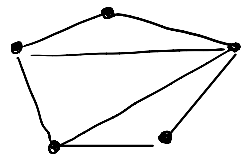
Review

- Spin systems:

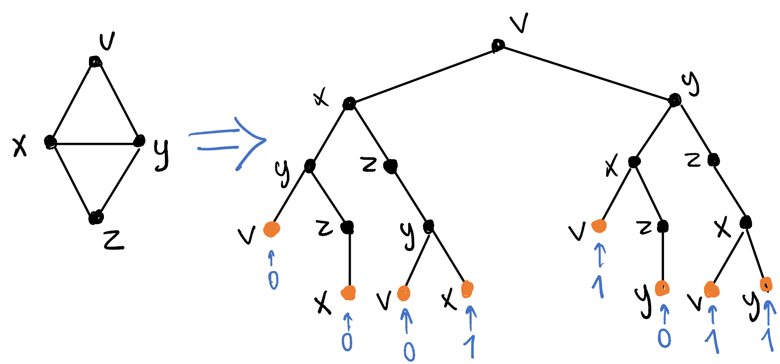
$$\mu(x) \propto \prod_v \phi_v(x_v) \cdot \prod_{u \sim v} \phi_{uv}(x_u, x_v)$$



μ on $[q]^V$

Examples: coloring, hardcore, Ising, ...

- Self-avoiding walk tree: $[q=2]$



Special vertices: first self-intersection

set to 1 if $edge_1 < edge_2$ } arb. ordering edges around each vertex

set to 0 if $edge_1 > edge_2$

- Thm: marginal of v equal in tree & graph.

  [works for any $q=2$ system] I'll sketch again

- Correlation decay: (truncate the tree; it's fine)

  If we condition vertices S to be σ_1 / σ_2

  at least goes to 0

  $$d_{TV}(X_v | \sigma_1, X_v | \sigma_2) \le \text{func}(d(v, \cdot))$$

  S for weak spatial mixing

  $\{u \in S \mid \sigma_1(u) \neq \sigma_2(u)\}$ for strong spatial mixing

* tree equivalence means: SSM for trees $\Rightarrow$ SSM for graphs

- In the hardcore model with $\lambda < (1-\delta) \frac{(\Delta-1)^{(\Delta-1)}}{(\Delta-2)^{\Delta}}$ we have SSM with exponential decay.

- Weitz's ALG: use tree recursion on truncated tree

  - Estimate $\mathbb{P}[X_{v_i} = 0 \mid X_{v_1} = \cdots = X_{v_{i-1}} = 0]$
  - Multiply estimates & invert

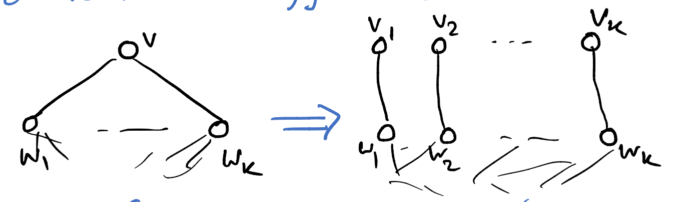
Depth: $O(\lg n)$ Runtime: $\Delta^{O(\lg n)}$ ← poly(n) for $\Delta = O(1)$

Today:

- Roots of polynomials
- Barvinok's method

Graphs ⇒ Trees

[Weitz used for alg] [Godsil for matchings]

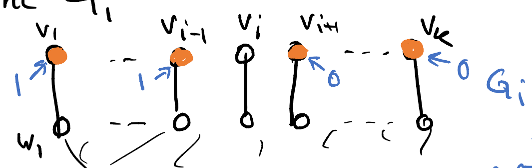


G G'

set new λs so that: $\lambda^{(G')}_{v_1} \times \dots \times \lambda^{(G')}_{v_k} = \lambda^{(G)}_{v}$

- $$\frac{\mathbb{P}_G[v=1]}{\mathbb{P}_G[v=0]} = \frac{\mathbb{P}_{G'}[v_1=\dots=v_k=1]}{\mathbb{P}_{G'}[v_1=\dots=v_k=0]}$$

- Define G_i:



$$\frac{\mathbb{P}_G[v=1]}{\mathbb{P}_G[v=0]} = \prod_{i=1}^{k} \frac{\mathbb{P}_{G_i}[v_i=1]}{\mathbb{P}_{G_i}[v_i=0]} = \prod_i \frac{\mathbb{P}_{G'}[\overbrace{11\cdots1}^{v_i\text{'s}}0\cdots0]}{\mathbb{P}_{G'}[\underbrace{11\cdots1\,0}_{v_i\text{'s}}0\cdots0]}$$

telescoping cancellation

$$\Rightarrow \mathbb{P}_G[v=0] = 1 \Big/ \left(1 + \prod_i \frac{\mathbb{P}_{G_i}[v_i=1]}{\mathbb{P}_{G_i}[v_i=0]}\right)$$

