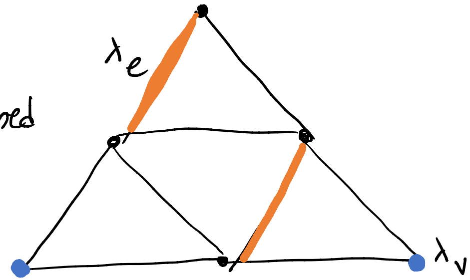
Review

- Monomer - Dimer

  $*\lambda_v$ can be absorbed



$$\mu(\text{matching } M) = \prod_{e \in M} \lambda_e \prod_{v \notin M} \lambda_v$$

[Jerrum-Sinclair]

Thm: Metropolis relaxes in $\text{poly}(n, \max\{\lambda_e\})$.

Corollary: When $\frac{PM}{\text{near-PM}} > \frac{1}{\text{Poly}(n)}$ we can sample PMs in poly(n) time.

- Bipartite perfect matchings:

  Let $\Omega_S = \{\text{matchings with monomers} = S\}$.

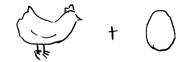
[Jerrum-Sinclair-Vigoda]

Thm: Metropolis on $\Omega = \Omega_\emptyset \cup \bigcup_{a,b} \Omega_{a,b}$ with

$$\mu(M) \propto \text{weight}(M) / \sum_{M' \in \Omega_S} \text{weight}(M') \quad \text{for } M \in \Omega_S$$

can be approx.

relaxes in poly(n) time for bipartite graphs.

- The  problem: $\sum_{M' \in \Omega_S} \text{weight}(M')$!

  Solution: Pick approx weights w:

  $$\text{weight}(M) = \lambda^M \cdot w(S) \quad \text{for } M \in \Omega_S$$

  $\lambda_0, w_0 \leftarrow K_{n,n}$ with all edges of weight 1.

  $$\lambda_0, w_0 \xmapsto[\text{change } \lambda\text{s slightly}]{} \lambda_1, w_0 \xmapsto[\text{sample \& Monte Carlo}]{} \lambda_1, w_1 \xmapsto[\text{Change } \lambda\text{s slightly}]{} \lambda_2, w_1 \mapsto \cdots$$

Plan for Today

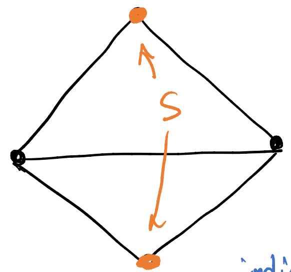
Break from Markov chains:

- Correlation decay
- Weitz's algorithm for the hardcore model

Hardcore Model

$$\mu(S) \propto \begin{cases} 0 & \text{if } S \text{ not ind.} \\ \lambda^{|S|} & \text{if } S \text{ ind.} \end{cases}$$

we can have different λ_v more generally



$S \in \{0,1\}^n$ indicator

Thm: $\lambda < \frac{1}{\Delta} \Rightarrow$ Dobrushin $\Rightarrow$ fast mixing of Glauber

[Weitz]
Thm: When $\lambda < (1-\delta)\lambda_c(\Delta) \Rightarrow$ FPTAS (deterministic)

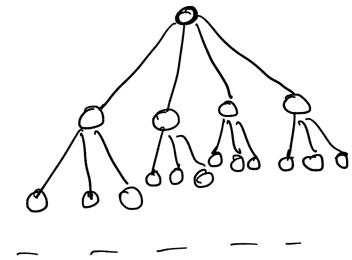
$$\lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^{\Delta}} \simeq \frac{e}{\Delta}$$

Note: This doesn't imply Glauber mixes fast :(
We will prove this later in the course.

[Sly]
Thm: For $\lambda > (1+\delta)\cdot\lambda_c(\Delta)$, FPRAS is NP-hard.

Question: Where is $\lambda_c(\Delta)$ coming from?

$\underbrace{(\Delta-1)\text{-branching}}_{\text{except for root, all nodes have } \deg = \Delta}$ trees



Suppose we have one of these trees of large height. How much do leaves influence the root?

$$d_{TV}\begin{pmatrix} \text{root} \mid \text{leaves in config}_1, \\ \text{root} \mid \text{leaves in config}_2 \end{pmatrix} = ?$$

Correlation Decay: As height $\to \infty$, the $d_{TV} \to 0$.

