CS 263: Counting and Sampling

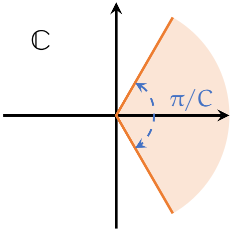
Nima Anari

Stanford University

slides for

Correlation Decay

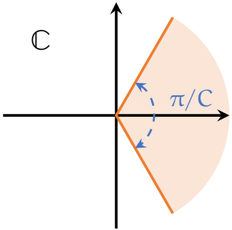
Review



Polynomial $g(z_1, \ldots, z_n)$ is C sector stable if for all $z_1, \ldots, z_n \in$ sector

$$g(z_1, \ldots, z_n) \neq 0.$$

Review



Polynomial $g(z_1, \ldots, z_n)$ is C sector stable if for all $z_1, \ldots, z_n \in$ sector

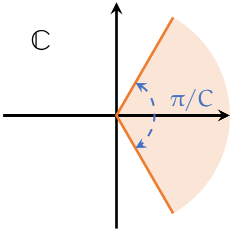
$$g(z_1, \ldots, z_n) \neq 0.$$

[Gårding]

Half-plane stable $\implies$

$$\log g(z_1, \ldots, z_n) \text{ concave}$$

Review



Polynomial $g(z_1, \ldots, z_n)$ is C sector stable if for all $z_1, \ldots, z_n \in$ sector

$$g(z_1, \ldots, z_n) \neq 0.$$

[Gårding]

Half-plane stable $\Longrightarrow$

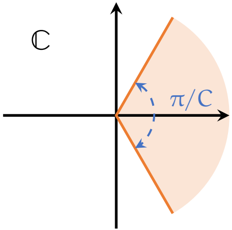
$$\log g(z_1, \ldots, z_n) \text{ concave}$$

[Alimohammadi-A-Shiragur-Vuong]

C sector stable $\Longrightarrow$

$$\log g(\sqrt[2C]{z_1}, \ldots, \sqrt[2C]{z_n}) \text{ concave}$$

Review



Polynomial $g(z_1, \ldots, z_n)$ is C sector stable if for all $z_1, \ldots, z_n \in$ sector

$$g(z_1, \ldots, z_n) \neq 0.$$

[Gårding]

Half-plane stable $\implies$

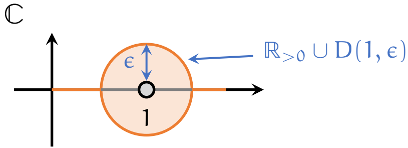
$$\log g(z_1, \ldots, z_n) \text{ concave}$$

[Alimohammadi-A-Shiragur-Vuong]

C sector stable $\implies$

$$\log g(\sqrt[2C]{z_1}, \ldots, \sqrt[2C]{z_n}) \text{ concave}$$

Sufficient region for SI:

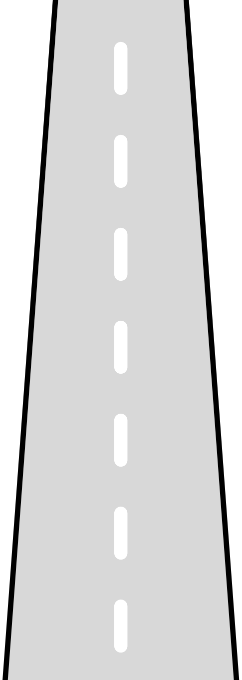


Correlation Decay

- Marginals on trees
- Self-avoiding walk tree
- Weak and strong spatial mixing

HDX via Correlation Decay

- Tree vs graph polynomials

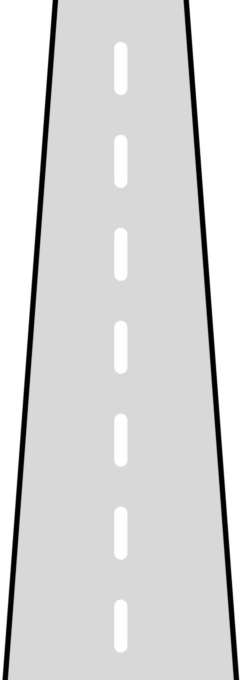


Correlation Decay

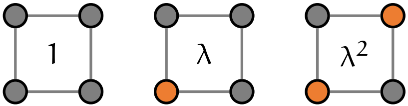
- Marginals on trees
- Self-avoiding walk tree
- Weak and strong spatial mixing

HDX via Correlation Decay

- Tree vs graph polynomials

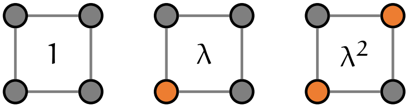


Hardcore model



$\mu(\text{ind set } S) \propto \lambda^{|S|}$

Hardcore model

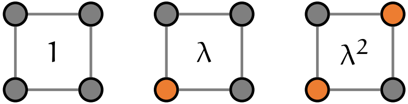


$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹

max ind set is **NP**-hard

Hardcore model



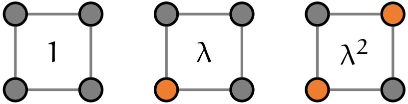
$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹

  max ind set is **NP**-hard

- For what λ is it easy to sample?

Hardcore model



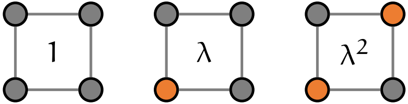
$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹

  max ind set is NP-hard

- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$

Hardcore model



$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

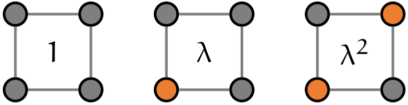
- Large λ is hard. ☹

  max ind set is **NP**-hard

- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

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

Hardcore model



$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard.

