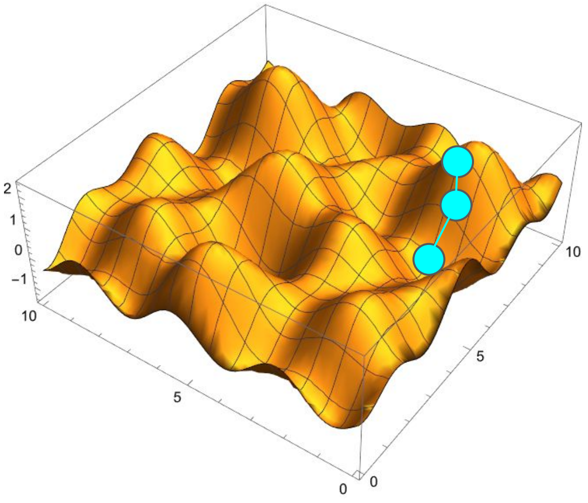
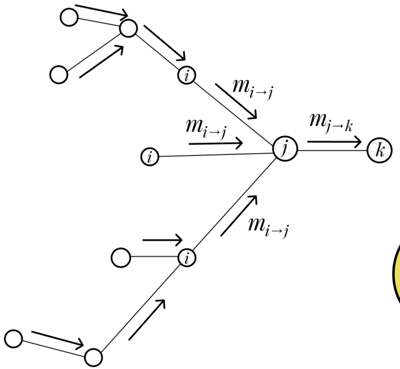
Treelike Constant-Time Dynamics for General First-Order Methods



F(x)



Chris Jones

(Bocconi University □ UC Davis)

Outline

1. The problem of effective dynamics
2. Approach via low-degree polynomials
3. Treelike dynamics

available: [Jones-Pesenti ’25] for GOE setting
in progress: [Gorini-Jones-Kunisky-Pesenti ’25+]

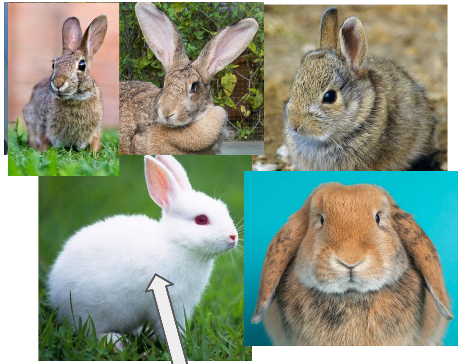
Theorist's view of ML

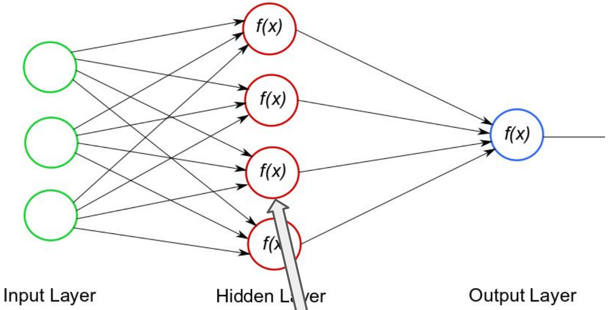
The three ingredients of machine learning:

1. Training data

2. Choice of model to fit (e.g. neural network)

3. Training algorithm





$$\mathbf{\Theta}^{t+1} = \mathbf{\Theta}^t - \alpha \nabla F(\mathbf{\Theta}^t), \quad t \geq 0$$

$$F(\mathbf{\Theta}) = \frac{1}{2} \sum_{i=1}^{N} (\langle x_i, \mathbf{\Theta} \rangle - y_i)^2$$

