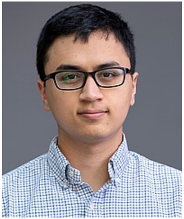
Algorithmic thresholds for random perceptron models

Brice Huang (Stanford → Yale)

Joint work with **Mark Sellke** (Harvard/OpenAI) and **Nike Sun** (MIT)



February 9, 2026 | Stanford probability seminar

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

Proof ideas for hardness, recap and generalization

Optimal algorithms via IAMP

Applications and conclusion

Projection pursuit

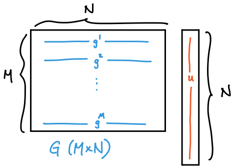
Projection pursuit (Kruskal 72, Friedman Tukey 74, Diaconis Freedman 84):

Projection pursuit

Projection pursuit (Kruskal 72, Friedman Tukey 74, Diaconis Freedman 84): cloud of M points $\boldsymbol{g}^1, \dots, \boldsymbol{g}^M$ in dim N. **What low-dim projections are possible?**

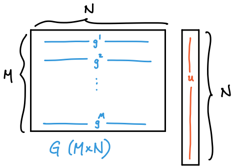
Projection pursuit

Projection pursuit (Kruskal 72, Friedman Tukey 74, Diaconis Freedman 84): cloud of M points $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^M$ in dim N. **What low-dim projections are possible?**



Projection pursuit

Projection pursuit (Kruskal 72, Friedman Tukey 74, Diaconis Freedman 84): cloud of M points $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^M$ in dim N. **What low-dim projections are possible?**

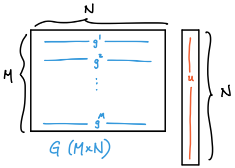


Formal statement (1D): let $\boldsymbol{G}$ be $M \times N$ iid gaussian with rows $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^M$. For $\boldsymbol{u} \in \mathbb{S}^{N-1}$, **what are the possible measures**

$$\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{u}) \equiv \frac{1}{M} \sum_{a=1}^{M} \delta(\boldsymbol{g}^a, \boldsymbol{u})\,?$$

Projection pursuit

Projection pursuit (Kruskal 72, Friedman Tukey 74, Diaconis Freedman 84): cloud of M points $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^M$ in dim N. **What low-dim projections are possible?**



Formal statement (1D): let $\boldsymbol{G}$ be $M \times N$ iid gaussian with rows $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^M$. For $\boldsymbol{u} \in \mathbb{S}^{N-1}$, **what are the possible measures**

$$\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{u}) \equiv \frac{1}{M} \sum_{a=1}^{M} \delta(\boldsymbol{g}^a, \boldsymbol{u}) \,?$$

$M/N \to \infty$: **all** low-dim proj $\to$ standard gaussian (DF84).
$M/N = \alpha \asymp 1$: $\exists$ non-gaussian projections (Bickel Kur Nadler 18).

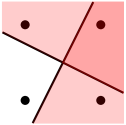
Statistical foundations of projection pursuit

A key challenge in analyzing high-dimensional data is to extract meaningful low-dimensional structures...with a limited amount of high-dimensional data, the results of projection pursuit and related ICA methods should be interpreted with great care...one may find projections with highly non-Gaussian components that have no statistical or scientific significance. (Bickel Kur Nadler 18)

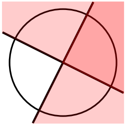
What is an "interesting" projection? We cannot expect universal agreement on what constitutes an "interesting" projection. A projection in which the data separate into disinct, meaningful clusters would certainly be interesting. But there are also interesting features that are not of the distinct cluster type (e.g. an edge, or jump of density, at the boundary of some region). (Huber 85)

The perceptron model

Perceptron: intersection of $\Sigma_N = \{-1, +1\}^N$ (**Ising**) or $\Sigma_N = N^{1/2}\mathbb{S}^{N-1}$ (**spherical**) with M iid random half-spaces:



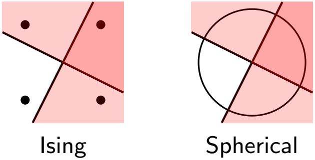
Ising



Spherical

The perceptron model

Perceptron: intersection of $\Sigma_N = \{-1, +1\}^N$ (**Ising**) or $\Sigma_N = N^{1/2}\mathbb{S}^{N-1}$ (**spherical**) with M iid random half-spaces:

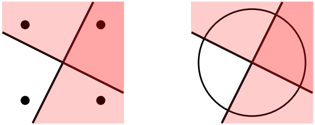


Formally, for $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, and $\kappa \in \mathbb{R}$ fixed,

$$S(\boldsymbol{G}) \equiv \left\{ \boldsymbol{x} \in \Sigma_N : \frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}} \geqslant \kappa \text{ for all } a \leqslant M \right\}.$$

The perceptron model

Perceptron: intersection of $\Sigma_N = \{-1,+1\}^N$ (**Ising**) or $\Sigma_N = N^{1/2}\mathbb{S}^{N-1}$ (**spherical**) with M iid random half-spaces:



Ising Spherical

Formally, for $\boldsymbol{G} \in \mathbb{R}^{M\times N}$ iid gaussian, and $\kappa \in \mathbb{R}$ fixed,

$$S(\boldsymbol{G}) \equiv \left\{ \boldsymbol{x} \in \Sigma_N : \frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}} \geqslant \kappa \text{ for all } a \leqslant M \right\}.$$

Basic question: for what values of $\alpha = M/N$ is the random set $S(\boldsymbol{G})$ (non)empty whp?
$\leftrightarrow$ variant of projection pursuit problem, with $\boldsymbol{x} = \sqrt{N}\boldsymbol{u}$.

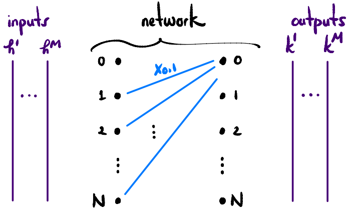
Connections to other problems

The perceptron is related to several other problems of interest:

Connections to other problems

The perceptron is related to several other problems of interest:

- Toy model of single-layer **neural network** (Gardner 88):
synaptic weights $\leftrightarrow$ $\boldsymbol{x}$, random patterns $\leftrightarrow$ rows $\boldsymbol{g}^a$ of $\boldsymbol{G}$.



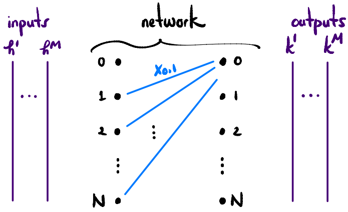
Network maps $h \mapsto k$,

$$k_i = \mathrm{sgn}\Big(\sum_{j \neq i} x_{i,j}\, h_j\Big)$$

Connections to other problems

The perceptron is related to several other problems of interest:

• Toy model of single-layer **neural network** (Gardner 88):
synaptic weights $\leftrightarrow$ $\boldsymbol{x}$, random patterns $\leftrightarrow$ rows $\boldsymbol{g}^a$ of $\boldsymbol{G}$.



Network maps $h \mapsto k$,

$$k_i = \mathrm{sgn}\Big(\sum_{j \neq i} x_{i,j}\, h_j\Big)$$

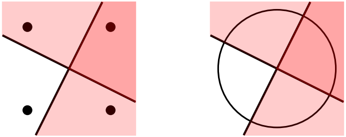
• **Discrepancy** minimization: symmetric model with constraints $(\boldsymbol{g}^a, \boldsymbol{x})/N^{1/2} \in [-\kappa, \kappa]$ (Aubin Perkins Zdeborová 19; Spencer 85, Bansal 10, Lovett Meka 15, Rothvoss 17, Eldan Singh 18).

Prior work on statics of perceptron model

Recall: for $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, and $\kappa \in \mathbb{R}$ fixed,

$$S(\boldsymbol{G}) \equiv \left\{ \boldsymbol{x} \in \Sigma_N : \frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}} \geqslant \kappa \text{ coordinatewise} \right\}.$$

Many basic questions about typical size and structure of $S(\boldsymbol{G})$.