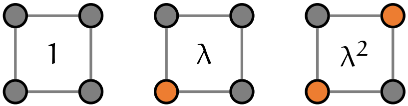
  max ind set is NP-hard

- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

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

- [Sly'10]: NP-hard for $\lambda > \lambda_c(\Delta)$

Hardcore model



$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹

  max ind set is NP-hard

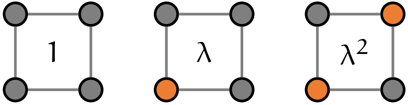
- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

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

- [Sly'10]: NP-hard for $\lambda > \lambda_c(\Delta)$

- Where is λ_c coming from?

Hardcore model



$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹

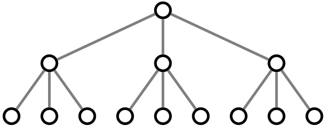
  max ind set is NP-hard

- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

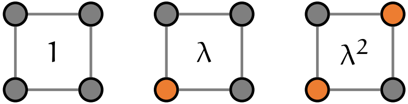
$$\lambda < \lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^{\Delta}} \simeq \frac{e}{\Delta}$$

- [Sly'10]: NP-hard for $\lambda > \lambda_c(\Delta)$

- Where is λ_c coming from?
- Correlation decay threshold on $(\Delta-1)$-branching tree:



Hardcore model



$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard.

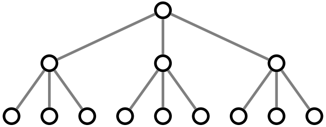
  max ind set is NP-hard

- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

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

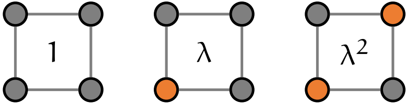
- [Sly'10]: NP-hard for $\lambda > \lambda_c(\Delta)$

- Where is λ_c coming from?
- Correlation decay threshold on $(\Delta - 1)$-branching tree:



- Do leaves influence the root?

Hardcore model



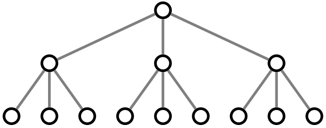
$$\mu(\text{ind set } S) \propto \lambda^{|S|}$$

- Large λ is hard. ☹
  max ind set is NP-hard
- For what λ is it easy to sample?
- [Dobrushin]'s condition: when $\lambda < \frac{1}{\Delta}$
- [Weitz'06]: easy when

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

- [Sly'10]: NP-hard for $\lambda > \lambda_c(\Delta)$

- Where is λ_c coming from?
- Correlation decay threshold on $(\Delta-1)$-branching tree:



- Do leaves influence the root?

Correlation decay

For any two configs σ, σ' for leaves

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(\text{height})$$

where $f(\text{height}) \to 0$ as height $\to \infty$.

[Weitz]

Strong form of correlation decay on trees $\implies$ there is an FPTAS for Δ-bounded degree graphs.

[Weitz]

Strong form of correlation decay on trees $\implies$ there is an FPTAS for Δ-bounded degree graphs.

- Note: this generalizes to other “repulsive” two-spin systems.

[Weitz]

Strong form of correlation decay on trees $\implies$ there is an FPTAS for Δ-bounded degree graphs.

- Note: this generalizes to other "repulsive" two-spin systems.
- [Weitz]'s algorithm estimates marginals $\mathbb{P}[X_v]$ via local computation.

[Weitz]

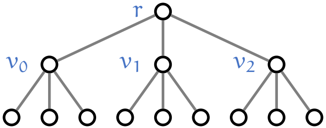
Strong form of correlation decay on trees $\implies$ there is an FPTAS for Δ-bounded degree graphs.

- Note: this generalizes to other "repulsive" two-spin systems.
- [Weitz]'s algorithm estimates marginals $\mathbb{P}[X_v]$ via local computation.
- Multiplying estimated marginals approximates partition function $\sum_{\text{ind } S} \lambda^S \simeq$

$$\frac{1}{\mathbb{P}[X_{v_1} = 0] \cdot \mathbb{P}[X_{v_2} = 0 \mid X_{v_1} = 0] \cdots \mathbb{P}[X_{v_n} = 0 \mid X_{v_1} = \cdots = X_{v_{n-1}} = 0]}$$

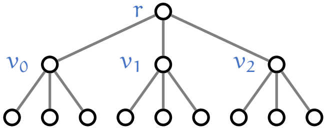
Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

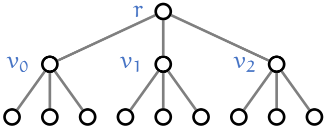
If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

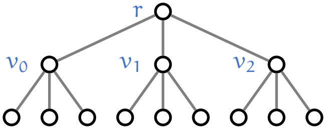
with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

Proof:

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

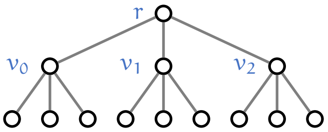
Proof:

- We have

$$\mathbb{P}[X_{v_i} = 0 \mid X_r = 0] = p_{v_i}$$

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1 + \lambda p_{v_1} \cdots p_{v_d}}$$

Proof:

- We have

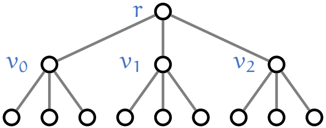
$$\mathbb{P}[X_{v_i} = 0 \mid X_r = 0] = p_{v_i}$$

- Once we condition on root, children become independent, so

$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 0] = \prod_i p_{v_i}$$

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

Proof:

- We have

$$\mathbb{P}[X_{v_i} = 0 \mid X_r = 0] = p_{v_i}$$

- Once we condition on root, children become independent, so

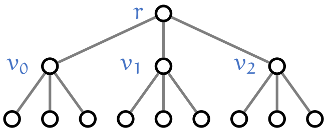
$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 0] = \prod_i p_{v_i}$$

- On the other hand

$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 1] = 1$$

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

Proof:

- We have

$$\mathbb{P}[X_{v_i} = 0 \mid X_r = 0] = p_{v_i}$$

- Once we condition on root, children become independent, so

$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 0] = \prod_i p_{v_i}$$

- On the other hand

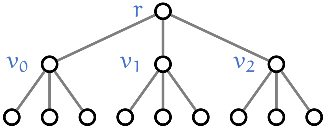
$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 1] = 1$$

- We also have

$$\frac{\mathbb{P}[X_r = 1 | X_{\text{children}} = 0]}{\mathbb{P}[X_r = 0 | X_{\text{children}} = 0]} = \lambda$$

Marginals on trees

- On trees, (conditional) marginals can be computed recursively.



