Charis Efthymiou

University of Warwick, UK.

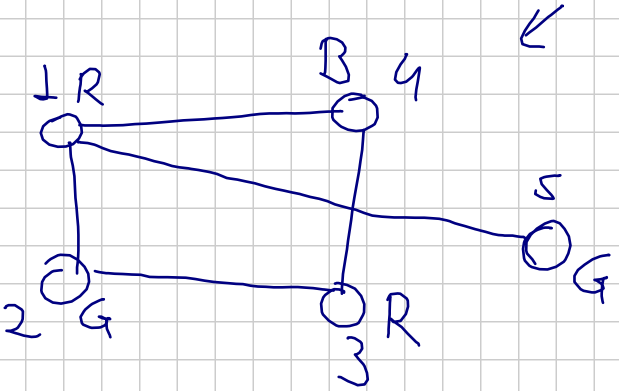
Algorithms: <u>Efficient Algorithms</u>

Input: $\boxed{G=(V,E)}$, int k, $b \in \mathbb{R}$

Running time $= O(n^c)$ c: small number ($c = 2, 3$)

Colouring Problem

$G=(V,E)$



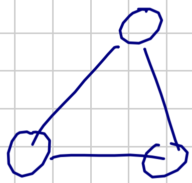
set of colours $C = [R, G, B]$

σ is proper 3-colouring of G iff

for $\{u,w\} \in E$ $\sigma(u) \neq \sigma(w)$

* $G = (V, E)$, integer $k > 0$

- G might not have a proper k-colouring
  "G is not k-colourable"

Example  $k = 2$

- Even if G is k-colourable, it might
  "computationally hard" to find one.
  $\Downarrow$
  "we believe that there no efficient alg."

- If k: (# colours) is "large",
  then G is both k-colourable
  "easy" to find a k-colouring

E.g. $k \geq \Delta + 1$
$\uparrow$ maximum degree

Can we count efficiently the # proper k-colouring of G?

Equivalent Question:

Can we generate efficiently <u>Uniformly at Random</u> a k-colouring of G?

$\boxed{\text{Focus}}$

<u>SAMPLING</u> Problem

Solve the problem efficiently.

"k- colouring Model"

$\sigma \in V^{E}$ ← set of colours

$$\mu_G(\sigma) \propto \prod_{\{u,w\}\in E} \psi(\sigma_u, \sigma_w)$$

$$\psi = \begin{cases} 0 & \text{if } \sigma_u = \sigma_w \\ 1 & \text{otherwise} \end{cases}$$

"Potts model with inverse temperature $\beta \in \mathbb{R} \cup \{-\infty\}$"

$$\mu_G(\sigma) \propto \prod_{\{u,w\}\in E} \psi(\sigma_u, \sigma_w)$$

$$\psi = \begin{cases} e^{\beta} & \text{if } \sigma_u = \sigma_w \\ 1 & \text{otherwise} \end{cases}$$

* $\beta < 0$: Antiferromagnetic
* $\beta > 0$ ferromagnetic case

$k = 2 \implies$ Ising Model

Gibbs distribution

→ Generating samples from μ. computationally hard.

→ Generate <u>approximate sample</u>

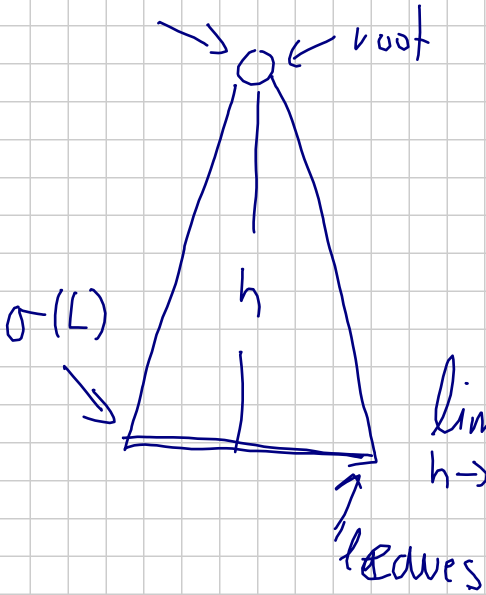
$\Rightarrow$

$$\|\mu_{alg} - \mu_G\| = \text{"small"}$$

ℓ_1-distance $\qquad\qquad$ $O(n^{-c})$

→ Find regions of the parameters of μ_G where we can have efficient approximate sampling.

When μ_G Exhibits Spatial Mixing



Δ-ary tree.

$k \geq \Delta + 1$

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

Sampling symmetric Gibbs distributions on the sparse random graph and hypergraph

Charis Efthymiou
University of Warwick

Seminar on Geometry, Probability, and Computing.

GeomProbComp April, 2022

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 $\mathcal{S}$
  - 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 $\sigma \in \mathcal{S}^V$ 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)

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

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)

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

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
- the **range of parameters** of μ in which we can get "**good**" approximations of μ

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 $d > 1$

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 $d > 1$
  - ... 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 $d > 1$
  - … 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 $d > 1$
  - … 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 $d > 1$
  - ... that is $m = \Theta(n)$

Sampling Problem on $G(n, m)$

- focus on approximate sampling
- use concepts from **physics** for better algorithms

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 $d > 1$
  - ... that is $m = \Theta(n)$

Sampling Problem on $G(n, m)$

- focus on approximate sampling
- use concepts from **physics** for better algorithms
- **Cavity Method**

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
- Lovasz Local Lemma

Popular approaches to sampling problem

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

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

The idea

The idea

The sampling algorithm

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

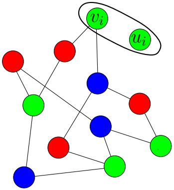
$G_0, G_1, \ldots, G_r = G$

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$



The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

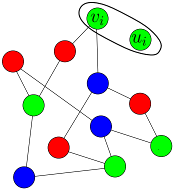
$G_0, G_1, \ldots, G_r = G$

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$



`Update`

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

The idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

Output: $\boldsymbol{\sigma}_r$

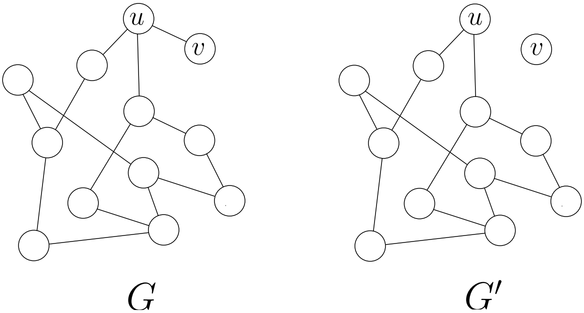
Example from the past

Example from the past

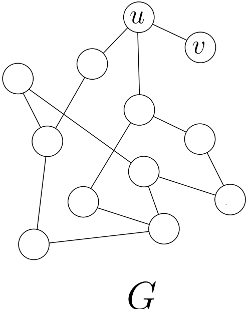
Example with the Colouring Model

Observation

Observation

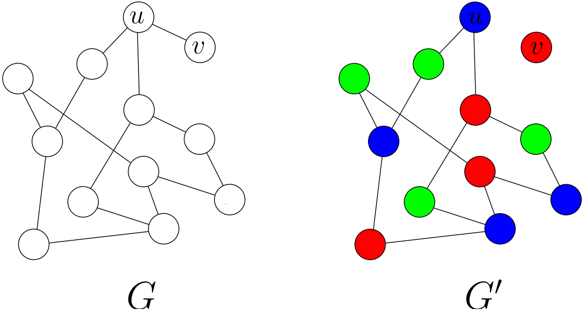


Observation



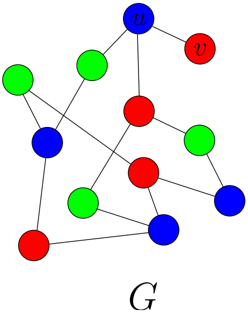
A random colouring of G can be seen as a random colouring of the **simpler** G' conditional that v, u receive **different colours**.

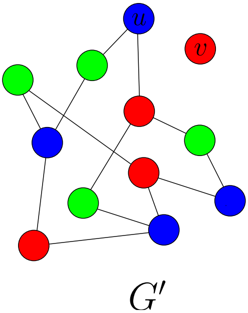
Observation



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

Observation





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

Aim at ....

`Update`

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

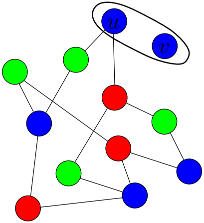
Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.

Aim at ....

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.

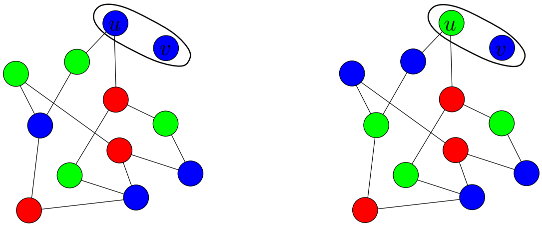


Aim at ....

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.