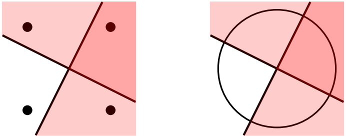


Prior work on statics of perceptron model

Recall: for $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, and $\kappa \in \mathbb{R}$ fixed,

$$S(\boldsymbol{G}) \equiv \left\{ \boldsymbol{x} \in \Sigma_N : \frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}} \geqslant \kappa \text{ coordinatewise} \right\}.$$

Many basic questions about typical size and structure of $S(\boldsymbol{G})$.



For example the **capacity problem**: does $S(\boldsymbol{G})$ go from nonempty to empty (whp) around a critical threshold $M/N \sim \alpha_{\text{cap}}$?

(Gardner Derrida 88, Krauth Mézard 89, Shcherbina Tirozzi 03, Kim Roche 98, Talagrand 00, Stojnic 13, Ding Sun 18, H. 24)

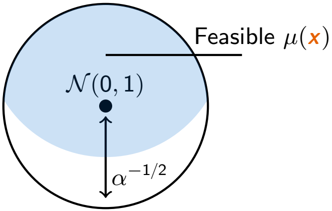
Prior work on statics of projection pursuit

Montanari Zhou 22: for $M/N = \alpha$, whp all feasible values of

$$\mu(\boldsymbol{x}) \equiv \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right),$$

over $\boldsymbol{x} \in \sqrt{N}\mathbb{S}^{N-1}$ satisfy

$$W_2(\mu(\boldsymbol{x}), \mathcal{N}(0,1)) \leqslant \alpha^{-1/2}$$



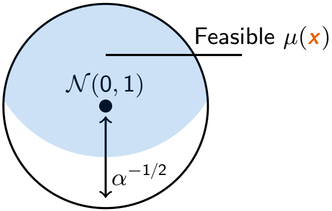
Prior work on statics of projection pursuit

Montanari Zhou 22: for $M/N = \alpha$, whp all feasible values of

$$\mu(\boldsymbol{x}) \equiv \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right),$$

over $\boldsymbol{x} \in \sqrt{N}\mathbb{S}^{N-1}$ satisfy

$$W_2(\mu(\boldsymbol{x}), \mathcal{N}(0,1)) \leqslant \alpha^{-1/2}$$



Montanari Zhou 24: conjectured description of feasible set via physics methods (in terms of Parisi-type formula)

Three formulations of the algorithmic problem

Given $M/N = \alpha$, $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, **do algorithmically**:

Three formulations of the algorithmic problem

Given $M/N = \alpha$, $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, **do algorithmically**:

- **Projection pursuit:** given target measure μ, find $\boldsymbol{x}$ such that

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \approx \mu \text{ (in } W_2).$$

Three formulations of the algorithmic problem

Given $M/N = \alpha$, $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, **do algorithmically**:

- **Projection pursuit:** given target measure μ, find $\boldsymbol{x}$ such that

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \approx \mu \text{ (in } W_2).$$

- **Constraint satisfaction:** given $\boldsymbol{I} \subseteq \mathbb{R}$, find $\boldsymbol{x}$ such that

$$\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}} \in \boldsymbol{I} \text{ for all } a \leqslant M\,.$$

Three formulations of the algorithmic problem

Given $M/N = \alpha$, $\boldsymbol{G} \in \mathbb{R}^{M\times N}$ iid gaussian, **do algorithmically**:

- **Projection pursuit:** given target measure μ, find $\boldsymbol{x}$ such that

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \approx \mu \text{ (in } W_2).$$

- **Constraint satisfaction:** given $\boldsymbol{I} \subseteq \mathbb{R}$, find $\boldsymbol{x}$ such that

$$\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}} \in \boldsymbol{I} \text{ for all } a \leqslant M\,.$$

- **Optimization:** given $\varphi : \mathbb{R} \to \mathbb{R}$, find $\boldsymbol{x}$ to optimize

$$H(\boldsymbol{x}) = \sum_{a=1}^{M} \varphi\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right)\,.$$

Three formulations of the algorithmic problem

Given $M/N = \alpha$, $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ iid gaussian, **do algorithmically**:

- **Projection pursuit:** given target measure μ, find $\boldsymbol{x}$ such that

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \approx \mu \text{ (in } W_2).$$

- **Constraint satisfaction:** given $\boldsymbol{I} \subseteq \mathbb{R}$, find $\boldsymbol{x}$ such that

$$\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}} \in \boldsymbol{I} \text{ for all } a \leqslant M\,.$$

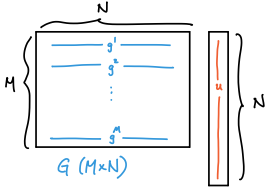
- **Optimization:** given $\varphi : \mathbb{R} \to \mathbb{R}$, find $\boldsymbol{x}$ to optimize

$$H(\boldsymbol{x}) = \sum_{a=1}^{M} \varphi\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right)\,.$$

Can consider $\boldsymbol{x} \in \{-1, +1\}^N$ **(Ising)** of $\boldsymbol{x} \in \sqrt{N}\mathbb{S}^{N-1}$ **(spherical)**.

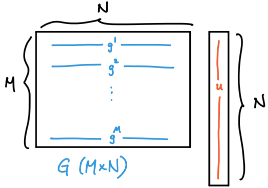
Algorithmic projection pursuit

Main result (HSS 25$^+$): for $M/N = \alpha$, we characterize the set of all measures EmpDist$(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by a broad class of **algorithms** $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, for $\boldsymbol{x}$ in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$.



Algorithmic projection pursuit

Main result (HSS 25$^+$): for $M/N = \alpha$, we characterize the set of all measures EmpDist$(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by a broad class of **algorithms** $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, for $\boldsymbol{x}$ in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$.



For $\boldsymbol{x} \in \sqrt{N}\mathbb{S}^{N-1}$, optimal algorithms obtained independently Montanari Zhou 24; our work gives matching hardness bound.

Theorem is phrased for **projection pursuit**, but implies algorithmic thresholds for **constraint satisfaction** and **optimization** settings.

Prior algorithmic results

Many works on both algorithms and hardness:

- Negative spherical perceptron ($\boldsymbol{I} = [\kappa, \infty)$ with $\kappa < 0$): Montanari Zhong Zhou 21, El Alaoui Sellke 20.

- Symmetric Ising perceptron ($\boldsymbol{I} = [-\kappa, \kappa]$): Bansal Spencer 20, Abbe Li Sly 21, Gamarnik Perkins Kızıldağ Xu 22+23.

- Asymmetric Ising perceptron ($\boldsymbol{I} = [\kappa, \infty)$): Kim Roche 98, Li Schramm Zhou 24.

We address these models in a unified way and close several gaps between algorithms and hardness results.

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

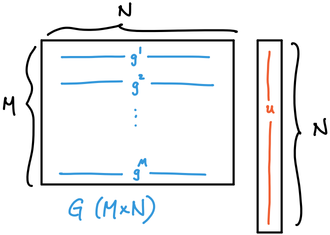
Proof ideas for hardness, recap and generalization

Optimal algorithms via IAMP

Applications and conclusion

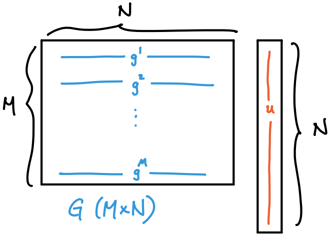
Problem restatement

Recall: $\boldsymbol{G}$ is $M \times N$, iid gaussian, with rows $\boldsymbol{g}^1, \ldots, \boldsymbol{g}^N$.



Problem restatement

Recall: $\boldsymbol{G}$ is $M \times N$, iid gaussian, with rows $\boldsymbol{g}^1, \dots, \boldsymbol{g}^N$.



We want to characterize **the set of all possible measures**

$$\mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right)$$

achievable by **algorithms** $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$ (in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$).

