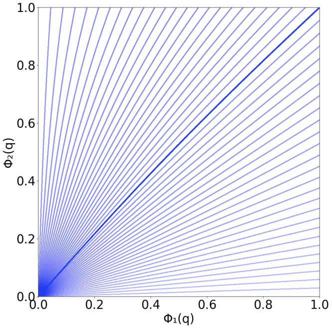
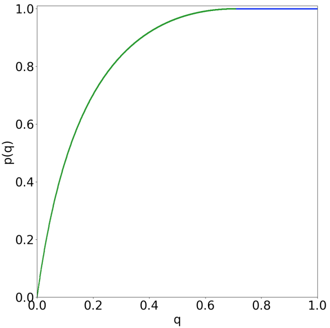


Image of optimal $\Phi : [0,1] \to [0,1]^2$ in bold

Variational Problem Example: Small Linear Term

Consider $(\lambda_1, \lambda_2) = (1/3, 2/3)$

$$\xi(q_1, q_2) = (\lambda_1 q_1)^2 + (\lambda_1 q_1)(\lambda_2 q_1) + (\lambda_2 q_1)^2 + (\lambda_1 q_1)^4 + (\lambda_1 q_1)(\lambda_2 q_2)^3 + 0.05(\lambda_1 q_1) + 0.5(\lambda_2 q_2)$$



Optimal $p : [0,1] \to [0,1]$

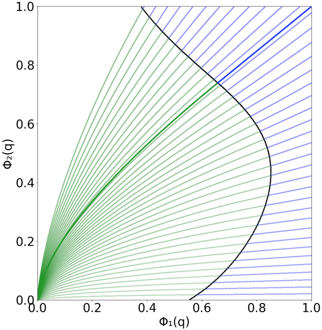
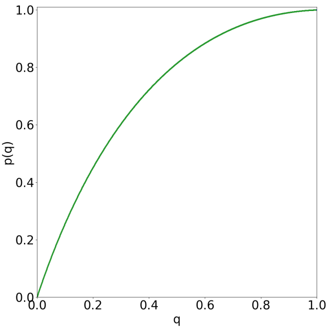


Image of optimal $\Phi : [0,1] \to [0,1]^2$ in bold

Variational Problem Example: Large Linear Term

Consider $(\lambda_1, \lambda_2) = (1/3, 2/3)$

$$\xi(q_1, q_2) = (\lambda_1 q_1)^2 + (\lambda_1 q_1)(\lambda_2 q_1) + (\lambda_2 q_1)^2 + (\lambda_1 q_1)^4 + (\lambda_1 q_1)(\lambda_2 q_2)^3 + 0.2(\lambda_1 q_1) + 1.8(\lambda_2 q_2)$$



Optimal $p : [0,1] \to [0,1]$

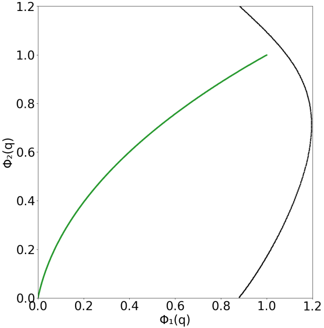


Image of optimal $\Phi : [0,1] \to [0,1]^2$