Sampling symmetric Gibbs distributions on sparse random graphs with contiguity

Charis Efthymiou
University of Warwick

Workshop on Inference Problems
Goethe University of Frankfurt, 02/09/2020

Gibbs distribution

Gibbs distribution

- spin configurations on the vertices of a graph

Gibbs distribution

- spin configurations on the vertices of a graph
  - graph G=(V,E) and set of spins $\mathcal{S}$

Gibbs distribution

- spin configurations on the vertices of a graph
  - graph G=(V,E) and set of spins $\mathcal{S}$
  - configuration space $\mathcal{S}^V$

Gibbs distribution

- spin configurations on the vertices of a graph
  - graph G=(V,E) and set of spins $\mathcal{S}$
  - configuration space $\mathcal{S}^V$
- for each configuration σ specify $\texttt{weight}(\sigma)$

Gibbs distribution

- spin configurations on the vertices of a graph
  - graph G=(V,E) and set of spins $\mathcal{S}$
  - configuration space $\mathcal{S}^V$
- for each configuration σ specify $\texttt{weight}(\sigma)$
- configuration σ is assigned probability measure

$$\mu(\sigma) \propto \texttt{weight}(\sigma)$$

Example

Example

Potts model

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$
- for each $\sigma \in \mathcal{S}^V$ we have (σ is a q-colouring)

$$\texttt{weight}(\sigma) = \exp(\beta \times \#\text{monochromatic-edges})$$

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$
- for each $\sigma \in \mathcal{S}^V$ we have (σ is a q-colouring)

$$\texttt{weight}(\sigma) = \exp(\beta \times \#\text{monochromatic-edges})$$

- for $\beta < 0$ penalises monochromatic edges - antiferromagnetic

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$
- for each $\sigma \in \mathcal{S}^V$ we have (σ is a q-colouring)

$$\texttt{weight}(\sigma) = \exp(\beta \times \#\text{monochromatic-edges})$$

- for $\beta < 0$ penalises monochromatic edges - antiferromagnetic
- for $\beta > 0$ favours monochromatic edges - ferromagnetic

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$
- for each $\sigma \in \mathcal{S}^V$ we have (σ is a q-colouring)

$$\texttt{weight}(\sigma) = \exp(\beta \times \#\text{monochromatic-edges})$$

- for $\beta < 0$ penalises monochromatic edges - antiferromagnetic
- for $\beta > 0$ favours monochromatic edges - ferromagnetic

Remarks

- for $q = 2$ we have the Ising model

Example

Potts model

- $G = (V, E)$, $\mathcal{S} = \{1, 2, \ldots, q\}$ and $\beta \in \mathbb{R} \cup \{\pm\infty\}$
- for each $\sigma \in \mathcal{S}^V$ we have (σ is a q-colouring)

$$\texttt{weight}(\sigma) = \exp(\beta \times \#\text{monochromatic-edges})$$

- for $\beta < 0$ penalises monochromatic edges - antiferromagnetic
- for $\beta > 0$ favours monochromatic edges - ferromagnetic

Remarks

- for $q = 2$ we have the Ising model
- for $\beta = -\infty$ we have the Colouring model

Efficient sampling

Efficient sampling

For the Gibbs distribution μ on $G = (V, E)$, generate *efficiently* the configuration $\boldsymbol{\sigma} \sim \mu$

Efficient sampling

For the Gibbs distribution μ on $G = (V, E)$, generate *efficiently* the configuration $\boldsymbol{\sigma} \sim \mu$

- worst-case the problem is computationally hard

Efficient sampling

For the Gibbs distribution μ on $G = (V, E)$, generate *efficiently* the configuration $\boldsymbol{\sigma} \sim \mu$

- worst-case the problem is computationally hard
- generate efficiently $\boldsymbol{\sigma}$ which is distributed "close" to μ

Efficient sampling

For the Gibbs distribution μ on $G = (V, E)$, generate *efficiently* the configuration $\boldsymbol{\sigma} \sim \mu$

- worst-case the problem is computationally hard
- generate efficiently $\boldsymbol{\sigma}$ which is distributed "close" to μ
- focus on the range of parameters of μ in which we can get "good" approximate samples

The case of $G(n, m)$

The case of $G(n, m)$

The sparse random graph

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$
- we focus on fixed $d > 0$ and $n \to \infty$

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$
- we focus on fixed $d > 0$ and $n \to \infty$
  - ... that is $m = \Theta(n)$

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$
- we focus on fixed $d > 0$ and $n \to \infty$
  - ... that is $m = \Theta(n)$

Sampling Problem on $G(n, m)$

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$
- we focus on fixed $d > 0$ and $n \to \infty$
  - ... that is $m = \Theta(n)$

Sampling Problem on $G(n, m)$

- focus on approximate sampling

The case of $G(n, m)$

The sparse random graph

$G(n, m)$ is the random graph on n vertices and m edges

- expected degree d, i.e. $m = \frac{dn}{2}$
- we focus on fixed $d > 0$ and $n \to \infty$
  - ... that is $m = \Theta(n)$

Sampling Problem on $G(n, m)$

- focus on approximate sampling
- use concepts from physics and related problems on random graphs to get better sampling algorithms

Popular approaches to sampling problem

Popular approaches to sampling problem

- Markov Chain Monte Carlo method

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
- Message Passing Algorithms

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
- Message Passing Algorithms
- Weitz's Algorithm

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
- Message Passing Algorithms
- Weitz's Algorithm
- Barvinok's approach

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
- Message Passing Algorithms
- Weitz's Algorithm
- Barvinok's approach
- See next talk for another ...

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
- Message Passing Algorithms
- Weitz's Algorithm
- Barvinok's approach
- See next talk for another ...

Our approach has nothing to do with all the above ...

Example from the past

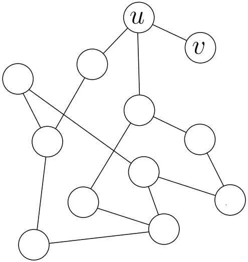
Example with the Colouring Model

Example with the Colouring Model

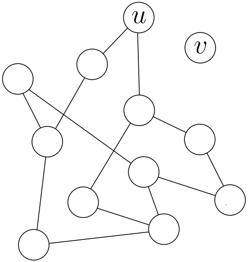
- from (Efthymiou 2012).

Observation

Observation

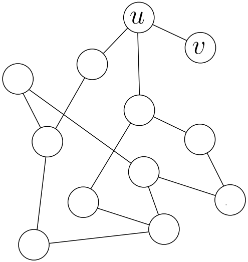


G

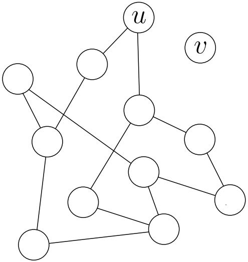


G'

Observation



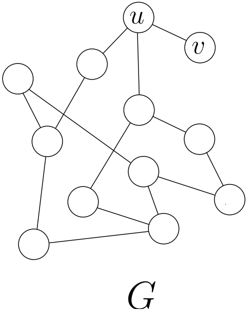
G



G'

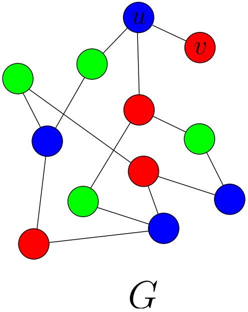
A random colouring of G can be seen as a random colouring of the **simpler** G' conditional that v, u receive **different colours**.

Observation



A random colouring of G can be seen as a random colouring of the **simpler** G' conditional that v, u receive **different colours**.

Observation



A random colouring of G can be seen as a random colouring of the **simpler** G' conditional that v, u receive **different colours**.

Suppose that ....

Update

Input: **random** q-colouring of G and the **vertices** v, u.

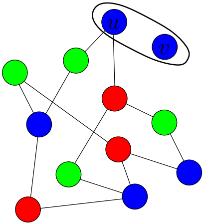
Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.

Suppose that ....

Update

Input: **random** q-colouring of G and the **vertices** v, u.

Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.

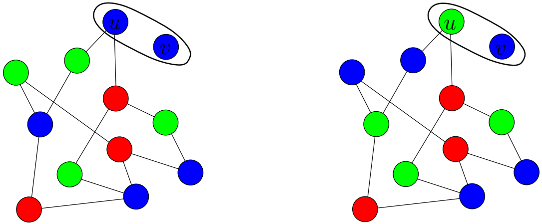


Suppose that ....

Update

Input: **random** q-colouring of G and the **vertices** v, u.

Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.

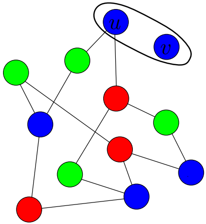


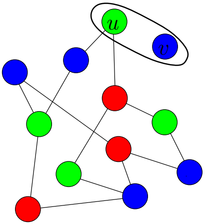
Suppose that ....

Update

Input: **random** q-colouring of G and the **vertices** v, u.

Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.





Be careful...

We can not change the colours of the vertices **arbitrarily**.

`Update` for sampling . . .

Update for sampling . . .

The sampling algorithm

Update for sampling . . .

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

Update for sampling . . .

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

Update for sampling . . .

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Update for sampling . . .

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and Update to get $\boldsymbol{\sigma}_{i+1}$

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

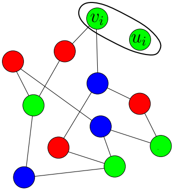
$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$



Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

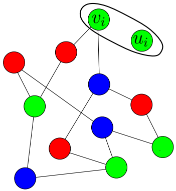
$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$



UPDATE

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

Update for sampling ...

The sampling algorithm

Input: $G = (V, E)$ $q > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

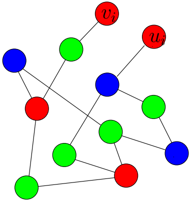
Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

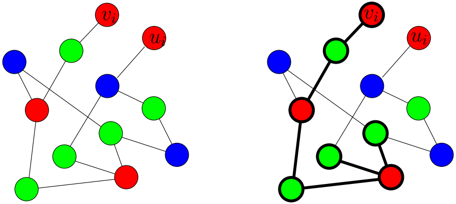
Output: $\boldsymbol{\sigma}_r$, the colouring of G_r

How does `Update` look like for $G(n, m)$?

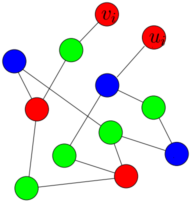
How does `Update` look like for $G(n, m)$?



How does `Update` look like for $G(n, m)$?

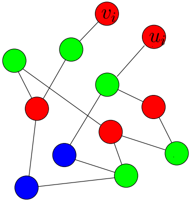


How does `Update` look like for $G(n, m)$?

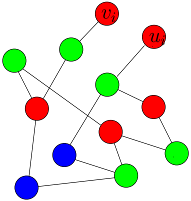


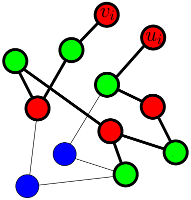
... why approximate sampling?

... why approximate sampling?

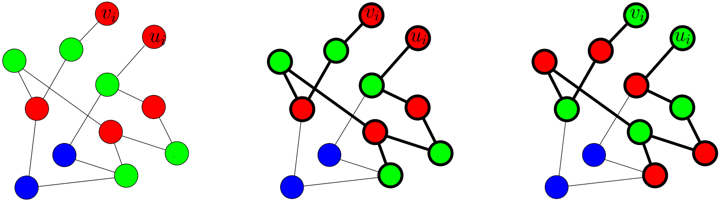


… why approximate sampling?

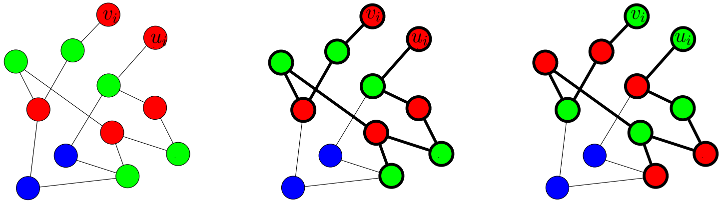




... why approximate sampling?



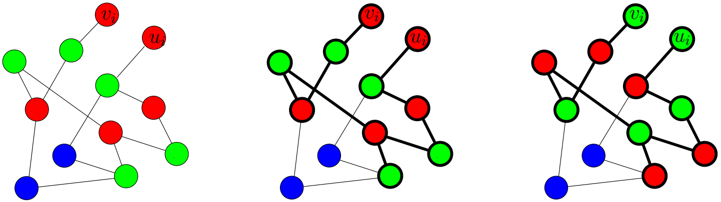
… why approximate sampling?



Failure

When both v_i and u_i change colour `Update` fails

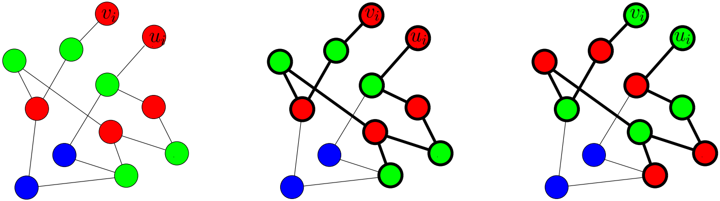
... why approximate sampling?



Failure Vs Approximation

Because of the failures `Update` is an approximation algorithm

... why approximate sampling?

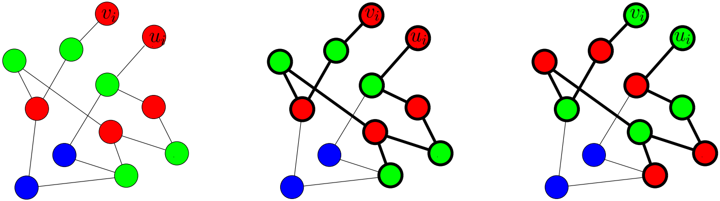


Failure Vs Approximation

Because of the failures `Update` is an approximation algorithm

- the output is approximately Gibbs distributed

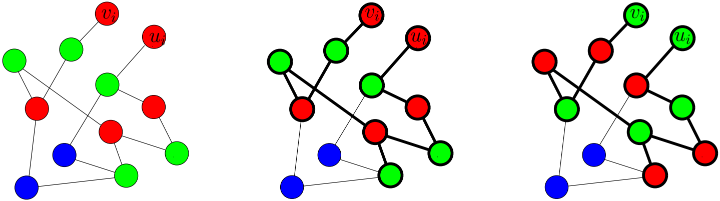
… why approximate sampling?



Error for `Update`

≈ the probability of failure

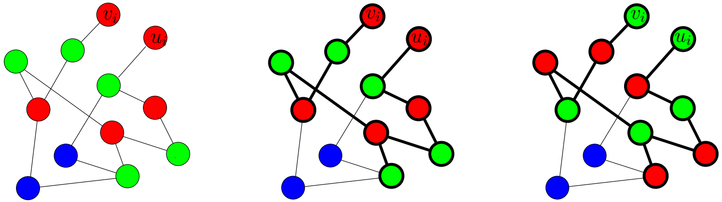
... why approximate sampling?



Approximation Sampler

The sampling algorithm that uses `Update` is approximation too

... why approximate sampling?



Approximation Sampler

The sampling algorithm that uses `Update` is approximation too

$$\text{output error} \approx \text{there is a failure is some iteration}$$

Some intuition for $G(n, m)$

Some intuition for $G(n, m)$

- for certain values of q the approach yields good approximation

Some intuition for $G(n, m)$

- for certain values of q the approach yields good approximation
- almost all pairs v_i, u_i are far away

Some intuition for $G(n, m)$

- for certain values of q the approach yields good approximation
- almost all pairs v_i, u_i are far away
    - failure implies that we have an *extensive* chain

Some intuition for $G(n, m)$

- for certain values of q the approach yields good approximation
- almost all pairs v_i, u_i are far away
  - failure implies that we have an *extensive* chain
- care should be taken for v_i, u_i are at short distance

Some intuition for $G(n, m)$

- for certain values of q the approach yields good approximation
- almost all pairs v_i, u_i are far away
  - failure implies that we have an *extensive* chain
- care should be taken for v_i, u_i are at short distance
  - the update for such pairs is different (didn't show that)

Performance of Sampler

Theorem (Efthymiou 2016)

For $\epsilon > 0$, for large $d > 0$ and $q \geq (1+\epsilon)d$ we have the following: With probability $1 - o(1)$ over the instances of $\boldsymbol{G}(n, m)$ the algorithm generates a q-colouring of $\boldsymbol{G}$ which is distributed within total variation distance $n^{-\Omega(1)}$ from the q-colouring model.