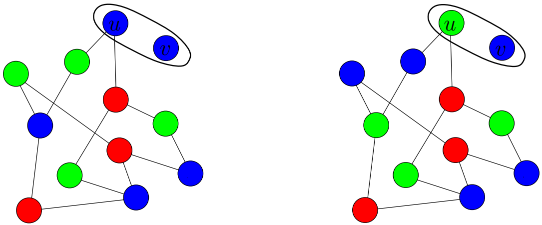


Aim at ....

Update

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

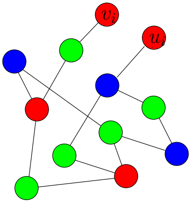
Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.



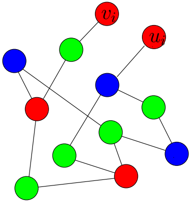
Be careful...
We can not change the colours of the vertices **arbitrarily**.

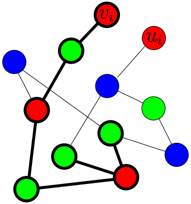
How does `Update` look like?

How does `Update` look like?

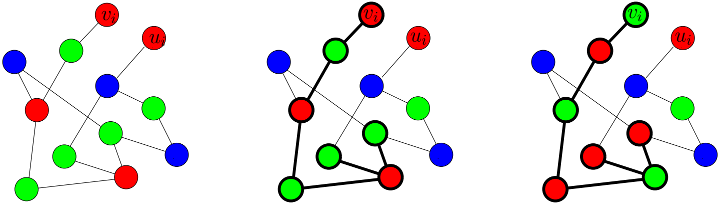


How does Update look like?



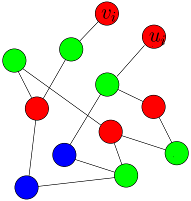


How does `Update` look like?

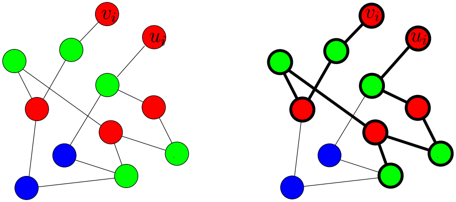


... why approximate sampling?

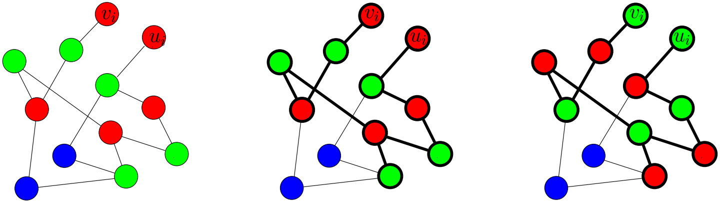
... why approximate sampling?



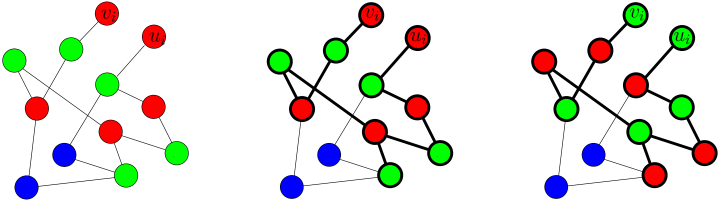
... why approximate sampling?



… why approximate sampling?



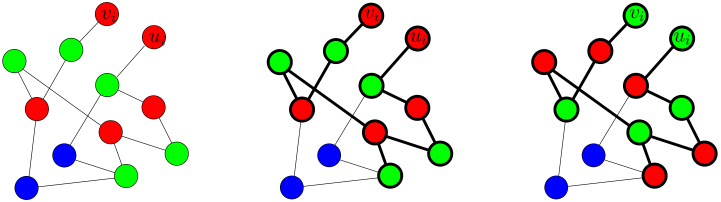
… why approximate sampling?



Failure

When both v_i and u_i change colour `Update` fails

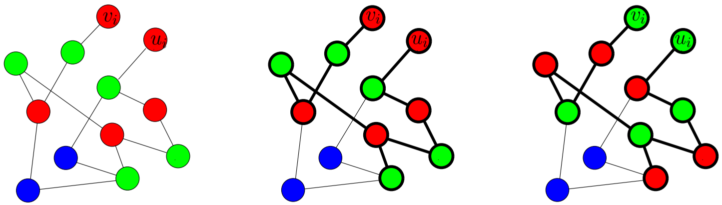
… why approximate sampling?



Failure Vs Approximation

Because of the failures `Update` is an approximation algorithm

... why approximate sampling?

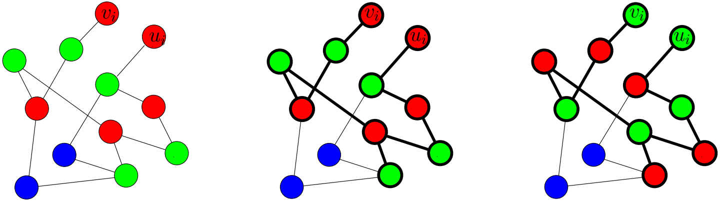


Failure Vs Approximation

Because of the failures `Update` is an approximation algorithm

- the output is *approximately* Gibbs distributed

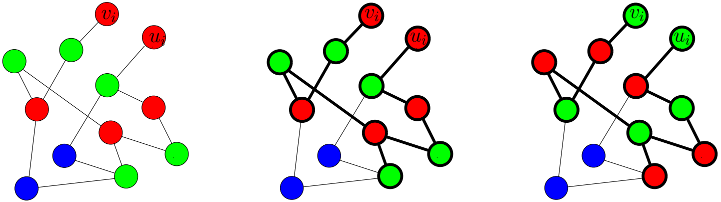
... why approximate sampling?



ℓ_1-error for `Update`

- having a perfect sample at the input
- ℓ_1-error $\approx$ the probability of failure

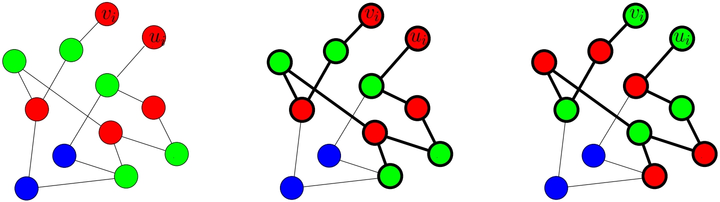
… why approximate sampling?



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

$$\ell_1\text{-error} \approx \mathrm{Prob}[\text{there is a failure is some iteration}]$$

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)

Some Remarks

Some Remarks

- The idea was proposed in [Efthymiou 2012]
  - specific to graph colourings
  - further improved in [Efthymiou 2016]
    - we need $q > d + 1$

Some Remarks

- The idea was proposed in [Efthymiou 2012]
  - specific to graph colourings
  - further improved in [Efthymiou 2016]
    - we need $q > d + 1$
- [Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang 2020]
  - Potts model in random regular graphs
  - the algorithm for ferromagnetic Potts apply to $G(n, m)$

Some Remarks

- The idea was proposed in [Efthymiou 2012]
  - specific to graph colourings
  - further improved in [Efthymiou 2016]
    - we need $q > d + 1$
- [Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang 2020]
  - Potts model in random regular graphs
  - the algorithm for ferromagnetic Potts apply to $G(n, m)$
- all previous approaches are **special** to the sampled distribution

Some Remarks

- The idea was proposed in [Efthymiou 2012]
  - specific to graph colourings
  - further improved in [Efthymiou 2016]
    - we need $q > d + 1$
- [Blanca, Galanis, Goldberg, Stefankovic, Vigoda, Yang 2020]
  - Potts model in random regular graphs
  - the algorithm for ferromagnetic Potts apply to $G(n, m)$
- all previous approaches are **special** to the sampled distribution
- Aim here: the distribution to be a **parameter** of the algorithm

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

The same idea

The same idea

The sampling algorithm

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

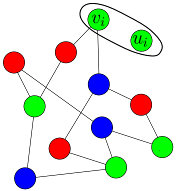
$G_0, G_1, \ldots, G_r = G$

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$



The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

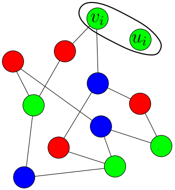
$G_0, G_1, \ldots, G_r = G$

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$



`Update`

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

The same idea

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

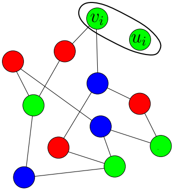
$G_0, G_1, \ldots, G_r = G$

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

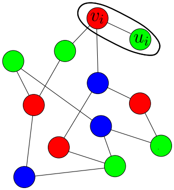
$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$



Update



The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

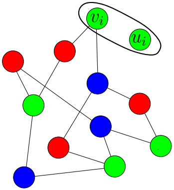
−get G_i from G_{i+1} by deleting the random edge $\{v_i, u_i\}$

−G_0 is **empty**

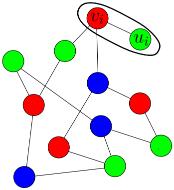
Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_{i-1}$ to generate **efficiently** $\boldsymbol{\sigma}_i$

Output: $\boldsymbol{\sigma}_r$



Update



Approach

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting ...