Input

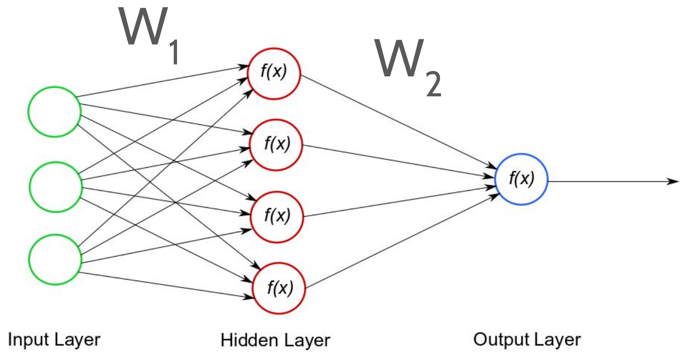
State of the algorithm

The algorithm

Run the algorithm on the data to iteratively update the model

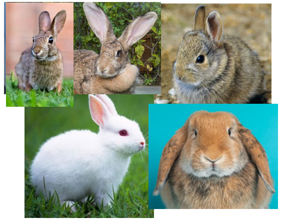
Theorist's view of ML

Viewpoint: the model evolves through training as a high-dimensional dynamical system



Initially:

(W_1, W_2) ~ random weight matrices



$$\boldsymbol{\Theta}^{t+1} = \boldsymbol{\Theta}^t - \alpha \nabla F(\boldsymbol{\Theta}^t), \quad t \geq 0$$

$$F(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{i=1}^{N}(\langle \mathbf{x}_i, \boldsymbol{\Theta}\rangle - y_i)^2$$

(W_1, W_2) follow SGD dynamics on input

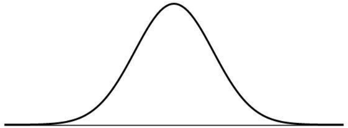
Central question: what is the explicit trajectory of (W_1, W_2)?

Theorist's view of ML

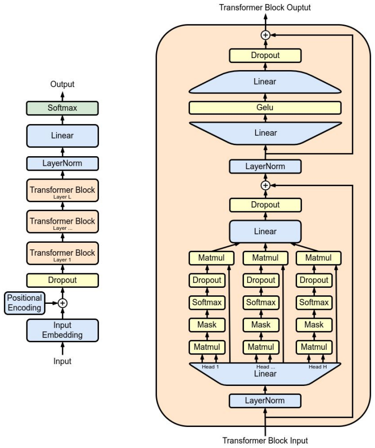
1. The input is not worst-case, like in complexity theory

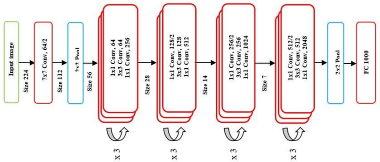
    Instead modeled as random / average-case / statistical

```
>>> The cat takes a nap.
>>> I eat an apple.
```



2. Model design: the "main" challenge





3. The algorithm is usually a simple iterative optimization algorithm

Physicist's view of ML

Physicists have studied dynamical systems of particles for ≥2 years

Simple interaction rules ⇔ simple iterative algorithm
Evolution of particles ⇔ algorithm's trajectory
Energy minimization ⇔ gradient descent
Large number of particles (statistical physics) ⇔ large high-dim data, large models

Key physical insight: large random systems exhibit effective dynamics

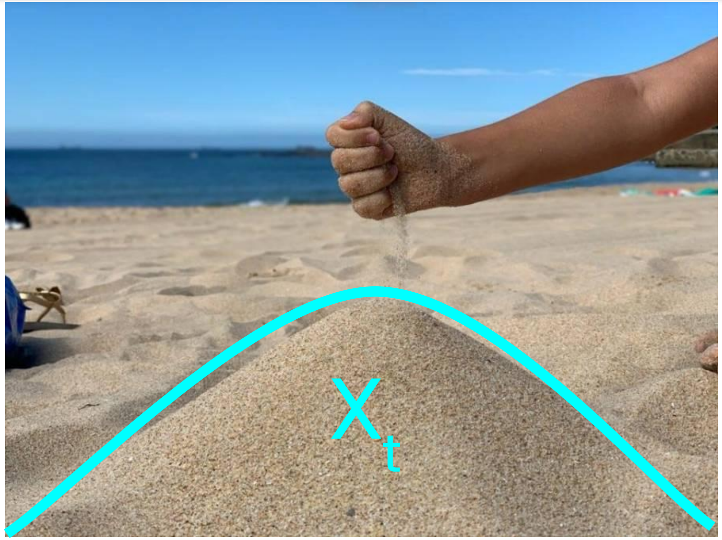
Effective dynamics metatheorem

As the size of a random, smoothly-interacting dynamical system $n \to \infty$, the effect of individual particles "averages out", so that the dynamical system's trajectory approximately follows an asymptotic distributional equation