Gibbs Tree Uniqueness

Gibbs Tree Uniqueness

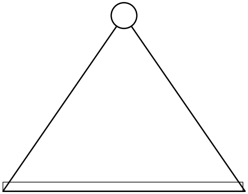
Remark

$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree

Gibbs Tree Uniqueness

Remark

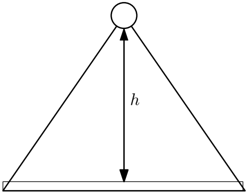
$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



Gibbs Tree Uniqueness

Remark

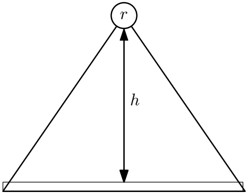
$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



Gibbs Tree Uniqueness

Remark

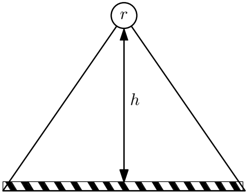
$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



Gibbs Tree Uniqueness

Remark

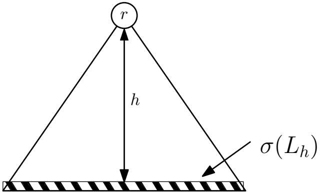
$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



Gibbs Tree Uniqueness

Remark

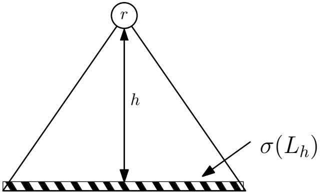
$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



Gibbs Tree Uniqueness

Remark

$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree

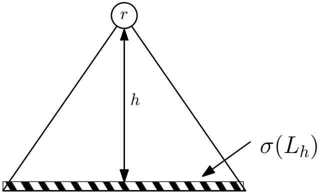


$$||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

Gibbs Tree Uniqueness

Remark

$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree

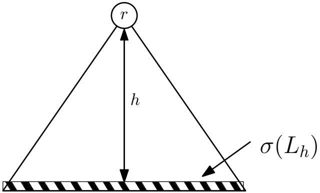


$$\lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

Gibbs Tree Uniqueness

Remark

$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree

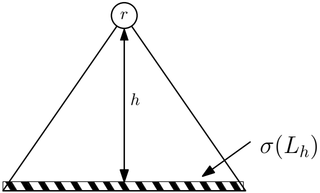


$$\lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}} = \begin{cases} 0 \\ \delta > 0 \end{cases}$$

Gibbs Tree Uniqueness

Remark

$q > d \Rightarrow$ **Gibbs uniqueness** for the colourings on the d-ary tree



$$\lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}} = \begin{cases} 0 \\ \delta > 0 \end{cases}$$

Gibbs Uniqueness $\iff \lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}} = 0$

Beyond colourings

Beyond colourings

similar approach other distributions

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution
- the previous approach is *tailored* to colourings

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution
- the previous approach is *tailored* to colourings
  - already fails for the interesting cases of Potts model

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution
- the previous approach is *tailored* to colourings
  - already fails for the interesting cases of Potts model
- (Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang '20)

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution
- the previous approach is *tailored* to colourings
  - already fails for the interesting cases of Potts model
- (Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang '20)
  - use here the sampler for the random cluster model

Beyond colourings

similar approach other distributions

- add edges, use update, failure etc ...
- general update independent of the distribution
- the previous approach is *tailored* to colourings
  - already fails for the interesting cases of Potts model
- (Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang '20)
  - use here the sampler for the random cluster model
  - special to distribution, too

Symmetric Gibbs distributions

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model
  - including colourings

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model
  - including colourings
- k not-all-equal SAT

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model
    - including colourings
- k not-all-equal SAT
- k-spin model for $k \geq 2$ even integer

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model
  - including colourings
- k not-all-equal SAT
- k-spin model for $k \geq 2$ even integer
  - spin-glass distribution

Symmetric Gibbs distributions

propose a sampler for symmetric distributions

Includes . . .

- Ising model
- Potts model
    - including colourings
- k not-all-equal SAT
- k-spin model for $k \geq 2$ even integer
    - spin-glass distribution

Remark

The above are for both graphs and hypergraphs

Range of parameters

Range of parameters

The aim is the whole tree uniqueness region

Range of parameters

The aim is the whole tree uniqueness region

- for many cases we only have conjectures about the region

Range of parameters

The aim is the whole tree uniqueness region

- for many cases we only have conjectures about the region
  - can't use the convergence of distributional recursions

Range of parameters

The aim is the whole tree uniqueness region

- for many cases we only have conjectures about the region
  - can't use the convergence of distributional recursions
- circumvent the above by exploiting **contiguity**

Range of parameters

The aim is the whole tree uniqueness region

- for many cases we only have conjectures about the region
  - can't use the convergence of distributional recursions
- circumvent the above by exploiting **contiguity**
  - from network inference algorithms e.g. for the Stochastic Block Model

Approach

Approach

Setting . . .

Approach

Setting . . .

- symmetric Gibbs distribution
  - . . . e.g. antiferromagnetic Ising or Potts

Approach

Setting . . .

- symmetric Gibbs distribution
  - . . . e.g. antiferromagnetic Ising or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$ for some edge e

Approach

Setting ...

- symmetric Gibbs distribution
  - ... e.g. antiferromagnetic Ising or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$ for some edge e
  - ... assume that both are of **high girth**

Approach

Setting ...

- symmetric Gibbs distribution
  - ... e.g. antiferromagnetic Ising or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$ for some edge e
  - ... assume that both are of **high girth**
- Gibbs distributions μ and μ' on G and G', resp.

Approach

Setting ...

- symmetric Gibbs distribution
  - ... e.g. antiferromagnetic Ising or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$ for some edge e
  - ... assume that both are of **high girth**
- Gibbs distributions μ and μ' on G and G', resp.
- configuration $\boldsymbol{\sigma}$ distributed as in μ

Approach

Setting . . .

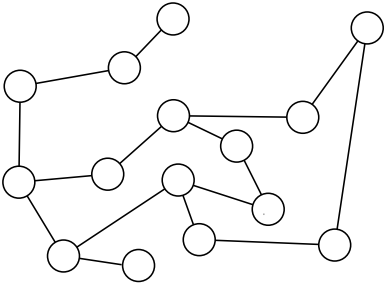
- symmetric Gibbs distribution
  - . . . e.g. antiferromagnetic Ising or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$ for some edge e
  - . . . assume that both are of **high girth**
- Gibbs distributions μ and μ' on G and G', resp.
- configuration $\boldsymbol{\sigma}$ distributed as in μ

Objective

Generate efficiently $\boldsymbol{\tau}$ distributed close to μ'

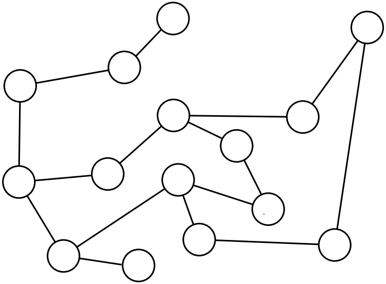
Ideal Solution

Ideal Solution

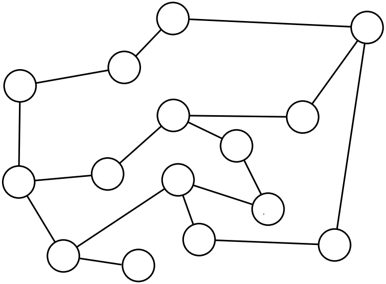


G

Ideal Solution

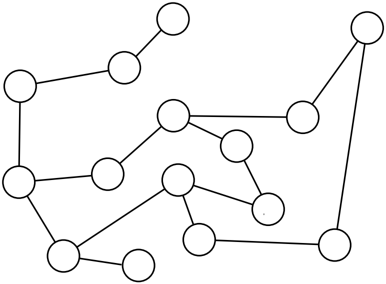


G



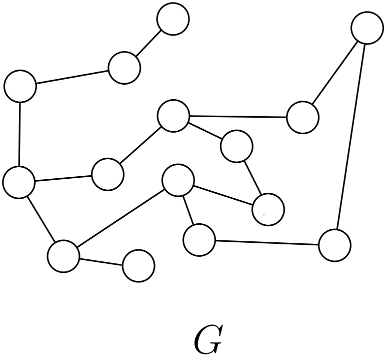
G'

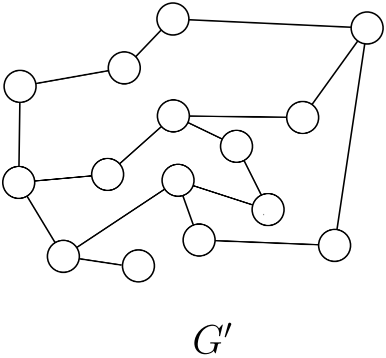
Ideal Solution



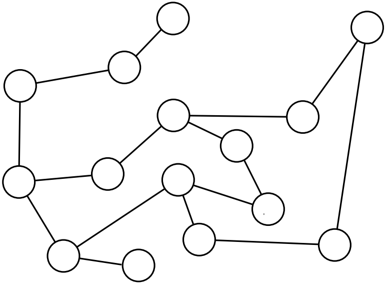
G G'

Ideal Solution

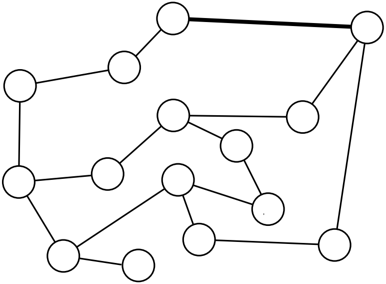




Ideal Solution

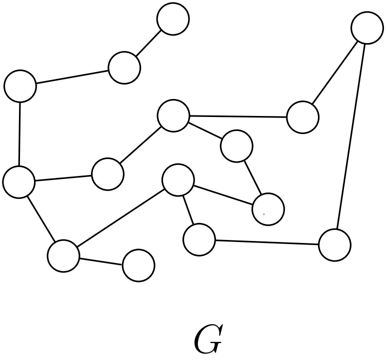


G

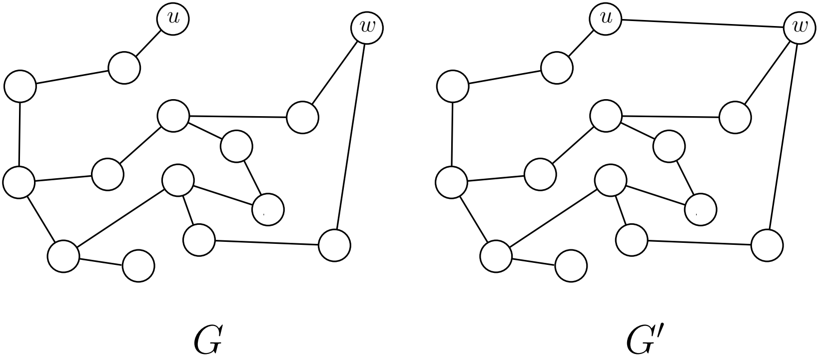


G'

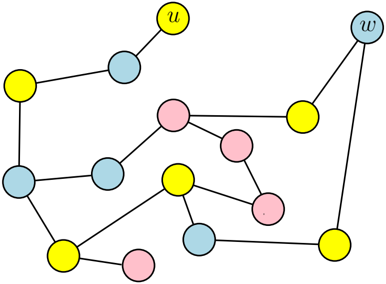
Ideal Solution



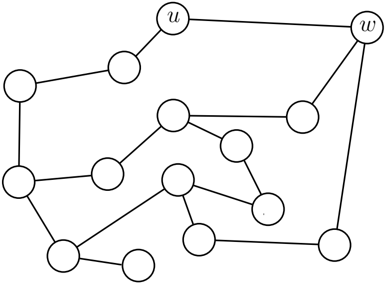
Ideal Solution



Ideal Solution

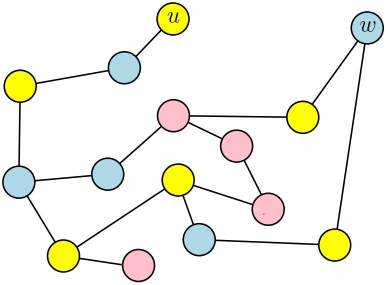


(G, σ)

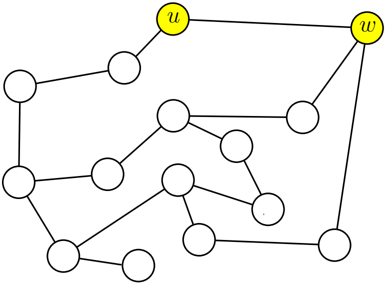


G'

Ideal Solution

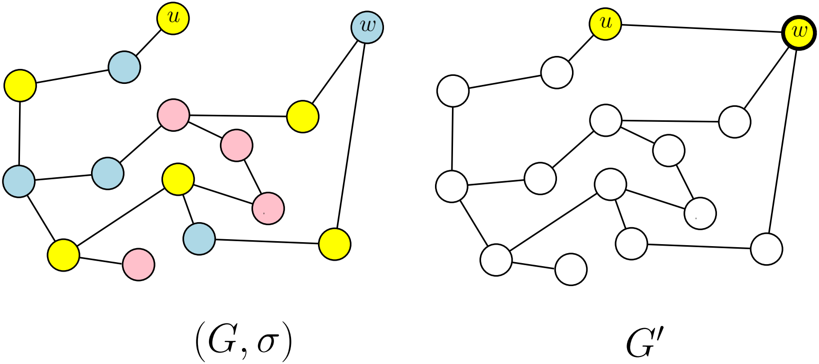


(G, σ)



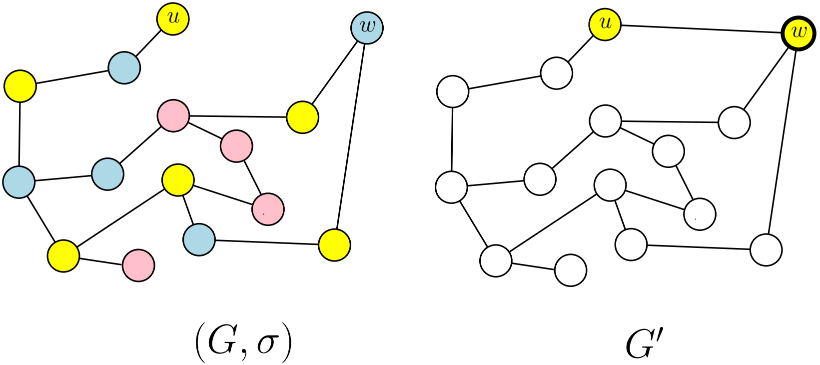
G'

Ideal Solution



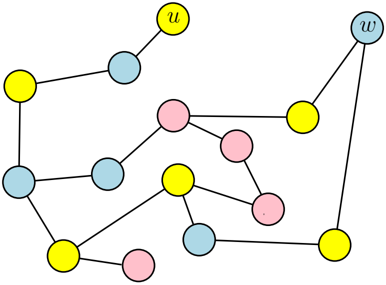
vertex w is a **disagreement** with spins $\{\texttt{blue}, \texttt{yellow}\}$

Ideal Solution

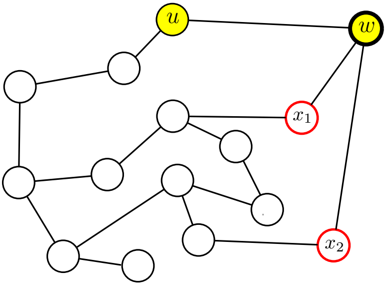


iteratively visit the vertices of G' one by one and decide their configuration at τ

Ideal Solution



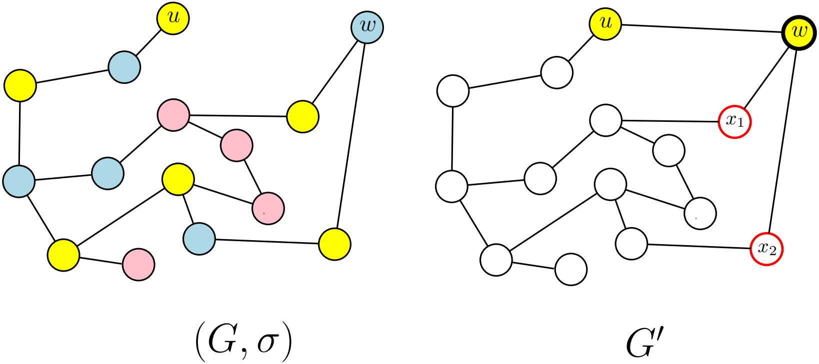
(G, σ)



G'

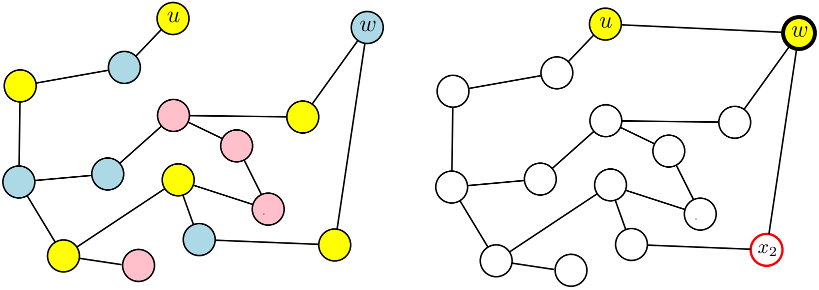
look for z's, neighbours of w with $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

Ideal Solution



look for z's, neighbours of w with $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

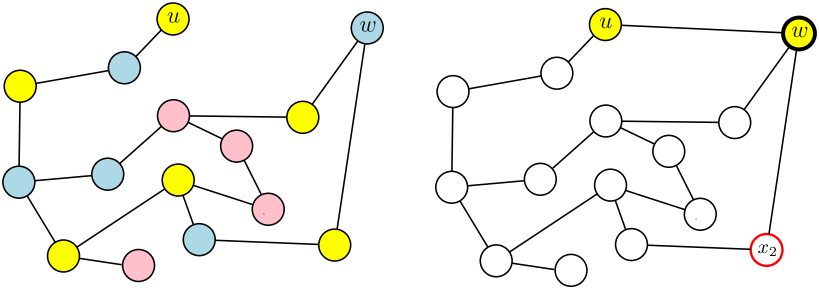
Ideal Solution



(G, σ) G'

pick x_2 and decide $\tau(x_2)$ such that $\tau(x_2) \in \{\texttt{blue}, \texttt{yellow}\}$