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting ...

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

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting ...

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

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting ...

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

Approach

The challenge is to define `Update`, ... generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting ...

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

Approach

The challenge is to define `Update`, … generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting …

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

The challenge is to define `Update`, … generate $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}_{i-1}$

Setting …

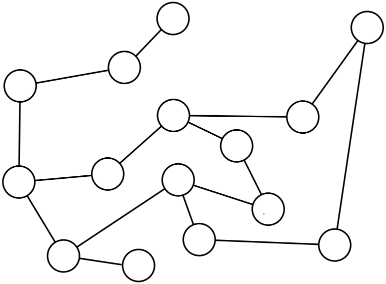
- symmetric Gibbs distribution
  - … e.g. antiferromagnetic Ising, or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$
  - … assume that both are of **high girth**
- Gibbs distributions μ and μ' on G and G', resp.
- configuration $\boldsymbol{\sigma}$ distributed as in μ

Objective

Generate efficiently $\boldsymbol{\tau}$ distributed (approximately) as in μ'

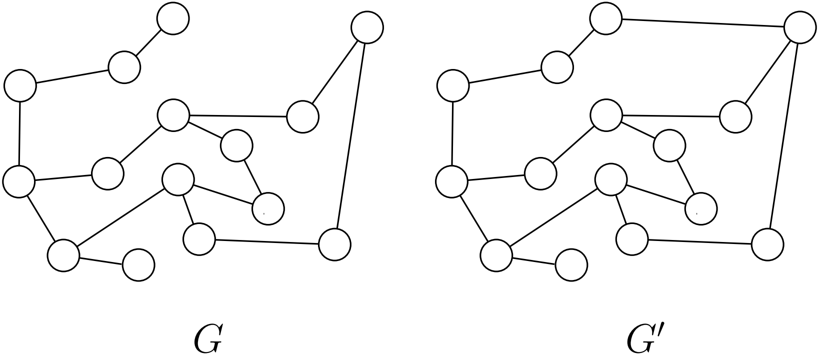
Coupling-Based Solution

Coupling-Based Solution

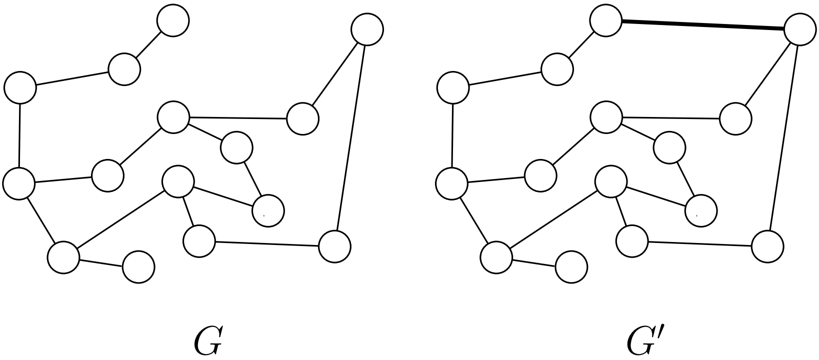


G

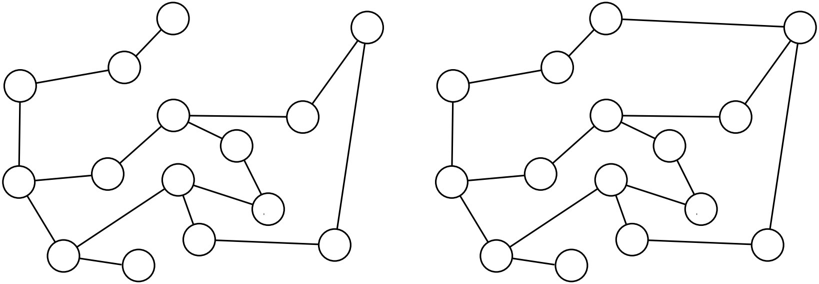
Coupling-Based Solution



Coupling-Based Solution

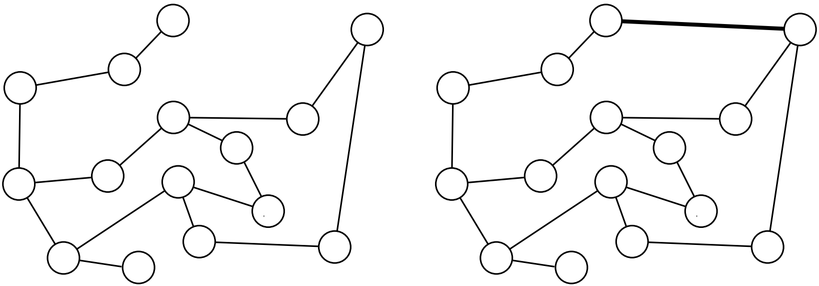


Coupling-Based Solution



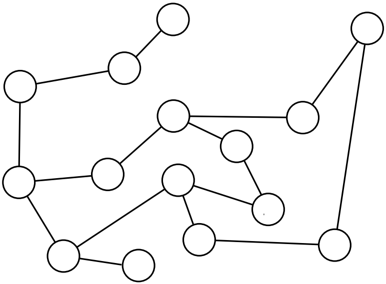
G $\quad$ G'

Coupling-Based Solution

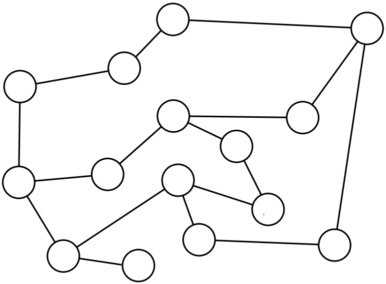


G $\qquad$ G'

Coupling-Based Solution

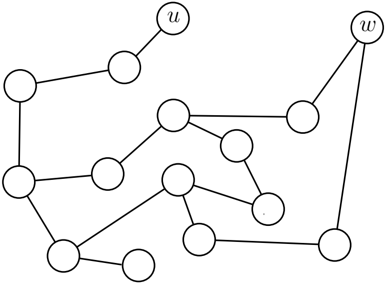


G

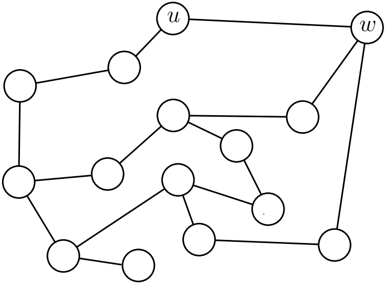


G'

Coupling-Based Solution

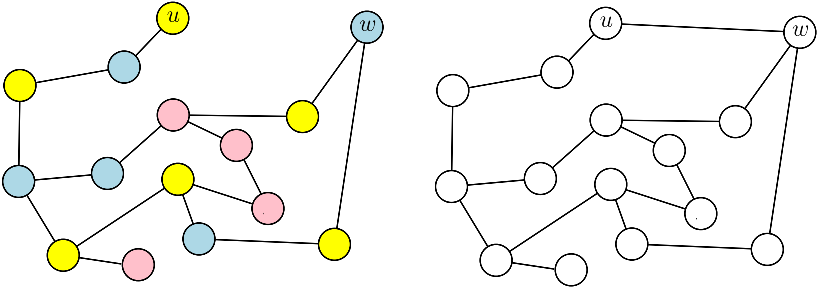


G



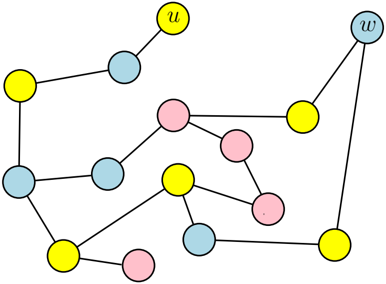
G'

Coupling-Based Solution

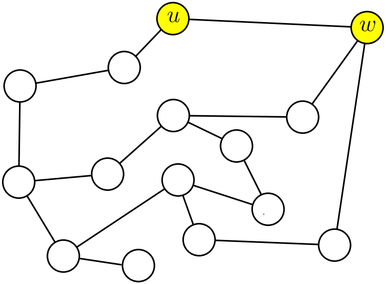


(G, σ) G'

Coupling-Based Solution

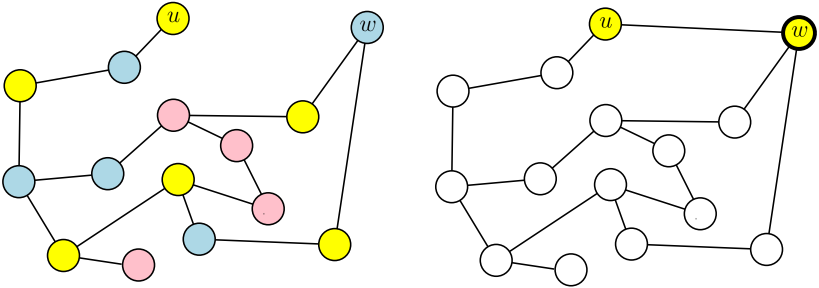


(G, σ)