Recursive computation

If $p_v = \mathbb{P}[X_v = 0]$ in v's subtree, and $d = \Delta - 1$:

with conditioned leaves

$$p_r = \frac{1}{1+\lambda p_{v_1} \cdots p_{v_d}}$$

Proof:

- We have

$$\mathbb{P}[X_{v_i} = 0 \mid X_r = 0] = p_{v_i}$$

- Once we condition on root, children become independent, so

$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 0] = \prod_i p_{v_i}$$

- On the other hand

$$\mathbb{P}[X_{\text{children}} = 0 \mid X_r = 1] = 1$$

- We also have

$$\frac{\mathbb{P}[X_r = 1 \mid X_{\text{children}} = 0]}{\mathbb{P}[X_r = 0 \mid X_{\text{children}} = 0]} = \lambda$$

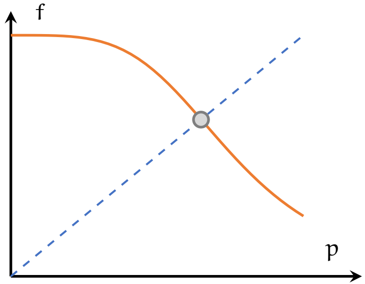
- Combining all gives

$$\frac{\mathbb{P}[X_r = 1]}{\mathbb{P}[X_r = 0]} = \lambda p_{v_1} \cdots p_{v_d}$$

What happens in the symmetric case where all leaves conditioned the same?

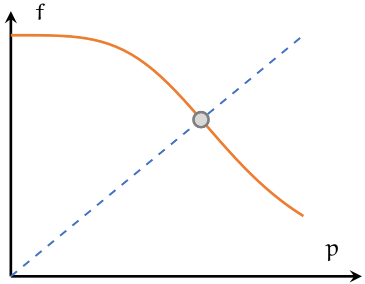
What happens in the symmetric case where all leaves conditioned the same?

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



What happens in the symmetric case where all leaves conditioned the same?

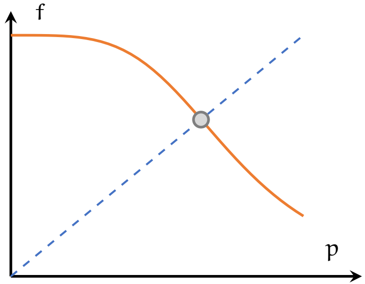
$$p \mapsto f(p) = \frac{1}{1+\lambda p^d}$$



- There is fixed point p^*.

What happens in the symmetric case where all leaves conditioned the same?

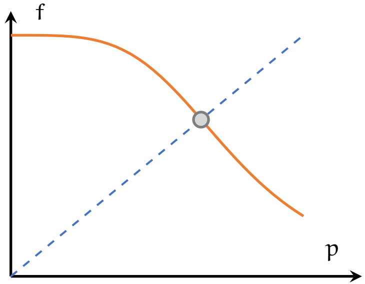
$$p \mapsto f(p) = \frac{1}{1+\lambda p^d}$$



- There is fixed point p^*.
- λ_c: when $|\text{derivative}| = 1$ at p^*

What happens in the symmetric case where all leaves conditioned the same?

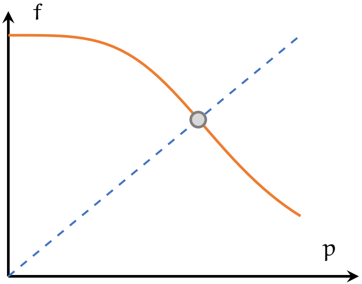
$$p \mapsto f(p) = \frac{1}{1+\lambda p^d}$$



- There is fixed point p^*.
- λ_c: when |derivative| $= 1$ at p^*
- Exercise: $=$ promised value.

What happens in the symmetric case where all leaves conditioned the same?

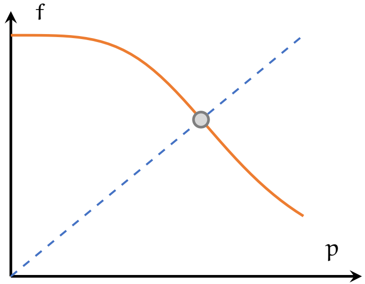
$$p \mapsto f(p) = \frac{1}{1+\lambda p^d}$$



- There is fixed point p^*.
- λ_c: when |derivative| $= 1$ at p^*
- Exercise: $=$ promised value.
- When $\lambda < \lambda_c$ we have $|f'(p^*)| < 1$. In this case, p^* is attractive.

What happens in the symmetric case where all leaves conditioned the same?

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

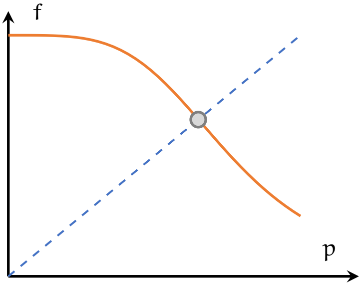


- There is fixed point p^*.
- λ_c: when $|\text{derivative}| = 1$ at p^*
- Exercise: $=$ promised value.