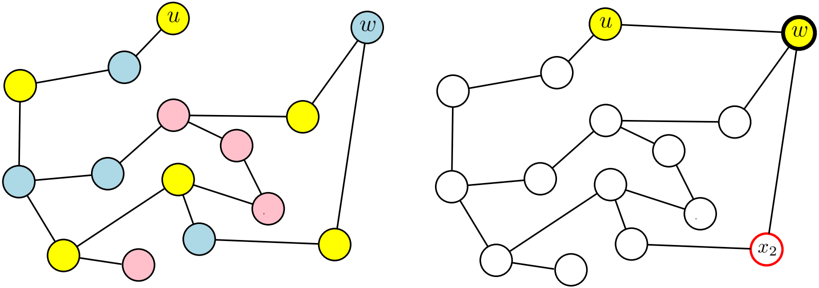
Ideal Solution



(G, σ) $\quad$ G'

we want to avoid a disagreement at x_2

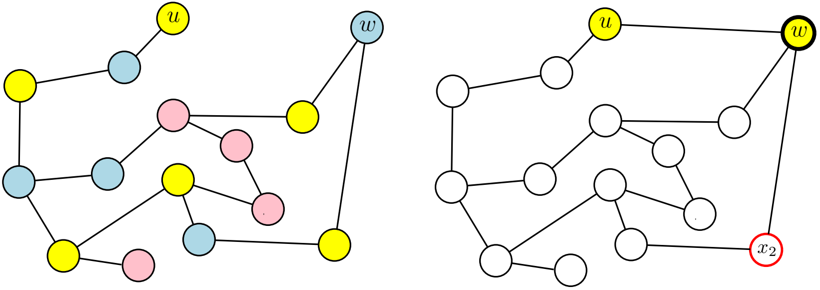
Ideal Solution



(G, σ) G'

the probability of disagreement is minimised by **coupling maximally** the marginals of μ and μ' on x_2

Ideal Solution

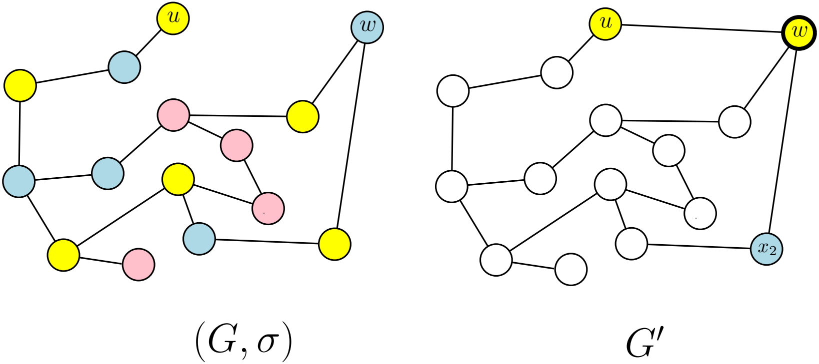


(G, σ) G'

maximal coupling

$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mu'_{x_2}(\sigma(x_2) \mid \tau(\{u,w\}))}{\mu_{x_2}(\sigma(x_2) \mid \sigma(\{u,w\}))}\right\}.$$

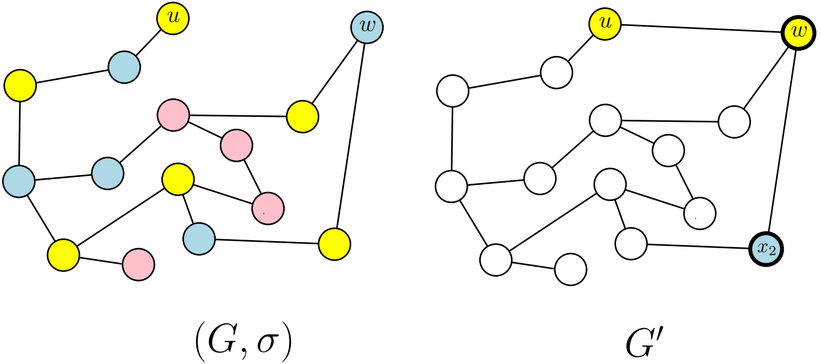
Ideal Solution



maximal coupling

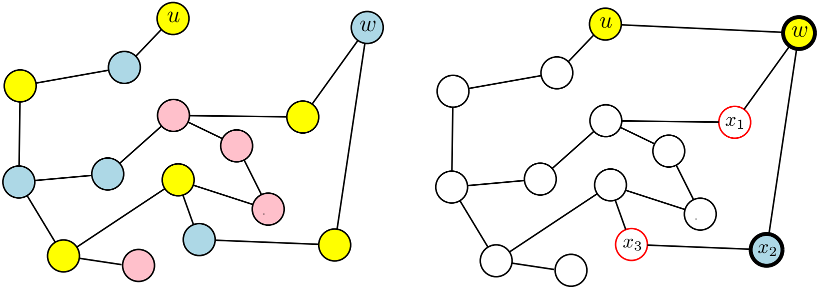
$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mu'_{x_2}(\sigma(x_2) \mid \tau(\{u,w\}))}{\mu_{x_2}(\sigma(x_2) \mid \sigma(\{u,w\}))}\right\}.$$

Ideal Solution



the disagreement set now is $\{w, x_2\}$

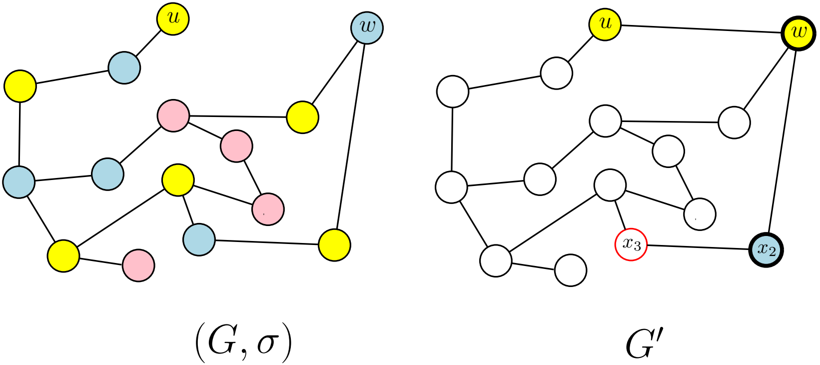
Ideal Solution



(G, σ) $\quad\quad$ G'

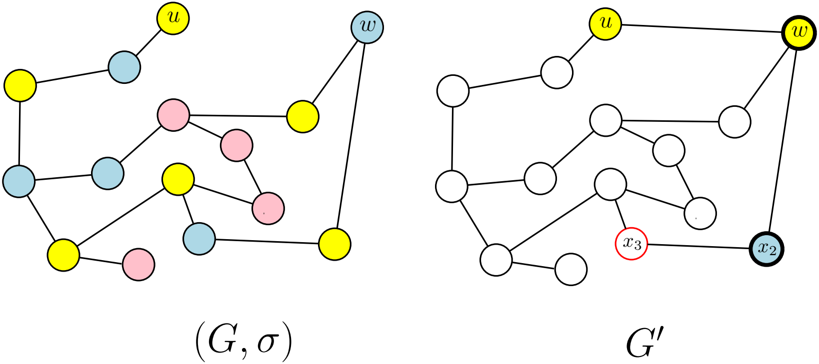
look for vertices z next to the disagreements such that
$\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

Ideal Solution



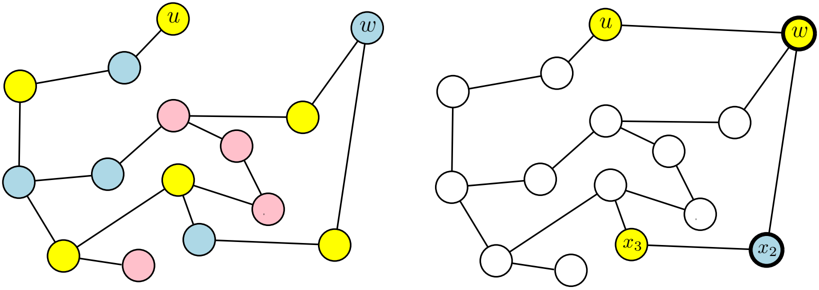
choose x_3 and repeat as before . . .

Ideal Solution



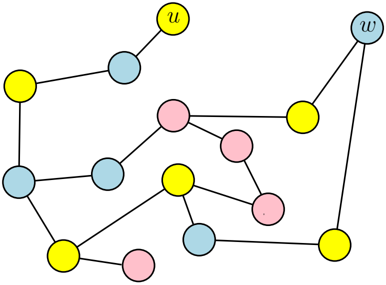
$$\Pr[\tau(x_3) = \texttt{yellow}] = \max\left\{0, 1 - \frac{\mu'_{x_3}(\sigma(x_3) \mid \tau(\{u,w,x_2\}))}{\mu_{x_3}(\sigma(x_3) \mid \sigma(\{u,w,x_2\}))}\right\}.$$

Ideal Solution

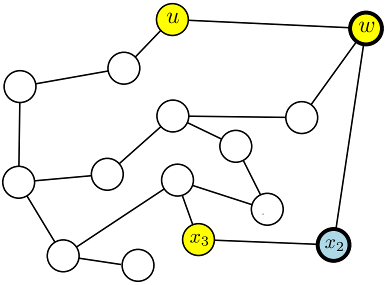


(G, σ) G'

Ideal Solution



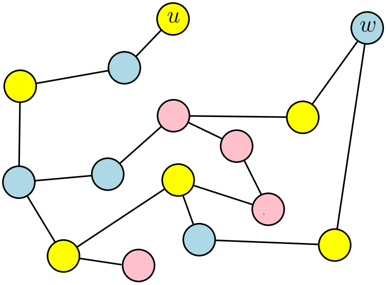
(G, σ)



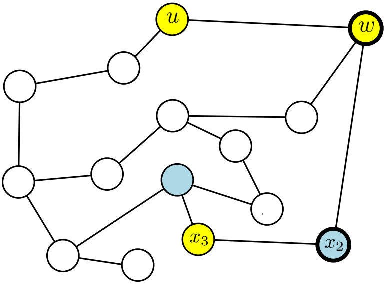
G'

repeat in the same way for the rest of the vertices

Ideal Solution



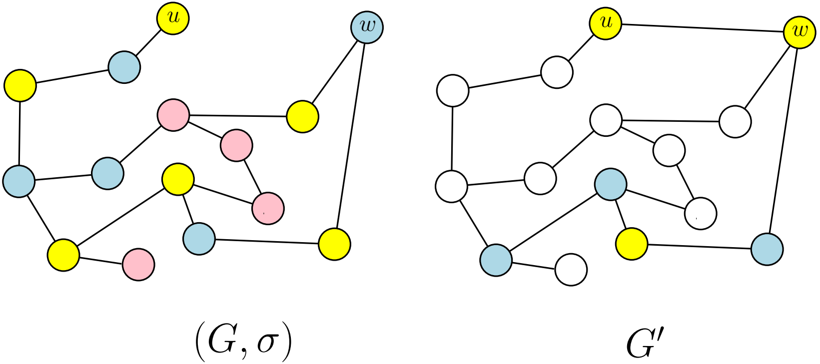
(G, σ)



G'

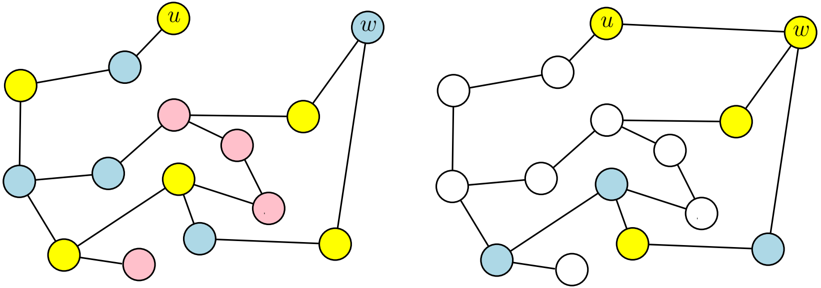
repeat in the same way for the rest of the vertices

Ideal Solution



repeat in the same way for the rest of the vertices

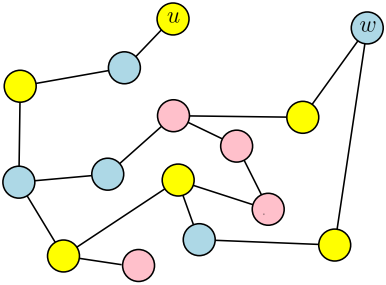
Ideal Solution



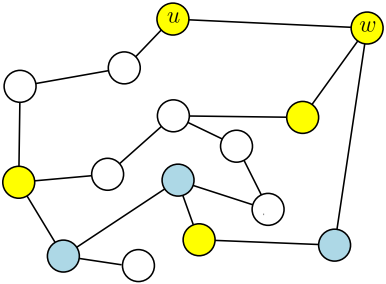
(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

Ideal Solution



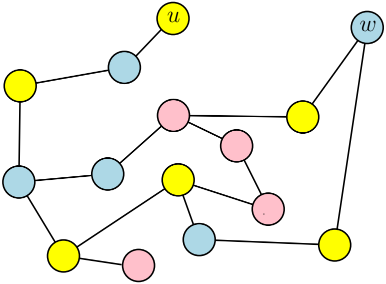
(G, σ)



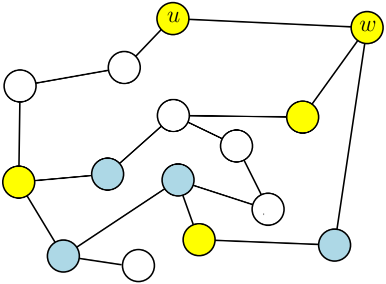
G'

repeat in the same way for the rest of the vertices

Ideal Solution



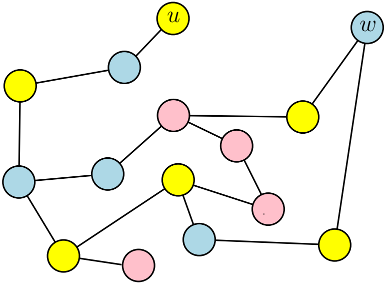
(G, σ)



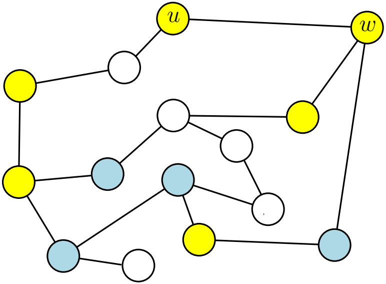
G'

repeat in the same way for the rest of the vertices

Ideal Solution



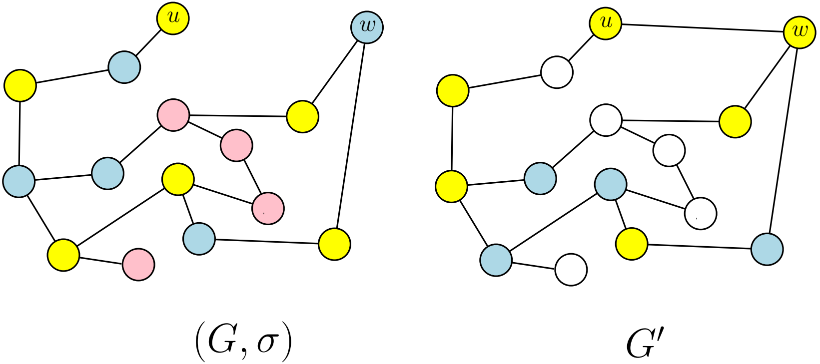
(G, σ)



G'

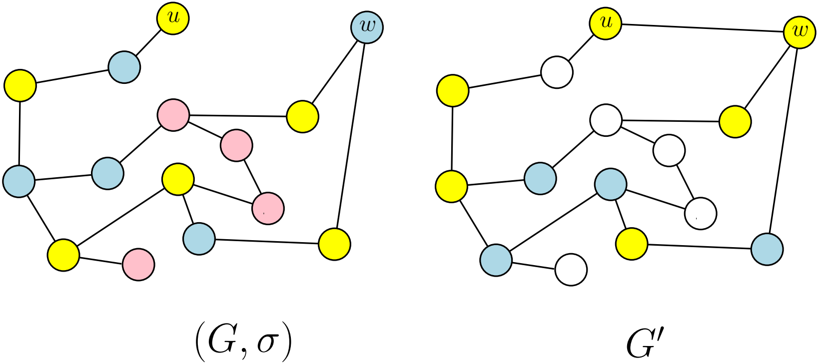
repeat in the same way for the rest of the vertices

Ideal Solution



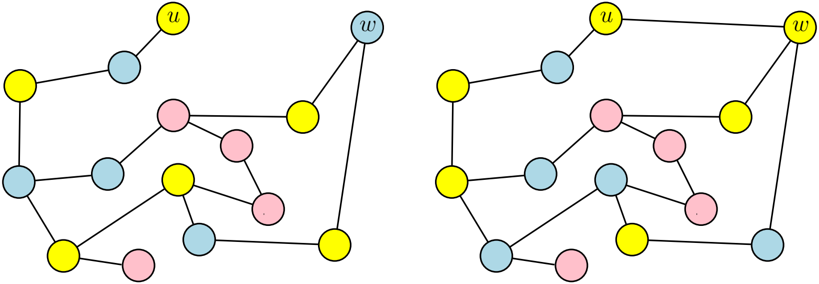
disagreement cannot propagate any more

Ideal Solution



the remaining vertices keep their assignments

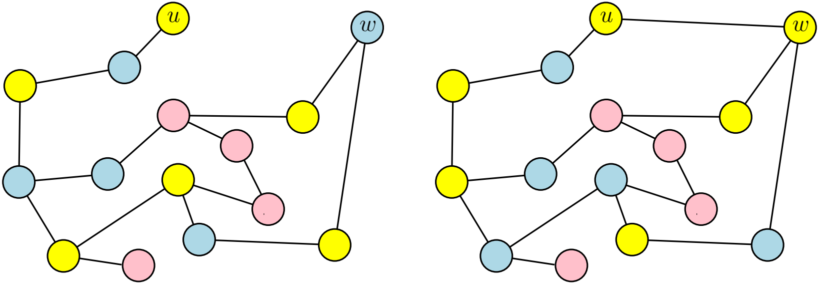
Ideal Solution



(G, σ) (G', τ)

the remaining vertices keep the initial assignments.