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2\,.$$

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2 \,.$$

Philosophy: hard to analyze "all poly-time algorithms" without solving $P = NP$. Instead, two main approaches:

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2\,.$$

Philosophy: hard to analyze "all poly-time algorithms" without solving $P = NP$. Instead, two main approaches:

- Reduce from a postulated hard problem (e.g. SAT)

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2 \,.$$

Philosophy: hard to analyze "all poly-time algorithms" without solving $P = NP$. Instead, two main approaches:

- Reduce from a postulated hard problem (e.g. SAT)

- Prove unconditional result about restricted class of computation

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2 \,.$$

Philosophy: hard to analyze "all poly-time algorithms" without solving $P = NP$. Instead, two main approaches:

- Reduce from a postulated hard problem (e.g. SAT)
(usual approach in **worst-case** problem setups)

- Prove unconditional result about restricted class of computation

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2 \,.$$

Philosophy: hard to analyze “all poly-time algorithms” without solving $P = NP$. Instead, two main approaches:

- Reduce from a postulated hard problem (e.g. SAT) (usual approach in **worst-case** problem setups)

- Prove unconditional result about restricted class of computation (common in **average-case** problems, like ours)

Informal result

Sharp algorithmic threshold for Lipschitz algorithms

$$\|\mathcal{A}(\boldsymbol{G}) - \mathcal{A}(\boldsymbol{G}')\|_2 \leqslant O(1)\|\boldsymbol{G} - \boldsymbol{G}'\|_2 \,.$$

Philosophy: hard to analyze "all poly-time algorithms" without solving $P = NP$. Instead, two main approaches:

- Reduce from a postulated hard problem (e.g. SAT)
(usual approach in **worst-case** problem setups)

- Prove unconditional result about restricted class of computation
(common in **average-case** problems, like ours)

Lipschitz algs include: grad desc, Langevin, AMP for $O(1)$ time, any $O(1)$-order method for $O(1)$ time, but not SOS or low-deg polynomials.

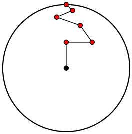
Informal result

Broader program: algorithmic thresholds in **random optimization / search problems**, e.g. SK and p-spin glass models

Informal result

Broader program: algorithmic thresholds in **random optimization / search problems**, e.g. SK and p-spin glass models

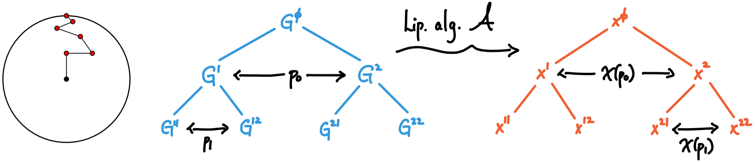
- Optimal **algorithms** following **incremental approximate message passing (IAMP)** framework (Montanari 18).



Informal result

Broader program: algorithmic thresholds in **random optimization / search problems**, e.g. SK and p-spin glass models

- Optimal **algorithms** following **incremental approximate message passing (IAMP)** framework (Montanari 18).



- Optimal **hardness** following **branching overlap gap property** framework (Gamarnik Sudan 14, H. Sellke 22).

Statement of our main result

Statement of our main result

Main theorem (HSS 25^+)

For Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, the set $S^{\mathrm{alg}}(\alpha)$ of achievable

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right),$$

is given by an **explicit stochastic control problem*** *(SDE subject to coefficient constraints), for $\boldsymbol{x}$ in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$.*

Statement of our main result

Main theorem (HSS 25^+)

For Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, the set $S^{\mathrm{alg}}(\alpha)$ of achievable

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M}\delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right),$$

is given by an **explicit stochastic control problem*** *(SDE subject to coefficient constraints), for $\boldsymbol{x}$ in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$.*

$$S^{\mathrm{alg}}(\alpha) = \left\{\mathrm{Law}(X_1) \; : \; X_1 = \int_0^1 b_t\, dt + \sigma_t\, dB_t, \right.$$
$$\left. (b,\sigma) \in \text{admissible set}\right\}$$

Statement of our main result

Main theorem (HSS 25^+)

For Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, the set $S^{\mathrm{alg}}(\alpha)$ of achievable

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right),$$

is given by an **explicit stochastic control problem*** *(SDE subject to coefficient constraints), for $\boldsymbol{x}$ in $\sqrt{N}\mathbb{S}^{N-1}$ or $\{-1,+1\}^N$.*

$$S^{\mathrm{alg}}(\alpha) = \left\{ \mathrm{Law}(X_1) \;:\; X_1 = \int_0^1 b_t \, dt + \sigma_t \, dB_t, \right. \\ \left. (b, \sigma) \in \text{admissible set} \right\}$$

*SDE will be shown in coming slides for various problem settings (spherical/Ising, symmetric/asymmetric).

Simplified theorem for **symmetric** spherical perceptron

Simplified theorem for **symmetric** spherical perceptron

Simplest case: for spherical perceptron, $M/N = \alpha$, consider

$$\mu^{\mathsf{sym}}(\boldsymbol{x}) \equiv \mathsf{EmpDist}\left(\frac{|\boldsymbol{G}\boldsymbol{x}|}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{|(\boldsymbol{g}^a, \boldsymbol{x})|}{\sqrt{N}}\right).$$

Simplified theorem for **symmetric** spherical perceptron

Simplest case: for spherical perceptron, $M/N = \alpha$, consider

$$\mu^{\mathrm{sym}}(\boldsymbol{x}) \equiv \mathrm{EmpDist}\left(\frac{|\boldsymbol{G}\boldsymbol{x}|}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{|(\boldsymbol{g}^a, \boldsymbol{x})|}{\sqrt{N}}\right).$$

Consider the **1D controlled SDE** (with $X_0 = 0$, σ_t adapted)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

Simplified theorem for **symmetric** spherical perceptron

Simplest case: for spherical perceptron, $M/N = \alpha$, consider

$$\mu^{\mathrm{sym}}(\boldsymbol{x}) \equiv \mathrm{EmpDist}\left(\frac{|\boldsymbol{G}\boldsymbol{x}|}{\sqrt{N}}\right) \equiv \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{|(\boldsymbol{g}^a, \boldsymbol{x})|}{\sqrt{N}}\right).$$

Consider the **1D controlled SDE** (with $X_0 = 0$, σ_t adapted)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1\,. \end{cases}$$

Theorem (HSS 25^+; **symmetric** spherical perceptron)

For the spherical perceptron, the measures $\mu^{\mathrm{sym}}(\boldsymbol{x})$ *achievable by Lipschitz algorithms are precisely of the form* $\mathrm{Law}(|X_1|)$ *for* X *satisfying the above controlled SDE.*

Depiction of controlled SDE

1D controlled SDE (with $X_0 = 0$, σ_t progressive)

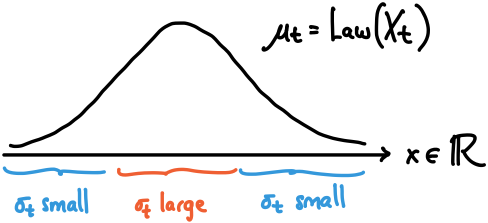
$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

Depiction of controlled SDE

1D controlled SDE (with $X_0 = 0$, σ_t progressive)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

That is, X_t evolves 'by default' like standard BM, but we can modify the local diffusivity σ_t subject to the **budget constraint**.

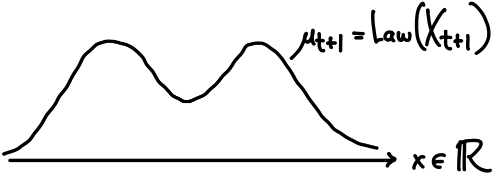


Depiction of controlled SDE

1D controlled SDE (with $X_0 = 0$, σ_t progressive)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

That is, X_t evolves 'by default' like standard BM, but we can modify the local diffusivity σ_t subject to the **budget constraint**.

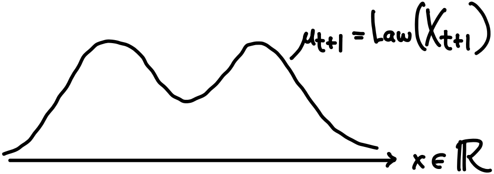


Depiction of controlled SDE

1D controlled SDE (with $X_0 = 0$, σ_t progressive)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

That is, X_t evolves 'by default' like standard BM, but we can modify the local diffusivity σ_t subject to the **budget constraint**.



❗ We give a stochastic description, but the resulting evolution of $\mu_t = \mathrm{Law}(X_t)$ is **deterministic**.