- When $\lambda < \lambda_c$ we have $|f'(p^*)| < 1$. In this case, p^* is attractive.
- When $\lambda > \lambda_c$ we have $|f'(p^*)| > 1$. In this case, p^* is repulsive.

What happens in the symmetric case where all leaves conditioned the same?

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



- There is fixed point p^*.
- λ_c: when |derivative| $= 1$ at p^*
- Exercise: $=$ promised value.

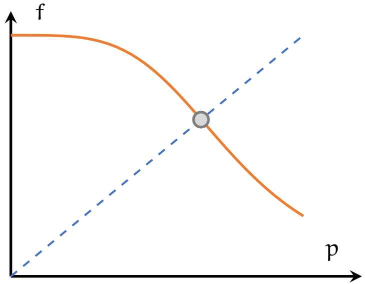
- When $\lambda < \lambda_c$ we have $|f'(p^*)| < 1$. In this case, p^* is attractive.
- When $\lambda > \lambda_c$ we have $|f'(p^*)| > 1$. In this case, p^* is repulsive.
- When attractive, for $p_0 \in$ basin of attraction:

  $$f^{\text{height}}(p_0) \to p^*$$

  Luckily for this f, basin of attraction is all of $[0, 1]$.

What happens in the symmetric case where all leaves conditioned the same?

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

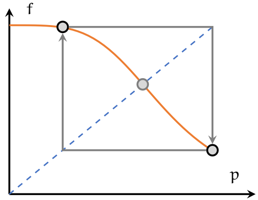


- There is fixed point p^*.
- λ_c: when |derivative| $= 1$ at p^*
- Exercise: $=$ promised value.

- When $\lambda < \lambda_c$ we have $|f'(p^*)| < 1$. In this case, p^* is attractive.
- When $\lambda > \lambda_c$ we have $|f'(p^*)| > 1$. In this case, p^* is repulsive.
- When attractive, for $p_0 \in$ basin of attraction:

  $$f^{\text{height}}(p_0) \to p^*$$

  Luckily for this f, basin of attraction is all of $[0, 1]$.
- When repulsive, oscillating:



- When repulsive, there is no hope of correlation decay. 🙁

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

a bit magical

$$g = \psi \circ f \circ \psi^{-1}$$

has $|g'| \leqslant 1 - \epsilon$ everywhere.

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

a bit magical

$$g = \psi \circ f \circ \psi^{-1}$$

has $|g'| \leqslant 1 - \epsilon$ everywhere.
- Since g is globally contracting, and $f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi$, for any two starting p_0, p_0':

$$\left| f^h(p_0) - f^h(p_0') \right| \leqslant C(1-\epsilon)^h$$

exponential decay

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

  a bit magical

$$g = \psi \circ f \circ \psi^{-1}$$

  has $|g'| \leqslant 1 - \epsilon$ everywhere.

- Since g is globally contracting, and

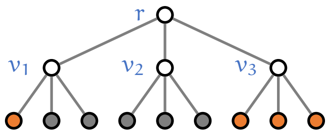
$$f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi,$$

  for any two starting p_0, p_0':

$$\left| f^h(p_0) - f^h(p_0') \right| \leqslant C(1 - \epsilon)^h$$

  exponential decay

- What happens for asymmetric conditionings?



- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

a bit magical

$$g = \psi \circ f \circ \psi^{-1}$$

has $|g'| \leqslant 1 - \epsilon$ everywhere.
- Since g is globally contracting, and

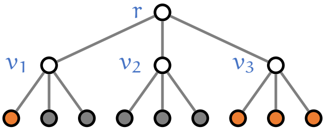
$$f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi,$$

for any two starting p_0, p_0':

$$\left| f^h(p_0) - f^h(p_0') \right| \leqslant C(1-\epsilon)^h$$

exponential decay

- What happens for asymmetric conditionings?



- Now f is multivariate:

$$p_r = f(p_{v_1}, \ldots, p_{v_d})$$

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

  a bit magical

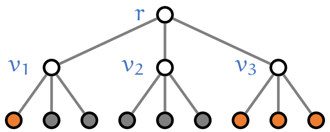
$$g = \psi \circ f \circ \psi^{-1}$$

  has $|g'| \leqslant 1 - \epsilon$ everywhere.
- Since g is globally contracting, and $f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi$, for any two starting p_0, p_0':

$$\left|f^h(p_0) - f^h(p_0')\right| \leqslant C(1-\epsilon)^h$$

  exponential decay

- What happens for asymmetric conditionings?



- Now f is multivariate:

$$p_r = f(p_{v_1}, \ldots, p_{v_d})$$

- Luckily, the same univariate transformation works:

$$g = \psi \circ f(\psi^{-1}(x_1), \ldots, \psi^{-1}(x_d))$$

  will have $\|\nabla g\|_1 \leqslant 1 - \epsilon$ everywhere.

- When repulsive, there is no hope of correlation decay. ☹
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. ☹
- Luckily, there is transformation ψ, such that

a bit magical

$$g = \psi \circ f \circ \psi^{-1}$$

has $|g'| \leqslant 1 - \epsilon$ everywhere.
- Since g is globally contracting, and

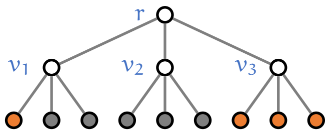
$$f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi,$$

for any two starting p_0, p_0':

$$\left| f^h(p_0) - f^h(p_0') \right| \leqslant C(1 - \epsilon)^h$$

exponential decay

- What happens for asymmetric conditionings?