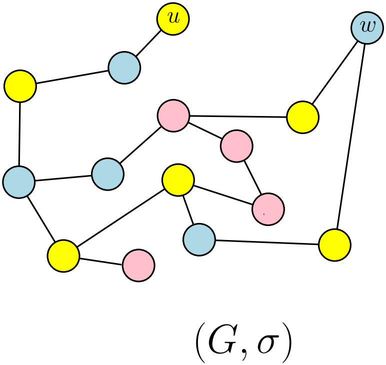
Ideal Solution

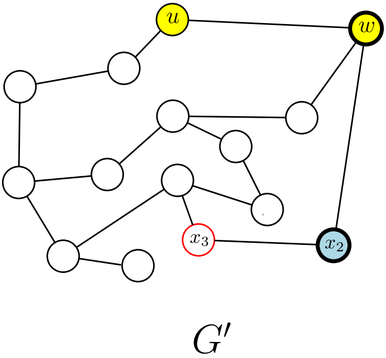


(G, σ) (G', τ)

the approach generates a **perfect** sample from μ'

Ideal Solution

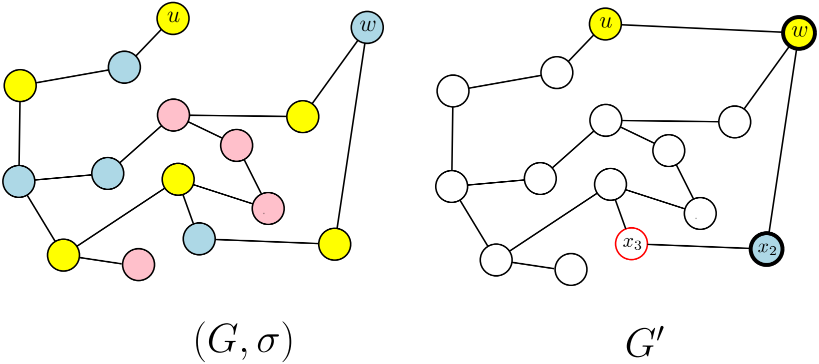




The catch ...

$$\Pr[\tau(x_3) = \texttt{yellow}] = \max\left\{0, 1 - \frac{\mu'_{x_3}(\sigma(x_3) \mid \tau(\{u,w,x_2\}))}{\mu_{x_3}(\sigma(x_3) \mid \sigma(\{u,w,x_2\}))}\right\}.$$

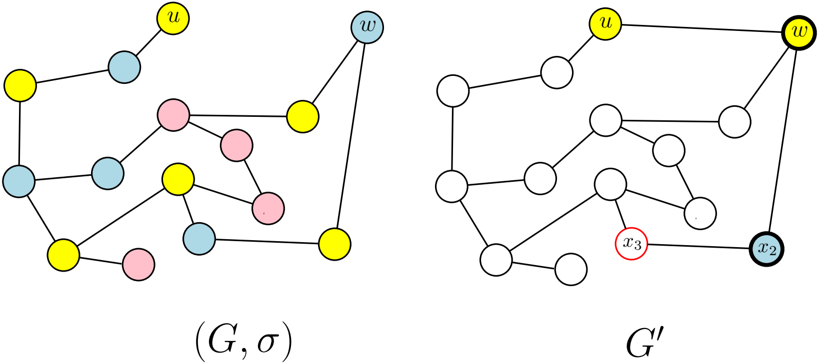
Ideal Solution



The catch ...

we need to compute $\mu'_{x_3}(\sigma(x_3) \mid \sigma(\{u, w, x_2\}))$ efficiently

Ideal Solution

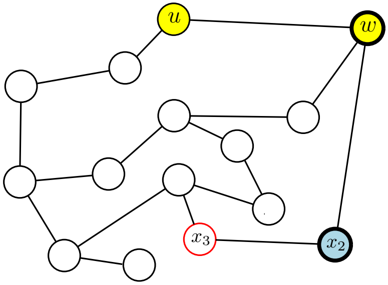


The idea . . .

replace the Gibbs marginals with “good” approximations

Some Intuition

Some Intuition

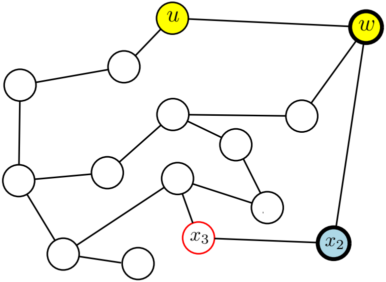


G'

Desideratum ...

compute efficiently the Gibbs marginal $\mu'_{x_3}(\sigma(x_3) \mid \sigma(\{u, w, x_2\}))$

Some Intuition

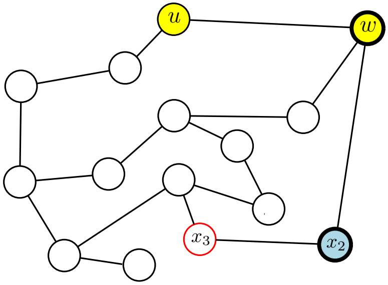


G'

Remark

the marginal $\mu'_{x_3}(\sigma(x_3) \mid \sigma(\{u, w, x_2\}))$ is a "complicated object"

Some Intuition

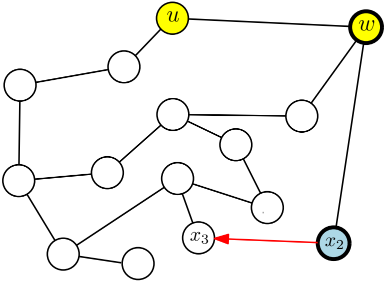


G'

Observation ...

influences form vertices with configuration make the Gibbs marginal at x_3 complicated

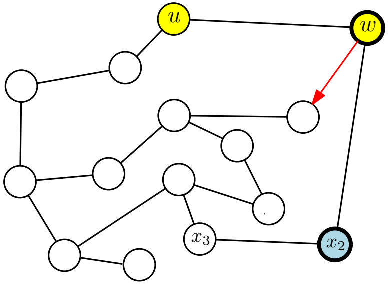
Some Intuition



G'

influence from the configuration at x_2

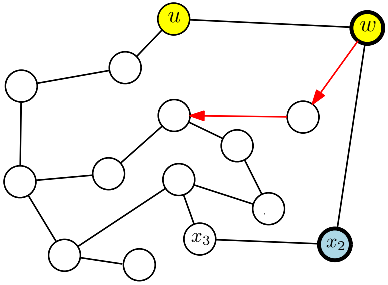
Some Intuition



G'

influence from the configuration at w

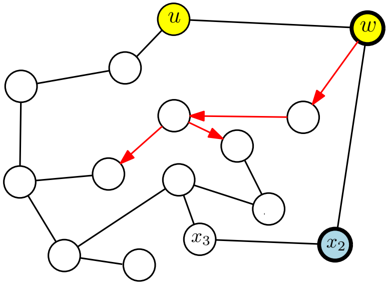
Some Intuition



G'

influence from the configuration at w

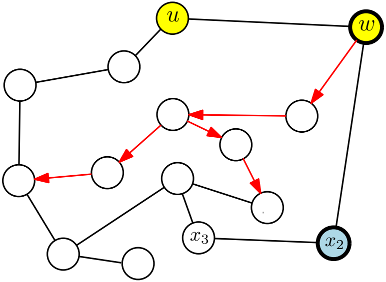
Some Intuition



G'

influence from the configuration at w

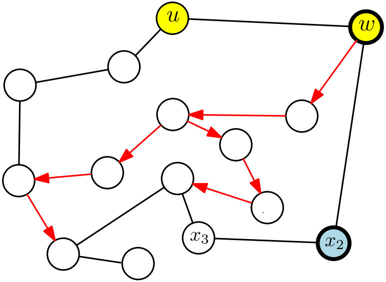
Some Intuition



G'

influence from the configuration at w

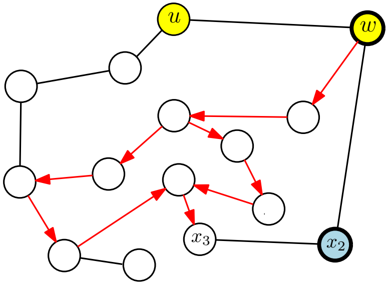
Some Intuition



G'

influence from the configuration at w

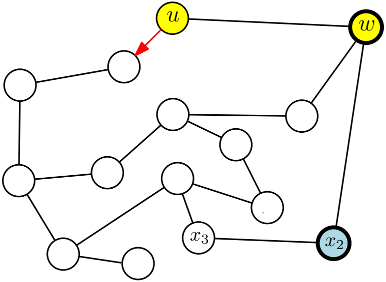
Some Intuition



G'

influence from the configuration at w

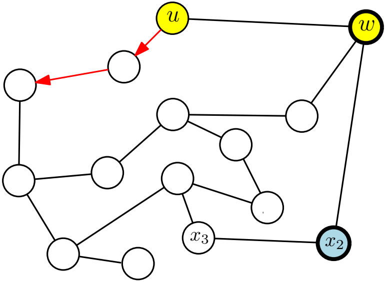
Some Intuition



G'

influence from the configuration at u

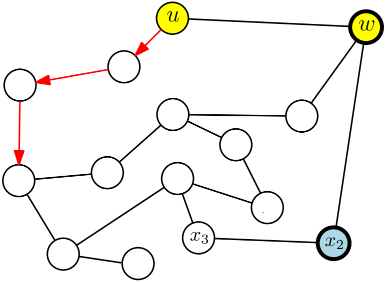
Some Intuition



G'

influence from the configuration at u

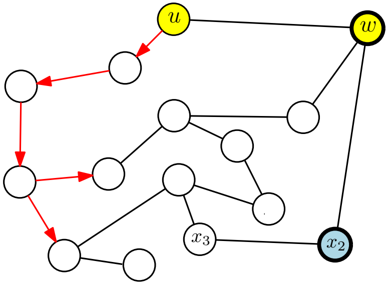
Some Intuition



G'

influence from the configuration at u

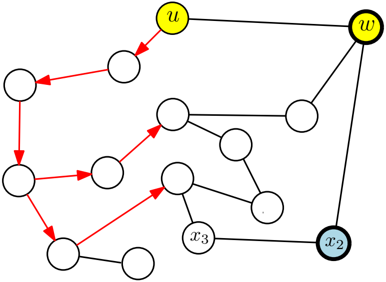
Some Intuition



$$G'$$

influence from the configuration at u

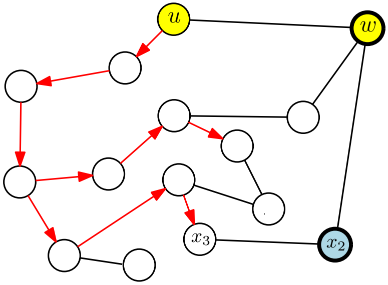
Some Intuition



G'

influence from the configuration at u

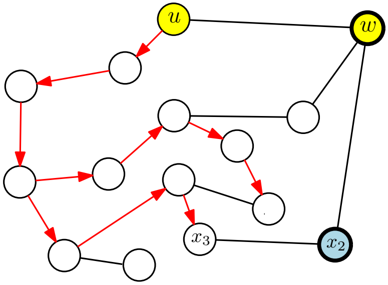
Some Intuition



G'

influence from the configuration at u

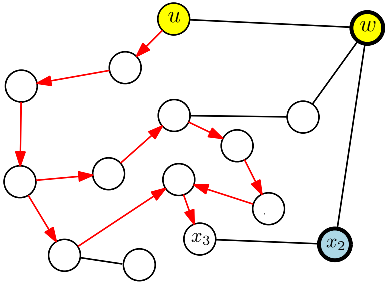
Some Intuition



G'

influence from the configuration at u

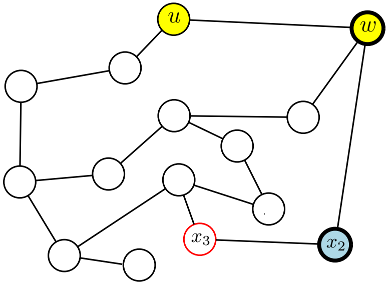
Some Intuition



G'

influence from the configuration at u

Some Intuition

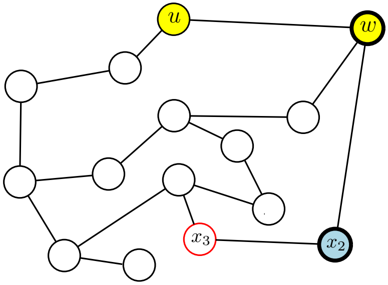


G'

However . . .

in most cases all but one vertex are far away (**girth**)

Some Intuition

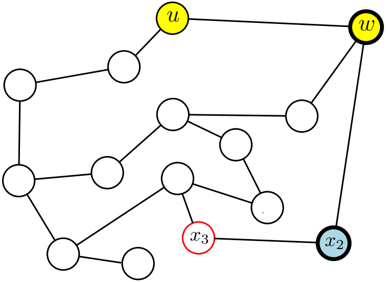


G'

Choosing the appropriate parameters . . .

the influences from distance are very weak & in most cases only one vertex influences the marginal

Some Intuition

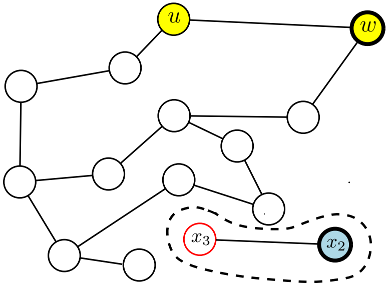


G'

Compute marginal but ...

ignore the influence on x_3 from u and w

Some Intuition

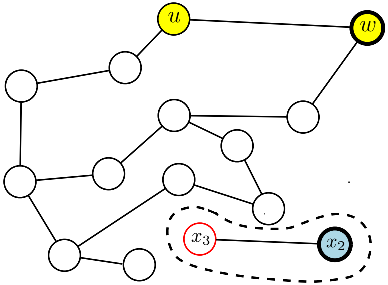


G'

Effectively

use the marginal of x_3 on the graph within the dashed curve

Some Intuition

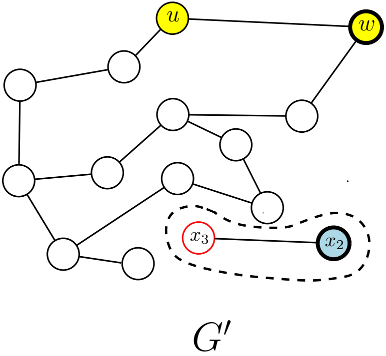


G'

Remarks

the "simplified" marginal on x_3 is trivial & is computed fast

Some Intuition

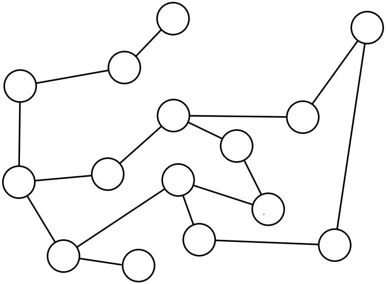


Remarks

our marginal is also called **broadcasting probability**

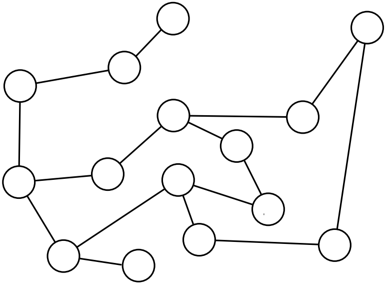
To sum up . . .

To sum up . . .

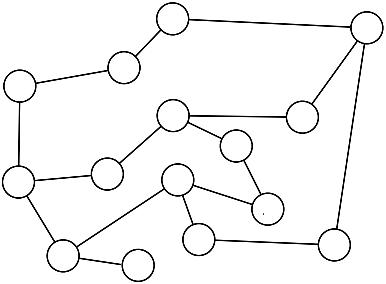


G

To sum up . . .

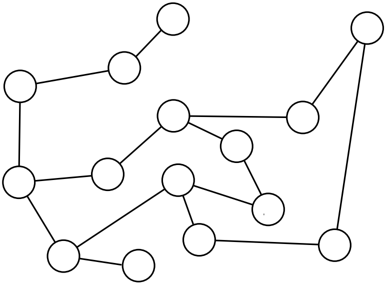


G

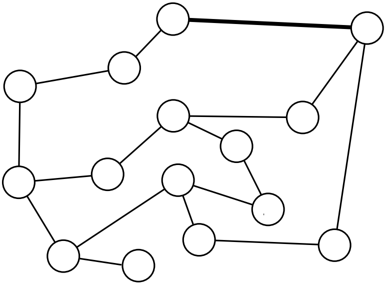


G'

To sum up . . .

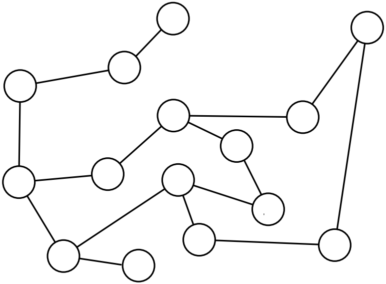


G

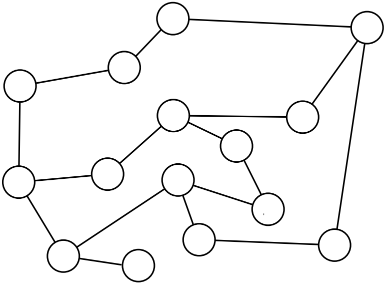


G'

To sum up . . .

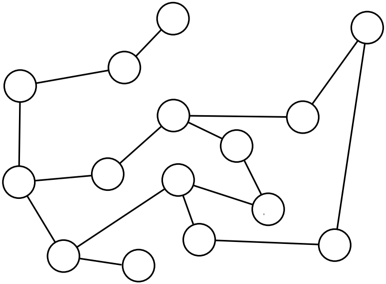


G

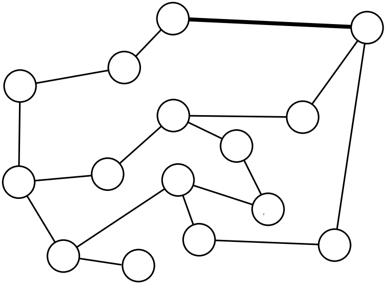


G'

To sum up . . .