This happens when $\lambda < \lambda_c(\Delta)$.

[Weitz]:

Correlation decay on $(\Delta-1)$-branching tree

$\Rightarrow$ FPTAS on all graphs with $\deg \leq \Delta$.

↳ no MCMC, deterministic

Note: This has been generalized to anti-ferromagnetic two-spin systems.

e.g. Ising model

Algorithm: (think of μ on $\{0,1\}^n$)

- Approx $\mathbb{P}[x_{v_1}=0]$ ← we will show how
- Approx $\mathbb{P}[x_{v_2}=0 \mid x_{v_1}=0]$ ← these are similar

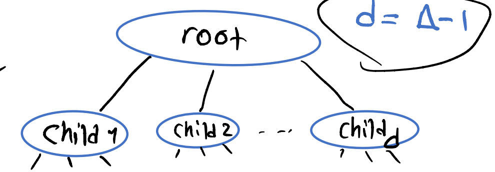
$\vdots$

- Approx $\mathbb{P}[x_{v_n}=0 \mid x_{v_1}=\cdots=x_{v_{n-1}}=0]$
- Multiply everything together and invert!

[Weitz]: Marginals can be estimated well from local neighborhood!

Correlation Decay on Trees

p_v: $\mathbb{P}[x_v=0]$ in v's subtree



Claim: $$p_{root} = \frac{1}{1+\lambda \prod_i p_{child\ i}}$$

Proof:

$$p_{root} = \frac{1}{1+\frac{\mathbb{P}[root=1]}{\mathbb{P}[root=0]}}$$

cond. ind.

$$\mathbb{P}[children=0 \mid root=0] = \prod p_{child\ i}$$

$$\mathbb{P}[children=0 \mid root=1] = 1$$

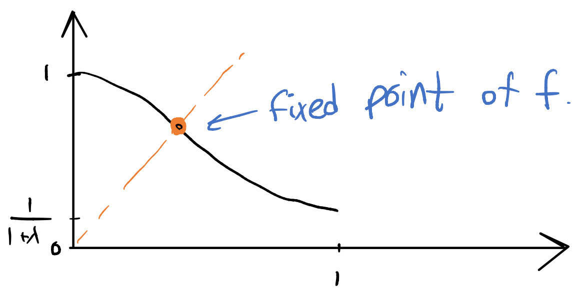
$$\frac{\mathbb{P}[root=1 \mid children=0]}{\mathbb{P}[root=0 \mid children=0]} = \lambda$$

$$\Rightarrow \frac{\mathbb{P}[root=1]}{\mathbb{P}[root=0]} = \lambda \cdot \prod p_{child\ i}$$

What happens for the symmetric case?

$$P_{\text{child } i} = P_{\text{child } j}$$

$$P \xmapsto{f} \frac{1}{1+\lambda \underbrace{p^d}}$$



$\lambda_c(\Delta)$: when $|\text{derivative}| = 1$ at fixed point!

HW: Show this happens for $\lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^{\Delta}}$

For $\lambda < \lambda_c(\Delta)$ we have $|\text{derivative}| < 1$!

For $\lambda > \lambda_c(\Delta)$ " " $|\text{derivative}| > 1$!

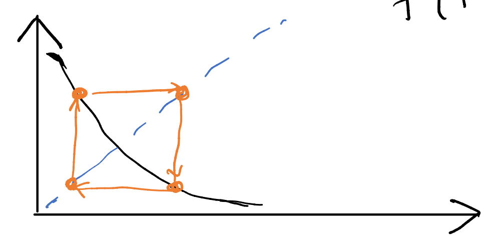
- When fixed point is attractive, the marginal of root converges to a unique distribution as height $\to \infty$.

$|\text{derivative}| < 1 \implies \exists$ basin of attraction

In this case basin of attraction = everywhere

- When fixed point is repulsive, the marginal of root oscillates based on height parity.

$f(f(x)) = x$



This is related to uniqueness of Gibbs measure on infinite trees.