G'

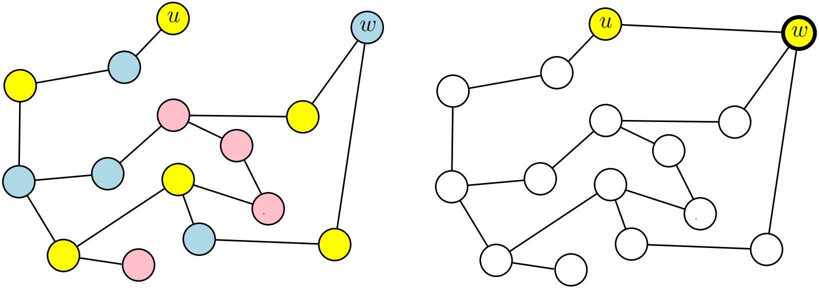
Coupling-Based Solution



(G, σ) $\qquad$ G'

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

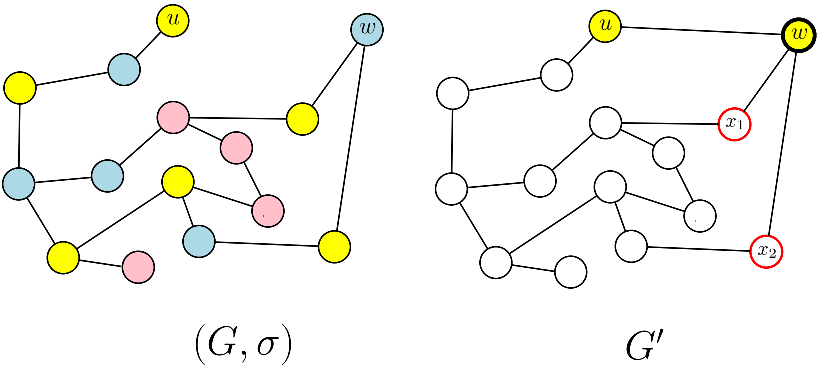
Coupling-Based Solution



(G, σ) G'

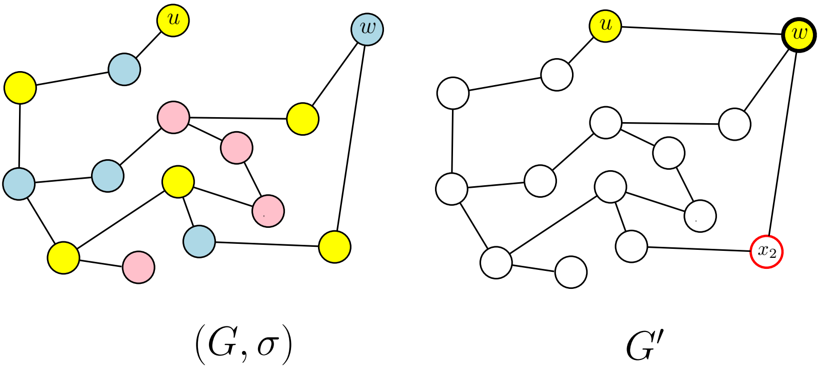
iteratively visit each vertex in G' and decide its configuration at τ

Coupling-Based Solution



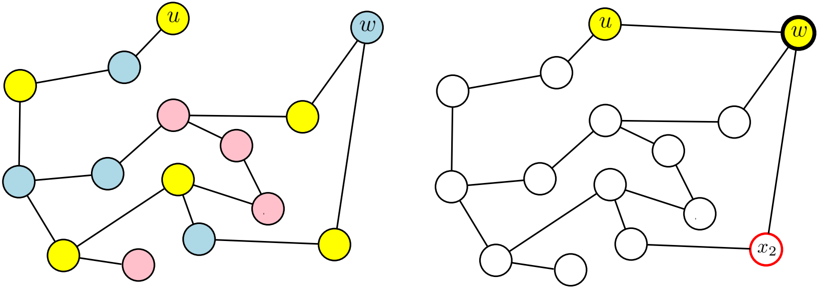
Priority to z's with $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$, next to disagreement.

Coupling-Based Solution



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

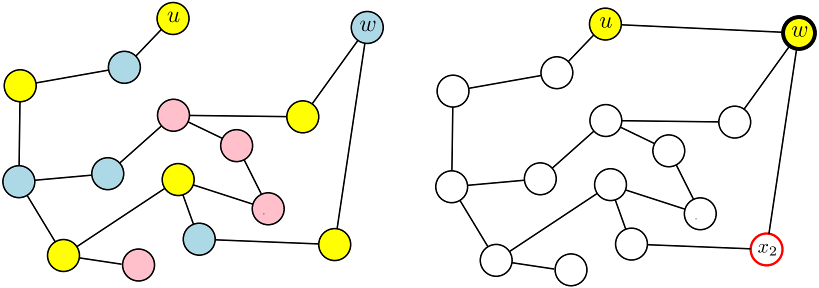
Coupling-Based Solution



(G, σ) $\qquad$ G'

the probability of disagreement is minimised by using **coupling maximally**

Coupling-Based Solution

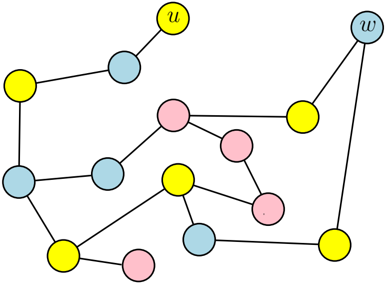


(G, σ) $\qquad$ G'

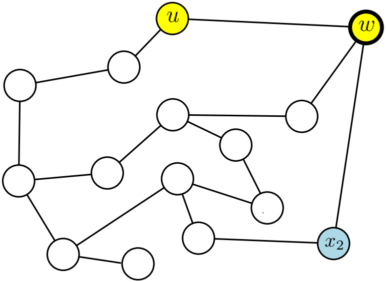
maximal coupling

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

Coupling-Based Solution



(G, σ)

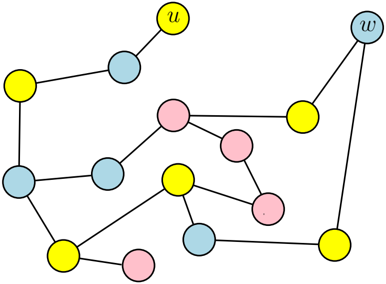


G'

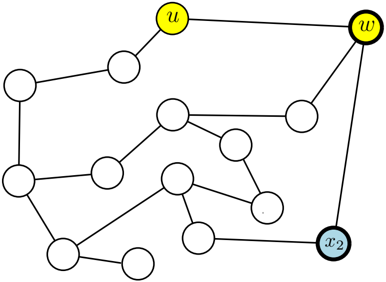
maximal coupling

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

Coupling-Based Solution



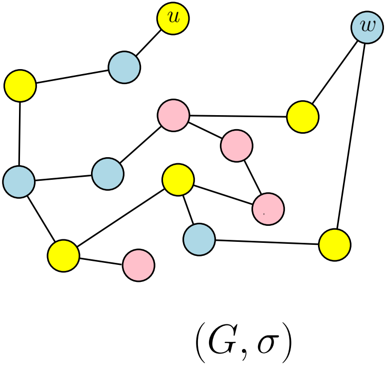
(G, σ)

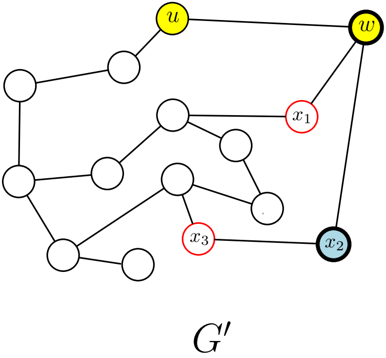


G'

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

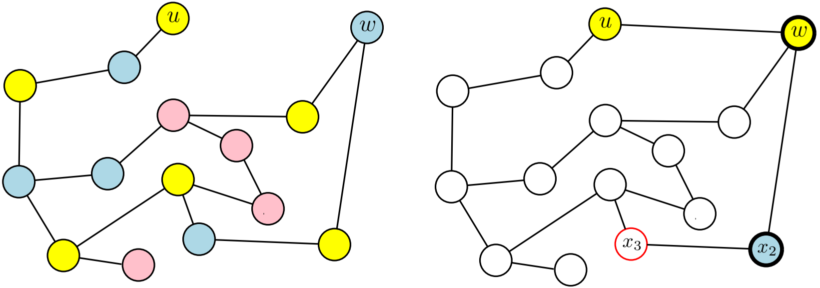
Coupling-Based Solution





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

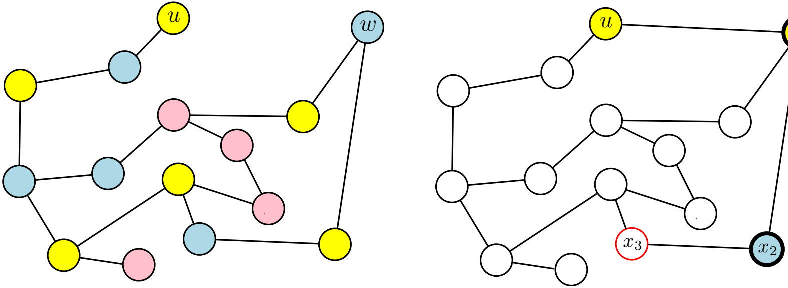
Coupling-Based Solution



(G, σ) $\qquad$ G'

choose x_3 and repeat as before . . .