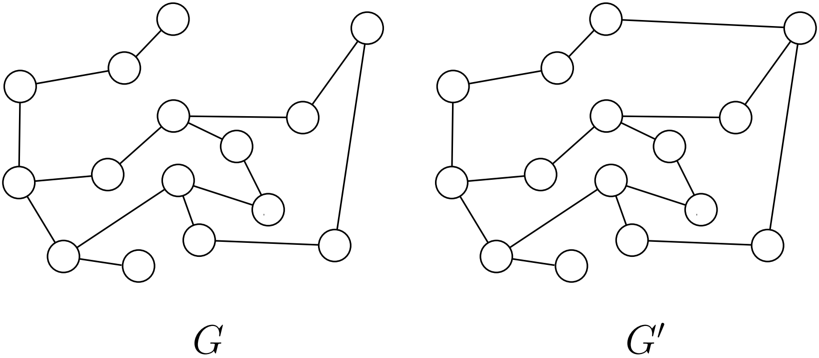


G

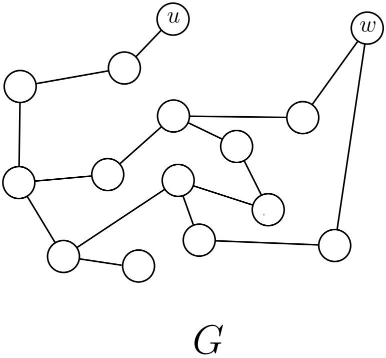


G'

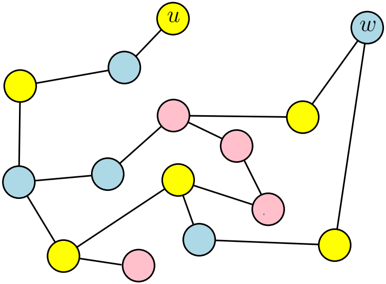
To sum up . . .



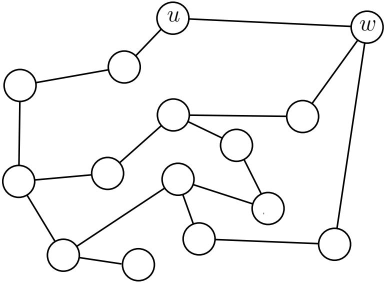
To sum up . . .



To sum up . . .

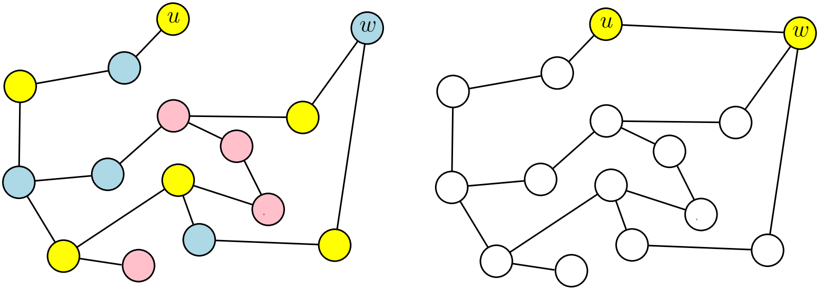


(G, σ)



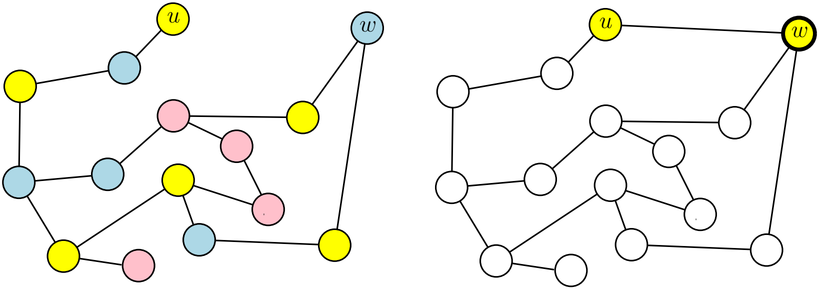
G'

To sum up . . .



(G, σ) G'

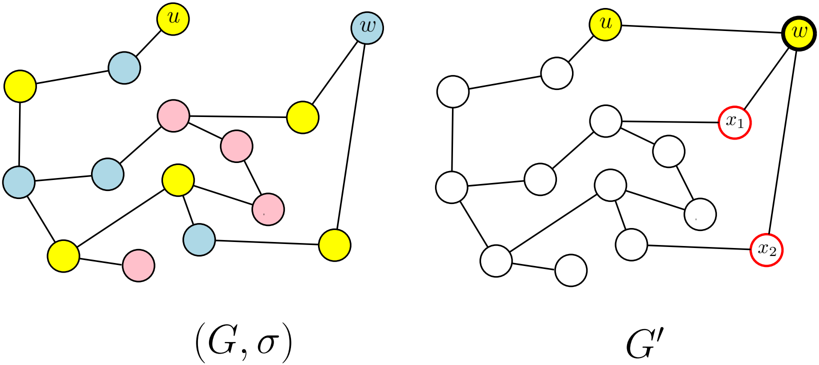
To sum up . . .



(G, σ) $\qquad$ G'

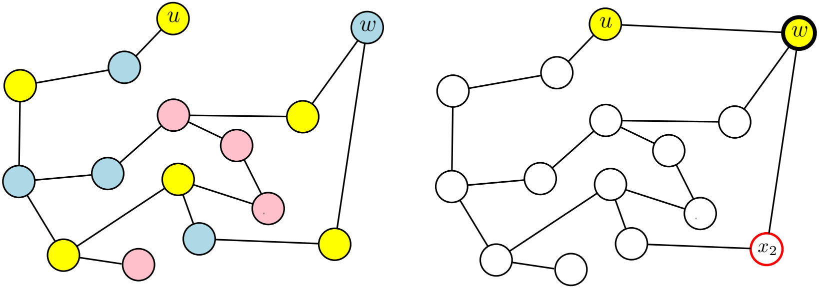
vertex w is a **disagreement** with spins $\mathcal{D} = \{\texttt{blue}, \texttt{yellow}\}$

To sum up . . .



look for z, neighbour of w with $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

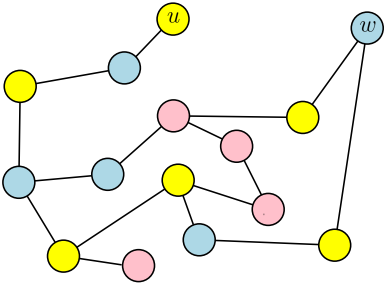
To sum up . . .



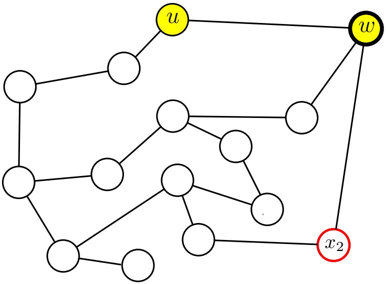
(G, σ) G'

pick x_2 and decide $\tau(x_2)$ such that $\tau(x_2) \in \{\texttt{blue}, \texttt{yellow}\}$

To sum up . . .



(G, σ)

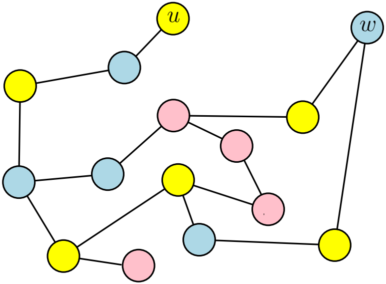


G'

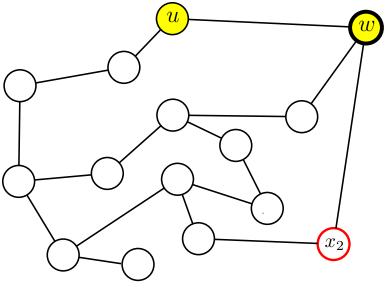
maximal coupling of broadcasting probabilities

$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \tau(w))}{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \sigma(w))}\right\}$$

To sum up . . .



(G, σ)

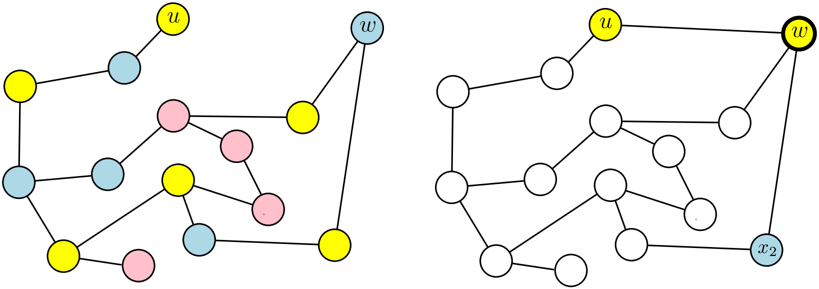


G'

maximal coupling of broadcasting probabilities

$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \tau(w))}{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \sigma(w))}\right\}$$

To sum up . . .

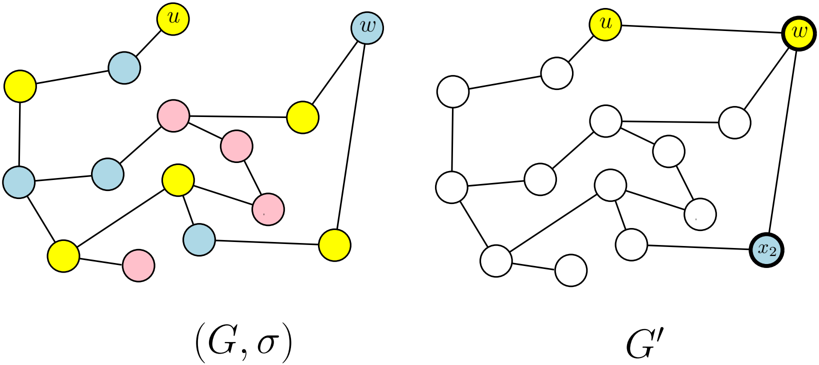


(G, σ) $\qquad$ G'

maximal coupling of broadcasting probabilities

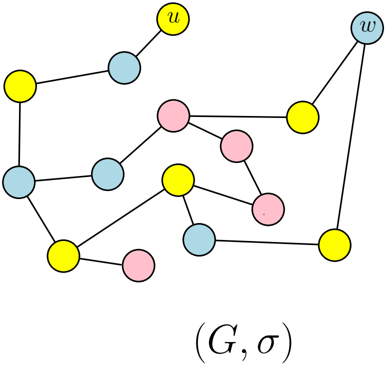
$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \tau(w))}{\mathfrak{m}_{x_2}(\sigma(x_2) \mid \sigma(w))}\right\}$$

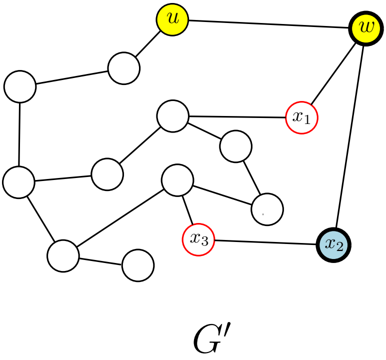
To sum up . . .



the disagreement set is $\{w, x_2\}$

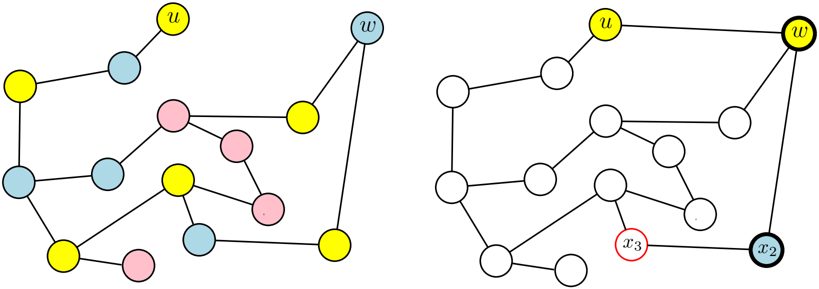
To sum up . . .





look for vertices z next to the disagreements such that $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

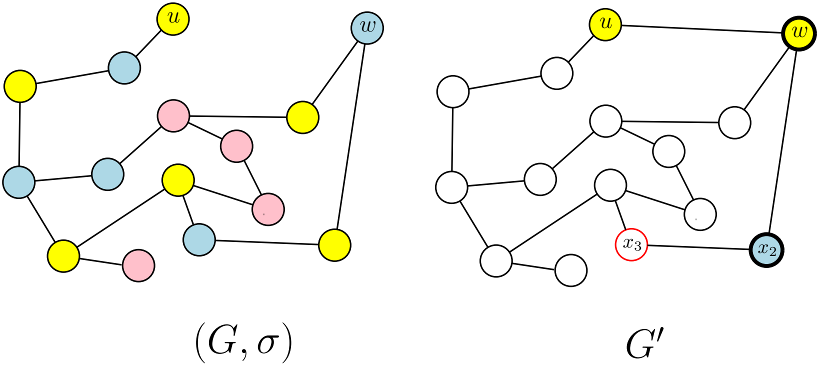
To sum up . . .



(G, σ) $\qquad$ G'

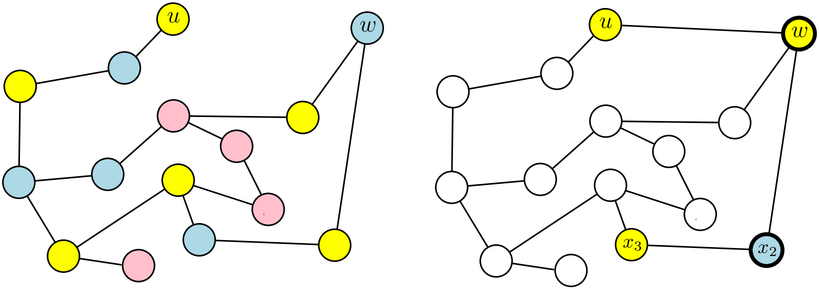
choose x_3 and repeat as before . . .

To sum up . . .



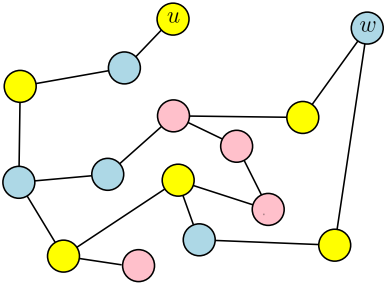
$$\Pr[\tau(x_3) = \texttt{yellow}] = \max\left\{0, 1 - \frac{\mathfrak{m}_{x_3}(\sigma(x_3) \mid \tau(x_2))}{\mathfrak{m}_{x_3}(\sigma(x_3) \mid \sigma(x_2))}\right\}.$$

To sum up . . .

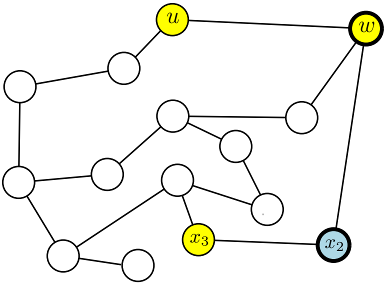


(G, σ) G'

To sum up . . .



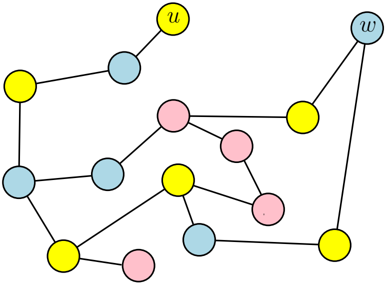
(G, σ)



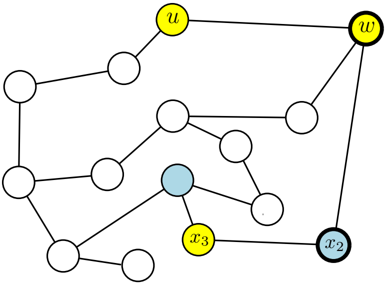
G'

repeat in the same way for the rest of the vertices

To sum up . . .



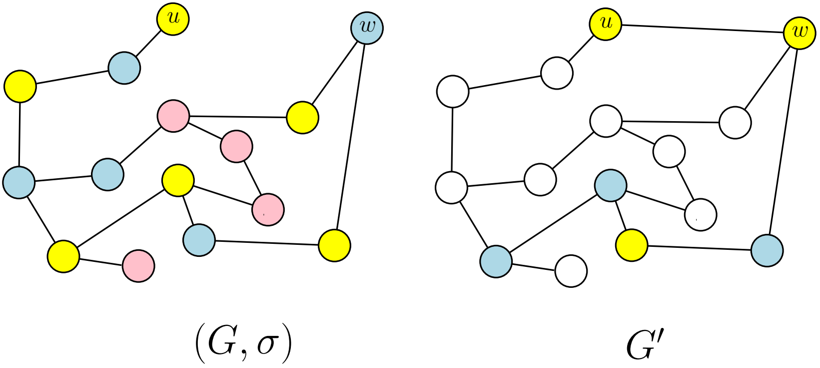
(G, σ)



G'

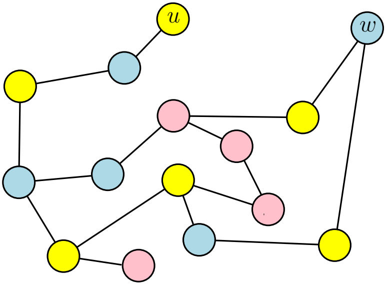
repeat in the same way for the rest of the vertices

To sum up . . .

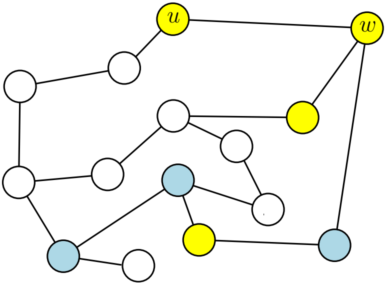


repeat in the same way for the rest of the vertices

To sum up . . .



