CS 260: Geometry of Polynomials in Algorithm Design

Instructor: Nima Anari

Stanford University

Logistics

- Classroom: Wallenberg (Building 160) Room 314
- Time: Tuesdays and Thursdays, 1:30-2:50pm
- Office hours: Tuesdays after class and by appointment

Logistics

- Classroom: Wallenberg (Building 160) Room 314
- Time: Tuesdays and Thursdays, 1:30-2:50pm
- Office hours: Tuesdays after class and by appointment
- Website: `https://nimaanari.com/cs260-winter2020`

  Website is currently empty because of technical difficulties. ☹

Logistics

- Classroom: Wallenberg (Building 160) Room 314
- Time: Tuesdays and Thursdays, 1:30-2:50pm
- Office hours: Tuesdays after class and by appointment
- Website: `https://nimaanari.com/cs260-winter2020`

  Website is currently empty because of technical difficulties. ☹
- If you have not officially signed up for the class, but would like to receive announcements, email me at `anari@cs.stanford.edu`.

Evaluation

1. Course project and presentation (teams of up to 2)
2. Brief survey of a paper/papers on a topic
3. Two sets of homework (light)

Letter Grade

One of these combinations:

- 1
- 2 + 3

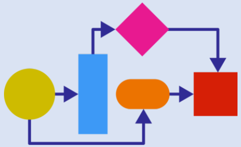
CR

One of these combinations:

- 1
- 2
- 3

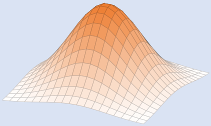
Topics for 1 or 2: suggestions will go on the website; chat with me.