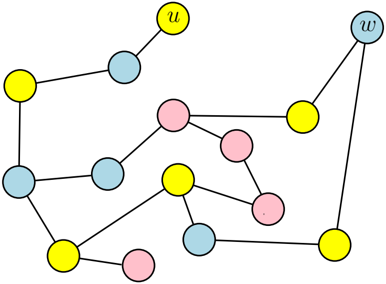
Coupling-Based Solution



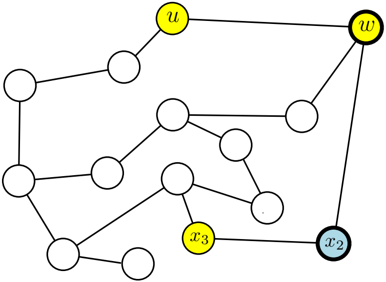
(G, σ) $\qquad$ G'

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

Coupling-Based Solution

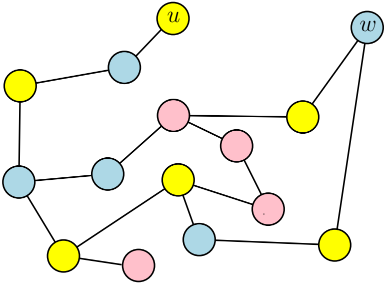


(G, σ)

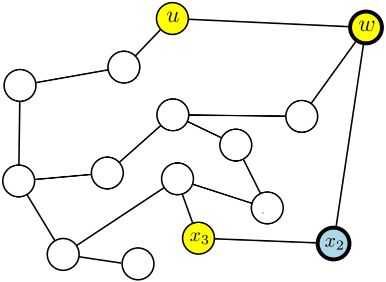


G'

Coupling-Based Solution



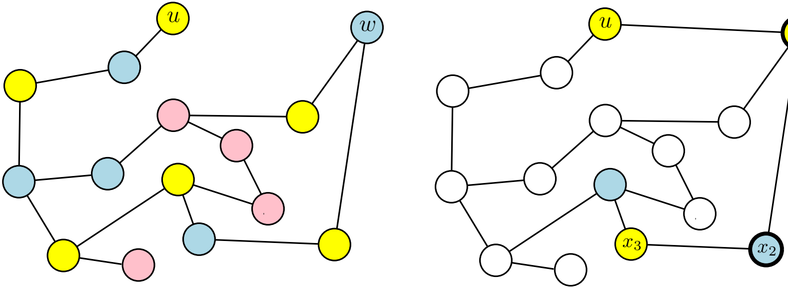
(G, σ)



G'

repeat in the same way for the rest of the vertices

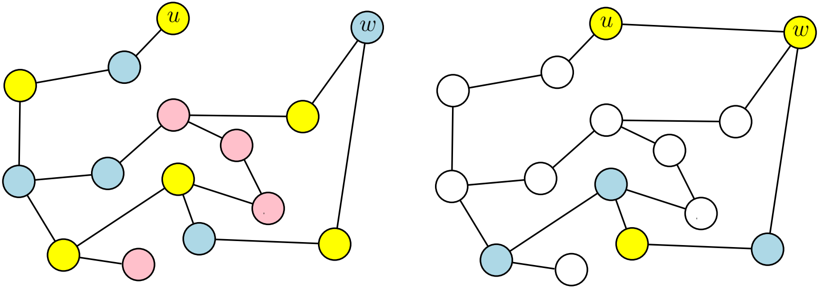
Coupling-Based Solution



(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

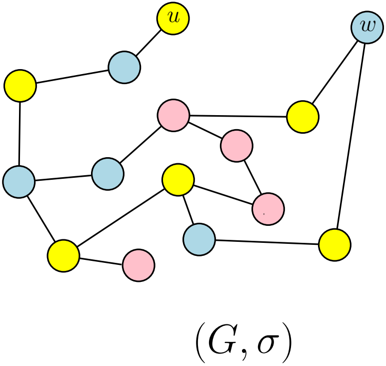
Coupling-Based Solution

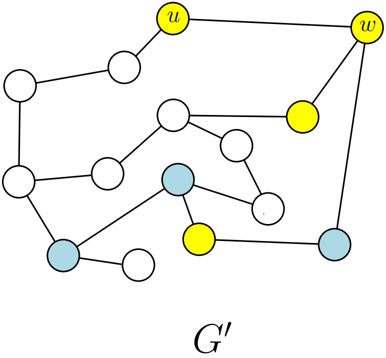


(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

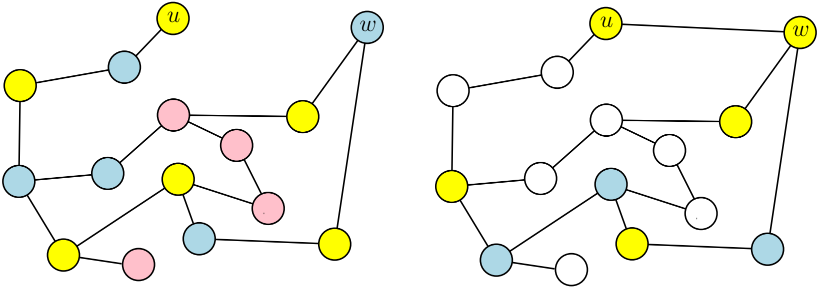
Coupling-Based Solution





repeat in the same way for the rest of the vertices

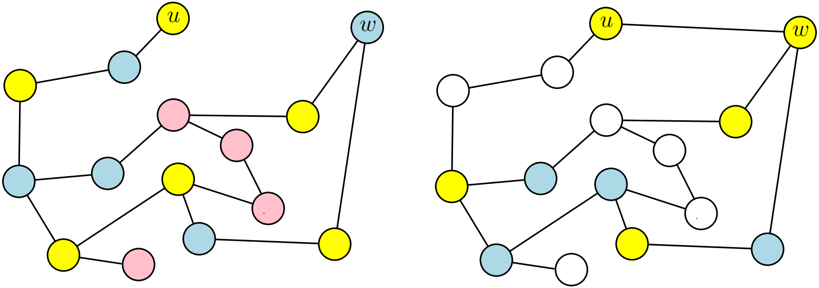
Coupling-Based Solution



(G, σ) $\qquad\qquad$ G'

repeat in the same way for the rest of the vertices

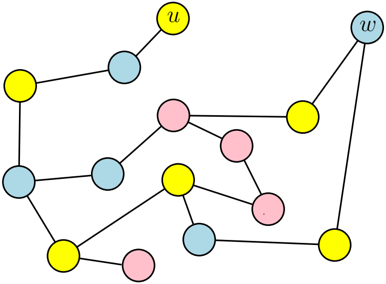
Coupling-Based Solution



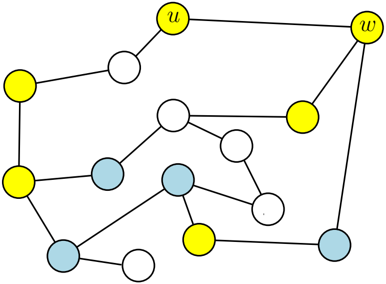
(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

Coupling-Based Solution



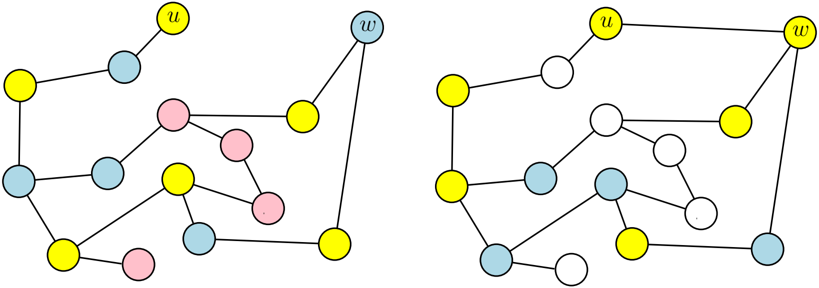
(G, σ)



G'

repeat in the same way for the rest of the vertices

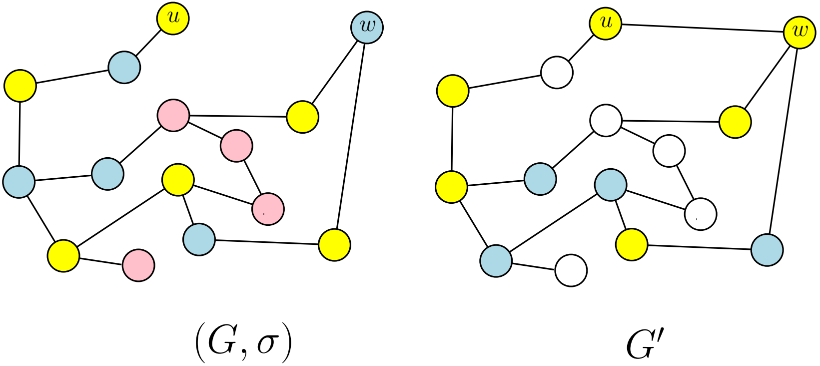
Coupling-Based Solution



(G, σ) $\qquad\qquad$ G'