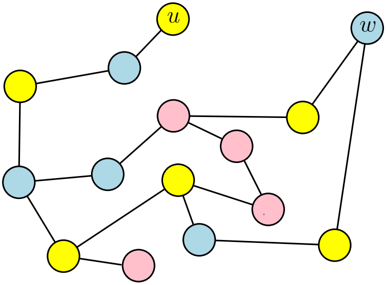
(G, σ)



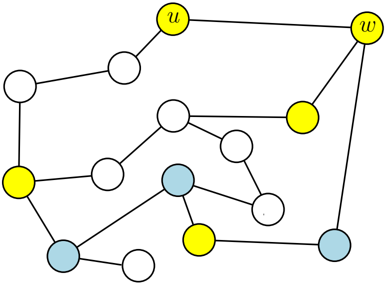
G'

repeat in the same way for the rest of the vertices

To sum up . . .



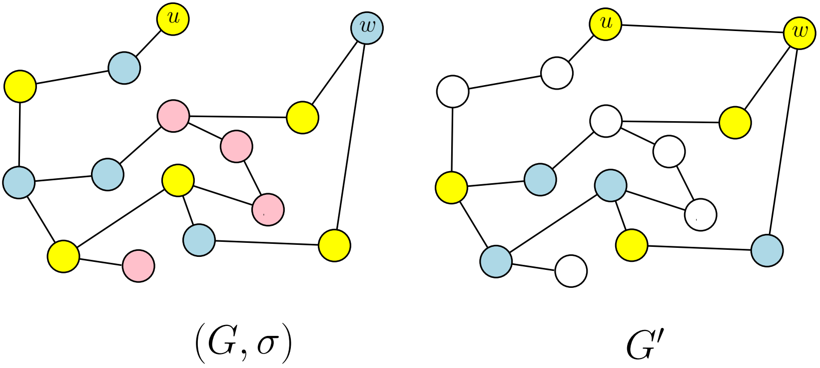
(G, σ)



G'

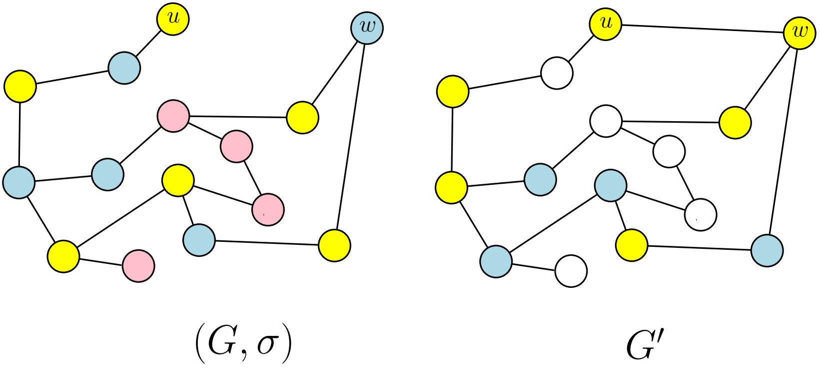
repeat in the same way for the rest of the vertices

To sum up ...



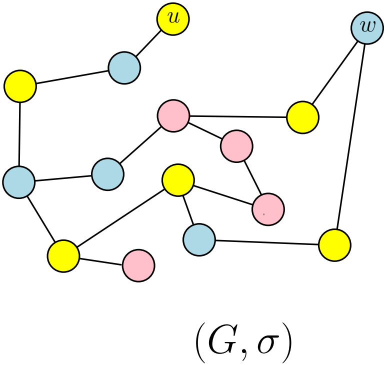
repeat in the same way for the rest of the vertices

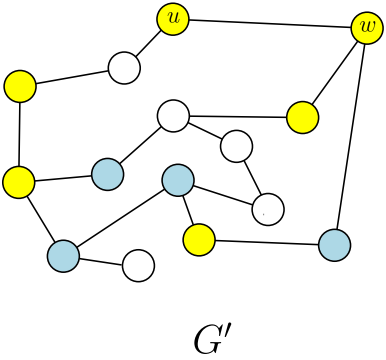
To sum up . . .



repeat in the same way for the rest of the vertices

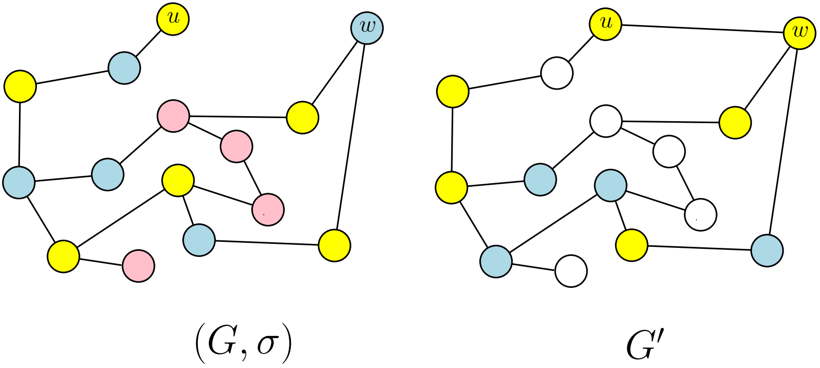
To sum up . . .





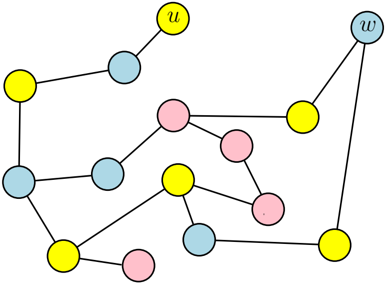
the disagreements cannot propagate any more

To sum up . . .

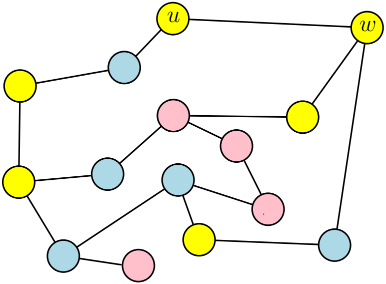


the remaining vertices keep the same assignment

To sum up . . .



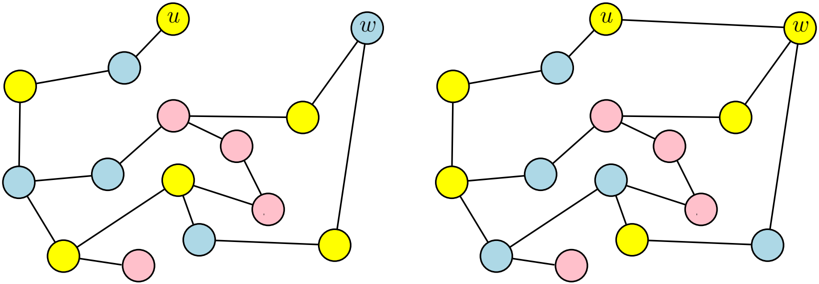
(G, σ)



(G', τ)

the remaining vertices keep their assignments.

To sum up . . .

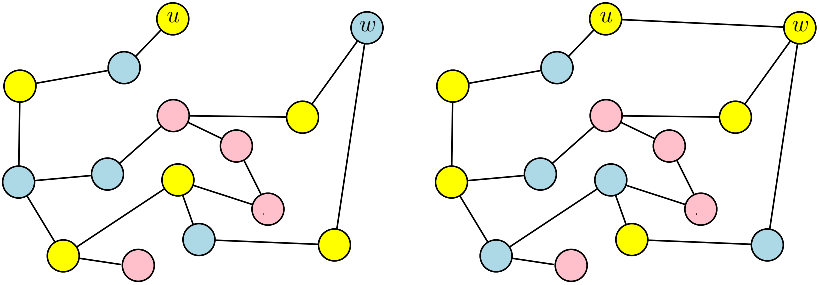


(G, σ) (G', τ)

The catch . . .

the process is not allowed to “self interact”

To sum up . . .

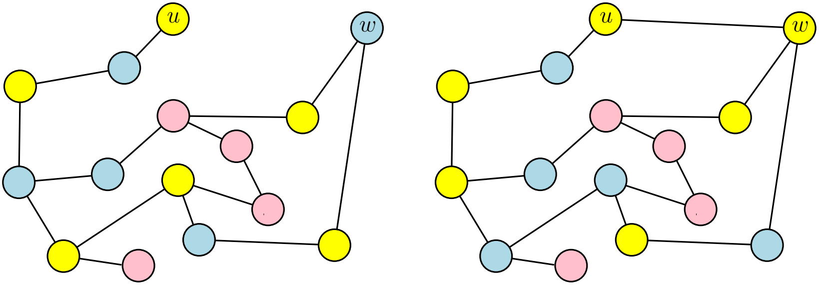


(G, σ) (G', τ)

Self interaction

- update neighbours of u
- the vertices whose assignment change induce a cycle

To sum up . . .



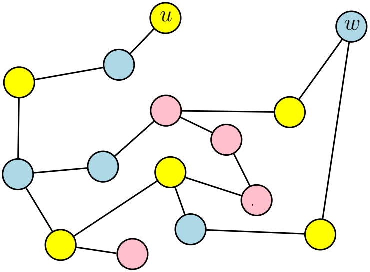
(G, σ) (G', τ)

Failure

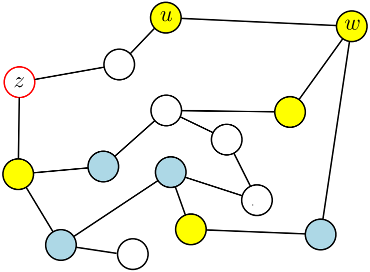
failure is when the process self-interacts.

Example of failure

Example of failure

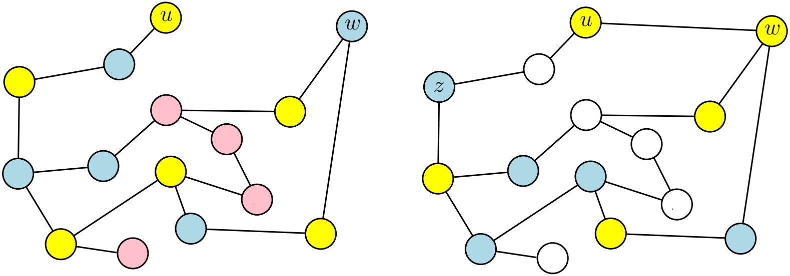


(G, σ)



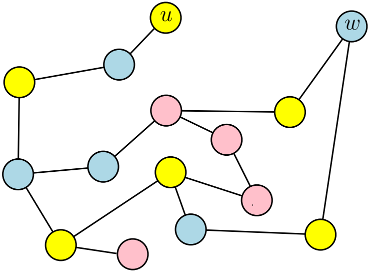
G'

Example of failure

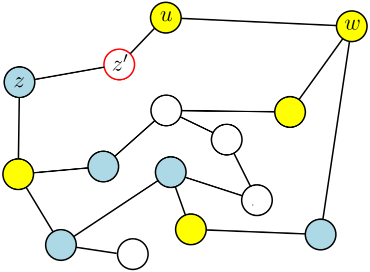


(G, σ) G'

Example of failure



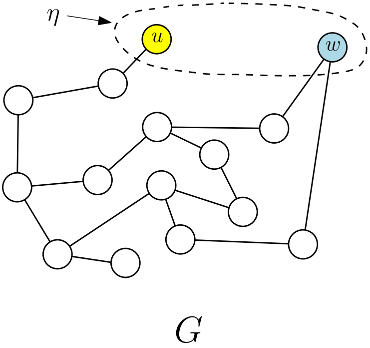
(G, σ)

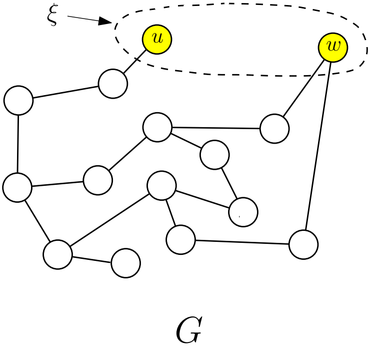


G'

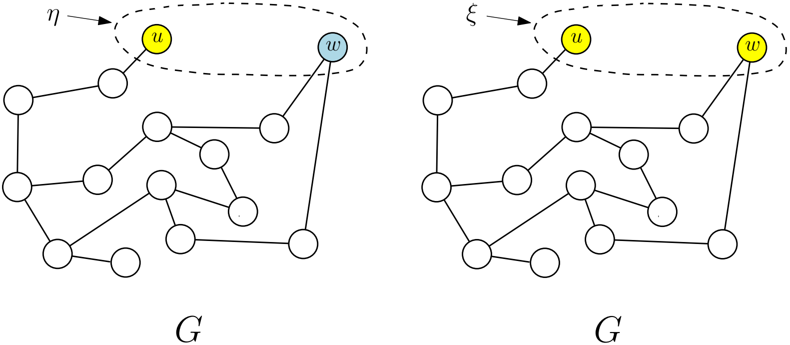
Failure Vs Approximation

Failure Vs Approximation





Failure Vs Approximation

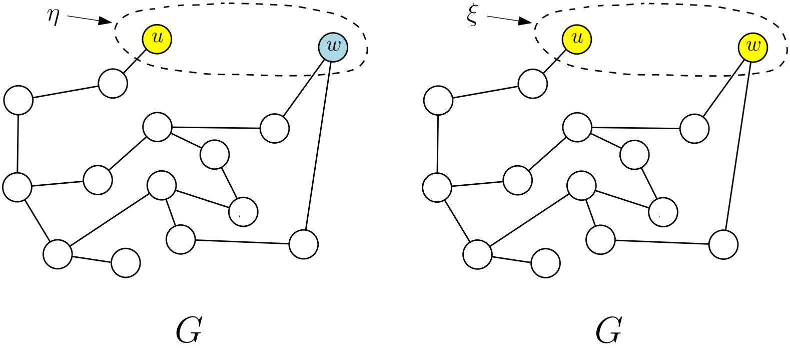


Accuracy

we have `Update` on input $\boldsymbol{\sigma}$ distributed as in $\mu(\cdot \mid \eta)$

- ν is the distribution of the output of `Update`
- compare ν with $\mu(\cdot \mid \xi)$

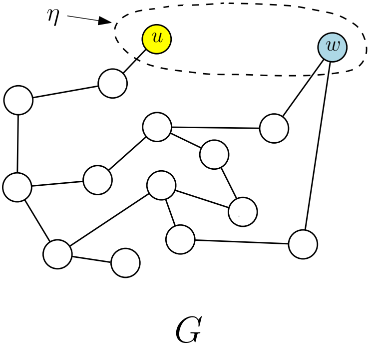
Failure Vs Approximation



Accuracy

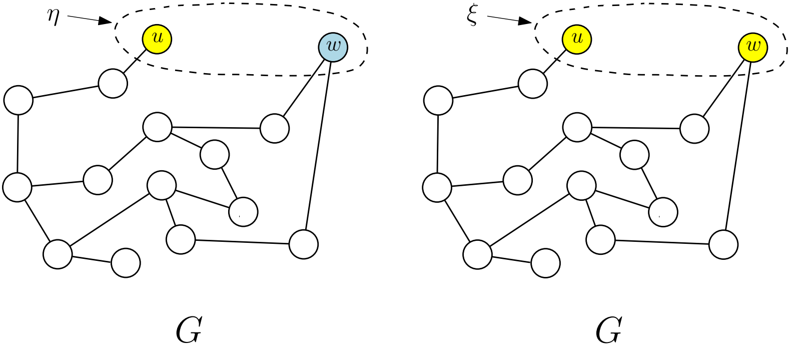
$$||\nu - \mu(\cdot \mid \xi)||_{tv} = ???$$

Failure Vs Approximation



we handle the update as a (probabilistic) map

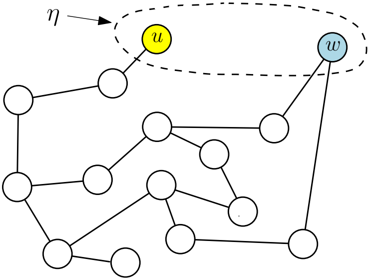
Failure Vs Approximation



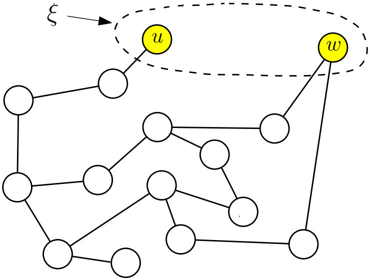
For configurations σ, τ

$\mathrm{P}_{\eta,\xi}(\sigma, \tau) :=$ Probability that update generates τ given input σ

Failure Vs Approximation



G

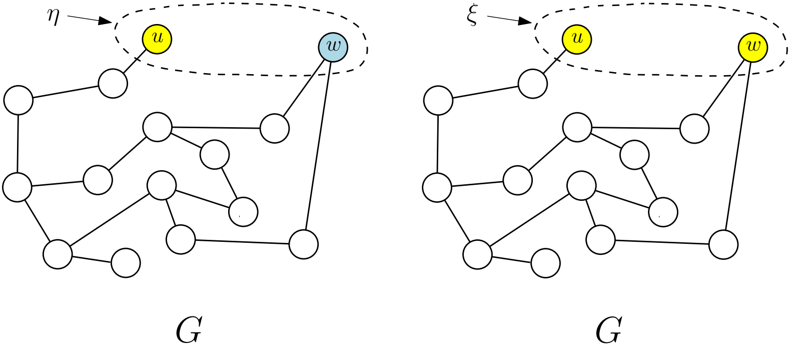


G

Reverse mapping

we can define the reverse mapping (right to left)