- Now f is multivariate:

$$p_r = f(p_{v_1}, \ldots, p_{v_d})$$

- Luckily, the same univariate transformation works:

$$g = \psi \circ f(\psi^{-1}(x_1), \ldots, \psi^{-1}(x_d))$$

will have $\|\nabla g\|_1 \leqslant 1 - \epsilon$ everywhere.
- Thus g contracts $\|\cdot\|_\infty$ by $1 - \epsilon$.

- When repulsive, there is no hope of correlation decay. 🙁
- When attractive, although $|f'(p^*)| < 1$, $|f'(\cdot)|$ is not globally bounded by 1. 🙁
- Luckily, there is transformation ψ, such that

$$g = \psi \circ f \circ \psi^{-1}$$

a bit magical

has $|g'| \leqslant 1 - \epsilon$ everywhere.
- Since g is globally contracting, and

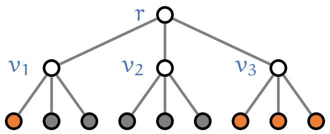
$$f \circ f \circ \cdots \circ f = \psi^{-1} \circ g \circ g \circ \cdots \circ g \circ \psi,$$

for any two starting p_0, p_0':

$$\left| f^h(p_0) - f^h(p_0') \right| \leqslant C(1-\epsilon)^h$$

exponential decay

- What happens for asymmetric conditionings?



- Now f is multivariate:

$$p_r = f(p_{v_1}, \ldots, p_{v_d})$$

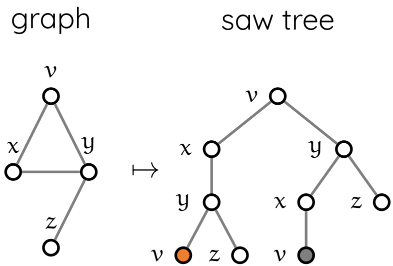
- Luckily, the same univariate transformation works:

$$g = \psi \circ f(\psi^{-1}(x_1), \ldots, \psi^{-1}(x_d))$$

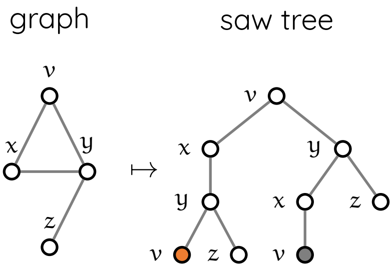
will have $\|\nabla g\|_1 \leqslant 1 - \epsilon$ everywhere.
- Thus g contracts $\|\cdot\|_\infty$ by $1 - \epsilon$.
- Correlation decay on trees exactly below λ_c. 🙂

What about general graphs?

Self-avoiding walk tree



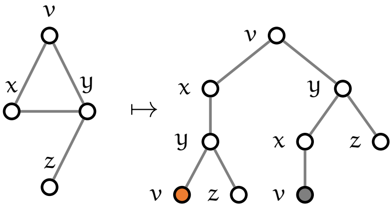
Self-avoiding walk tree



- Tree $=$ paths from root in graph. If cycle formed, end and condition.

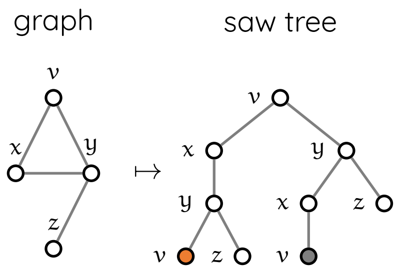
Self-avoiding walk tree

graph saw tree



- Tree $=$ paths from root in graph. If cycle formed, end and condition.
- Fix arb ordering on edges. When cycle formed, condition to $0/1$ based on order of incoming vs. outgoing edge.

Self-avoiding walk tree

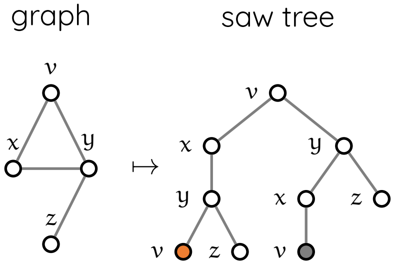


Lemma [Godsil, Weitz, . . .]

Root marginal same on self-avoiding walk tree and graph.

- Tree = paths from root in graph. If cycle formed, end and condition.
- Fix arb ordering on edges. When cycle formed, condition to $0/1$ based on order of incoming vs. outgoing edge.

Self-avoiding walk tree

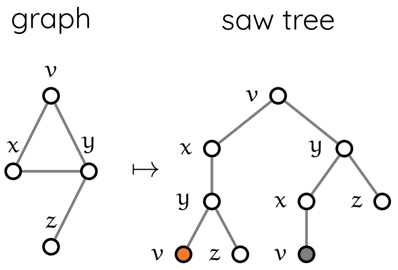


> **Lemma** [Godsil, Weitz, ...]
>
> Root marginal same on self-avoiding walk tree and graph.

- Tree = paths from root in graph. If cycle formed, end and condition.
- Fix arb ordering on edges. When cycle formed, condition to $0/1$ based on order of incoming vs. outgoing edge.
- This means we can compute marginals on tree.

exponentially large though

Self-avoiding walk tree



- Tree = paths from root in graph. If cycle formed, end and condition.
- Fix arb ordering on edges. When cycle formed, condition to $0/1$ based on order of incoming vs. outgoing edge.

Lemma [Godsil, Weitz, . . .]

Root marginal same on self-avoiding walk tree and graph.

- This means we can compute marginals on tree.

  exponentially large though

- Correlation decay allows us to cut the tree height by conditioning nodes at level $\simeq \log n$ arbitrarily.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{\mathsf{TV}}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0, 1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{\mathsf{TV}}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{\mathsf{TV}}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

- SSM allows conditioning nearby verts, as long as we do it consistently.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

- SSM allows conditioning nearby verts, as long as we do it consistently.
- Usually we want exponentially decaying $f(\cdot)$.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

- SSM allows conditioning nearby verts, as long as we do it consistently.
- Usually we want exponentially decaying $f(\cdot)$.