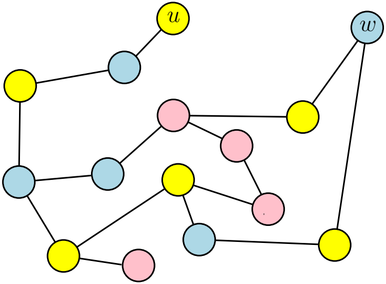
disagreement cannot propagate any more

Coupling-Based Solution

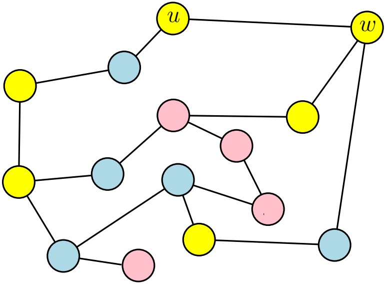


the remaining vertices keep the initial assignments.

Coupling-Based Solution



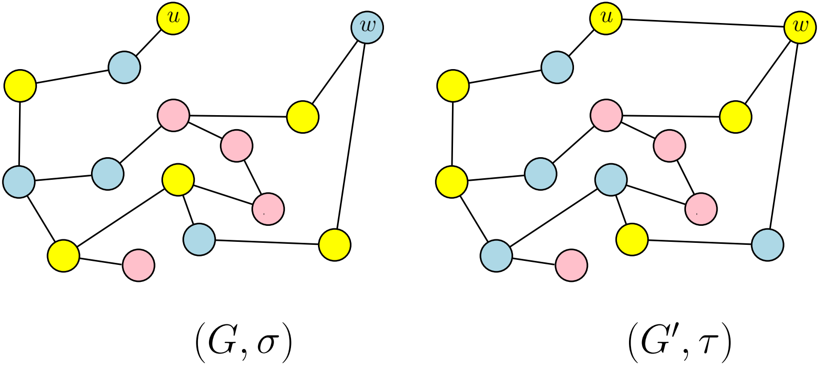
(G, σ)



(G', τ)

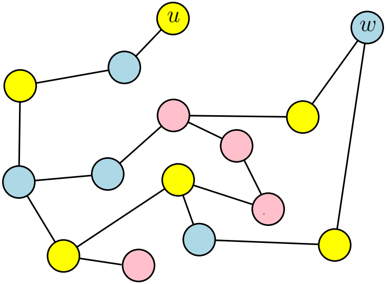
the remaining vertices keep the initial assignments.

Coupling-Based Solution

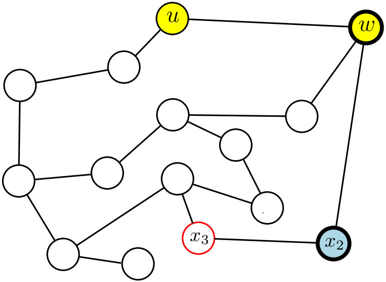


the approach generates a **perfect** sample from μ'

Coupling-Based Solution



(G, σ)

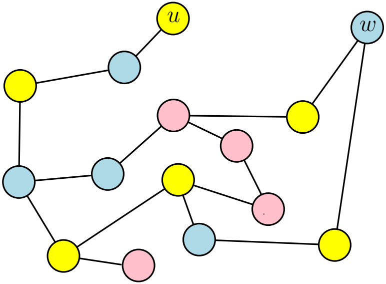


G'

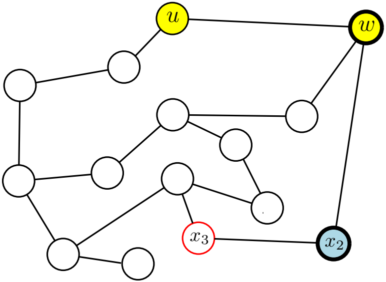
The catch ...

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

Coupling-Based Solution



(G, σ)

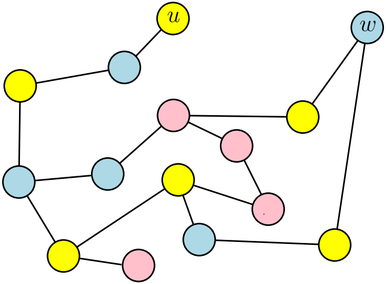


G'

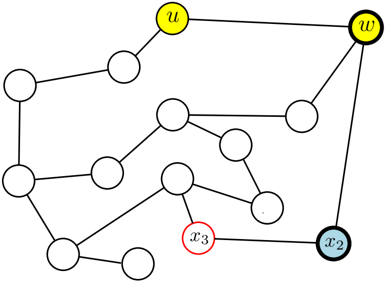
The catch . . .

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

Coupling-Based Solution



(G, σ)



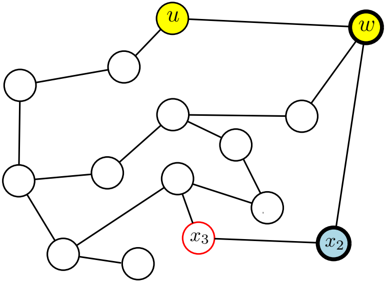
G'

The idea . . .

replace the Gibbs marginals with "good" approximations that can be computed efficiently

Some Intuition . . .

Some Intuition . . .

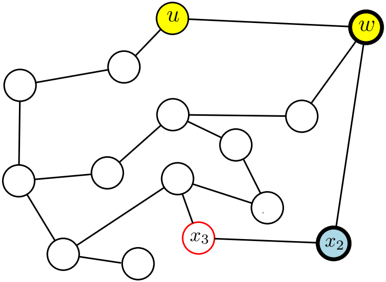


G'

Observation . . .

influences from vertices with fixed configuration make the Gibbs marginals at x_3 **too complicated** an object

Some Intuition . . .

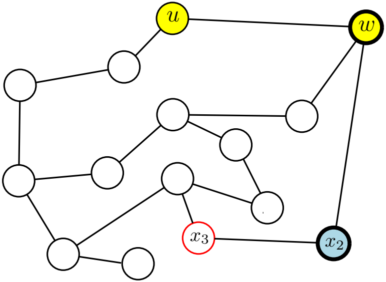


G'

However . . .

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

Some Intuition . . .

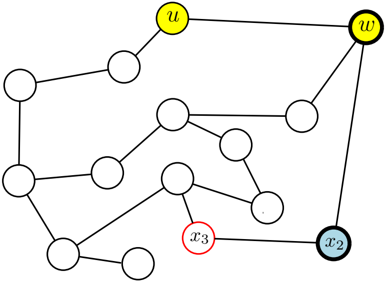


G'

Choosing the appropriate parameters . . .

essentially only one vertex influences the marginal

Some Intuition . . .

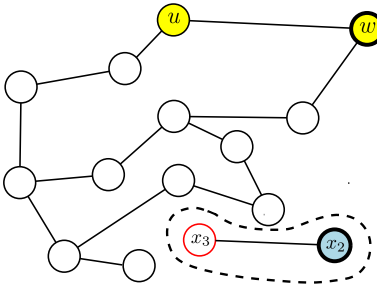


G'

Compute marginal but . . .

ignore the influence on x_3 from u and w

Some Intuition . . .

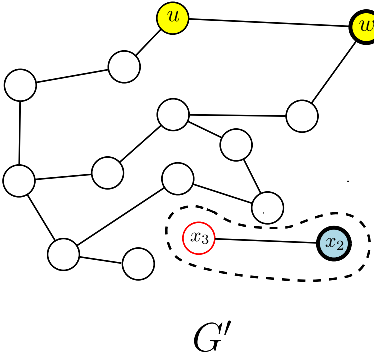


G'

Effectively

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

Some Intuition . . .

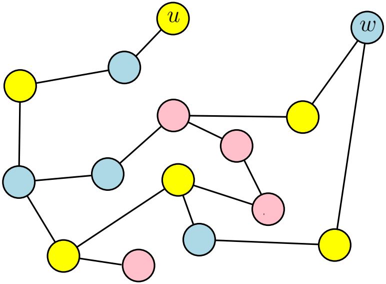


Remark

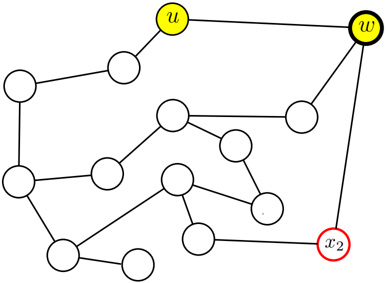
we can compute the "simplified" marginal at x_3 in $O(1)$ steps

To sum up . . .