General theorem for **(asymmetric)** spherical perceptron

General theorem for **(asymmetric)** spherical perceptron

Consider 1D controlled SDE (with $X_0 = 0$, b_t and σ_t progressive)

$$\begin{cases} dX_t = \sqrt{p'(t)}\; b_t \, dt + \sqrt{p(t) + tp'(t)}\; \sigma_t \, dB_t \\ \text{subject to } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1, \end{cases}$$

with $p : [0,1] \to [0,1]$ inc diffble ($p \equiv 1$ recovers symmetric SDE).

General theorem for **(asymmetric)** spherical perceptron

Consider 1D controlled SDE (with $X_0 = 0$, b_t and σ_t progressive)

$$\begin{cases} dX_t = \sqrt{p'(t)}\; b_t\, dt + \sqrt{p(t) + tp'(t)}\; \sigma_t\, dB_t \\ \text{subject to } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1, \end{cases}$$

with $p : [0,1] \to [0,1]$ inc diffble ($p \equiv 1$ recovers symmetric SDE).

Theorem (HSS 25$^+$; **general** spherical perceptron)

For the **(asymmetric)** *spherical perceptron, the measures $\mu(\boldsymbol{x})$ achievable by Lipschitz algorithms are precisely of the form* Law(X_1) *for X satisfying the above controlled SDE.*

General theorem for **(asymmetric)** spherical perceptron

Consider 1D controlled SDE (with $X_0 = 0$, b_t and σ_t progressive)

$$\begin{cases} dX_t = \sqrt{p'(t)}\; b_t\, dt + \sqrt{p(t) + tp'(t)}\; \sigma_t\, dB_t \\ \text{subject to } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1, \end{cases}$$

with $p : [0,1] \to [0,1]$ inc diffble ($p \equiv 1$ recovers symmetric SDE).

Theorem (HSS 25^+; **general** spherical perceptron)

For the **(asymmetric)** *spherical perceptron, the measures* $\mu(\boldsymbol{x})$ *achievable by Lipschitz algorithms are precisely of the form* $\mathrm{Law}(X_1)$ *for* X *satisfying the above controlled SDE.*

Analogous result for Ising perceptron (later).

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

Proof ideas for hardness, recap and generalization

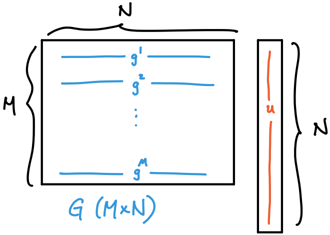
Optimal algorithms via IAMP

Applications and conclusion

Method: analysis of Doob martingale

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

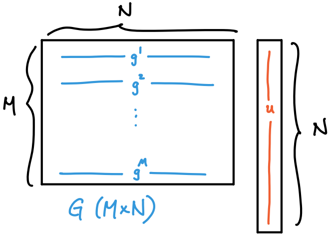
$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M}\delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$



Method: analysis of Doob martingale

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$

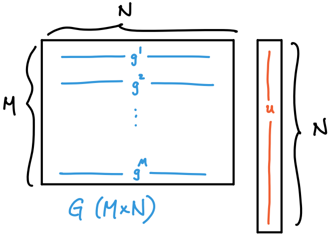


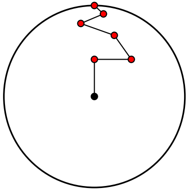
- **Matrix Brownian motion** $\boldsymbol{G}_t$, $t \in [0, 1]$, with $\boldsymbol{G}_1 = \boldsymbol{G}$.

Method: analysis of Doob martingale

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{\langle \boldsymbol{g}^a, \boldsymbol{x}\rangle}{\sqrt{N}}\right).$$





- **Matrix Brownian motion** $\boldsymbol{G}_t$, $t \in [0,1]$, with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

Overview of proof idea: SDE

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

Overview of proof idea: SDE

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

The SDE in our main theorem describes the evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t\boldsymbol{x}_t}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}_t^a, \boldsymbol{x}_t)}{\sqrt{N}}\right).$$

Overview of proof idea: SDE

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1) | \boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

The SDE in our main theorem describes the evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t\boldsymbol{x}_t}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}_t^a, \boldsymbol{x}_t)}{\sqrt{N}}\right).$$

We will argue that $\mu_t = \mathrm{Law}(X_t)$ for X solving the SDE.

Overview of proof idea: SDE

Recall: for Lipschitz algorithms $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$, want to understand

$$\mu(\boldsymbol{x}) = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}\boldsymbol{x}}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a, \boldsymbol{x})}{\sqrt{N}}\right).$$

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

The SDE in our main theorem describes the evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t\boldsymbol{x}_t}{\sqrt{N}}\right) = \frac{1}{M}\sum_{a=1}^{M} \delta\left(\frac{(\boldsymbol{g}^a_t, \boldsymbol{x}_t)}{\sqrt{N}}\right).$$

We will argue that $\mu_t = \mathrm{Law}(X_t)$ **for** X **solving the SDE.**

Therefore, $\mu_1 = \mathrm{Law}(X_1) \leftrightarrow$ algorithmically achievable $\mu(\boldsymbol{x})$.

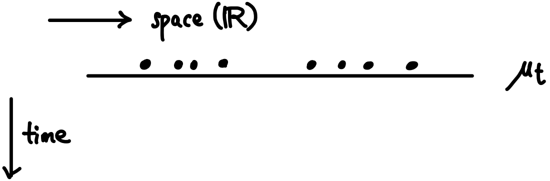
SDE describing evolving measure

Tracking the evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

SDE describing evolving measure

Tracking the evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

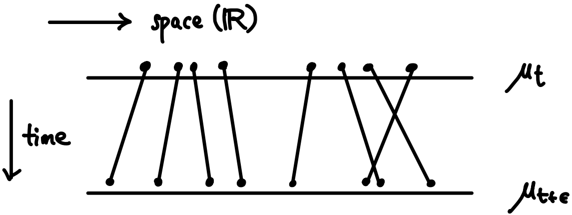
At time t we have particles at the M entries of $\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N}$:



SDE describing evolving measure

Tracking the evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

At time t we have particles at the M entries of $\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N}$:

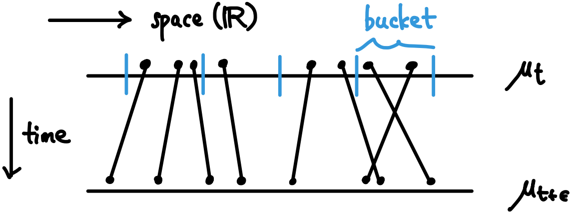


After time ϵ they move to new positions given by $\boldsymbol{G}_{t+\epsilon} \boldsymbol{x}_{t+\epsilon}/\sqrt{N}$.

SDE describing evolving measure

Tracking the evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

At time t we have particles at the M entries of $\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N}$:



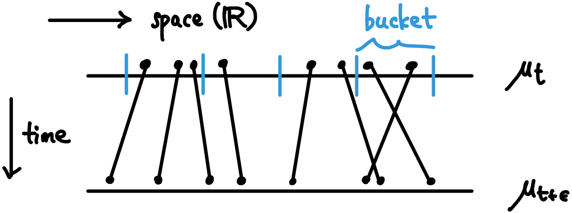
After time ϵ they move to new positions given by $\boldsymbol{G}_{t+\epsilon} \boldsymbol{x}_{t+\epsilon}/\sqrt{N}$.

Discretize space into "buckets" to relate above to a stochastic control problem, with SDE coeffs $\approx$ (bucket drift, bucket variance).

SDE describing evolving measure

Tracking the evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

At time t we have particles at the M entries of $\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N}$:



After time ϵ they move to new positions given by $\boldsymbol{G}_{t+\epsilon} \boldsymbol{x}_{t+\epsilon}/\sqrt{N}$.

Discretize space into "buckets" to relate above to a stochastic control problem, with SDE coeffs $\approx$ (bucket drift, bucket variance).

! Matching Fokker–Planck eqns; **not** a pathwise description.