Effective dynamics

Effective dynamics metatheorem

As the size of a random, smoothly-interacting dynamical system $n \to \infty$, the effect of individual particles "averages out", so that the dynamical system's trajectory approximately follows an asymptotic distributional equation



1. *Effective dynamics*

The problem of effective dynamics

1. (Existence) What assumptions on the algorithm and the input imply that the algorithm's state has effective dynamics X_t as $n \to \infty$?
2. (Universality) What parameters of the input characterize X_t?
3. (Calculation and analysis) What is X_t? What is $\lim_{t \to \infty} X_t$?

General First-Order Methods (GFOM) [CMW'20]

Input: $A \in \mathbb{R}^{n \times n}$

General First-Order Methods (GFOM)

Iteratively compute $\mathbf{x}_t \in \mathbb{R}^n$ via two allowed operations ($\mathbf{x}_0 = \vec{1}$):

1. Multiply by A: $\mathbf{x}_{t+1} = A\mathbf{x}_t$
2. Apply componentwise nonlinearity:

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \ldots, \mathbf{x}_0)$$

Linear operation

Nonlinear operation

Polynomial GFOM: the nonlinearities f_t are polynomials

Includes Approximate Message Passing (AMP)

1. *Effective dynamics*

[Celentano Montanari Wu '20]

Results

We study existence, universality, and explicit computation of effective dynamics X_t for GFOM using low-degree polynomial techniques

Theorem [JP'25, GJKP'25+].
Let $A \in \mathbb{R}^{n\times n}$ be an orthogonally-invariant random matrix with $\|A\| < O(1)$.
Let $\mathbf{x}_t \in \mathbb{R}^n$ be polynomial GFOM iterates, $t=O(1)$, and E_t = empirical r.v. of $\mathbf{x}_t$.
Then $E_t \to X_t$ in distribution where X_t is the treelike asymptotic state.

Corollary: new proof of Orthogonal AMP state evolution + derivation of Onsager correction

Theorem-in-progress [GJKP'25+].
Let $A \in \mathbb{R}^{n\times n}$ be the Walsh-Hadamard, discrete sine transform, or discrete cosine transform matrix (with first row+column deleted). Let $\mathbf{x}_t$ be polynomial GFOM iterates, $t = O(1)$. Then X_t exists and matches the regular ROM.

Outline

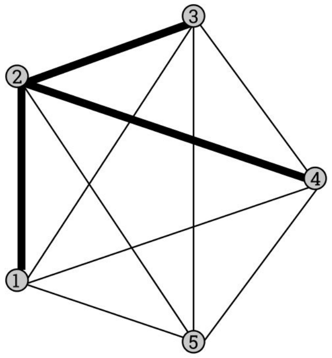
1. The problem of effective dynamics
2. Approach via low-degree polynomials
3. Treelike dynamics

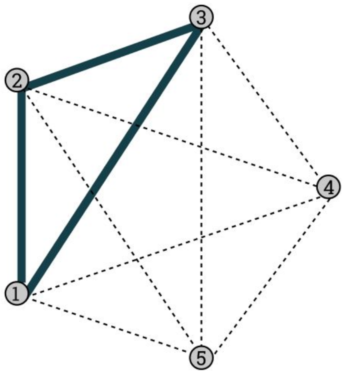
Algorithms as polynomials

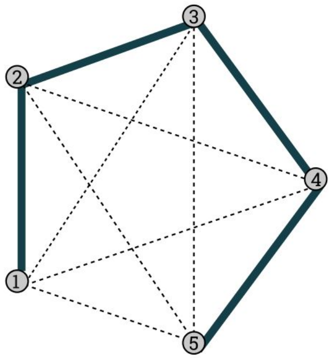
We analyze algorithms by expressing them as multivariate polynomials in the input $A \in \mathbb{R}^{n \times n}$