To sum up . . .



(G, σ)

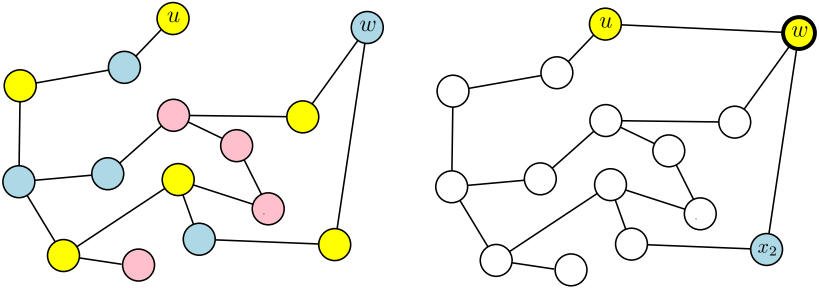


G'

"maximal coupling"

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

To sum up ...



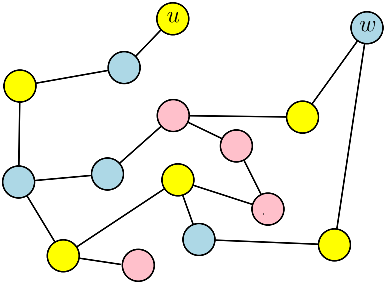
(G, σ)

G'

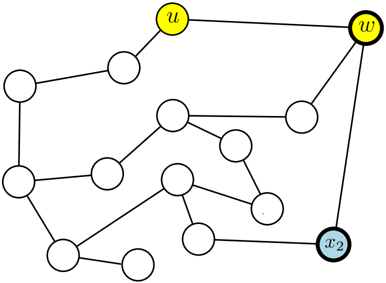
"maximal coupling"

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

To sum up . . .



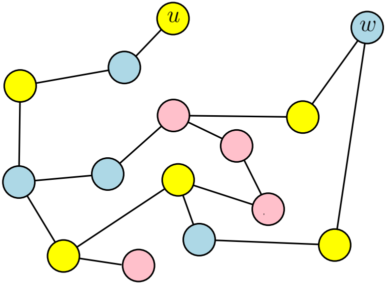
(G, σ)



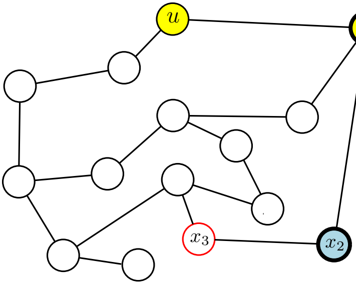
G'

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

To sum up . . .



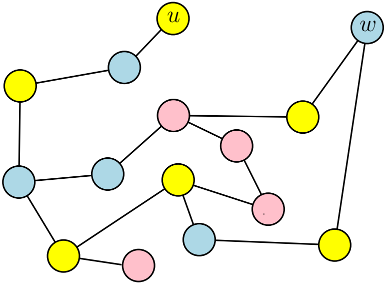
(G, σ)



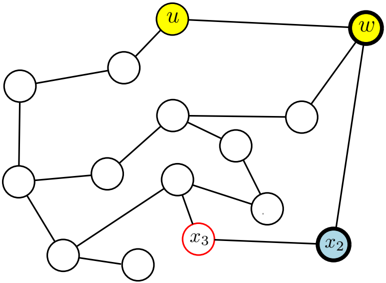
G'

choose x_3 and repeat as before . . .

To sum up . . .



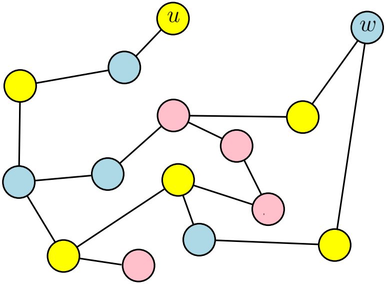
(G, σ)



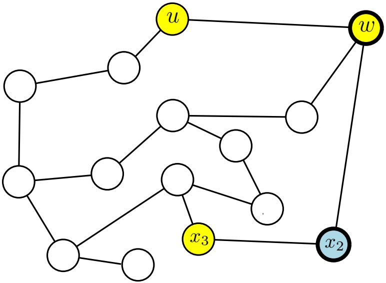
G'

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

To sum up . . .

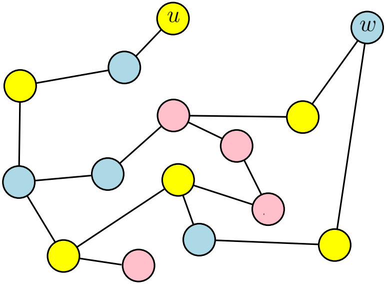


(G, σ)

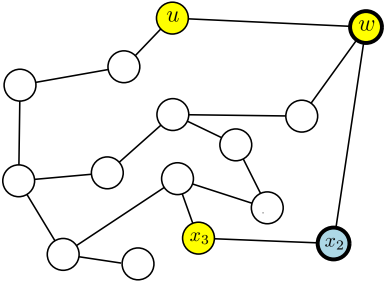


G'

To sum up ...



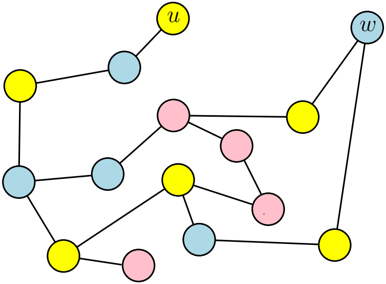
(G, σ)



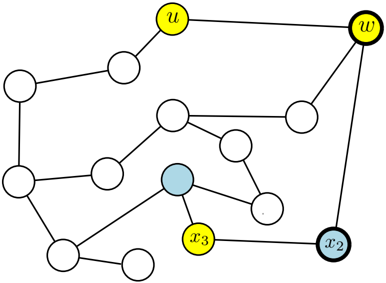
G'

repeat in the same way for the rest of the vertices

To sum up . . .



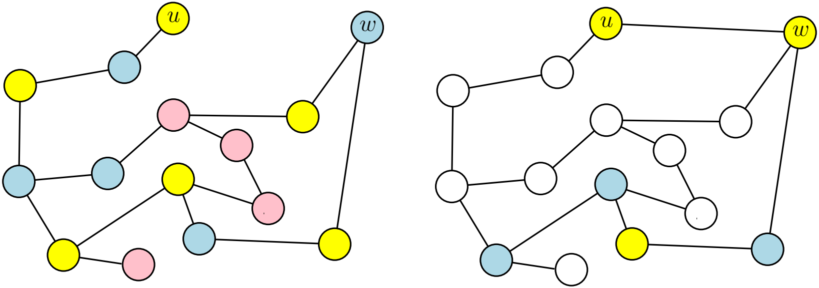
(G, σ)



G'

repeat in the same way for the rest of the vertices

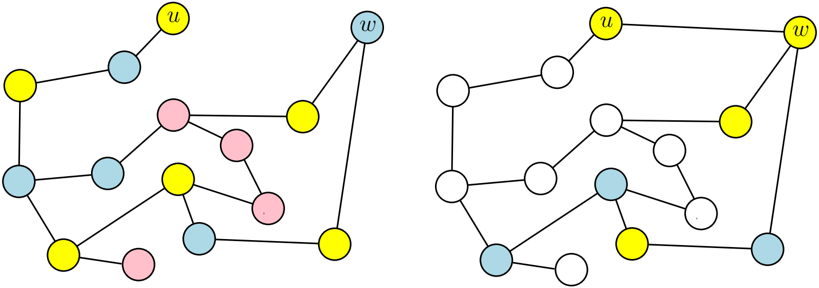
To sum up ...



(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

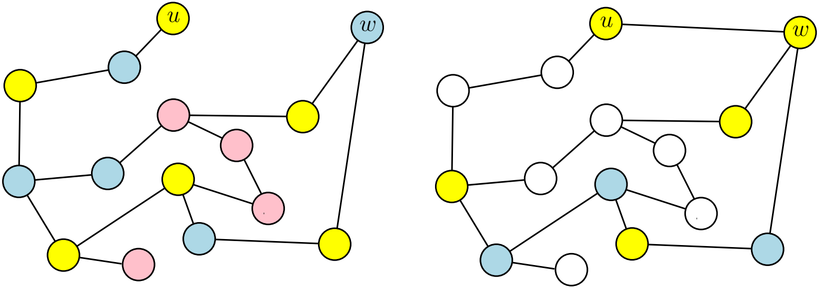
To sum up . . .



(G, σ) $\qquad\qquad$ G'

repeat in the same way for the rest of the vertices

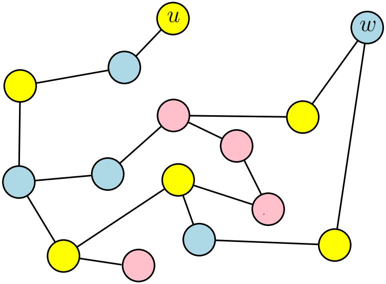
To sum up . . .