Algorithms



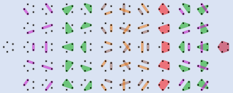
+

Probability

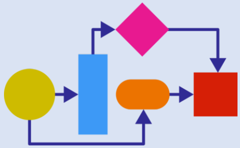


+

Combinatorics

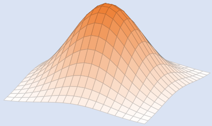


Algorithms



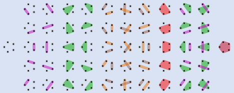
+

Probability



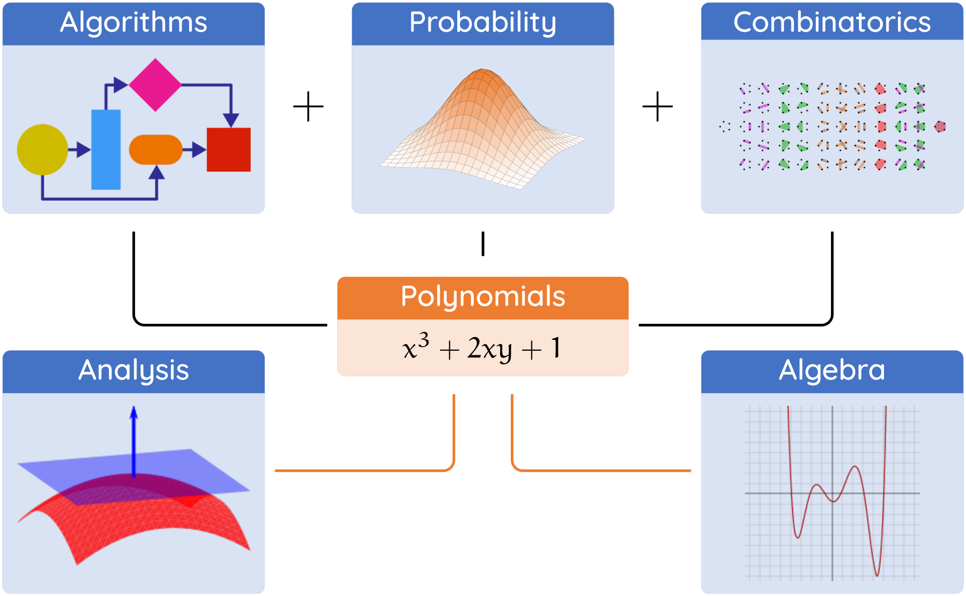
+

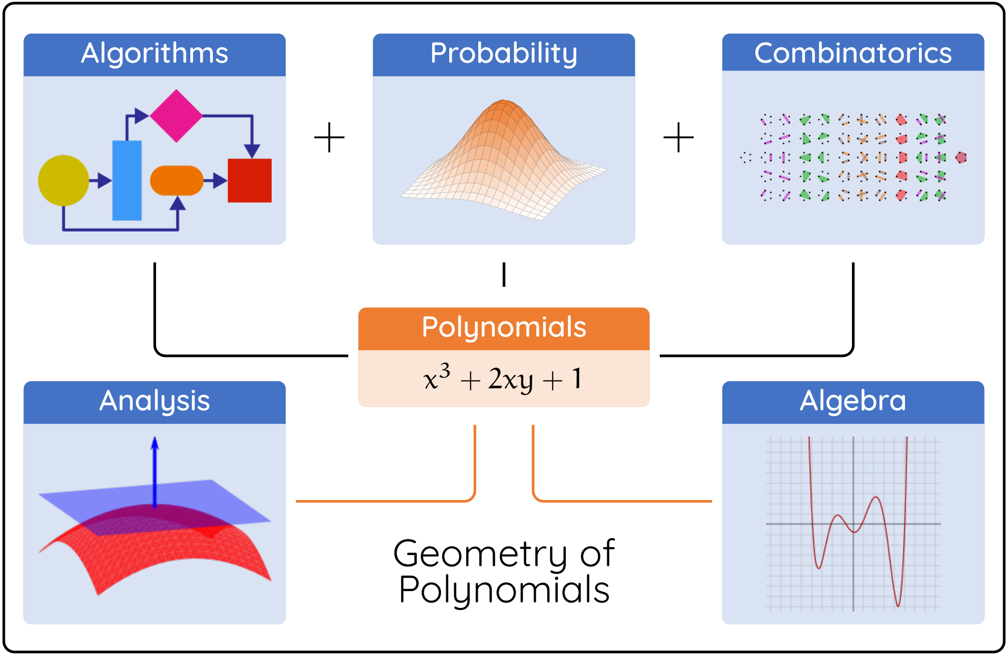
Combinatorics



Polynomials

$$x^3 + 2xy + 1$$


Algorithms
+
Probability
+
Combinatorics
Polynomials
$x^3 + 2xy + 1$
Analysis
Algebra


Algorithms
+
Probability
+
Combinatorics
Polynomials
$x^3 + 2xy + 1$
Analysis
Algebra
Geometry of Polynomials

Main Paradigm

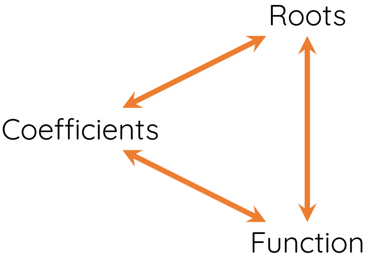
Different ways of looking at a polynomial:

- $g(z) = a_0 + a_1 z + \cdots + a_d z^d$
- $g(z) = c(z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_d)$
- $g : \mathbb{C} \to \mathbb{C}$ or $g : \mathbb{R} \to \mathbb{R}$

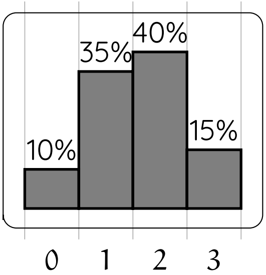
Main Paradigm

Different ways of looking at a polynomial:

- $g(z) = a_0 + a_1 z + \cdots + a_d z^d$
- $g(z) = c(z - \lambda_1)(z - \lambda_2)\cdots(z - \lambda_d)$
- $g : \mathbb{C} \to \mathbb{C}$ or $g : \mathbb{R} \to \mathbb{R}$



Example: Coin Flips $\leftrightarrow$ Univariate Polynomials



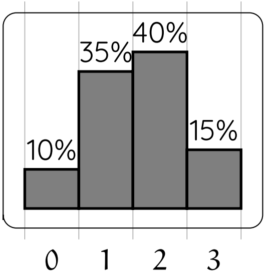
Does this look like #heads dist. in independent coin flips?

Example: Coin Flips $\leftrightarrow$ Univariate Polynomials

Recipe to Verify Coin Flipness

Compute the roots of $g(z)$ and verify that all are real (none are complex).

$$g(z) := 0.10 + 0.35z + 0.40z^2 + 0.15z^3$$



Does this look like #heads dist. in independent coin flips?

Example: Coin Flips $\leftrightarrow$ Univariate Polynomials

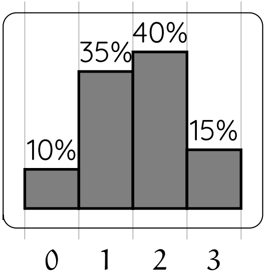
Recipe to Verify Coin Flipness

Compute the roots of $g(z)$ and verify that all are real (none are complex).

$$g(z) := 0.10 + 0.35z + 0.40z^2 + 0.15z^3$$

- $g(z)$ can be factorized:

$$g(z) = \underbrace{(0.5z + 0.5)}_{\text{coin flip}}\underbrace{(0.5z + 0.5)}_{\text{coin flip}}\underbrace{(0.6z + 0.4)}_{\text{coin flip}}$$



Does this look like #heads dist. in independent coin flips?

Example: Coin Flips $\leftrightarrow$ Univariate Polynomials

Recipe to Verify Coin Flipness

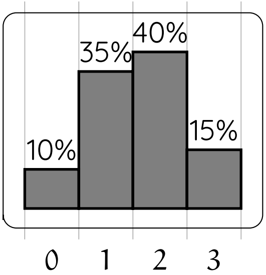
Compute the roots of $g(z)$ and verify that all are real (none are complex).

$$g(z) := 0.10 + 0.35z + 0.40z^2 + 0.15z^3$$

- $g(z)$ can be factorized:

$$g(z) = \underbrace{(0.5z + 0.5)}_{\text{coin flip}} \underbrace{(0.5z + 0.5)}_{\text{coin flip}} \underbrace{(0.6z + 0.4)}_{\text{coin flip}}$$

- When this happens, $\log(g(z))$ becomes concave over $\mathbb{R}_{\geq 0}$.