There is a transformation φ such that $\underline{\varphi \circ f \circ \varphi^{-1}}$ has |derivative| bounded by $1-\varepsilon$ everywhere.

HW

$$f(x) = \frac{1}{1+\lambda x^d}$$

$\varphi(f(\varphi^{-1}(x))) = x$

$\varphi^{-1}(x)$ f.p. of f.

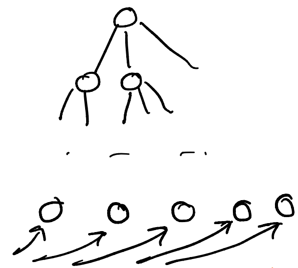
$g(x) = \varphi \circ f \circ \varphi^{-1}$ ← absolute contraction

$$f \circ f \circ \cdots \circ f(x) = \varphi^{-1} \circ \underbrace{g \circ g \circ g \cdots \circ g}_{g \text{ is a contraction}} \circ \varphi(x)$$

Note: In two-spin systems a similar calculation defines uniqueness threshold on trees.

Open Problem: For |spins| > 2, e.g., coloring, everything is harder and in many cases open.

General Trees



arbitrary boundary conditions.

$$P_{root} = f_d(P_{child_1}, P_{child_2}, \ldots, P_{child_d})$$

multi-variate version of f.

Using the same transformed iteration

$$g_d := \varphi \circ f_d(\varphi^{-1}(x_1), \varphi^{-1}(x_2), \ldots, \varphi^{-1}(x_d))$$

Then g_d is still a contraction.

$$\|\nabla g_d\|_1 < 1-\varepsilon \text{ at all points}$$

- This implies that if

$$\| P_{leaves} - P'_{leaves} \|_\infty < C \Longrightarrow$$

$$| P_{root} - P'_{root} |_\infty < (1-\delta)^{height} C$$

$$g_d(x) - g_d(x') = \int_0^1 \langle \nabla g_d(tx' + (1-t)x), x'-x \rangle dt$$

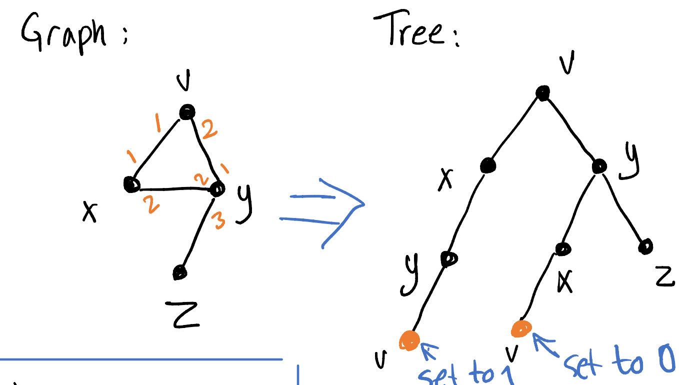
$$\leq \int \|\nabla g_d\|_1 \cdot \|x'-x\|_\infty dt \leq \max\{\|\nabla g\|_1\} \cdot \|x'-x\|_\infty$$

- Regardless of how we set leaves the root converges exponentially fast to the same marginal.

- So far: If $\lambda < \lambda_c(\Delta)$ then trees have correlation decay

- Question: How do we go from trees to general graphs?

Self-Avoiding Walk Tree



Weitz: The marginal of v on graph = marginal on tree

- We have to condition highlighted vertices in tree carefully.

- Every node except those highlighted represents a path from v that doesn't intersect itself.

- Highlighted nodes: Cycles going back to v.

- Fix ordering on tree traversal (edges going out of vertex)

- Cycle verts:
  - * 1 if closing edge > starting edge
  - * 0 otherwise

Spatial Mixing

Let us condition a subset $S \subseteq V$ of vertices to have σ_1, σ_2 values.

Weak Spatial Mixing:

$$d_{TV}(\text{root} \mid \sigma_1, \text{root} \mid \sigma_2) \leq \text{func}(\text{dist}(\text{root}, S))$$

exponentially decay

Strong Spatial Mixing:

$$d_{TV}(\text{root} \mid \sigma_1, \text{root} \mid \sigma_2) \leq \text{func}(\text{dist}(\text{root}, S'))$$

where $S' = \{ v \mid \sigma_1(v) \neq \sigma_2(v) \}$

Strong Spatial Mixing allows us to condition on nearby vertices, as long as we do so consistently.