Symmetric spherical perceptron, single bucket

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)

For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)}{\|\Delta \boldsymbol{x}_t\|}$$

$$\mathrm{var}(B) = \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)^2}{\|\Delta \boldsymbol{x}_t\|^2}$$

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)

For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)}{\|\Delta \boldsymbol{x}_t\|}$$

$$\mathrm{var}(B) = \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)^2}{\|\Delta \boldsymbol{x}_t\|^2}$$

What drifts and variances are possible?

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)

For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)}{\|\Delta\boldsymbol{x}_t\|} \approx 0 \quad \text{(concentration)},$$

$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)^2}{\|\Delta\boldsymbol{x}_t\|^2}$$

What drifts and variances are possible?

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)

For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)}{\|\Delta \boldsymbol{x}_t\|} \approx 0 \quad (\text{concentration}),$$

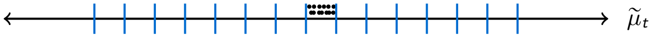
$$\mathrm{var}(B) = \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)^2}{\|\Delta \boldsymbol{x}_t\|^2}$$

What drifts and variances are possible?

Simplest case of **one bucket** $B = [M]$:

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)



For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)}{\|\Delta \boldsymbol{x}_t\|} \approx 0 \quad \text{(concentration)},$$

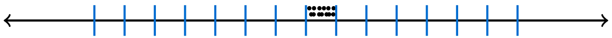
$$\mathrm{var}(B) = \frac{1}{|B|} \sum_{a \in B} \frac{(\boldsymbol{g}^a, \Delta \boldsymbol{x}_t)^2}{\|\Delta \boldsymbol{x}_t\|^2}$$

What drifts and variances are possible?

Simplest case of **one bucket** $B = [M]$:

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)



For each spatial bucket B, bucket drift and variance

$$\mathrm{dr}(B) \equiv \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)}{\|\Delta\boldsymbol{x}_t\|} \approx 0 \quad (\text{concentration})\,,$$

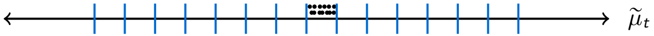
$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)^2}{\|\Delta\boldsymbol{x}_t\|^2} \leqslant \Big\| \boldsymbol{W}^B \equiv \frac{1}{|B|}\sum_{a\in B}(\boldsymbol{g}^a)^{\otimes 2}\Big\|_{\mathrm{op}}\,.$$

What drifts and variances are possible?

Simplest case of **one bucket** $B = [M]$:

Symmetric spherical perceptron, single bucket

For **symmetric** model, tracking $\widetilde{\mu}_t = \mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ suffices.
(Reparametrization: idea is $\boldsymbol{x}_t$ only moves for $t \approx 1$.)



For each spatial bucket B, bucket drift and variance

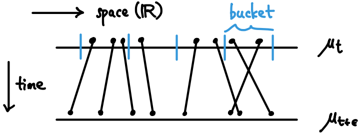
$$\mathrm{dr}(B) \equiv \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)}{\|\Delta\boldsymbol{x}_t\|} \approx 0 \quad \text{(concentration)}\,,$$

$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a\in B}\frac{(\boldsymbol{g}^a, \Delta\boldsymbol{x}_t)^2}{\|\Delta\boldsymbol{x}_t\|^2} \leqslant \left\|\boldsymbol{W}^B \equiv \frac{1}{|B|}\sum_{a\in B}(\boldsymbol{g}^a)^{\otimes 2}\right\|_{\mathrm{op}}.$$

What drifts and variances are possible?

Simplest case of **one bucket** $B = [M]$: Marchenko–Pastur law gives $(\sqrt{\mathrm{var(B)}} - 1)^2 \leqslant 1/\alpha$, equivalent to our budget constraint.

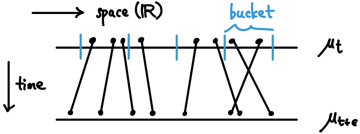
Symmetric spherical perceptron, many buckets



Tracking EmpDist$(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ with **multiple buckets**: $\boldsymbol{u} = \frac{\Delta \boldsymbol{x}_t}{\|\Delta \boldsymbol{x}_t\|}$,

$$\mathrm{var}(B) = \frac{1}{|B|} \sum_{a \in B} (\boldsymbol{g}^a, \boldsymbol{u})^2 = (\boldsymbol{u}, \boldsymbol{W}^B \boldsymbol{u}) \,.$$

Symmetric spherical perceptron, many buckets

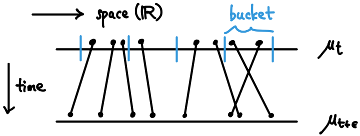


Tracking $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N})$ with **multiple buckets**: $\boldsymbol{u} = \frac{\Delta \boldsymbol{x}_t}{\|\Delta \boldsymbol{x}_t\|}$,

$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a \in B}(\boldsymbol{g}^a, \boldsymbol{u})^2 = (\boldsymbol{u}, \boldsymbol{W}^B \boldsymbol{u})\,.$$

Want to characterize feasible tuples $(\mathrm{var}(B))_{\text{buckets } B}$

Symmetric spherical perceptron, many buckets



Tracking EmpDist($\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N}$) with **multiple buckets**: $\boldsymbol{u} = \frac{\Delta \boldsymbol{x}_t}{\|\Delta \boldsymbol{x}_t\|}$,

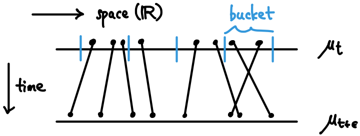
$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a \in B} (\boldsymbol{g}^a, \boldsymbol{u})^2 = (\boldsymbol{u}, \boldsymbol{W}^B \boldsymbol{u}) .$$

Want to characterize feasible tuples $(\mathrm{var}(B))_{\text{buckets } B}$

For any weights $\gamma_B \in \mathbb{R}$, **free probability** gives optimal value

$$\sum_B \gamma_B \mathrm{var}(B) \leqslant \left\| \sum_B \gamma_B \boldsymbol{W}^B \right\|_{\mathrm{op}} .$$

Symmetric spherical perceptron, many buckets



Tracking EmpDist($\boldsymbol{G}\boldsymbol{x}_t/\sqrt{N}$) with **multiple buckets**: $\boldsymbol{u} = \frac{\Delta \boldsymbol{x}_t}{\|\Delta \boldsymbol{x}_t\|}$,

$$\mathrm{var}(B) = \frac{1}{|B|}\sum_{a\in B}(\boldsymbol{g}^a, \boldsymbol{u})^2 = (\boldsymbol{u}, \boldsymbol{W}^B \boldsymbol{u})\,.$$

Want to characterize feasible tuples $(\mathrm{var}(B))_{\text{buckets } B}$

For any weights $\gamma_B \in \mathbb{R}$, **free probability** gives optimal value

$$\sum_B \gamma_B \mathrm{var}(B) \leqslant \left\| \sum_B \gamma_B \boldsymbol{W}^B \right\|_{\mathrm{op}}.$$

Intersection of all γ-constraints corresponds to **budget constraint**

$$\sum_B \frac{|B|}{M}\left(\sqrt{\mathrm{var}(B)} - 1\right)^2 \leqslant \frac{1}{\alpha}\,.$$

Symmetric spherical perceptron, SDE description

Recap: controlled SDE (with $X_0 = 0$, σ_t progressive)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

Theorem (HSS 25^+; **symmetric** spherical perceptron)

For the spherical perceptron, the measures $\mu^{\mathsf{sym}}(\boldsymbol{x})$ achievable by Lipschitz algorithms are precisely of the form $\mathrm{Law}(|X_1|)$ *for X satisfying the above controlled SDE.*

The last few slides explain the proof of the hardness bound.

Symmetric spherical perceptron, SDE description

Recap: controlled SDE (with $X_0 = 0$, σ_t progressive)

$$\begin{cases} dX_t = \sigma_t \, dB_t \,, \text{ subject to} \\ \textbf{budget constraint } \mathbb{E}\Big[\big(\sigma_t - 1\big)^2\Big] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1 \,. \end{cases}$$

Theorem (HSS 25^+; **symmetric** spherical perceptron)

For the spherical perceptron, the measures $\mu^{\mathrm{sym}}(\boldsymbol{x})$ *achievable by Lipschitz algorithms are precisely of the form* $\mathrm{Law}(|X_1|)$ *for* X *satisfying the above controlled SDE.*

The last few slides explain the proof of the hardness bound.