Does this look like #heads dist. in independent coin flips?

Motivations

Study of polynomials, their root locations, and related properties:

Motivations

Study of polynomials, their root locations, and related properties:

- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, $\ldots$].

Motivations

Study of polynomials, their root locations, and related properties:

- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, $\ldots$].
- **Control Theory**: root-free regions of polynomials $\leftrightarrow$ stability of LTI systems and differential equations [Routh'1876, Hurwitz'1895, Gårding'1959].

Motivations

Study of polynomials, their root locations, and related properties:

- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, . . .].
- **Control Theory**: root-free regions of polynomials $\leftrightarrow$ stability of LTI systems and differential equations [Routh'1876, Hurwitz'1895, Gårding'1959].
- **Continuous Optimization**: hyperbolic programming.

Motivations

Study of polynomials, their root locations, and related properties:

- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, ...].
- **Control Theory**: root-free regions of polynomials $\leftrightarrow$ stability of LTI systems and differential equations [Routh'1876, Hurwitz'1895, Gårding'1959].
- **Continuous Optimization**: hyperbolic programming.
- **Combinatorial Optimization**: produce good-in-expectation answers and control deviation from the expectation (e.g., TSP).

Motivations

Study of polynomials, their root locations, and related properties:

- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, ...].
- **Control Theory**: root-free regions of polynomials $\leftrightarrow$ stability of LTI systems and differential equations [Routh'1876, Hurwitz'1895, Gårding'1959].
- **Continuous Optimization**: hyperbolic programming.
- **Combinatorial Optimization**: produce good-in-expectation answers and control deviation from the expectation (e.g., TSP).
- **Counting and Sampling**: analysis of Markov chains, variational methods, and deterministic expansions.

Motivations

Study of polynomials, their root locations, and related properties:

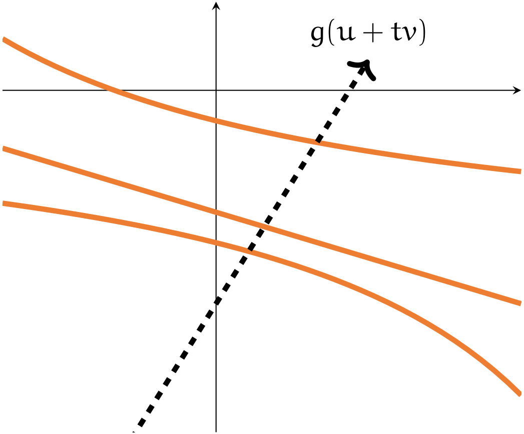
- **Statistical Mechanics**: roots of polynomials $\leftrightarrow$ phase transitions [Lee-Yang'1952, ...].
- **Control Theory**: root-free regions of polynomials $\leftrightarrow$ stability of LTI systems and differential equations [Routh'1876, Hurwitz'1895, Gårding'1959].
- **Continuous Optimization**: hyperbolic programming.
- **Combinatorial Optimization**: produce good-in-expectation answers and control deviation from the expectation (e.g., TSP).
- **Counting and Sampling**: analysis of Markov chains, variational methods, and deterministic expansions.

Main View in this Course

Properties of Polynomials $\leftrightarrow$ Efficiency of Algorithms

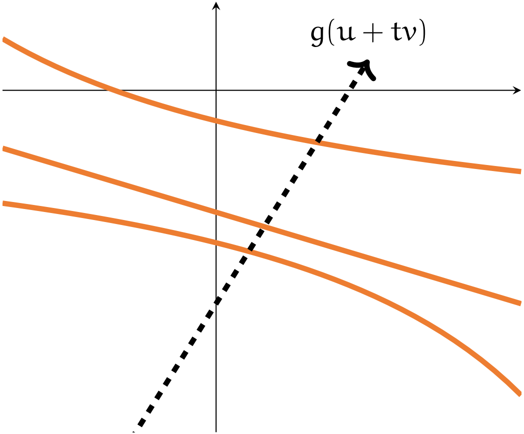
Example: Continuous Optimization

- Hyperbolic polynomial g: for some v and all u, $g(u+tv) \in \mathbb{R}[t]$ is real-rooted.



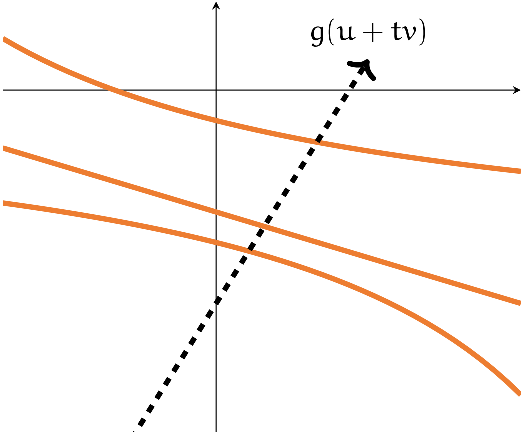
Example: Continuous Optimization

- Hyperbolic polynomial g: for some v and all u, $g(u+tv) \in \mathbb{R}[t]$ is real-rooted.
- Important example: $\det$ for symmetric matrices.