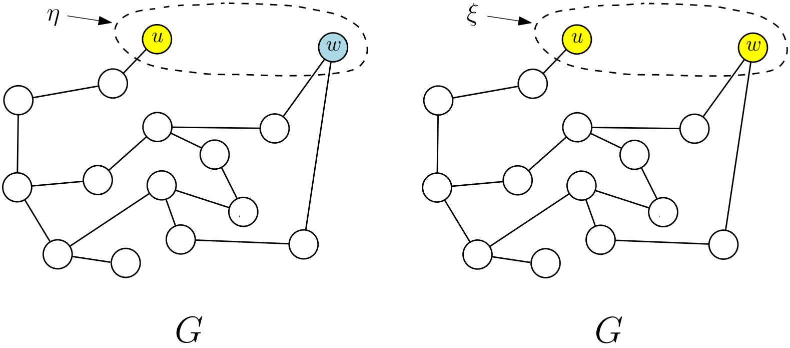
Failure Vs Approximation



For configurations σ, τ

$\mathrm{P}_{\xi,\eta}(\tau, \sigma) :=$ Probability that reverse update generates σ on input τ

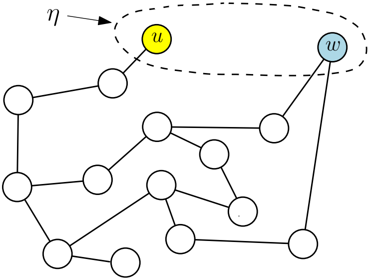
Failure Vs Approximation



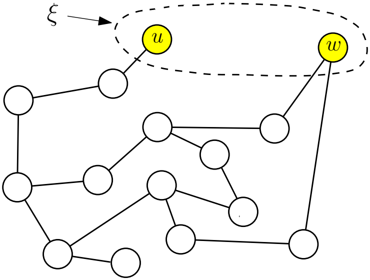
Transition probabilities

$P_{\eta,\xi}(\cdot,\cdot) \Rightarrow$ for the mapping from left to right
$P_{\xi,\eta}(\cdot,\cdot) \Rightarrow$ for the mapping from right to left

Failure Vs Approximation



G



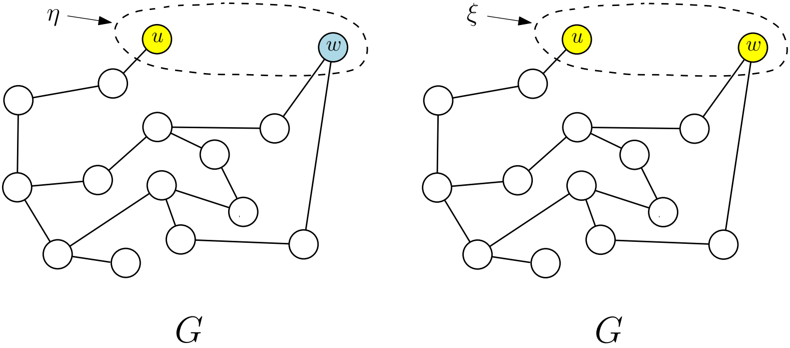
G

Detailed Balance Property

For any σ, τ we have

$$\mu(\sigma)\mathrm{P}_{\eta,\xi}(\sigma,\tau) = \mu(\tau)\mathrm{P}_{\xi,\eta}(\tau,\sigma)$$

Failure Vs Approximation

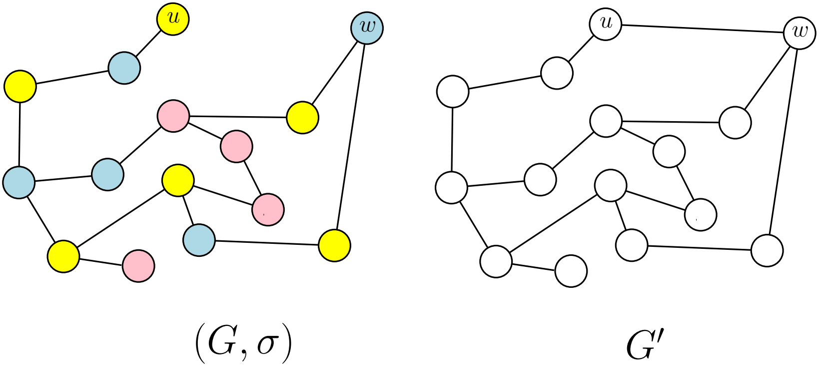


Using detailed balance we get ...

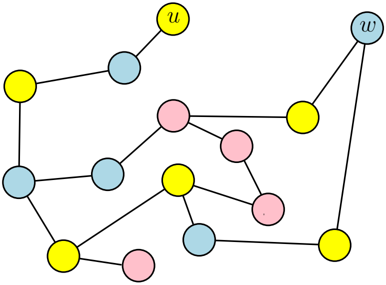
$$||\nu - \mu(\cdot \mid \xi)|| \approx \frac{1}{2}(\Pr[\texttt{Update Fails}] + \Pr[\texttt{Reverse Fails}])$$

Configuration for u and w

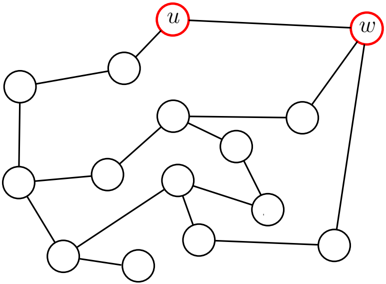
Configuration for u and w



Configuration for u and w

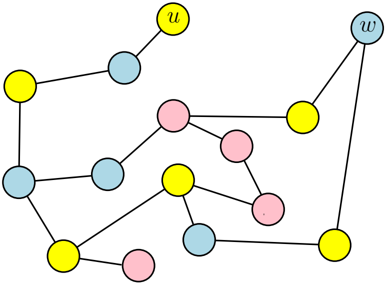


(G, σ)

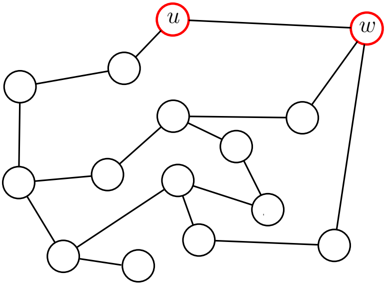


G'

Configuration for u and w



(G, σ)

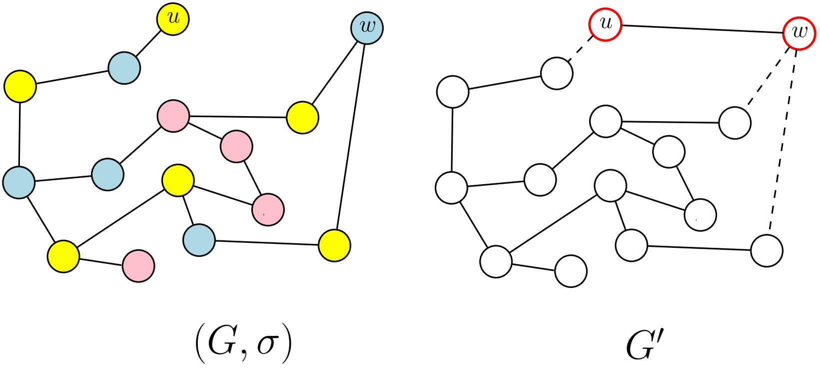


G'

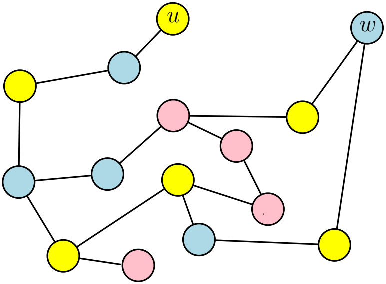
Remark

The choice of $\tau(u)$ and $\tau(w)$ in G' is oblivious to σ

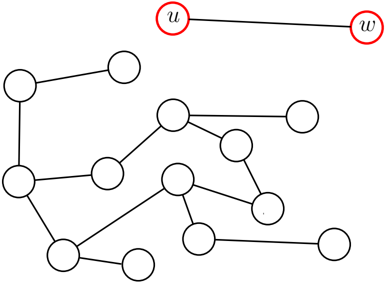
Configuration for u and w



Configuration for u and w

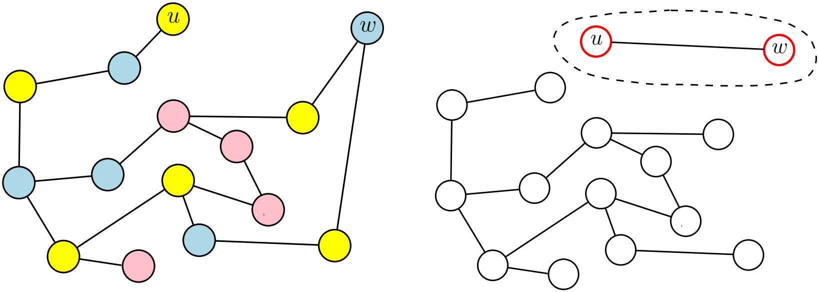


(G, σ)



G'

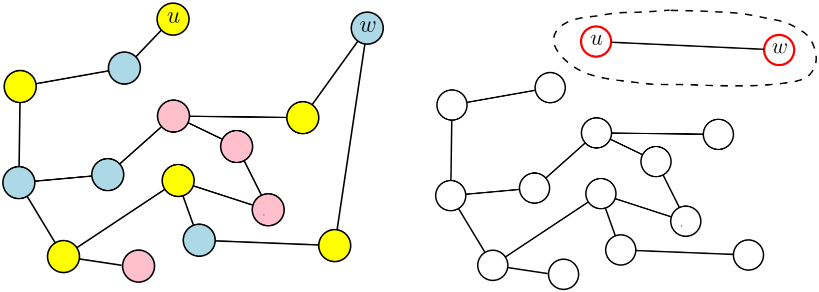
Configuration for u and w



(G, σ) G'

Sample from the distribution on the graph within the dashed lines

Configuration for u and w



(G, σ) $\qquad$ G'

Remarks

- introduces an extra error
- initial disagreement maybe ≥ 1

The algorithm

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$, the colouring of G_r

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$, the colouring of G_r

The error for the algorithm

$\approx$ probability of failure at some iteration

The iterative algorithm

The algorithm

Input: $G = (V, E)$ $k > 0$

$G_0, G_1, \ldots, G_r = G$

$-$get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ a random coloring of G_0

Iteratively: Use $\boldsymbol{\sigma}_i$ the colouring of G_i and `Update` to get $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$, the colouring of G_r

The time complexity

the time complexity is $O(|E|^2)$

- for each iteration we compute $O(|E|)$ broadcasting marginals
- we have $|E|$ iterations

From high girth to $\boldsymbol{G}(n, m)$

From high girth to $G(n, m)$

- we considered high girth graphs

From high girth to $G(n, m)$

- we considered high girth graphs
- typical instances of $G(n, m)$ are a bit different

From high girth to $G(n, m)$

- we considered high girth graphs
- typical instances of $G(n, m)$ are a bit different
  - there are short cycles far apart from each other

From high girth to $G(n, m)$

- we considered high girth graphs
- typical instances of $G(n, m)$ are a bit different
  - there are short cycles far apart from each other
- we won't discuss the challenges from the short cycles here ...

The parameters

The parameters

For which parameters of the Gibbs distribution on $G(n, m)$ do we get good approximations?

The parameters

For which parameters of the Gibbs distribution on $G(n, m)$ do we get good approximations?

- good approximation $\Rightarrow$ error $n^{-\Omega(1)}$

The parameters

For which parameters of the Gibbs distribution on $G(n, m)$ do we get good approximations?

- good approximation $\Rightarrow$ error $n^{-\Omega(1)}$
- Gibbs uniqueness condition

The parameters

For which parameters of the Gibbs distribution on $G(n, m)$ do we get good approximations?

- good approximation $\Rightarrow$ error $n^{-\Omega(1)}$
- Gibbs uniqueness condition
- tools based on Contiguity

Analysis

Analysis

Setting

Consider $\boldsymbol{G}(n, m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Analysis

Setting

Consider $G(n, m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Objective

The disagreements grow **subcritically**

Analysis

Setting

Consider $G(n, m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Objective

The disagreements grow **subcritically**

The randomness . . .

Setting

Consider $G(n, m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Objective

The disagreements grow **subcritically**

The randomness . . .

- the random graph $G(n, m)$

Analysis

Setting

Consider $G(n,m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Objective

The disagreements grow **subcritically**

The randomness ...

- the random graph $G(n,m)$
- confinguration $\boldsymbol{\sigma}$

Analysis

Setting

Consider $G(n, m)$ and $\boldsymbol{\sigma}$ distributed as in μ. Consider `Update` that starts from vertex v with initial assignment $\tau(v) \neq \boldsymbol{\sigma}(v)$

Objective

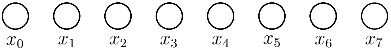
The disagreements grow **subcritically**

The randomness ...

- the random graph $G(n, m)$
- confinguration $\boldsymbol{\sigma}$
- random choices of `Update`

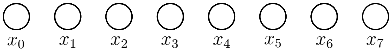
Analysis Sketch

Analysis Sketch



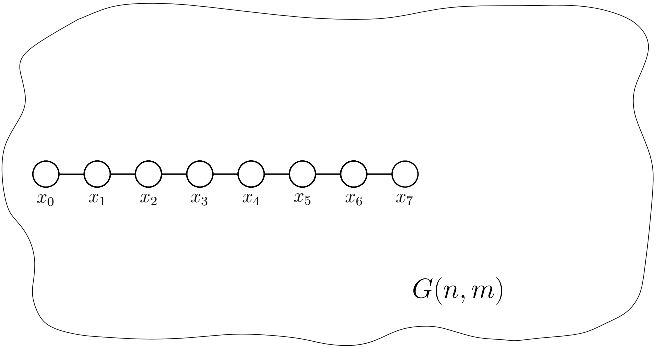
consider a permutation of vertices s.t. $x_0 = v$

Analysis Sketch



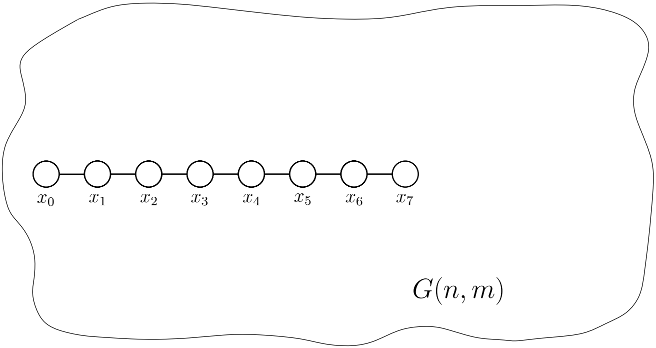
reveal the graph structure $G(n, m)$

Analysis Sketch



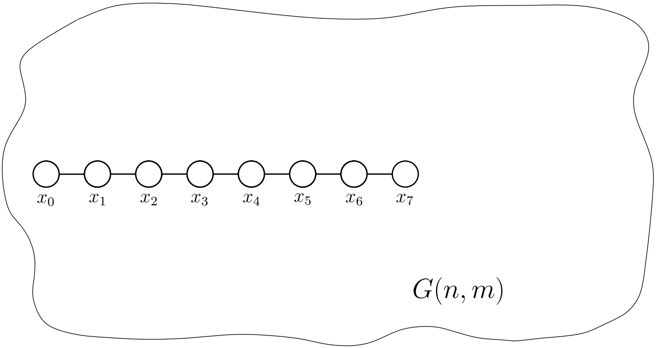
reveal the graph structure $\boldsymbol{G}(n, m)$

Analysis Sketch



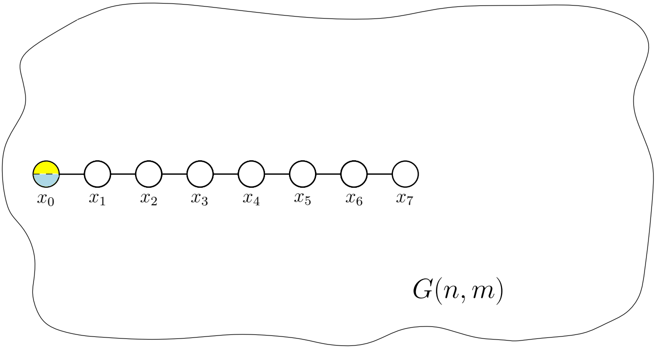
we care whether $x_0, \ldots, x_7$ forms a path in $\boldsymbol{G}(n, m)$

Analysis Sketch



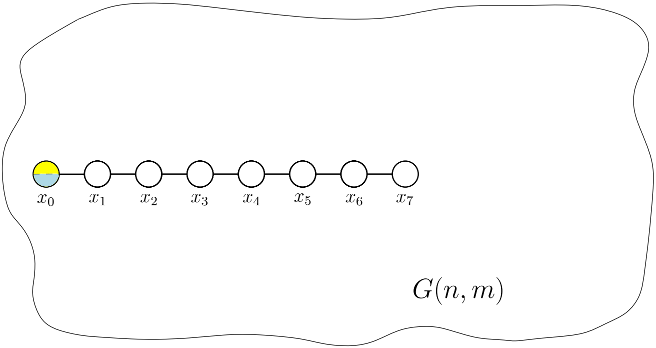
we set initial disagreement at x_0

Analysis Sketch



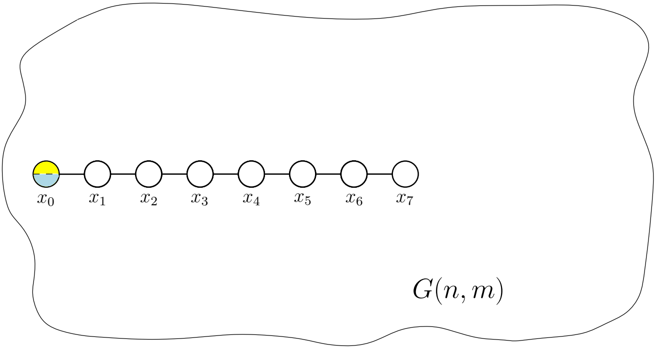
we set initial disagreement at x_0

Analysis Sketch



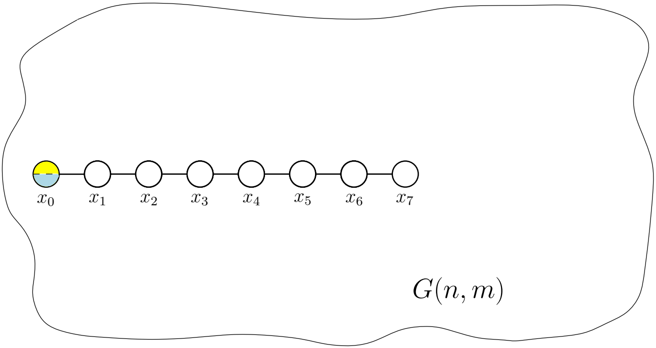
the two configurations at x_0 are from the input $\boldsymbol{\sigma}$ and he output $\boldsymbol{\tau}$ of `Update`