But note that

$$\frac{\mathbb{P}_{G_i}[v_i=1]}{\mathbb{P}_{G_i}[v_i=0]} = \frac{\mathbb{P}_{G_i}[v_i=1 \mid w_i=0]}{\mathbb{P}_{G_i}[v_i=0 \mid w_i=0]} \cdot \frac{\mathbb{P}_{G_i}[w_i=0 \mid v_i=0]}{\mathbb{P}_{G_i}[w_i=0 \mid v_i=1]}$$

$$= \lambda_{v_i} \cdot \mathbb{P}_{G_i}[w_i=0 \mid v_i=0] = \lambda_{v_i} \cdot \mathbb{P}_{G_i - v_i}[w_i=0]$$

So we get recursive formula

$$\mathbb{P}_G[v=0] = \frac{1}{1 + \left(\prod_i \cancel{\lambda_{v_i}}^{\,\lambda}\right) \prod_i \mathbb{P}_{G_i - v_i}[w_i=0]}$$

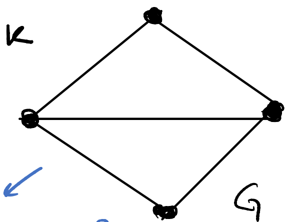
- This is the same as tree recursion.
- We do the same thing for each $G_i - v_i$ by expanding the vertex w_i and so on ---
- If we come back to some v_j it is conditioned the correct way in tree.

Matching Polynomial

Note: Polys and their roots show up a lot!

$m_k(G)$: matchings of size k

$P_G(z) = m_0 + m_1 z + \cdots + m_k z^k + \cdots$



$P_G = 1 + 5z + 2z^2$

Counting goal:

Estimate $P_G(1)$ ← we know FPRAS for this [Jerrum-Sinclair]

What we can do:

Compute $m_k(G)$ in $n^{O(k)}$ time! ← enumerate all

Barvinok's method:

Estimate $P_G(1)$ using just $m_0, \ldots, m_{O(\lg n)}$

when $\Delta = O(1)$ ← max degree

Note: This only gives $n^{O(\lg n)}$-time algs.

$\exists$ trick that makes it poly(n) [Patel-Regts]
↑ if time

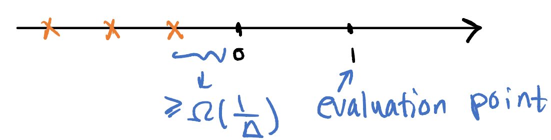
Note: Correlation decay also works [Bayati-Gamarnik-Katz-Nair-Tetali]

What is special about matchings?

- Roots of P_G ← this is the only thing Barvinok's method needs

Thm: P_G has negative real roots and they are $\Omega(\frac{1}{\Delta})$ in magnitude!

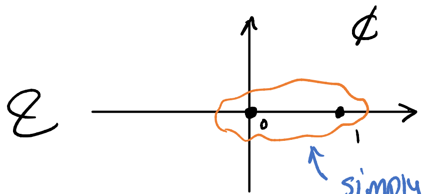
↑ will prove today



Open: Deterministic approx without $\Delta = O(1)$

Barvinok's Method

- Goal: estimate $p(1)$
- Assume: we know $p^{(i)}(0)$ for small i. ← derivative



↑ simply connected root-free region $\supseteq 0, 1$

$\Rightarrow$ Estimate $p(1)$ using $p^{(i)}(0)$ for $i \leq O(\lg n)$ (n: deg p)

Idea:

Truncate Taylor series of $\lg(p)$ around 0.

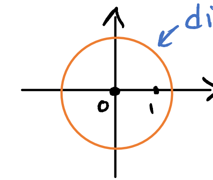
$$\lg p(z) = a_0 + a_1 z + \cdots + a_k z^k + \cdots$$

$$k!\, a_k = \frac{d^k}{dz^k}(\lg p)\Big|_{z=0} = \text{func}\left(p^{(0)}(0), \ldots, p^{(k)}(0)\right)$$

↘ By rules of calculus

What's the error in truncation?

- Complex analysis fact:
  Distance to nearest singularity (zeros of p for lg p) = Radius of convergence
- The error can only be bounded if

zero-free region = 

← disk of radius $e^{\delta} \simeq 1+\delta$

- We will see how to go beyond disks.