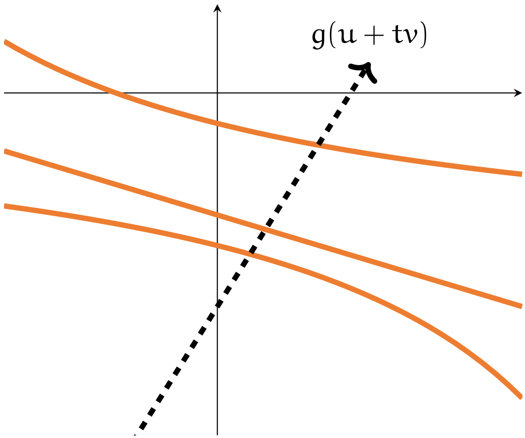
Example: Continuous Optimization

- Hyperbolic polynomial g: for some v and all u, $g(u + tv) \in \mathbb{R}[t]$ is real-rooted.
- Important example: $\det$ for symmetric matrices.
- Hyperbolicity $\implies$ region "above" roots convex.



Example: Continuous Optimization

- Hyperbolic polynomial g: for some v and all u, $g(u + tv) \in \mathbb{R}[t]$ is real-rooted.
- Important example: $\det$ for symmetric matrices.
- Hyperbolicity $\implies$ region "above" roots convex.
- Barrier: $\log(g)$ is a concave function "above" the roots. Basis for hyperbolic programming.



Example: Random Walks

For weighted graph $G = (V, E)$, consider the degree-2 polynomial

$$g(z_{v_1}, \ldots, z_{v_n}) = \sum_{\{u,v\} \in E} w(u, v) z_u z_v$$

Example: Random Walks

For weighted graph $G = (V, E)$, consider the degree-2 polynomial

$$g(z_{v_1}, \ldots, z_{v_n}) = \sum_{\{u,v\} \in E} w(u, v) z_u z_v$$

Log-Concavity $\leftrightarrow$ Mixing

Random walk on G mixes if and only if $\log g(z_{v_1}^{\alpha}, \ldots, z_{v_n}^{\alpha})$ is concave in a neighborhood of $(1, \ldots, 1)$ for some $\alpha > 1/2$.

Example: Random Walks

For weighted graph $G = (V, E)$, consider the degree-2 polynomial

$$g(z_{v_1}, \ldots, z_{v_n}) = \sum_{\{u,v\} \in E} w(u, v) z_u z_v$$

Log-Concavity $\leftrightarrow$ Mixing

Random walk on G mixes if and only if $\log g(z_{v_1}^{\alpha}, \ldots, z_{v_n}^{\alpha})$ is concave in a neighborhood of $(1, \ldots, 1)$ for some $\alpha > 1/2$.

- Generalizes to hypergraphs; high-dimensional expanders.

Example: Random Walks

For weighted graph $G = (V, E)$, consider the degree-2 polynomial

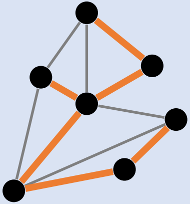
$$g(z_{v_1}, \ldots, z_{v_n}) = \sum_{\{u,v\} \in E} w(u, v) z_u z_v$$

Log-Concavity $\leftrightarrow$ Mixing

Random walk on G mixes if and only if $\log g(z_{v_1}^{\alpha}, \ldots, z_{v_n}^{\alpha})$ is concave in a neighborhood of $(1, \ldots, 1)$ for some $\alpha > 1/2$.

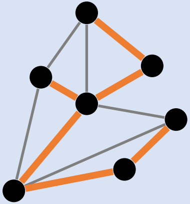
- Generalizes to hypergraphs; high-dimensional expanders.
- Efficient algorithms for sampling from combinatorial distributions.