Corollary of saw trees

SSM for trees $\implies$ SSM for graphs

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

- SSM allows conditioning nearby verts, as long as we do it consistently.
- Usually we want exponentially decaying $f(\cdot)$.

Corollary of saw trees

SSM for trees $\implies$ SSM for graphs

- The same analysis we had for trees actually shows strong spatial mixing for $\lambda < \lambda_c$.

Spatial mixing

Let us fix root, and condition subset $S \subseteq V$ of verts two ways: $\sigma, \sigma' \subseteq \{0,1\}^S$.

Weak spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid u \in S\}$.

Strong spatial mixing

For some $f(h) \to 0$ with $h \to \infty$:

$$d_{TV}(\text{root} \mid \sigma, \text{root} \mid \sigma') \leqslant f(h)$$

for $h = \min\{d(\text{root}, u) \mid \sigma(u) \neq \sigma'(u)\}$.

- SSM allows conditioning nearby verts, as long as we do it consistently.
- Usually we want exponentially decaying $f(\cdot)$.

Corollary of saw trees

SSM for trees $\implies$ SSM for graphs

- The same analysis we had for trees actually shows strong spatial mixing for $\lambda < \lambda_c$.
- For any fixed Δ, under SSM, we get an FPTAS.

- Reduce counting to marginal estimation.

- Reduce counting to marginal estimation.
- Form saw tree rooted at v, but truncate it at level $h = O(\log n)$.

- Reduce counting to marginal estimation.
- Form saw tree rooted at v, but truncate it at level $h = O(\log n)$.
- Assign arbitrary conditioning to last level.

- Reduce counting to marginal estimation.
- Form saw tree rooted at v, but truncate it at level $h = O(\log n)$.
- Assign arbitrary conditioning to last level.
- Compute root marginal on truncated tree via recursion.

- Reduce counting to marginal estimation.
- Form saw tree rooted at v, but truncate it at level $h = O(\log n)$.
- Assign arbitrary conditioning to last level.
- Compute root marginal on truncated tree via recursion.
- Size of tree

$$\leqslant \Delta^{O(\log n)} = \mathrm{poly}(n)$$

when Δ is constant. 🙂

- Reduce counting to marginal estimation.
- Form saw tree rooted at v, but truncate it at level $h = O(\log n)$.
- Assign arbitrary conditioning to last level.
- Compute root marginal on truncated tree via recursion.
- Size of tree

$$\leqslant \Delta^{O(\log n)} = \text{poly}(n)$$

  when Δ is constant. ☺
- Marginals are

$$\exp(-\Omega(\text{height})) = 1/\text{poly}(n)$$

  accurate. ☺

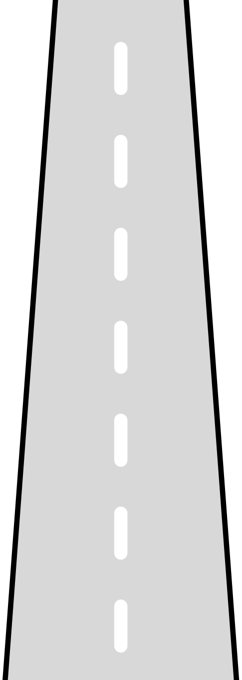
Need to show tree $\equiv$ graph ...

Correlation Decay

- Marginals on trees
- Self-avoiding walk tree
- Weak and strong spatial mixing

HDX via Correlation Decay

- Tree vs graph polynomials

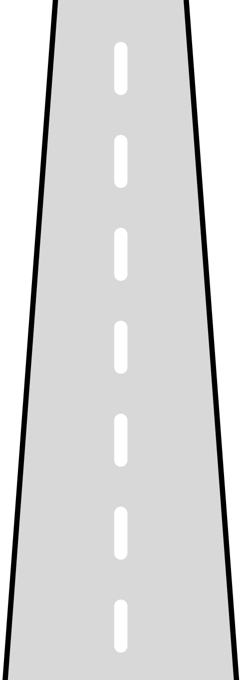


Correlation Decay

- Marginals on trees
- Self-avoiding walk tree
- Weak and strong spatial mixing

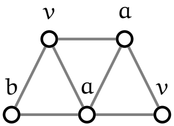
HDX via Correlation Decay

- Tree vs graph polynomials



Polynomials

- Assume we have a 2-spin system with labels on vertices:



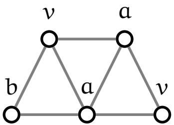
graph G, labels ℓ

$$\Omega = \{0,1\}^n$$

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

Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$$\Omega = \{0, 1\}^n$$

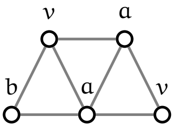
$$\mu(x) = \prod_v \phi_v(x_v) \cdot \prod_{u \sim v} \phi_{uv}(x_u, x_v)$$

- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$$\Omega = \{0,1\}^n$$

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

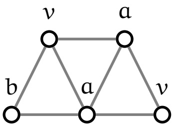
- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

- If pinned vertices, we exclude them from monomial.

Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$\Omega = \{0,1\}^n$

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

- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

- If pinned vertices, we exclude them from monomial.

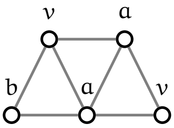
Lemma [Godsil, . . .]

Form the saw tree (inherit labels). There is a polynomial h such that

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$$\Omega = \{0,1\}^n$$

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

- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

- If pinned vertices, we exclude them from monomial.