Monomials in $A \in \mathbb{R}^{n \times n}$ correspond to graphs on {1,2,...,n}

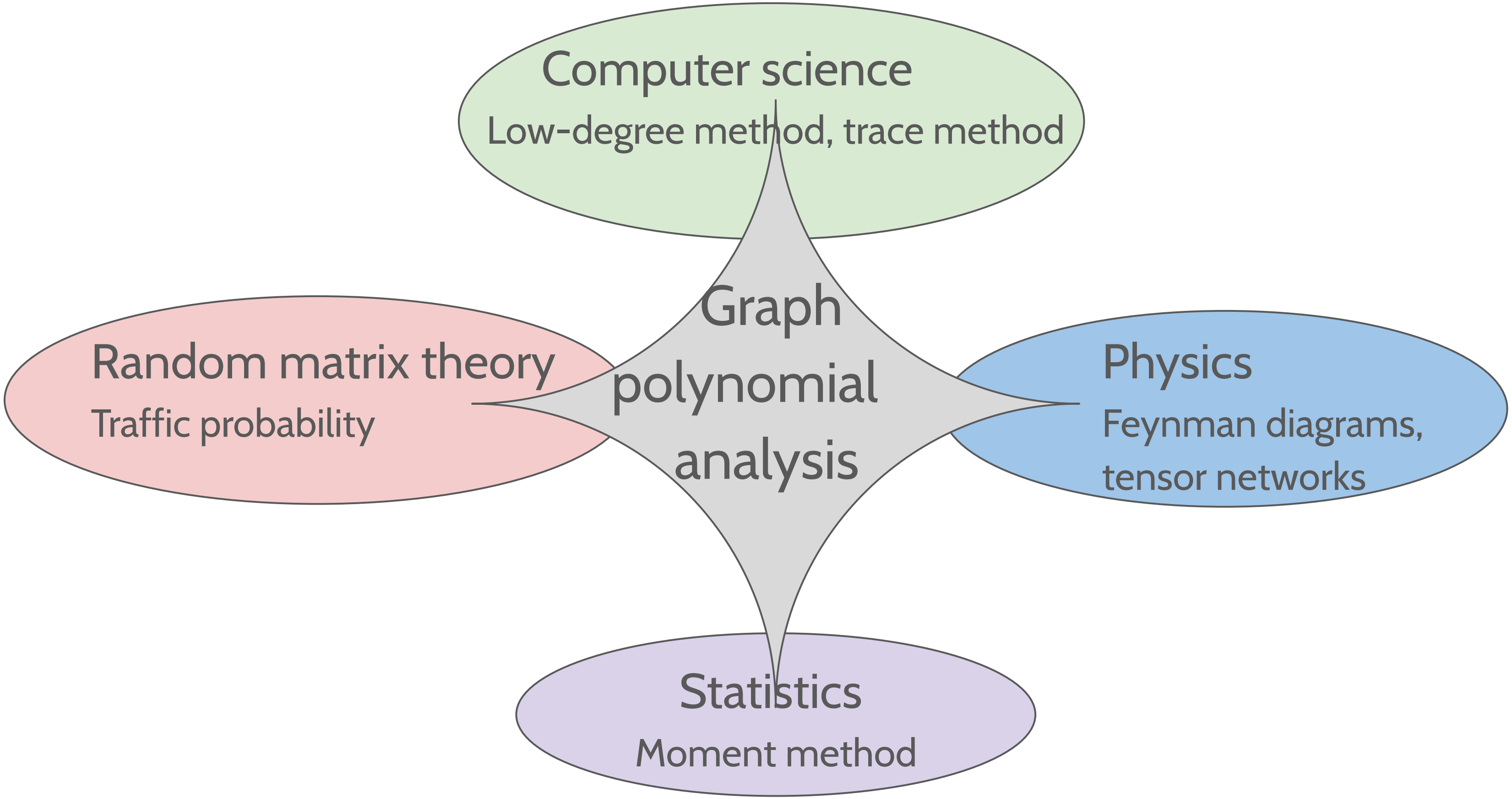
$$A_{12}A_{23}A_{24} \quad + 2 \quad A_{12}A_{23}A_{13} \quad - \quad A_{12}A_{23}A_{34}A_{45}$$