❗ Going from discretized processes to limiting stochastic control problems occupies large part of paper. See e.g. D. Lacker survey "Probabilistic compactification methods" for some of the ideas.

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

Proof ideas for hardness, recap and generalization

Optimal algorithms via IAMP

Applications and conclusion

General **(asymmetric)** spherical perceptron

For general spherical perceptron, track $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t / \sqrt{N})$:

$$\begin{aligned}\Delta(\boldsymbol{G}_t \boldsymbol{x}_t) &= (\Delta \boldsymbol{G}_t)(\Delta \boldsymbol{x}_t) + \boldsymbol{G}_t(\Delta \boldsymbol{x}_t) + (\Delta \boldsymbol{G}_t)\boldsymbol{x}_t \\ &= \text{drift} + \text{diffusivity} + \text{fixed diffusivity}\,,\end{aligned}$$

with first two terms controlled by choice of $\Delta \boldsymbol{x}_t$.

General **(asymmetric)** spherical perceptron

For general spherical perceptron, track $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

$$\begin{aligned}\Delta(\boldsymbol{G}_t\boldsymbol{x}_t) &= (\Delta\boldsymbol{G}_t)(\Delta\boldsymbol{x}_t) + \boldsymbol{G}_t(\Delta\boldsymbol{x}_t) + (\Delta\boldsymbol{G}_t)\boldsymbol{x}_t \\ &= \text{drift} + \text{diffusivity} + \text{fixed diffusivity}\,,\end{aligned}$$

with first two terms controlled by choice of $\Delta\boldsymbol{x}_t$. Define

$$\begin{aligned}\mathrm{dr}(B) &= \text{bucket average of drift} \\ \mathrm{var}(B) &= \text{bucket average of } [\text{diffusivity}^2] \\ \mathrm{cov}(B) &= \text{bucket average of } [\text{diffusivity} \cdot (\text{fixed diffusivity})]\end{aligned}$$

General **(asymmetric)** spherical perceptron

For general spherical perceptron, track $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$:

$$\begin{aligned}\Delta(\boldsymbol{G}_t \boldsymbol{x}_t) &= (\Delta \boldsymbol{G}_t)(\Delta \boldsymbol{x}_t) + \boldsymbol{G}_t(\Delta \boldsymbol{x}_t) + (\Delta \boldsymbol{G}_t)\boldsymbol{x}_t \\ &= \text{drift} + \text{diffusivity} + \text{fixed diffusivity}\,,\end{aligned}$$

with first two terms controlled by choice of $\Delta \boldsymbol{x}_t$. Define

$$\begin{aligned}\mathrm{dr}(B) &= \text{bucket average of drift} \\ \mathrm{var}(B) &= \text{bucket average of } [\text{diffusivity}^2] \\ \mathrm{cov}(B) &= \text{bucket average of } [\text{diffusivity} \cdot (\text{fixed diffusivity})]\end{aligned}$$

As before, **free probability** gives optimal value of

$$\sum_B \left\{ \boldsymbol{\theta}_B \mathrm{dr}(B)^2 + \boldsymbol{\varphi}_B \mathrm{cov}(B)^2 + \boldsymbol{\gamma}_B \mathrm{var}(B) \right\} \leqslant \left\| \textbf{spiked} \text{ random mtx} \right\|_{\mathrm{op}}.$$

Intersection of all $(\boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\gamma})$ constraints is our **budget constraint**.

General (asymmetric) spherical perceptron, recap

Recap: $p : [0,1] \to [0,1]$ increasing differentiable; controlled SDE

$$\begin{cases} dX_t = \sqrt{p'(t)}\; b_t\, dt + \sqrt{p(t) + tp'(t)}\; \sigma_t\, dB_t \\ \text{subject to the } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1. \end{cases}$$

Theorem (HSS 25^+; **general** spherical perceptron)

For the **(asymmetric)** *spherical perceptron, the measures $\mu(\boldsymbol{x})$ achievable by Lipschitz algorithms are precisely of the form* $\mathrm{Law}(X_1)$ *for X satisfying the above controlled SDE.*

In the previous slide we explained the basic proof strategy.
In the next slide we will clarify the role of the p function.

Radius schedule

Recap: $p : [0,1] \to [0,1]$ increasing differentiable; controlled SDE

$$\begin{cases} dX_t = \sqrt{p'(t)}\; b_t \, dt + \sqrt{p(t) + tp'(t)}\; \sigma_t \, dB_t \\ \text{subject to the } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1. \end{cases}$$

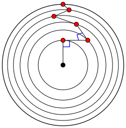
Radius schedule

Recap: $p : [0,1] \to [0,1]$ increasing differentiable; controlled SDE

$$\begin{cases} dX_t = \sqrt{p'(t)}\ b_t\, dt + \sqrt{p(t) + tp'(t)}\ \sigma_t\, dB_t \\ \text{subject to the } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1. \end{cases}$$

Interpretation of p: $p = \chi^{-1}$ where $\chi(t)$ is "radius schedule" of the Doob martingale $\boldsymbol{x}_t$:

$$\chi(t) = \frac{\mathbb{E}[\|\boldsymbol{x}_t\|^2]}{N}.$$



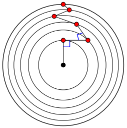
Radius schedule

Recap: $p : [0,1] \to [0,1]$ increasing differentiable; controlled SDE

$$\begin{cases} dX_t = \sqrt{p'(t)}\ b_t\, dt + \sqrt{p(t) + tp'(t)}\ \sigma_t\, dB_t \\ \text{subject to the } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1. \end{cases}$$

Interpretation of p: $p = \chi^{-1}$ where $\chi(t)$ is "radius schedule" of the Doob martingale $\boldsymbol{x}_t$:

$$\chi(t) = \frac{\mathbb{E}[\|\boldsymbol{x}_t\|^2]}{N}.$$



This is a key input to description of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t / \sqrt{N})$.

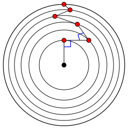
Radius schedule

Recap: $p : [0,1] \to [0,1]$ increasing differentiable; controlled SDE

$$\begin{cases} dX_t = \sqrt{p'(t)}\ b_t\, dt + \sqrt{p(t) + tp'(t)}\ \sigma_t\, dB_t \\ \text{subject to the } \textbf{budget constraint} \\ \mathbb{E}\left[b_t^2 + \dfrac{p(t) + tp'(t)}{p(t)} (\sigma_t - 1)^2 \right] \leqslant \dfrac{1}{\alpha} \text{ for all } 0 \leqslant t \leqslant 1. \end{cases}$$

Interpretation of p: $p = \chi^{-1}$ where $\chi(t)$ is "radius schedule" of the Doob martingale $\boldsymbol{x}_t$:

$$\chi(t) = \frac{\mathbb{E}[\|\boldsymbol{x}_t\|^2]}{N}\,.$$



This is a key input to description of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t \boldsymbol{x}_t/\sqrt{N})$.
If consider $\mu^{\mathrm{sym}}(\boldsymbol{x})$, suffices to take $b_t \equiv 0$ and $\chi \equiv 0$, i.e. $p \equiv 1$.