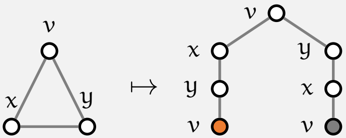
Lemma [Godsil, . . .]

Form the saw tree (inherit labels). There is a polynomial h such that

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

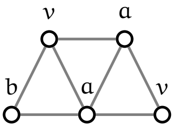
Example

Hardcore model with $\lambda = 1$:



Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$$\Omega = \{0,1\}^n$$

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

- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

- If pinned vertices, we exclude them from monomial.

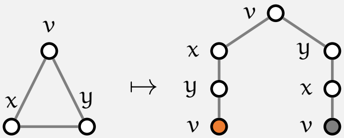
Lemma [Godsil, . . .]

Form the saw tree (inherit labels). There is a polynomial h such that

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

Example

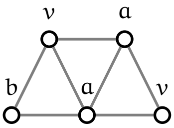
Hardcore model with $\lambda = 1$:



- $g_{\text{graph}} = 1 + z_v + z_x + z_y$

Polynomials

- Assume we have a 2-spin system with labels on vertices:



graph G, labels ℓ

$\Omega = \{0,1\}^n$

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

- Define a (non-hom) polynomial

$$g = \sum_x \mu(x) z_{\ell(1)}^{x_1} \cdots z_{\ell(n)}^{x_n}$$

- If pinned vertices, we exclude them from monomial.

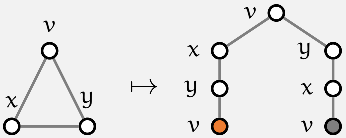
Lemma [Godsil, . . .]

Form the saw tree (inherit labels). There is a polynomial h such that

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

Example

Hardcore model with $\lambda = 1$:



- $g_{\text{graph}} = 1 + z_v + z_x + z_y$
- $g_{\text{tree}} = (1 + z_v + z_x + z_y)(1 + z_x)$

Corollary

Root marginals same on tree/graph.

Corollary

Root marginals same on tree/graph.

Proof:

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.
- We have for $g \in \{g_{\text{graph}}, g_{\text{tree}}\}$:

$$\mathbb{P}_{\text{graph/tree}}[x_v = 1] = \partial_{z_v} \log g|_{z=1}$$

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.
- We have for $g \in \{g_{\text{graph}}, g_{\text{tree}}\}$:

$$\mathbb{P}_{\text{graph/tree}}[x_v = 1] = \partial_{z_v} \log g|_{z=1}$$

- But observe that $\partial_{z_v} \log g_{\text{tree}} =$

$$\partial_{z_v} \log g_{\text{graph}} + \partial_{z_v} \underset{\substack{\uparrow \\ \text{this is } 0}}{\log h}$$

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.
- We have for $g \in \{g_{\text{graph}}, g_{\text{tree}}\}$:

$$\mathbb{P}_{\text{graph/tree}}[x_v = 1] = \partial_{z_v} \log g|_{z=1}$$

- But observe that $\partial_{z_v} \log g_{\text{tree}} =$

$$\partial_{z_v} \log g_{\text{graph}} + \partial_{z_v} \log h$$

this is 0

- It remains to prove the lemma:

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.
- We have for $g \in \{g_{\text{graph}}, g_{\text{tree}}\}$:

$$\mathbb{P}_{\text{graph/tree}}[x_v = 1] = \partial_{z_v} \log g|_{z=1}$$

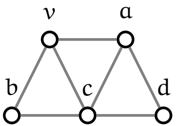
- But observe that $\partial_{z_v} \log g_{\text{tree}} =$

$$\partial_{z_v} \log g_{\text{graph}} + \partial_{z_v} \underbrace{\log h}_{\text{this is } 0}$$

- It remains to prove the lemma:

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

- Assume no univariate/vertex factors in μ for simplicity (e.g., $\lambda = 1$). They can be added later by change of variable $z_v \mapsto \lambda z_v$:



graph $G = ([n], E)$

$\Omega = \{0, 1\}^n$

$$\mu(x) = \prod_{u \sim v} \phi_{uv}(x_u, x_v)$$

Corollary

Root marginals same on tree/graph.

Proof:

- Use distinct labels for graph.
- If v is root, then z_v appears with $\deg \leqslant 1$ in g_{graph} and g_{tree}. So h has no z_v.
- We have for $g \in \{g_{\text{graph}}, g_{\text{tree}}\}$:

$$\mathbb{P}_{\text{graph/tree}}[x_v = 1] = \partial_{z_v} \log g|_{z=1}$$

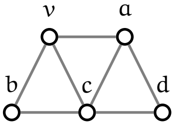
- But observe that $\partial_{z_v} \log g_{\text{tree}} =$

$$\partial_{z_v} \log g_{\text{graph}} + \underset{\text{this is } 0}{\underset{\uparrow}{\partial_{z_v} \log h}}$$

- It remains to prove the lemma:

$$g_{\text{tree}} = g_{\text{graph}} \cdot h$$

- Assume no univariate/vertex factors in μ for simplicity (e.g., $\lambda = 1$). They can be added later by change of variable $z_v \mapsto \lambda z_v$:



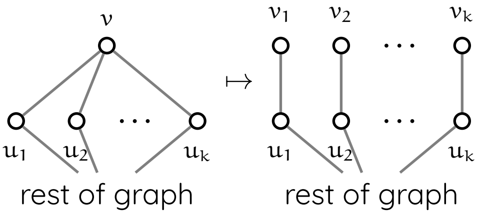
graph $G = ([n], E)$

$\Omega = \{0, 1\}^n$

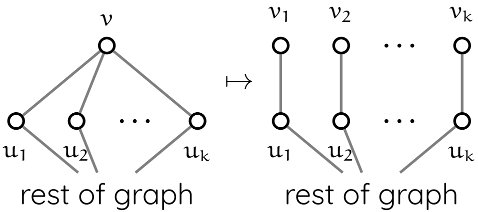
$$\mu(x) = \prod_{u \sim v} \phi_{uv}(x_u, x_v)$$

- We will prove via induction . . .