Polynomial analysis: a unified approach

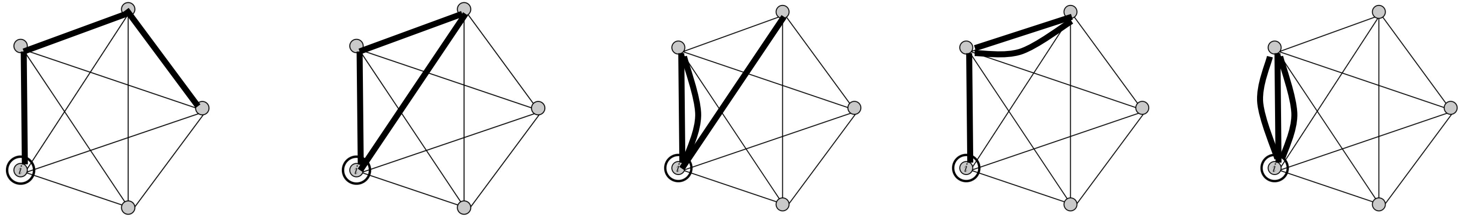


Algorithms as polynomials

Input: $A \in \mathbb{R}^{n \times n}$

<u>Warm-up</u> $\mathbf{x}_t = A^t \vec{1}$ (matrix power iteration)

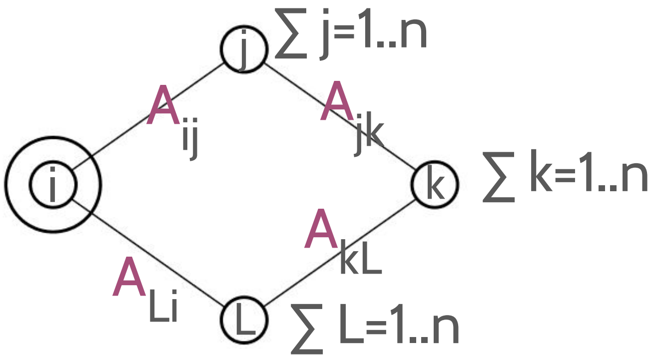
$$(\mathbf{x}_3)_i = \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{L=1}^{n} A_{ij}A_{jk}A_{kL} = \sum_{\substack{j,k,L=1 \\ \text{distinct}}}^{n} A_{ij}A_{jk}A_{kL} + \sum_{\substack{j,k,L=1 \\ i=L}}^{n} A_{ij}A_{jk}A_{kL} + \sum_{\substack{j,k,L=1 \\ i=k}}^{n} A_{ij}A_{jk}A_{kL} + \sum_{\substack{j,k,L=1 \\ j=L}}^{n} A_{ij}A_{jk}A_{kL} + \sum_{\substack{j,k,L=1 \\ i=k,j=L}}^{n} A_{ij}A_{jk}A_{kL}$$



Graph polynomials

Def: Given a rooted graph $\alpha = (V,E)$, the graph monomial $Z^{\alpha}(A)$ is the vector in $\mathbb{R}^n$ whose entries are

$$Z^{\alpha}(A)_i = \sum_{\substack{\varphi:\, V \to [n] \\ \text{injective} \\ \varphi(\text{root}) = i}} \prod_{(u,v) \in E} A_{\varphi(u)\varphi(v)}$$