Spanning Trees



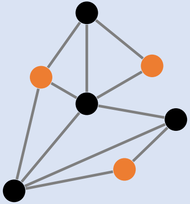
uniformly at random

Spanning Trees



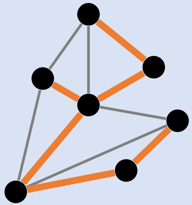
uniformly at random

Stable Sets



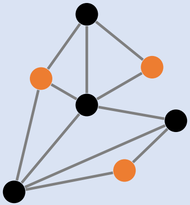
uniformly at random

Spanning Trees



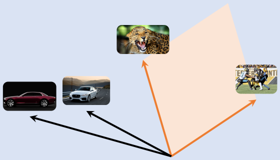
uniformly at random

Stable Sets



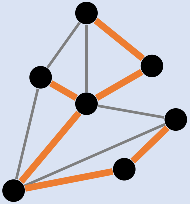
uniformly at random

Volume Based



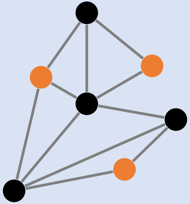
Prob $\propto$ volume

Spanning Trees



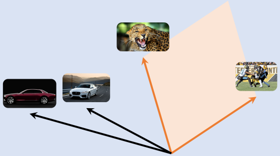
uniformly at random

Stable Sets



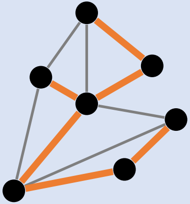
uniformly at random

Volume Based



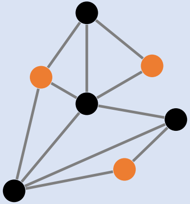
Prob $\propto$ volume

Spanning Trees



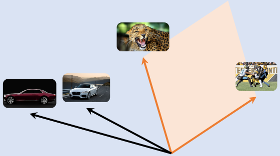
uniformly at random

Stable Sets



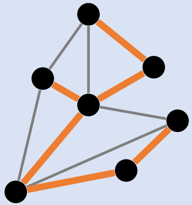
uniformly at random

Volume Based



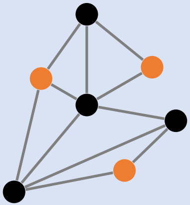
Prob $\propto$ volume

Spanning Trees



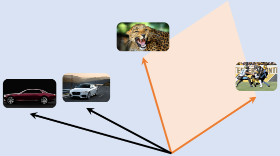
uniformly at random

Stable Sets



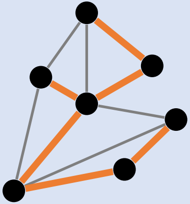
uniformly at random

Volume Based



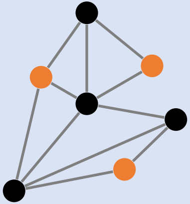
Prob $\propto$ volume

Spanning Trees



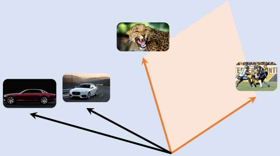
uniformly at random

Stable Sets



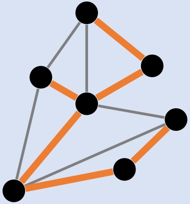
uniformly at random

Volume Based



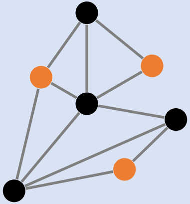
$\text{Prob} \propto \text{volume}^2$

Spanning Trees



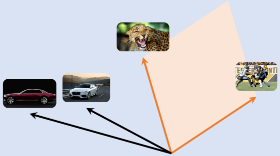
uniformly at random

Stable Sets



uniformly at random

Volume Based



Prob $\propto$ volume^{10}

Example: Gross Substitutes [Kelsey-Crawford'82]

Suppose an agent wants to buy some subset of t-shirts with prices p_1, p_2, p_3:

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Example: Gross Substitutes [Kelsey-Crawford'82]

Suppose an agent wants to buy some subset of t-shirts with prices p_1, p_2, p_3:

- Rational Agent: Buy subset with most utility. Exponentially large table. ☹

	$\$0$
	$\$20 - p_1$
	$\$10 - p_2$
	$\$30 - p_1 - p_2$
	$\$10 - p_3$
	$\$30 - p_1 - p_3$
	$\$10 - p_2 - p_3$
	$\$30 - p_1 - p_2 - p_3$

Example: Gross Substitutes [Kelsey-Crawford'82]

Suppose an agent wants to buy some subset of t-shirts with prices p_1, p_2, p_3:

- Rational Agent: Buy subset with most utility. Exponentially large table. 🙁
- Locally Rational Agent: Add, remove, or replace one t-shirt at a time to improve utility, until no more adjustment possible.

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Example: Gross Substitutes [Kelsey-Crawford'82]

Suppose an agent wants to buy some subset of t-shirts with prices p_1, p_2, p_3:

- Rational Agent: Buy subset with most utility. Exponentially large table.
- Locally Rational Agent: Add, remove, or replace one t-shirt at a time to improve utility, until no more adjustment possible.

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Gross Substitutes

Locally rational agent finds the globally optimal subset.


STANFORD
BERKELEY
STANFORD

Example: Gross Substitutes + Discrete Choice

- Noisily Rational Agent [Nobel Prize: McFadden'00]: Buy S with probability:

$$\mathbb{P}[S] \propto e^{\text{utility}(S)}.$$

Exponentially large lookup table. ☹

	$\$0$
	$\$20 - p_1$
	$\$10 - p_2$
	$\$30 - p_1 - p_2$
	$\$10 - p_3$
	$\$30 - p_1 - p_3$
	$\$10 - p_2 - p_3$
	$\$30 - p_1 - p_2 - p_3$

Example: Gross Substitutes + Discrete Choice

- Noisily Rational Agent [Nobel Prize: McFadden'00]: Buy S with probability:

$$\mathbb{P}[S] \propto e^{\text{utility}(S)}.$$

Exponentially large lookup table. ☹

Theorem

"Random" additions, removals, replacements of one item at a time converge to the true distribution in $\sim O(n \log n)$ steps for gross substitutes.