Ising perceptron

For Ising perceptron, we need to track joint evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t \boldsymbol{x}_t}{\sqrt{N}}\right), \quad \nu_t = \mathrm{EmpDist}(\boldsymbol{x}_t)\,.$$

Ising perceptron

For Ising perceptron, we need to track joint evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t \boldsymbol{x}_t}{\sqrt{N}}\right), \quad \nu_t = \mathrm{EmpDist}(\boldsymbol{x}_t)\,.$$

Symmetric case: $X_0 = Y_0 = 0$, $|Y_1| = 1$ a.s.,

$$\begin{cases} dX_t = \sigma_t\, dB_t,\ dY_t = w_t\, d\tilde{B}_t \text{ subj to } \textbf{budget constraint} \\ \mathbb{E}[(\sigma_t - 1)^2] \leqslant \dfrac{\mathbb{E}[w_t]^2}{\alpha},\ \mathbb{E}[w_t^2] = 1 \text{ for all } 0 \leqslant t \leqslant 1\,. \end{cases}$$

Ising perceptron

For Ising perceptron, we need to track joint evolution of

$$\mu_t = \mathrm{EmpDist}\left(\frac{\boldsymbol{G}_t \boldsymbol{x}_t}{\sqrt{N}}\right), \quad \nu_t = \mathrm{EmpDist}(\boldsymbol{x}_t)\,.$$

Symmetric case: $X_0 = Y_0 = 0$, $|Y_1| = 1$ a.s.,

$$\begin{cases} dX_t = \sigma_t \, dB_t,\ dY_t = w_t \, d\tilde{B}_t \text{ subj to } \textbf{budget constraint} \\ \mathbb{E}[(\sigma_t - 1)^2] \leqslant \dfrac{\mathbb{E}[w_t]^2}{\alpha},\ \mathbb{E}[w_t^2] = 1 \text{ for all } 0 \leqslant t \leqslant 1\,. \end{cases}$$

Theorem (HSS25$^+$, **symmetric** Ising perceptron)

For the Ising perceptron, the measures $\mu^{\mathrm{sym}}(\boldsymbol{x})$ *achievable by Lipschitz algorithms are precisely of the form* $\mathrm{Law}(|X_1|)$ *for* (X, Y) *satisfying the above controlled SDE.*

Analogous result for **(asymmetric)** Ising perceptron, omitted here.

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

Proof ideas for hardness, recap and generalization

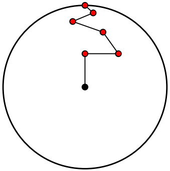
Optimal algorithms via IAMP

Applications and conclusion

Recap: Doob martingale, radius schedule

Recall: Lipschitz algorithm $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$.

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

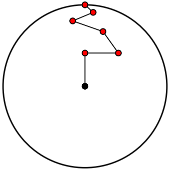


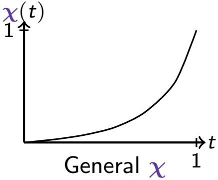
Recap: Doob martingale, radius schedule

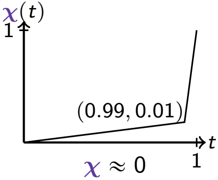
Recall: Lipschitz algorithm $\mathcal{A} : \boldsymbol{G} \mapsto \boldsymbol{x}$.

- **Matrix Brownian motion** $\boldsymbol{G}_t$ with $\boldsymbol{G}_1 = \boldsymbol{G}$.
- **Doob martingale** $\boldsymbol{x}_t = \mathbb{E}[\mathcal{A}(\boldsymbol{G}_1)|\boldsymbol{G}_t]$ with $\boldsymbol{x}_1 = \boldsymbol{x} = \mathcal{A}(\boldsymbol{G})$.

Process $\boldsymbol{x}_t$ has radius schedule $\chi(t) = \mathbb{E}(\|\boldsymbol{x}_t\|^2)/N = p^{-1}(t)$.







Matching IAMP algorithms

AMP: class of first-order algorithms

$$\begin{aligned} \boldsymbol{m}^{k+1} &= \boldsymbol{G}^{\top} f(\boldsymbol{n}^k, \ldots, \boldsymbol{n}^1) - \mathbf{ons}_{m,k} \,, \\ \boldsymbol{n}^k &= \boldsymbol{G} g(\boldsymbol{m}^k, \ldots, \boldsymbol{m}^1) - \mathbf{ons}_{n,k} \,. \end{aligned}$$

Limiting behavior well understood (Bolthausen 12, Bayati Montanari 11, many subsequent extensions).

Matching IAMP algorithms

AMP: class of first-order algorithms

$$\boldsymbol{m}^{k+1} = \boldsymbol{G}^\top f(\boldsymbol{n}^k, \ldots, \boldsymbol{n}^1) - \mathbf{ons}_{m,k}\,,$$
$$\boldsymbol{n}^k = \boldsymbol{G} g(\boldsymbol{m}^k, \ldots, \boldsymbol{m}^1) - \mathbf{ons}_{n,k}\,.$$

Limiting behavior well understood (Bolthausen 12, Bayati Montanari 11, many subsequent extensions).

IAMP (Montanari 19, Montanari Zhou 24): for iterates $\boldsymbol{m}^k$, $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{m}^k/\sqrt{N})$ simulates SDE with radius schedule $\chi \approx 0$, i.e. $p \approx 1$. Matches hardness result for symmetric perceptron.

Matching IAMP algorithms

AMP: class of first-order algorithms

$$\boldsymbol{m}^{k+1} = \boldsymbol{G}^\top f(\boldsymbol{n}^k, \dots, \boldsymbol{n}^1) - \mathbf{ons}_{m,k}\,,$$
$$\boldsymbol{n}^k = \boldsymbol{G} g(\boldsymbol{m}^k, \dots, \boldsymbol{m}^1) - \mathbf{ons}_{n,k}\,.$$

Limiting behavior well understood (Bolthausen 12, Bayati Montanari 11, many subsequent extensions).

IAMP (Montanari 19, Montanari Zhou 24): for iterates $\boldsymbol{m}^k$, EmpDist$(\boldsymbol{G}\boldsymbol{m}^k/\sqrt{N})$ simulates SDE with radius schedule $\chi \approx 0$, i.e. $p \approx 1$. Matches hardness result for symmetric perceptron.

Our new IAMP simulates SDE for general $\chi = p^{-1}$: k-th step uses $\boldsymbol{G}_{t(k)}$, where $\boldsymbol{G}_t$ is matrix Brownian motion from hardness analysis. Matches hardness result for general (asymmetric) perceptron.

Outline of talk:

Introduction

Main result: algorithmic threshold

Proof ideas for hardness, a simple case

Proof ideas for hardness, recap and generalization

Optimal algorithms via IAMP

Applications and conclusion

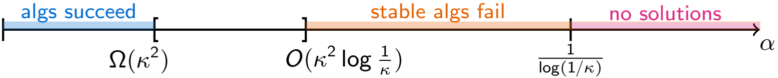
Algorithmic bounds in constraint satisfaction settings

Constraint satisfaction: find $\boldsymbol{x}$ with $\boldsymbol{G}\boldsymbol{x}/\sqrt{N} \in \boldsymbol{I}$ coordinatewise.

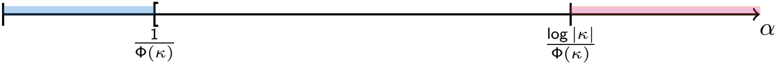
Algorithmic bounds in constraint satisfaction settings

Constraint satisfaction: find $\boldsymbol{x}$ with $\boldsymbol{G}\boldsymbol{x}/\sqrt{N} \in \boldsymbol{I}$ coordinatewise.

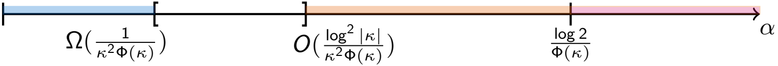
Symmetric Ising perceptron: $\boldsymbol{I} = [-\kappa, \kappa]$, $\kappa \ll 1$ Bansal Spencer 20
Gamarnik Perkins Kızıldağ Xu 22



Negative spherical perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Montanari Zhong Zhou 21



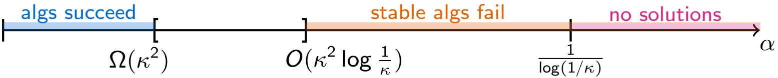
Negative Ising perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Li Schramm Zhou 24



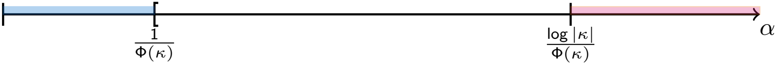
Algorithmic bounds in constraint satisfaction settings

Constraint satisfaction: find $\boldsymbol{x}$ with $\boldsymbol{G}\boldsymbol{x}/\sqrt{N} \in \boldsymbol{I}$ coordinatewise.