Algorithms as polynomials

Input: $A \in \mathbb{R}^{n \times n}$

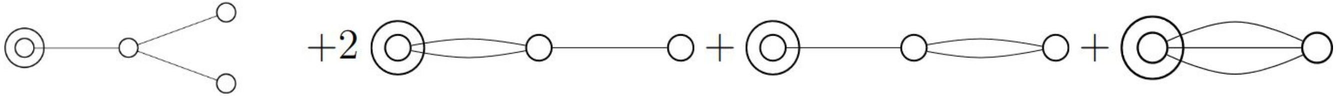
<u>Nonlinear example</u>

$\mathbf{x}_{t+1} = (A\,\mathbf{x}_t)^2$ with the square applied componentwise

$\mathbf{x}_0 = \vec{1}$

$$A(A\vec{1})^2_i = \sum^n_{j=1}\sum^n_{k=1}\sum^n_{L=1} A_{ij}A_{jk}A_{jL}$$

$$= \sum^n_{\substack{j,k,L=1 \\ \text{distinct}}} A_{ij}A_{jk}A_{jL} + 2\sum^n_{\substack{j,k=1 \\ \text{distinct}}} A_{ij}^2 A_{jk} + \sum^n_{\substack{j,k=1 \\ \text{distinct}}} A_{ij}A_{jk}^2 + \sum^n_{j=1} A_{ij}^3$$



Observation: the output $\mathbf{x}_t \in \mathbb{R}^n$ of a GFOM is S_n-equivariant ⇒ all monomials with the same shape have the same coefficient

Algorithms as polynomials

Theorem (universality). If $\mathbf{x}_t \in \mathbb{R}^n$ is a polynomial GFOM, t=O(1), then $\mathbf{x}_t$ can be expressed as a sum of O(1) graph monomials in A of size O(1).

GFOM algorithm

$\mathbf{x}_{t+1} = A\mathbf{x}_t$

or $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \ldots, \mathbf{x}_0)$

applied componentwise

So, the limiting values of the unrooted connected O(1)-size graph monomials $\frac{1}{n}Z^{\alpha}(A)$ are sufficient to specify the universality class of O(1)-time GFOM

- Generalizes the limiting spectral density, whose moments are specified by the cycle graph monomials

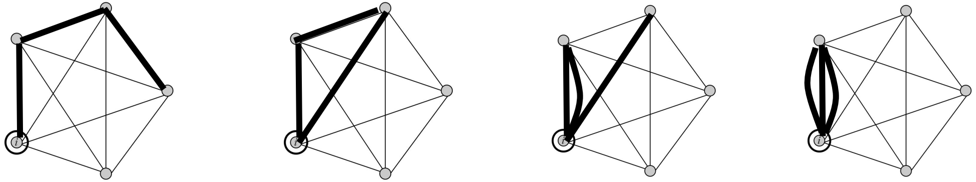
Outline

1. The problem of effective dynamics
2. Approach via low-degree polynomials
3. Treelike dynamics

Summary so far

Given $\mathbf{x}_t$ generated by GFOM, we express $\mathbf{x}_t$ in the basis of graph monomials

$$\mathbf{x}_t = 2\, Z^{\text{3-path}}(A) \quad + Z^{\text{triangle}}(A) \quad -10\, Z^{\alpha}(A) \quad + Z^{\text{triple edge}}(A) \quad + \ldots$$



To compute dynamics, we should compute the graph monomials for A and analyze the representation in this basis during the algorithm

Key idea: For many random matrices A, only the "treelike" graph monomials are asymptotically non-zero random vbls

Treelike dynamics

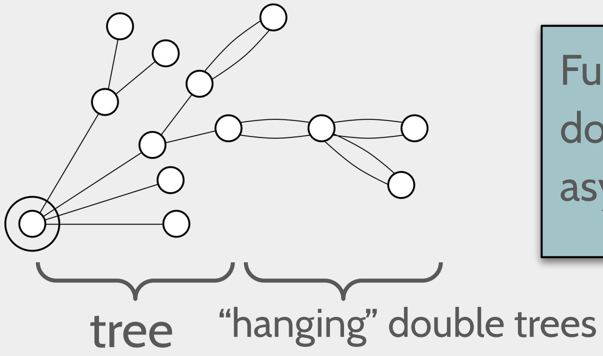
Theorem [JP'25, GJKP'25+].
Let $A \in \mathbb{R}^{n\times n}$ be an orthogonally-invariant random matrix with $||A|| < O(1)$.
Let α be an O(1)-size connected rooted graph. Then the empirical r.v. of $Z^{\alpha}(A) \to 0$ in distribution unless α is a tree with hanging cactuses.