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Connection to Polynomials

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Connection to Polynomials

- The following multivariate polynomial captures the distribution

$$g(z_1, z_2, z_3) = e^0 + e^{20} z_1 + \cdots + e^{30} z_1 z_2 z_3.$$

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Connection to Polynomials

- The following multivariate polynomial captures the distribution

$$g(z_1, z_2, z_3) = e^0 + e^{20} z_1 + \cdots + e^{30} z_1 z_2 z_3.$$

- This polynomial behaves like real-rooted univariate polynomials. In particular $\log g$ is concave over $\mathbb{R}^n_{\geqslant 0}$.
- Note: For univariate real-rooted polynomials

$$\log((0.5z + 0.5)(0.5z + 0.5)(0.6z + 0.4)) =$$

$$\log(0.5z + 0.5) + \log(0.5z + 0.5) + \log(0.6z + 0.4).$$

$\$0$

$\$20 - p_1$

$\$10 - p_2$

$\$30 - p_1 - p_2$

$\$10 - p_3$

$\$30 - p_1 - p_3$

$\$10 - p_2 - p_3$

$\$30 - p_1 - p_2 - p_3$

Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\text{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

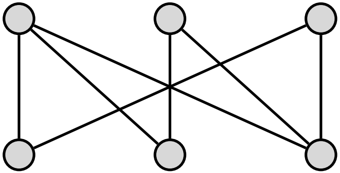
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\text{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.

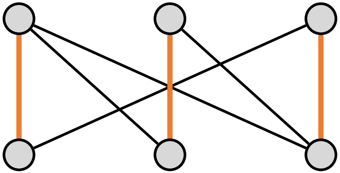
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\mathrm{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.

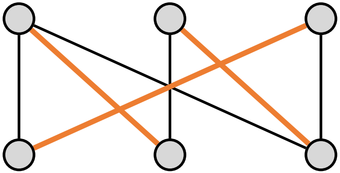
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\mathrm{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.

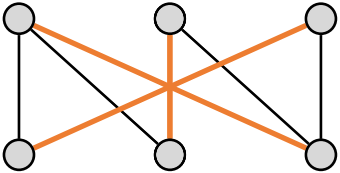
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\mathrm{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.

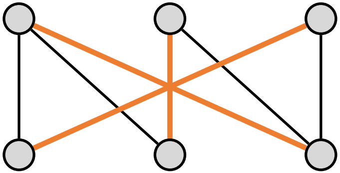
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

$$\mathrm{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.
- #P-hard even for $0/1$ matrices [Valiant'79].

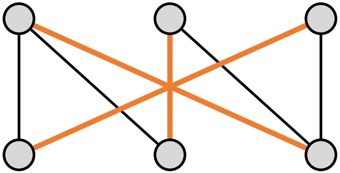
Example: Permanent

Main Problem

Given $n \times n$ matrix M compute:

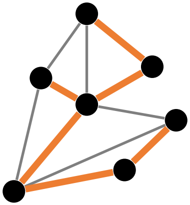
$$\text{per}(M) = \sum_{\text{permutation } \sigma} M_{1,\sigma(1)} M_{2,\sigma(2)} \cdots M_{n,\sigma(n)}.$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$



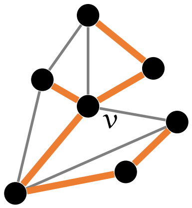
- Permanent of $0/1$ matrix $\equiv$ count of bipartite perfect matchings.
- #P-hard even for $0/1$ matrices [Valiant'79].
- When all of M is close to 1, Barvinok's method [on board ...]

Random Spanning Tree Degrees



Uniformly Random Spanning Tree

Random Spanning Tree Degrees



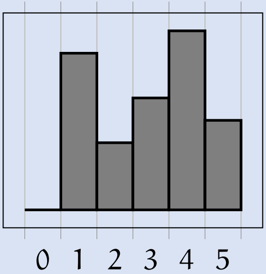
Uniformly Random Spanning Tree

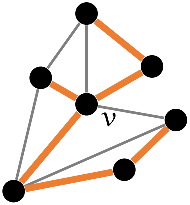
$\deg(v) = \#\{\text{edges adjacent to } v\}$

Random Spanning Tree Degrees

Shape of the Distribution

Can the dist. of $\deg(v)$ look like this?





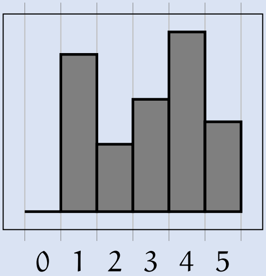
Uniformly Random Spanning Tree

$\deg(v) = \#\{\text{edges adjacent to } v\}$

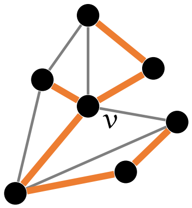
Random Spanning Tree Degrees

Shape of the Distribution

Can the dist. of $\deg(v)$ look like this?



▷ No, it has to be unimodal.



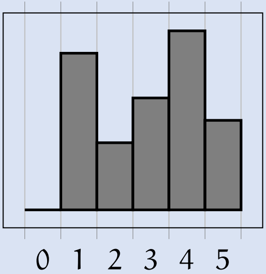
Uniformly Random Spanning Tree

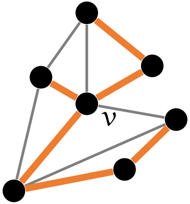
$\deg(v) = \#\{\text{edges adjacent to } v\}$

Random Spanning Tree Degrees

Shape of the Distribution

Can the dist. of $\deg(v)$ look like this?