Thm: For disks approx to $\lg p(1)$ has

$$\text{additive error} \leq \frac{2e^{-\delta k}}{\delta} \times n$$

↑ for k-truncation

Corollary: When $\delta = \text{const}$, enough to take $k \simeq \lg {}^{n}/_{\varepsilon}$ to get $(1+\varepsilon)$-approx to $p(1)$.

Proof:

$$p = b(z-\lambda_1)\cdots(z-\lambda_n) = c\left(1-\frac{z}{\lambda_1}\right)\cdots\left(1-\frac{z}{\lambda_n}\right)$$

roots

$$\lg p = \lg c + \lg\left(1-\frac{z}{\lambda_1}\right) + \cdots + \lg\left(1-\frac{z}{\lambda_n}\right)$$

$$\text{Taylor}_k(\lg p) = \lg c + \sum_i \text{Taylor}_k\left(\lg\left(1-\frac{z}{\lambda_i}\right)\right)$$

enough to bound additive error of each term

Taylor series for $\lg(1-x)$: (valid for $|x|<1$)

$$-x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \cdots$$

Error of Taylor_k:

$$\leq \sum_{i=k+1}^{\infty} \frac{|x|^i}{i} \leq \sum_{i=k+1}^{\infty} |x|^i \leq \frac{|x|^k}{1-|x|}$$

For $x = \frac{1}{\lambda_i}$ we have $|x| < e^{-\delta}$ so

$$\frac{|x|^k}{1-|x|} \leq \frac{e^{-\delta k}}{\delta/2} \simeq \frac{2e^{-\delta k}}{\delta}$$

Since there are n terms in sum

$$\text{overall error} \leq \frac{2e^{-\delta k}}{\delta} \times n$$

Extending beyond disks

Idea: Find polynomial ϕ s.t.

- $\phi(0)=0$
- $\phi(1)=1$
- $\phi(\text{disk of radius } 1+\delta) \subseteq$ zero-free region
- $\deg\phi$ not too large

Apply the Taylor series method to $p\circ\phi$.

Exercise: Still $(p\circ\phi)^{(i)}(0) = \text{func}(p^{(0)}(0), \ldots, p^{(i)}(0))$

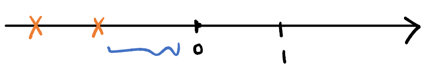
* How to design Φ:

Riemann Mapping:

$\exists$ "biholomorphic" Φ from $D(0, 1+\epsilon)$ to any "hole-free" region.

Take polynomial approximating this Φ and manipulate it a bit.

For matching polynomial:



$\geq \Omega\left(\frac{1}{\Delta}\right)$

$-2 \quad 0 \quad 1 \quad \Rightarrow \quad -\frac{1}{10\Delta} \quad 0 \quad 1$

$$\Phi(z) = \frac{z}{cz + (1-c)}$$

mobius map

$$-\frac{1}{10\Delta} = \frac{-2}{1-3c} \Rightarrow c = \frac{1-20\Delta}{3}$$

Take Φ's Taylor approx of deg $O(\Delta)$ and combine with linear map that sends $\Phi_{approx}(0), \Phi_{approx}(1)$ to $0, 1$.

Exercise: This works!

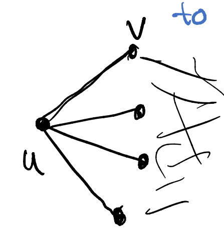
Roots of matching polynomial

Recursive formula for P_G:

$$P_G(z) = P_{G-u}(z) + z \sum_{v \sim u} P_{G-u-v}(z)$$

u unmatched

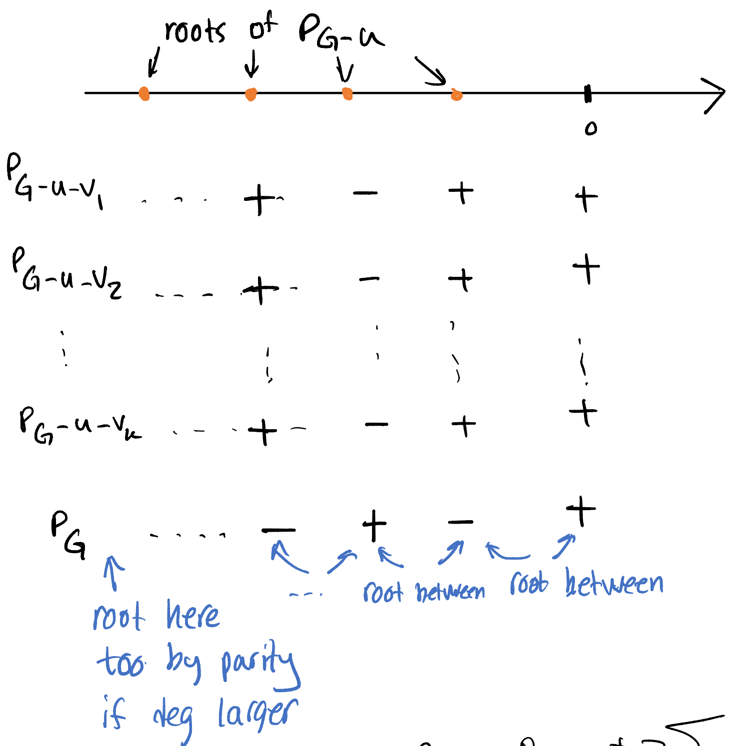
u matched to v



Claim: Roots of P_G and P_{G-u} real and interlace.

one between every two

Proof: We use induction. [Heilmann-Lieb]



$$P_G = P_{G-u} + z \sum_{v \sim u} P_{G-u-v}$$

How to show roots $\leq -\Omega(\frac{1}{\Delta})$?

Take $z \in (-\frac{1}{4\Delta}, 0]$. We will show inductively that

$$0 < P_{G-u}(z) < 2 P_G(z)$$

negative

$$P_G(z) = P_{G-u}(z) + z \cdot \sum_{v \sim u} P_{G-u-v}(z)$$

$$\geq P_{G-u}(z) + 2z \sum_{v \sim u} P_{G-u}(z)$$

$$= (1 + 2z\Delta) P_{G-u}(z) \geq \frac{1}{2} P_{G-u}(z) > 0$$

This means no roots $\in (-\frac{1}{4\Delta}, 0]$

For any fixed $z \in \mathbb{R}_{\geq 0}$, $p(z)$ on Δ-bounded-degree graphs has QFPTAS.

How do we remove the Q?

- Smarter way to compute m_k = # k-matchings.
  - * Brute-force: $n^{O(k)}$
  - * [Patel-Regts]: $\text{poly}(n) \cdot \Delta^{O(k)}$.
- We only go up to $k \simeq \lg n$:)
- Works with

  $\text{ind}(H, G)$: # induced copies of H in G.

  fixed graph

- Example: $\text{ind}(\;, G) = 2\#\text{edges in } G$.

We will work with functions

$$f : \text{Graphs} \to \mathbb{C}$$

that can be written as finite linear combination:

$$f(G) := c_1 \,\text{ind}(H_1, G) + \cdots + c_k \,\text{ind}(H_k, G)$$

Example:

- $f(G) = \#\text{vertices} = \text{ind}(\bullet, G)$
- $f(G) = \#\text{edges} = \text{ind}(\;, G)/2$
- $f(G) = \#\text{2-matchings} =$

$$\left[\text{ind}(\;)\cdot 3 + \text{ind}(\;)\cdot 2 + \cdots\right]/24$$

$\vdots$

The space of these functions is rich.

- $f, g \longmapsto f+g$
- $f, g \longmapsto f \cdot g$

This is because

$$\mathrm{ind}(H_1, G) \cdot \mathrm{ind}(H_2, G) = c_1(H_1 \mathrel{\overline{>}} H_2, G) + c_2(H_1 \mathrel{\times\!\!=} H_2, G) \pm \cdots$$

Call a function additive if

disjoint union.

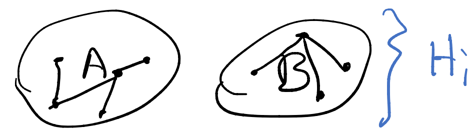
$$f(G_1 + G_2) = f(G_1) + f(G_2).$$

Theorem: Additive functions only need connected graphs H_i in

$$f(G) = c_1 \mathrm{ind}(H_1, G) + \cdots + c_k \mathrm{ind}(H_k, G)$$

Proof: Note that if H is connected $\mathrm{ind}(H, \cdot)$ is additive.

- Suppose $f = \sum c_i \mathrm{ind}(H_i, G)$
- Pick the disconnected H_i with the smallest # of edges.



- Then $f(H_i) - f(A) - f(B) = c_i - 0 - 0$
- This means $c_i = 0$

Key Observation: If H is connected, counting $\mathrm{ind}(H, G)$ can be done in $\mathrm{poly}(n)\, \Delta^{O(|H_i|)}$ time.