Analysis Sketch



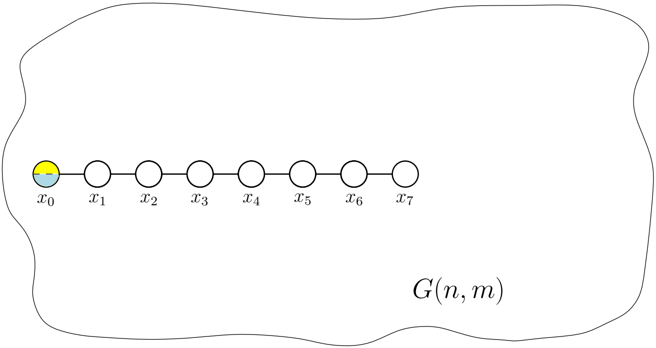
focus is on the **probability** that the disagreement propagates over the path $x_0, \ldots, x_7$

Analysis Sketch



in steps, for $x_1, x_2, \ldots$, we reveal the configurations on the vertices in the path

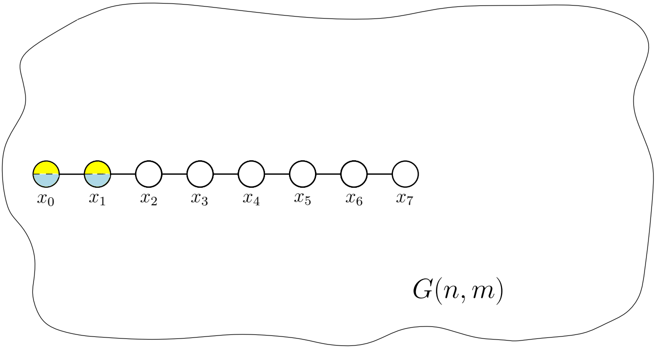
Analysis Sketch



Disagreement probability

the probability that the disagreement propagates one step further

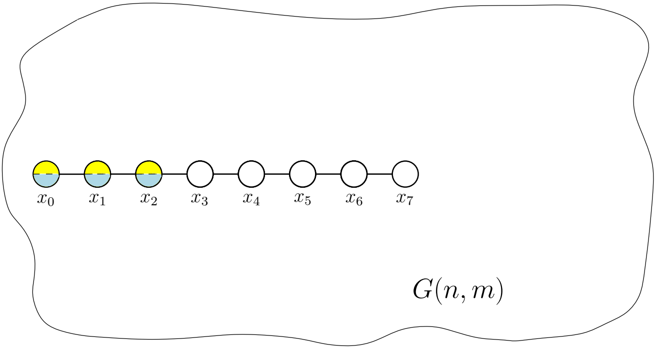
Analysis Sketch



Disagreement probability

the probability that the disagreement propagates one step further

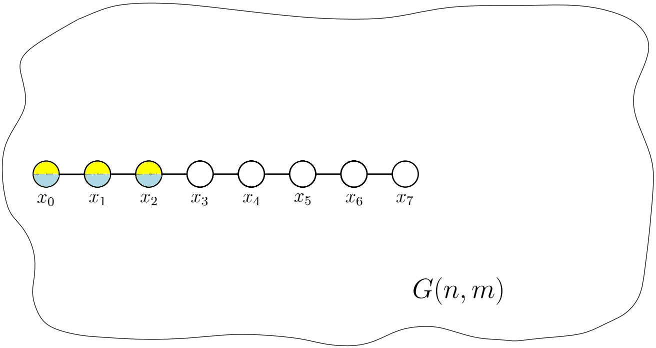
Analysis Sketch



Disagreement probability

the probability that the disagreement propagates one step further

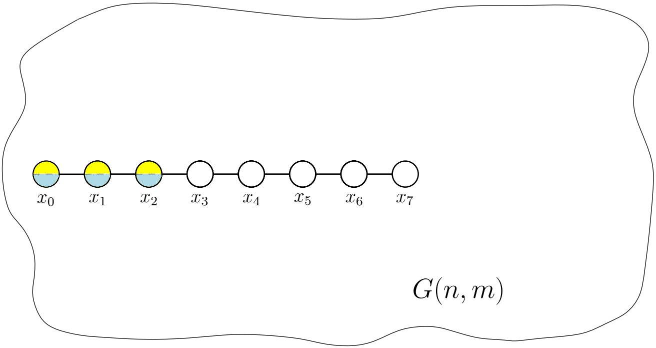
Analysis Sketch



Desideratum

at each step the disagreement probability $< 1/d$

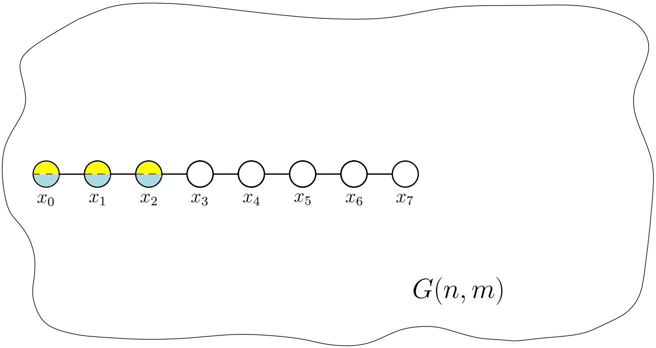
Analysis Sketch



Remark I

this probability depends on μ and the random choice of `Update`

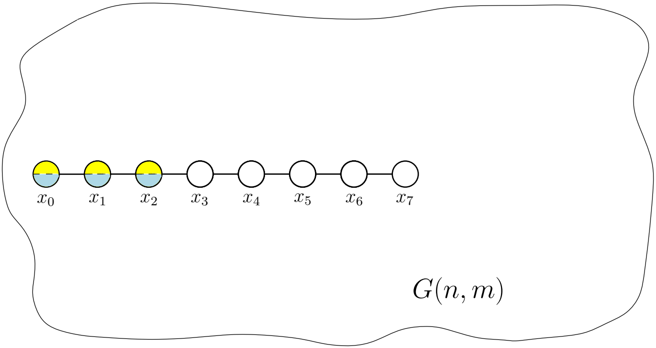
Analysis Sketch



Some magic

if Gibbs marginal at x_3 was close to the broadcasting probability

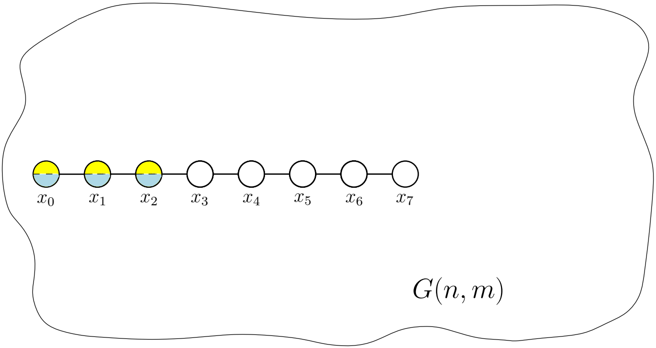
Analysis Sketch



Some magic

if Gibbs marginal at x_3 was close to the broadcasting probability
$\Rightarrow$ in the (conjectured) uniqueness region we have the desideratum

Analysis Sketch

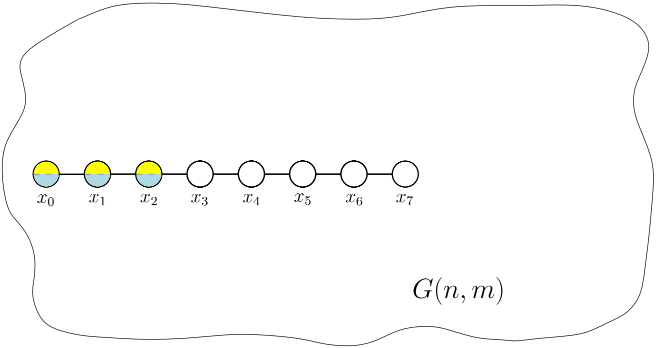


Some problems

if Gibbs marginal at x_3 was close to the broadcasting probability

$\Rightarrow$ in the (conjectured) uniqueness region we have the desideratum

Analysis Sketch



Contiguity to the rescue ...

The planted model

The planted model

Idea . . .

Reconsider the order of randomness

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of σ
3. choices of `Update`

Teacher-Student Model

1. generate σ^*
2. graph G^*
3. choices of `Update`

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

$\boldsymbol{\sigma}^*$ is a random q-partition of the vertex set, $\ldots q = |\mathcal{S}|$

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

$\boldsymbol{\sigma}^*$ is a random q-partition of the vertex set, ... $q = |\mathcal{S}|$

- the distribution of $\boldsymbol{\sigma}^*$ is **very simple**

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

G^* is a **weighted** random graph on n vertices, m edges

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

G^* is a **weighted** random graph on n vertices, m edges

- the weights depends on Gibbs distribution
- G^* depends on $\boldsymbol{\sigma}^*$

The planted model

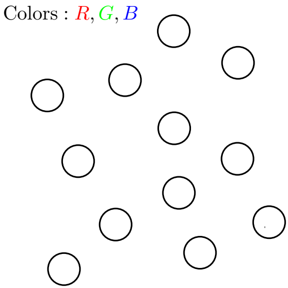
Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

Planting Colourings



The planted model

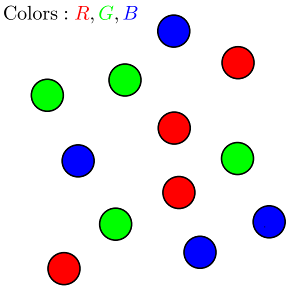
Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

Planting Colourings



The planted model

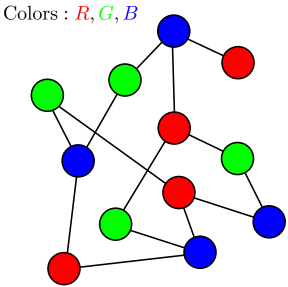
Uniform Model

1. random graph $\boldsymbol{G}(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph $\boldsymbol{G}^*$
3. choices of `Update`

Planting Colourings



The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

Also known ...

in network inference they call it **Stochastic Block Model**

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

`Update` for Teacher-Student

the input of the process is the pair $(G^*, \boldsymbol{\sigma}^*)$

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

`Update` for Teacher-Student

the input of the process is the pair $(G^*, \boldsymbol{\sigma}^*)$

- this process is **simpler** to analyse

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

`Update` for Teacher-Student

the input of the process is the pair $(G^*, \boldsymbol{\sigma}^*)$

- this process is **simpler** to analyse
- in uniqueness the disagreements grow subcritically (proof)

The planted model

Uniform Model

1. random graph $G(n, m)$
2. randomness of $\boldsymbol{\sigma}$
3. choices of `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. choices of `Update`

`Update` for Teacher-Student

the input of the process is the pair $(G^*, \boldsymbol{\sigma}^*)$

- this process is **simpler** to analyse
- in uniqueness the disagreements grow subcritically (proof)
- ... argue that this implies the same for the "real process"

Contiguity

Contiguity

- consider a Gibbs distribution
  - e.g., for Potts we need q and β

Contiguity

- consider a Gibbs distribution
  - e.g., for Potts we need q and β

Uniform pair $(\boldsymbol{G}, \boldsymbol{\sigma})$

- $\boldsymbol{G} = \boldsymbol{G}(n, m)$
- $\boldsymbol{\sigma} \sim \mu_{\boldsymbol{G}}$

Contiguity

- consider a Gibbs distribution
  - e.g., for Potts we need q and β

Uniform pair $(\boldsymbol{G}, \boldsymbol{\sigma})$

- $\boldsymbol{G} = \boldsymbol{G}(n, m)$
- $\boldsymbol{\sigma} \sim \mu_{\boldsymbol{G}}$

Planted pair $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$

- $\boldsymbol{\sigma}^*$ a $|\mathcal{S}|$-partion of vertices
- generate $\boldsymbol{G}^* = \boldsymbol{G}^*(\boldsymbol{\sigma}^*)$

Contiguity

Contiguity

Definition

We say that $(\boldsymbol{G}, \boldsymbol{\sigma})$ and $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$ are **mutual contiguous** when for any property $\mathcal{A}_n$ we have that

$$\lim_{n\to\infty} \Pr[(\boldsymbol{G}^*, \boldsymbol{\sigma}^*) \in \mathcal{A}_n] = 0 \quad \text{iff} \quad \lim_{n\to\infty} \Pr[(\boldsymbol{G}, \boldsymbol{\sigma}) \in \mathcal{A}_n] = 0.$$

Contiguity

Definition

We say that $(\boldsymbol{G}, \boldsymbol{\sigma})$ and $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$ are **mutual contiguous** when for any property $\mathcal{A}_n$ we have that

$$\lim_{n\to\infty} \Pr[(\boldsymbol{G}^*, \boldsymbol{\sigma}^*) \in \mathcal{A}_n] = 0 \quad \text{iff} \quad \lim_{n\to\infty} \Pr[(\boldsymbol{G}, \boldsymbol{\sigma}) \in \mathcal{A}_n] = 0.$$

Contiguity implies ...

the two distributions have the same typical properties

Back to sampling

Back to sampling

Update for Teacher-Student

the input of the process is the pair $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$

- this process is **simpler** to analyse
- in uniqueness the disagreements grow subcritically (proof)
- . . . the same holds for the "real process"

Back to sampling

`Update` for Teacher-Student

the input of the process is the pair $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$

- this process is **simpler** to analyse
- in uniqueness the disagreements grow subcritically (proof)
- ... the same holds for the "real process"
- the above is **true due to contiguity**

Overview of the result

High level

We proved that

For **symmetric** Gibbs distributions on $G(n, m)$ that

1. parameters in the (conjectured) Gibbs uniqueness
   - parametrised w.r.t. the expected degree $d > 0$
2. exhibit contiguity with the corresponding teacher-student model

there is an $O(n^2 \log n)$ time sampler such that the following holds: with probability $1 - o(1)$ over $G(n, m)$ the output error is $n^{-\Omega(1)}$.

Overview of the result

High level

We proved that

For **symmetric** Gibbs distributions on $G(n, m)$ that

1. parameters in the (conjectured) Gibbs uniqueness
   - parametrised w.r.t. the expected degree $d > 0$
2. exhibit contiguity with the corresponding teacher-student model

there is an $O(n^2 \log n)$ time sampler such that the following holds: with probability $1 - o(1)$ over $G(n, m)$ the output error is $n^{-\Omega(1)}$.

Uniqueness Vs Contiguity

contiguity is much weaker a notion than uniqueness

Results

Results

- antiferromagnetic Ising model

Results

- antiferromagnetic Ising model
- antiferromagnetic Potts model

Results

- antiferromagnetic Ising model
- antiferromagnetic Potts model
- random k-NAE-SAT solutions

Results

- antiferromagnetic Ising model
- antiferromagnetic Potts model
- random k-NAE-SAT solutions
- k-spin model for $k \geq 2$ even integer

Results

- antiferromagnetic Ising model
- antiferromagnetic Potts model
- random k-NAE-SAT solutions
- k-spin model for $k \geq 2$ even integer

Which parameters?

(conjectured) tree uniqueness parametrised w.r.t. the expected degree d

Results

- antiferromagnetic Ising model
- antiferromagnetic Potts model
- random k-NAE-SAT solutions
- k-spin model for $k \geq 2$ even integer

Contiguity

- Coja-Oghlan, Krzakala, Perkins, Zdeborova 2017
- Coja-Oghlan, Efthymiou, Jaafari, Kang, Kapetanopoulos 2017
- Coja-Oghlan, Kapetanopoulos, Muller, 2018

For exact statement of results . . .

On sampling symmetric Gibbs distributions on sparse random graphs and hypergraphs
https://arxiv.org/abs/2007.07145

Concluding Remarks

Concluding Remarks

- Presented a novel approximate sampling algorithm

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $\boldsymbol{G}(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $\boldsymbol{G}(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$
- performance in terms of the range of the parameters
  - for the anti-ferromagnetic distributions it outperform any other sampler for $G(n, m)$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$
- performance in terms of the range of the parameters
  - for the anti-ferromagnetic distributions it outperform any other sampler for $G(n, m)$
- applies to spin-glasses

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$
- performance in terms of the range of the parameters
  - for the anti-ferromagnetic distributions it outperform any other sampler for $G(n, m)$
- applies to spin-glasses
- uses concepts from other research areas

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$
- performance in terms of the range of the parameters
  - for the anti-ferromagnetic distributions it outperform any other sampler for $G(n, m)$
- applies to spin-glasses
- uses concepts from other research areas
  - contiguity, from network inference algorithm

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $G(n, m)$
  - **any** fixed expected degree $d > 0$
  - works for **any** symmetric Gibbs distribution
  - running time $O(n^2 \log n)$
  - accuracy $n^{-\Omega(1)}$
- performance in terms of the range of the parameters
  - for the anti-ferromagnetic distributions it outperform any other sampler for $G(n, m)$
- applies to spin-glasses
- uses concepts from other research areas
  - contiguity, from network inference algorithm
  - broadcasting models and probabilities

The end

Thank you!