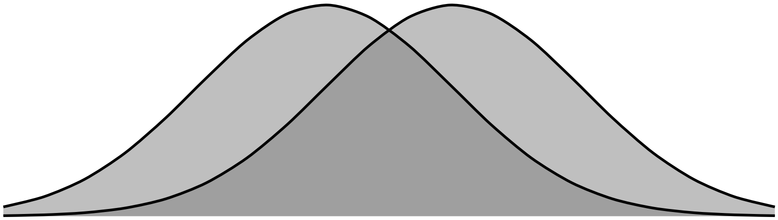
Uniformly Random Spanning Tree

$\deg(v) = \#\{\text{edges adjacent to } v\}$

- No, it has to be unimodal.
- It should actually look like #heads in some number of independent (biased) coin flips. Has to be very concentrated around the mean value.

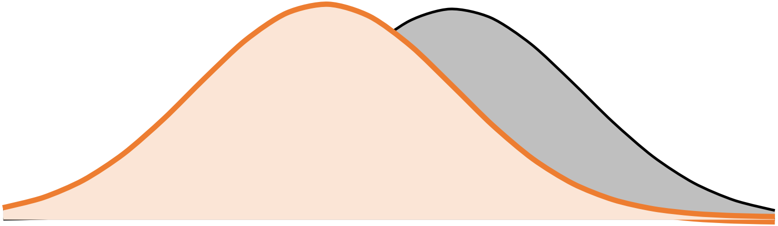
Test Your Intuition

- Two Gaussians. Are mixtures unimodal? Log-concave?



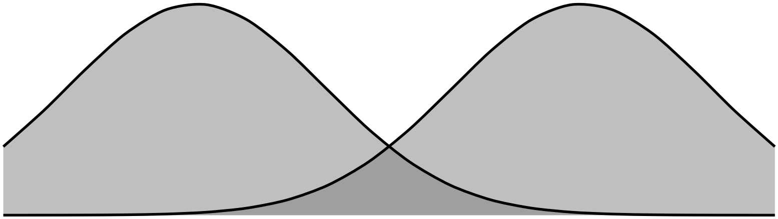
Test Your Intuition

- Two Gaussians. Are mixtures unimodal? Log-concave?



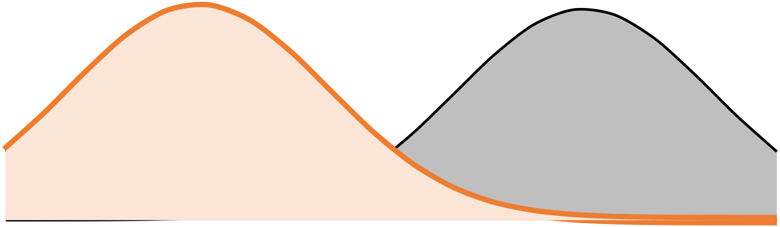
Test Your Intuition

- Two Gaussians. Are mixtures unimodal? Log-concave?



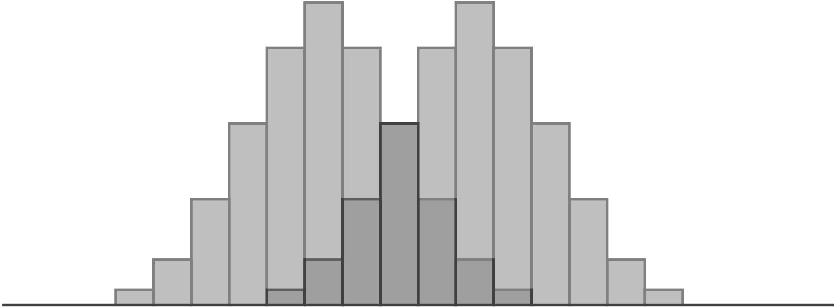
Test Your Intuition

- Two Gaussians. Are mixtures unimodal? Log-concave?



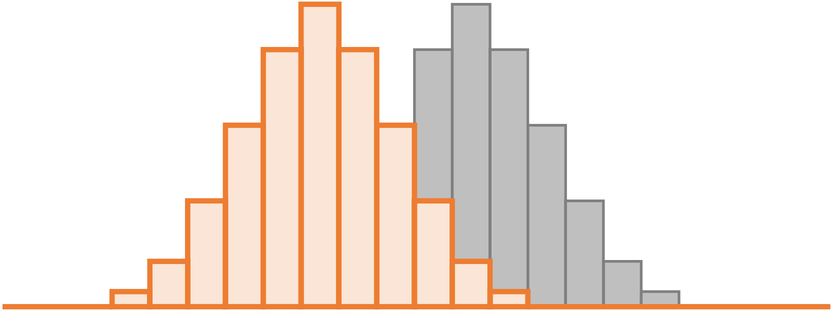
Test Your Intuition

- Two generalized binomials. Are mixtures unimodal? Log-concave?