The remaining empirical r.v.s of $Z^{\alpha}(A)$ converge to order 1 random variables.

Can be interpreted as a generalization of the cavity method assumption that BP occurs on a tree, generalized from GOE matrix to orthogonally-invariant random matrices

Tree approximation

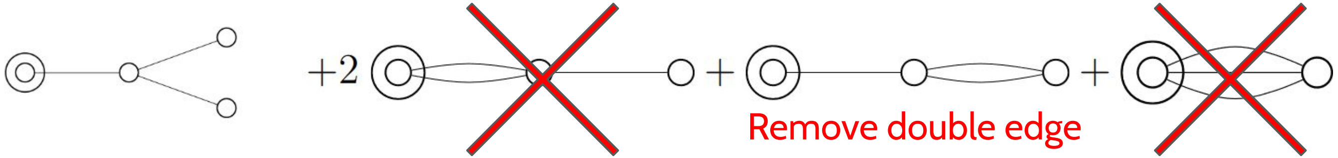
Theorem (classification of diagrams). Whp over $A \sim \mathcal{N}(0, 1/N)^{N\times N}$, the following diagrams are order 1 as $N \to \infty$:



Furthermore, hanging double trees can be asymptotically removed

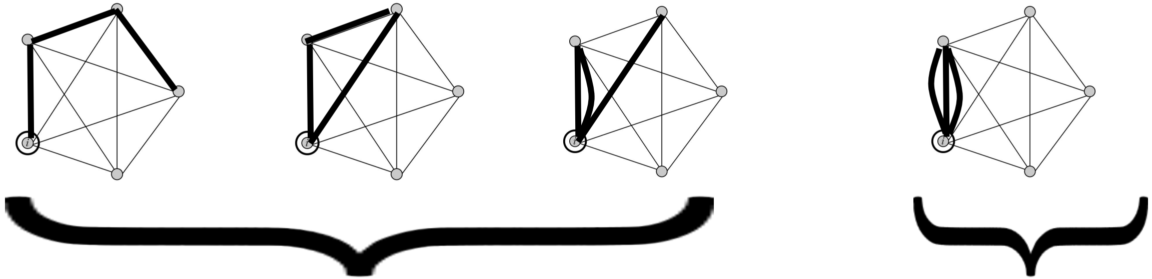
The remaining diagrams are order $1/\sqrt{N}$.

$A(A\,1)^2 =$



Treelike dynamics

$$\mathbf{x}_t = 2\, Z^{3\text{-path}}(A) + Z^{\text{triangle}}(A) - 10\, Z^{\alpha}(A) + Z^{\text{triple edge}}(A) + \ldots$$



Treelike graph monomials
O(1) magnitude

Non-treelike graph monomials
o(1) error terms

Tracking the evolution of the O(1) component is now much simpler because the GFOM is occurring on a tree, in a certain sense

- The treelike asymptotic state follows the evolution of the treelike graph monomials

Computing the graph monomials

To run the method, we need to compute the limiting values of all unrooted connected graph monomials $\frac{1}{n}Z^{\alpha}(A)$ for a given matrix $A \in \mathbb{R}^{n\times n}$

GOE/Wigner matrix
Easy exercise

Orthogonally-invariant random matrix
Either:
(1) Weingarten calculus
(2) Feynman diagram expansion, as in φ^4 matrix model. The treelike graph monomial limit is related to the 't Hooft limit

Walsh-Hadamard, DCT, DST matrix
Extend "fundamental theorem of graph monomials" from traffic probability