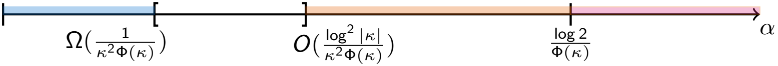
Symmetric Ising perceptron: $\boldsymbol{I} = [-\kappa, \kappa]$, $\kappa \ll 1$ Bansal Spencer 20
Gamarnik Perkins Kızıldağ Xu 22



Negative spherical perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Montanari Zhong Zhou 21



Negative Ising perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Li Schramm Zhou 24

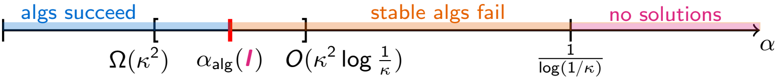


We identify $\alpha_{\text{alg}}(\boldsymbol{I})$ **exactly** in all these models (in terms of SDE).

Algorithmic bounds in constraint satisfaction settings

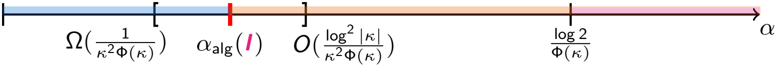
Constraint satisfaction: find $\boldsymbol{x}$ with $\boldsymbol{G}\boldsymbol{x}/\sqrt{N} \in \boldsymbol{I}$ coordinatewise.

Symmetric Ising perceptron: $\boldsymbol{I} = [-\kappa, \kappa]$, $\kappa \ll 1$ Bansal Spencer 20
Gamarnik Perkins Kızıldağ Xu 22



Negative spherical perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Montanari Zhong Zhou 21

Negative Ising perceptron: $\boldsymbol{I} = [\kappa, \infty)$, $\kappa \ll 0$ Li Schramm Zhou 24



We identify $\alpha_{\mathrm{alg}}(\boldsymbol{I})$ **exactly** in all these models (in terms of SDE).

Algorithmic thresholds in twelve asymptotic regimes

Our main theorem implies SDE description of $\alpha_{\mathsf{alg}}(\boldsymbol{I})$.

Algorithmic thresholds in twelve asymptotic regimes

Our main theorem implies SDE description of $\alpha_{\mathsf{alg}}(\boldsymbol{I})$. Can analyze to obtain asymptotic behavior of algorithmic thresholds:

		spherical	Ising
$\boldsymbol{I} = [-\kappa, \kappa]$	$\kappa \ll 1$	$1 + \Theta(\kappa)$	$\Theta(\kappa^2 \log(\kappa^{-1}))$
(symmetric)	$\kappa \gg 1$	$\Theta(\Phi(\kappa)^{-1})$	$\Theta(\Phi(\kappa)^{-1}\kappa^{-2} \log \kappa)$
$\boldsymbol{I} = \mathbb{R}\backslash[-\kappa, \kappa]$	$\kappa \ll 1$	$\kappa^{-1+o(1)}$	$\Theta(1)$
(symmetric)	$\kappa \gg 1$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
$\boldsymbol{I} = [\kappa, \infty)$	$\kappa \gg 0$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
(asymmetric)	$\kappa \ll 0$	$\Theta(\Phi(\lvert\kappa\rvert)^{-1})$	$\Theta(\Phi(\lvert\kappa\rvert)^{-1}\kappa^{-2} \log \lvert\kappa\rvert)$

where $\Phi = 1 -$ gaussian cdf.

Algorithmic thresholds in twelve asymptotic regimes

Our main theorem implies SDE description of $\alpha_{\text{alg}}(I)$. Can analyze to obtain asymptotic behavior of algorithmic thresholds:

		spherical	Ising
$I = [-\kappa, \kappa]$	$\kappa \ll 1$	$1 + \Theta(\kappa)$	$\Theta(\kappa^2 \log(\kappa^{-1}))$
(symmetric)	$\kappa \gg 1$	$\Theta(\Phi(\kappa)^{-1})$	$\Theta(\Phi(\kappa)^{-1}\kappa^{-2} \log \kappa)$
$I = \mathbb{R} \backslash [-\kappa, \kappa]$	$\kappa \ll 1$	$\kappa^{-1+o(1)}$	$\Theta(1)$
(symmetric)	$\kappa \gg 1$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
$I = [\kappa, \infty)$	$\kappa \gg 0$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
(asymmetric)	$\kappa \ll 0$	$\Theta(\Phi(\|\kappa\|)^{-1})$	$\Theta(\Phi(\|\kappa\|)^{-1}\kappa^{-2} \log \|\kappa\|)$

where $\Phi = 1 -$ gaussian cdf. One of our Ising results closes a gap between Bansal Spencer 20 and Gamarnik Perkins Kızıldağ Xu 22;

Algorithmic thresholds in twelve asymptotic regimes

Our main theorem implies SDE description of $\alpha_{\text{alg}}(I)$. Can analyze to obtain asymptotic behavior of algorithmic thresholds:

		spherical	Ising
$I = [-\kappa, \kappa]$	$\kappa \ll 1$	$1 + \Theta(\kappa)$	$\Theta(\kappa^2 \log(\kappa^{-1}))$
(symmetric)	$\kappa \gg 1$	$\Theta(\Phi(\kappa)^{-1})$	$\Theta(\Phi(\kappa)^{-1}\kappa^{-2} \log \kappa)$
$I = \mathbb{R} \backslash [-\kappa, \kappa]$	$\kappa \ll 1$	$\kappa^{-1+o(1)}$	$\Theta(1)$
(symmetric)	$\kappa \gg 1$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
$I = [\kappa, \infty)$	$\kappa \gg 0$	$(1 + o(1))\kappa^{-2}$	$(2/\pi + o(1))\kappa^{-2}$
(asymmetric)	$\kappa \ll 0$	$\Theta(\Phi(\|\kappa\|)^{-1})$	$\Theta(\Phi(\|\kappa\|)^{-1}\kappa^{-2} \log \|\kappa\|)$

where $\Phi = 1 -$ gaussian cdf. One of our Ising results closes a gap between Bansal Spencer 20 and Gamarnik Perkins Kızıldağ Xu 22; another closes a gap in Li Schramm Zhou 24.

Conclusion

- We obtain a sharp characterization (via controlled SDEs) of the measures $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by Lipschitz algorithms.

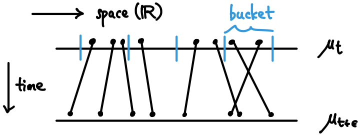
- Implies algorithmic threshold in optimization and constraint satisfaction problems.

Conclusion

- We obtain a sharp characterization (via controlled SDEs) of the measures $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by Lipschitz algorithms.

- Implies algorithmic threshold in optimization and constraint satisfaction problems.

- Hardness: SDE tracks evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t\boldsymbol{x}_t/\sqrt{N})$.



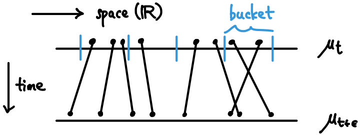
- Algorithms: new IAMP algorithm follows same SDE.

Conclusion

• We obtain a sharp characterization (via controlled SDEs) of the measures $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by Lipschitz algorithms.

• Implies algorithmic threshold in optimization and constraint satisfaction problems.

• Hardness: SDE tracks evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t\boldsymbol{x}_t/\sqrt{N})$.



• Algorithms: new IAMP algorithm follows same SDE.

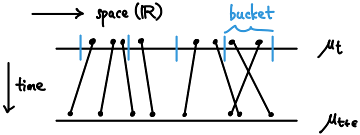
• Open problem: sharp algorithmic threshold in **sparse** random search/optimization problems, e.g. random (max)-k-SAT.

Conclusion

- We obtain a sharp characterization (via controlled SDEs) of the measures $\mathrm{EmpDist}(\boldsymbol{G}\boldsymbol{x}/\sqrt{N})$ achievable by Lipschitz algorithms.

- Implies algorithmic threshold in optimization and constraint satisfaction problems.

- Hardness: SDE tracks evolution of $\mu_t = \mathrm{EmpDist}(\boldsymbol{G}_t\boldsymbol{x}_t/\sqrt{N})$.



- Algorithms: new IAMP algorithm follows same SDE.

- Open problem: sharp algorithmic threshold in **sparse** random search/optimization problems, e.g. random (max)-k-SAT.

Thank you!