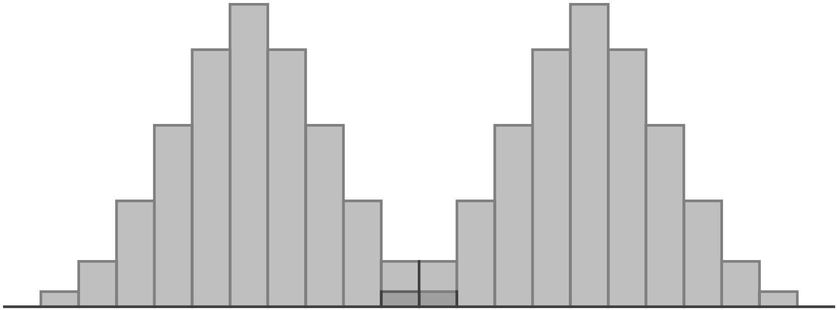
Test Your Intuition

- Two generalized binomials. Are mixtures unimodal? Log-concave?



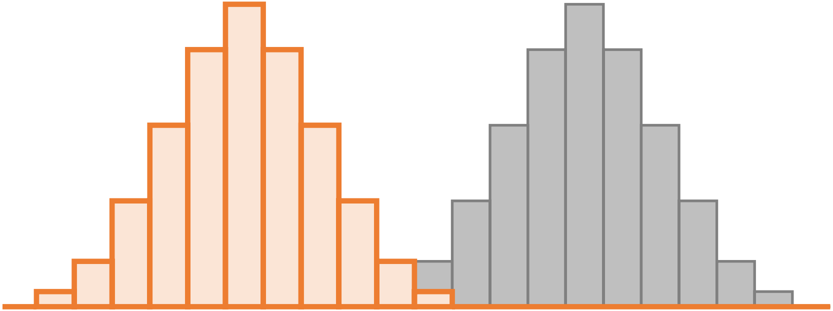
Test Your Intuition

- Two generalized binomials. Are mixtures unimodal? Log-concave?



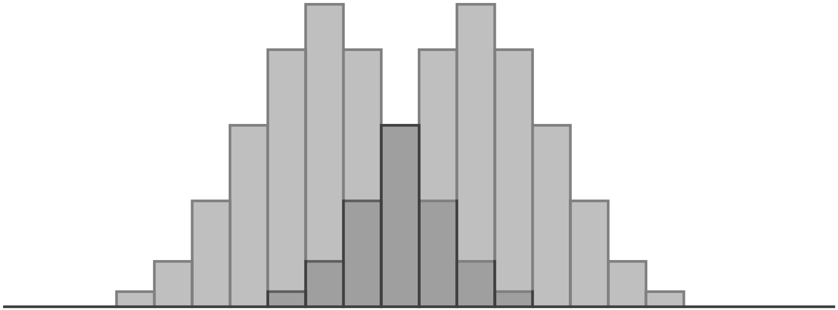
Test Your Intuition

- Two generalized binomials. Are mixtures unimodal? Log-concave?



Test Your Intuition

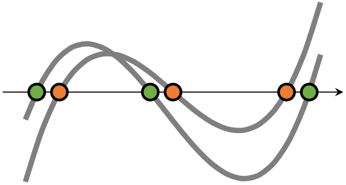
- Two generalized binomials. Are mixtures generalized binomials?



Mixtures of Polynomials

[Folklore, used by e.g., MSS'13]

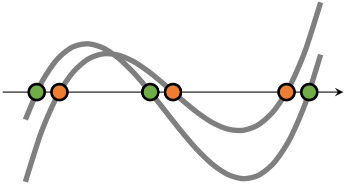
If $\alpha g_1(z) + \beta g_2(z)$ is real-rooted for all $\alpha, \beta > 0$, then roots of g_1, g_2 must have common interlacing.



Mixtures of Polynomials

[Folklore, used by e.g., MSS'13]

If $\alpha g_1(z) + \beta g_2(z)$ is real-rooted for all $\alpha, \beta > 0$, then roots of g_1, g_2 must have common interlacing.



- Corollary: If mixtures of μ, ν are always generalized binomials, then

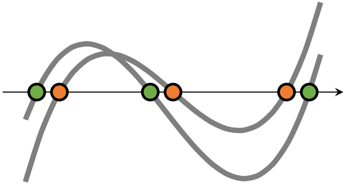
$$\mu = \text{Bernoulli}(p_1) + \cdots + \text{Bernoulli}(p_n),$$
$$\nu = \text{Bernoulli}(q_1) + \cdots + \text{Bernoulli}(q_n),$$

with $p_i, q_i \leqslant p_{i+1}, q_{i+1}$.

Mixtures of Polynomials

[Folklore, used by e.g., MSS'13]

If $\alpha g_1(z) + \beta g_2(z)$ is real-rooted for all $\alpha, \beta > 0$, then roots of g_1, g_2 must have common interlacing.



- Corollary: If mixtures of μ, ν are always generalized binomials, then

$$\mu = \text{Bernoulli}(p_1) + \cdots + \text{Bernoulli}(p_n),$$
$$\nu = \text{Bernoulli}(q_1) + \cdots + \text{Bernoulli}(q_n),$$

  with $p_i, q_i \leqslant p_{i+1}, q_{i+1}$.
- Corollary: The means of μ and ν can be off by $\leqslant 1$.

$$|(p_1 + \cdots + p_n) - (q_1 + \cdots + q_n)| \leqslant 1$$