General recipe for effective dynamics

1. Compute all of the graph monomials $\frac{1}{n} Z^{\alpha}(A)$ for unrooted connected graphs α

2. Invert the “moment problem” to obtain empirical r.v.s

    i.e. asymptotics of random vectors $Z^{\alpha}(A)$ for rooted graphs α

    Are only treelike graphs non-zero? 🌳

😎YOU ARE NOW IN ASYMPTOTIC SPACE😎

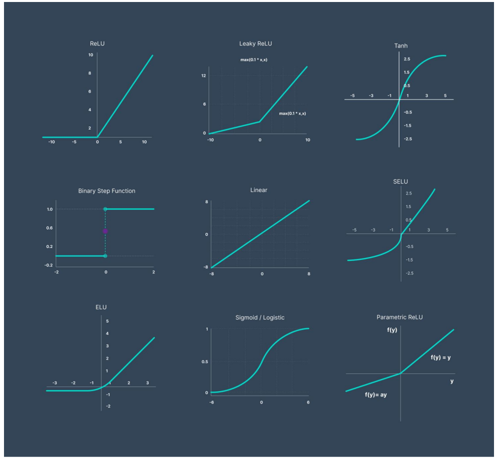
3. Analyze the algorithm’s trajectory through the asymptotic probability space

Conclusion

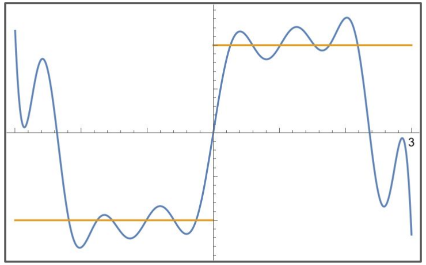
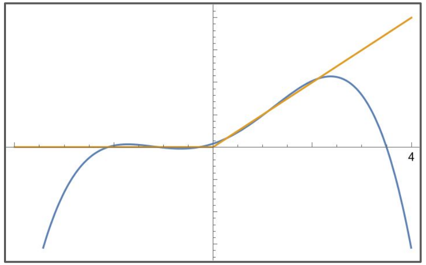
- We study the effective dynamics of GFOM using the tool of graph polynomials
- Natural route for proving existence and universality of dynamics, connections to physics and free cumulants
- For orthogonally-invariant random matrices, dynamics are asymptotically treelike
- Treelike dynamics derives the Onsager correction for AMP algorithms as “backtracking terms”

Thanks for listening!

Why low-degree polynomial algorithms?



Most iterative algorithms and neural networks use nonlinearities which are not polynomials

...yet in many cases, the nonlinearities can be approximated by polynomials

[Image from V7 Labs blog]

Some history

Low-degree algorithms: output = low-degree poly(input)

Low-degree ≈ O(1) or O(log n)

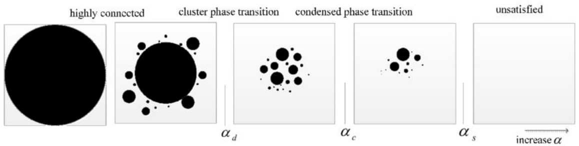
Late 2010s:

development of low-degree likelihood ratio (LDLR)

[Hopkins Steurer '17] [Hopkins '18] [Kunisky Wein Bandeira '19]

$$\max_{f \text{ deg } D} \frac{\mathbb{E}_{Y\sim\mathbb{P}}[f(Y)]}{\sqrt{\mathbb{E}_{Y\sim\mathbb{Q}}[f(Y)^2]}}$$

"Hard" regimes for LDLR match hard phases predicted by physics (low-degree conjecture)



Some history

Thinking about the class of low-degree algorithms originates from the Sum-of-Squares algorithm [Lasserre '01]

The "SoS hierarchy" is also called the "Lasserre hierarchy"

Development of Fourier analysis for Sum-of-Squares algorithms [BHKKMP'19, AMP'20, GJJPR'20, JPRTX'21, PR'22, JP'22, RT'23, JPRX'23, KPX'24]

Ahn
Ghosh
Jeronimo
Jones
Kothari
Medarametla
Potechin
Rajendran
Tulsiani
Xu

This work: use Fourier analytic technology to analyze iterative algorithms including GFOM and BP/AMP

[Barak Hopkins Kelner Kothari Moitra Potechin '19]

Tree approximation