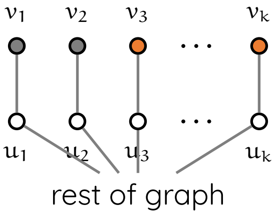
▷ Split root v:


v
u_1 u_2 $\cdots$ u_k
rest of graph
$\mapsto$
v_1 v_2 $\cdots$ v_k
u_1 u_2 $\cdots$ u_k
rest of graph

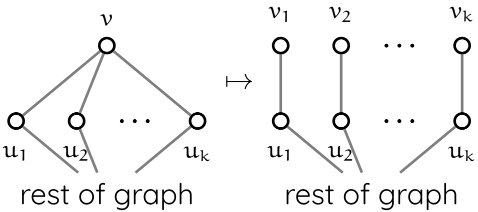
- Split root v:



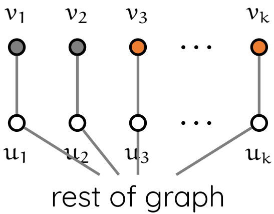
- Pinnings: pin $v_1, \ldots, v_i$ to 0 and $v_{i+1}, \ldots, v_k$ to 1. Let poly be g_i.



- Split root v:



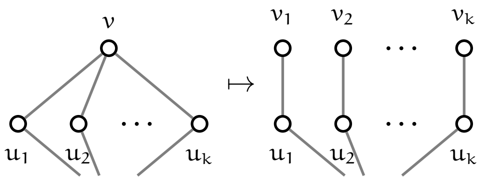
- Pinnings: pin $v_1, \ldots, v_i$ to 0 and $v_{i+1}, \ldots, v_k$ to 1. Let poly be g_i.



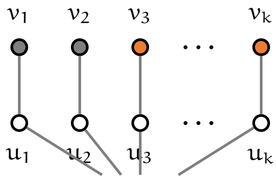
- We have

$$g_{\text{graph}} = z_v g_0 + g_k$$

- Split root v:



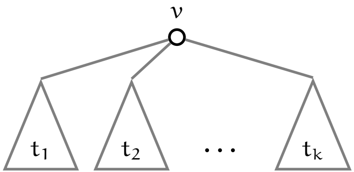
- Pinnings: pin $v_1, \ldots, v_i$ to 0 and $v_{i+1}, \ldots, v_k$ to 1. Let poly be g_i.



- We have

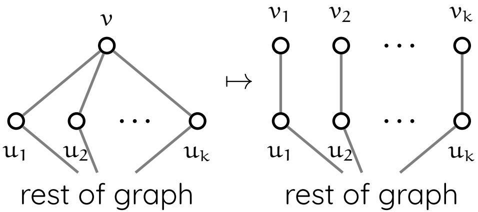
$$g_{\text{graph}} = z_v g_0 + g_k$$

- Now looking at tree

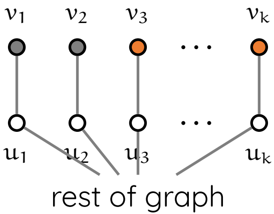


$$g_{\text{tree}} = z_v \prod_i r_i + \prod_i s_i$$

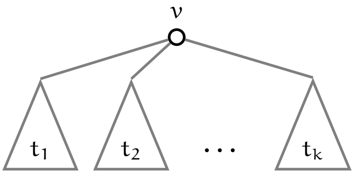
- Split root v:



- Pinnings: pin $v_1, \ldots, v_i$ to 0 and $v_{i+1}, \ldots, v_k$ to 1. Let poly be g_i.



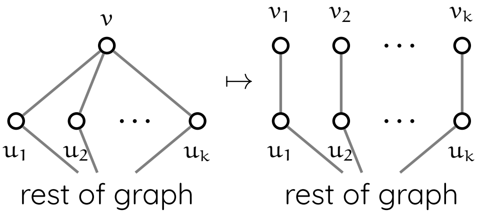
- We have
$$g_{\text{graph}} = z_v g_0 + g_k$$
- Now looking at tree



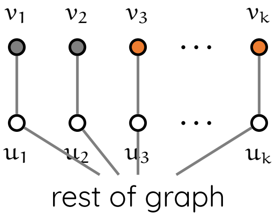
$$g_{\text{tree}} = z_v \prod_i r_i + \prod_i s_i$$

- By induction we have for each i
$$z_v r_i + s_i = h_i \cdot (z_v g_{i-1} + g_i)$$
i.e., $r_i = h_i g_{i-1}$ and $s_i = h_i g_i$.

- Split root v:



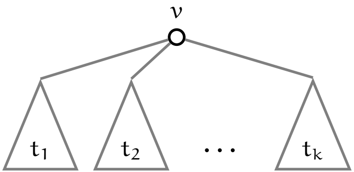
- Pinnings: pin $v_1, \ldots, v_i$ to 0 and $v_{i+1}, \ldots, v_k$ to 1. Let poly be g_i.



- We have

$$g_{\text{graph}} = z_v g_0 + g_k$$

- Now looking at tree



$$g_{\text{tree}} = z_v \prod_i r_i + \prod_i s_i$$

- By induction we have for each i

$$z_v r_i + s_i = h_i \cdot (z_v g_{i-1} + g_i)$$

i.e., $r_i = h_i g_{i-1}$ and $s_i = h_i g_i$.

- Now let $h = g_1 \cdots g_{k-1} \cdot h_1 \cdots h_k$:

$$h g_{\text{graph}} = z_v \prod_i r_i + \prod_i s_i$$

This finishes proof of [Weitz].