- Self-Avoiding Walk Representation:

  If trees have strong spatial mixing $\Rightarrow$ graphs have SSM.

- The tree analysis (recursion + $\|\nabla g\|_1$) actually shows strong spatial mixing. :)

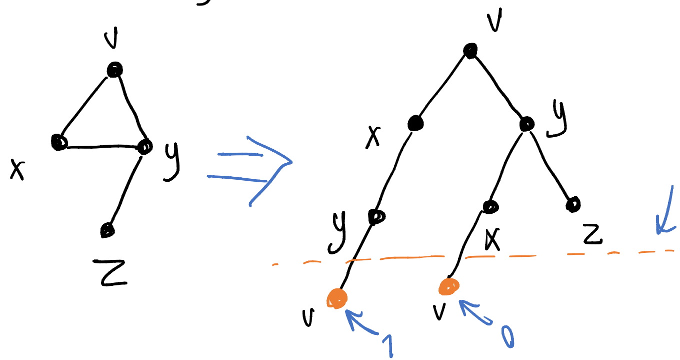
Strong Spatial Mixing $\Rightarrow$ Approx

- To compute marginal of v:
- Construct SAW tree, truncated at depth $O(\lg n)$.
- Assign arbitrary boundary conditions to free leaves.
- Use recursion to compute marginals

  $$\text{error} \leq (1-\delta)^{\lg n} = \frac{1}{\text{poly}(n)}$$:)

Weitz's Algorithm

- For vertex v, form the truncated self-avoiding walk tree.



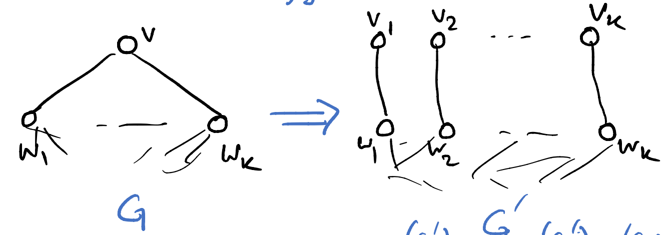
- Fix leaves arbitrarily.
- Recursively compute $Pr[X_v]$ in tree.

Analysis:
- Truncate in $O(\lg n)$ depth
  - Approximation error $\leq (1-\delta)^{O(\lg n)} = \frac{1}{\text{poly}(n)}$
  - Time: Size of tree $\approx (\Delta-1)^{O(\lg n)} = n^{O(\lg \Delta)} = \text{poly}(n)$

assuming this is constant

Graphs $\Rightarrow$ Trees

[Weitz used for alg.] [Godsil for matchings]

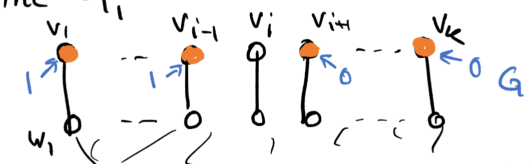


set new λs so that: $\lambda^{(G')}_{v_1} \times \cdots \times \lambda^{(G')}_{v_k} = \lambda^{(G)}_{v}$

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

telescoping cancellation

- Define G_i:



$$\frac{\mathbb{P}_G[v=1]}{\mathbb{P}_G[v=0]} = \prod_{i=1}^{k} \frac{\mathbb{P}_{G_i}[v_i=1]}{\mathbb{P}_{G_i}[v_i=0]} = \prod_i \frac{\mathbb{P}_{G'}[\overbrace{11\cdots1}^{v_i\text{'s}}\,0\cdots0]}{\mathbb{P}_{G'}[\underbrace{11\cdots1}_{v_i\text{'s}}\,00\cdots0]}$$

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

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

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

- 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 ---

Note: When we come back to a copy of v, say v_j, if $j<i$ we have set that copy in G_i to 1, and if $j>i$, we have set it to 0. This is why the special conditioning happens.