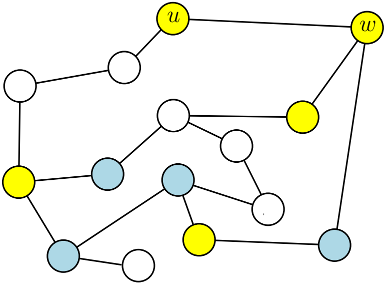
(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

To sum up . . .



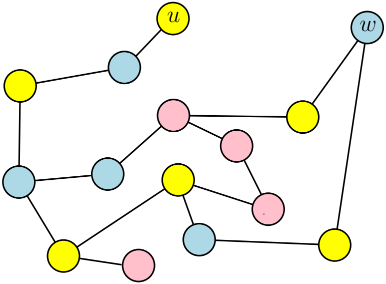
(G, σ)



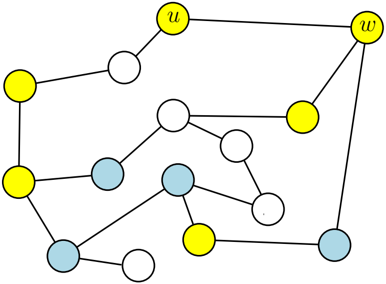
G'

repeat in the same way for the rest of the vertices

To sum up . . .



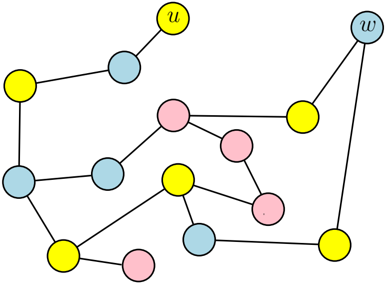
(G, σ)



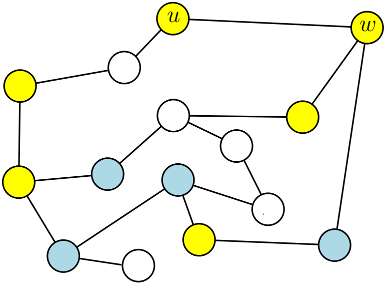
G'

repeat in the same way for the rest of the vertices

To sum up . . .



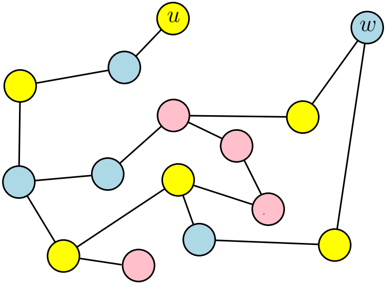
(G, σ)



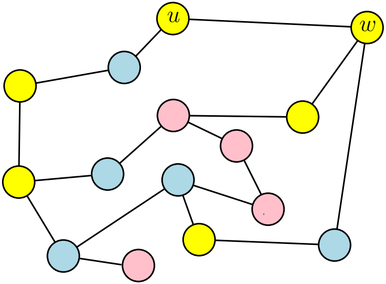
G'

when the disagreements cannot propagate any more
the remaining vertices keep the same assignment

To sum up . . .

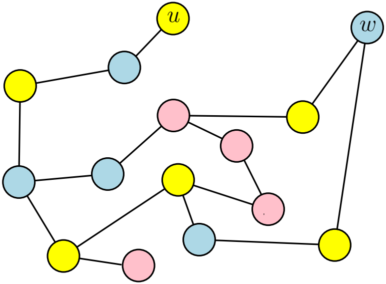


(G, σ)

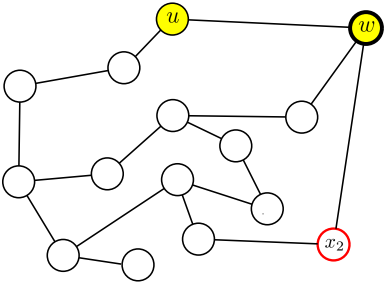


(G', τ)

To sum up ...



(G, σ)



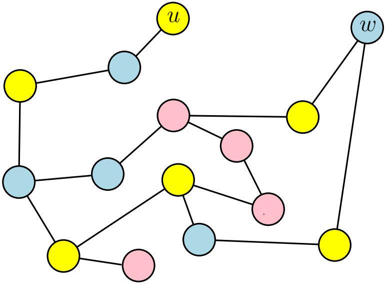
G'

... another catch

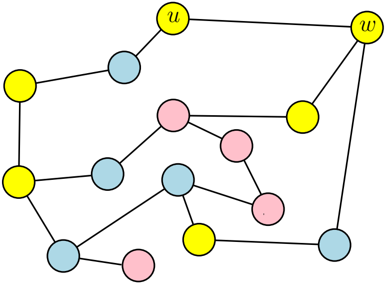
disagreements should not

- reach neighbours of the vertex u
- cover *all* the vertices of a cycle in G'

To sum up . . .



(G, σ)



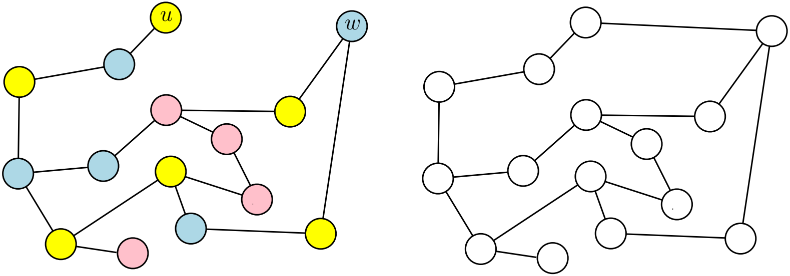
(G', τ)

Otherwise . . .

we have a **failure**!

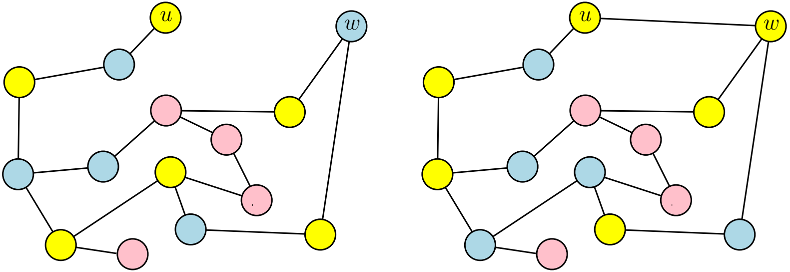
Failure Vs Approximation

Failure Vs Approximation



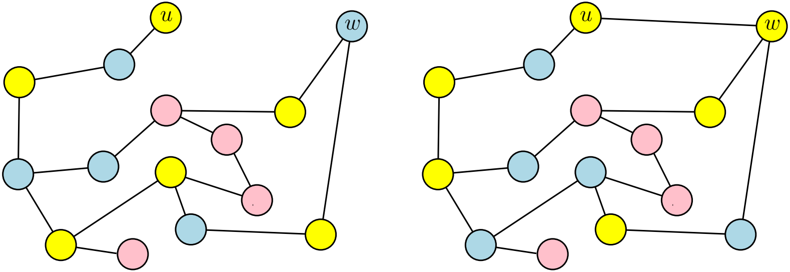
(G, σ) $\qquad\qquad$ G'

Failure Vs Approximation



(G, σ) (G', τ)

Failure Vs Approximation



(G, σ) (G', τ)

If $\boldsymbol{\sigma} \sim \mu(\cdot)$, then

$$||\mu_{\text{update}}(\cdot) - \mu'(\cdot)|| \leq \Pr[\ \texttt{Update}(\boldsymbol{\sigma})\ \text{Fails}]$$

The iterative algorithm

The iterative algorithm

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_i$ with `Update` to generate $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$

The iterative algorithm

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_i$ with `Update` to generate $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$

The ℓ_1 error for the algorithm

$\approx$ probability of failure at some iteration

The iterative algorithm

The sampling algorithm

Input: $G = (V, E)$, Gibbs distribution μ

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

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

$-G_0$ is **empty**

Generate $\boldsymbol{\sigma}_0$ according to the Gibbs distribution at G_0

Iteratively: use $\boldsymbol{\sigma}_i$ with `Update` to generate $\boldsymbol{\sigma}_{i+1}$

Output: $\boldsymbol{\sigma}_r$

The time complexity

the time complexity is $O(|E| \times |V|)$

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

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

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
  - 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 $n^{-\Omega(1)}$

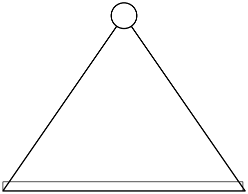
The parameters

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

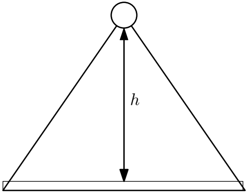
- good approximation $\Rightarrow$ error $n^{-\Omega(1)}$
- need to have **local changes** in the `Update`

Gibbs Tree Uniqueness

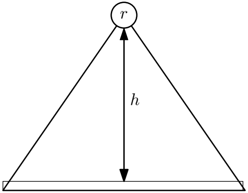
Gibbs Tree Uniqueness



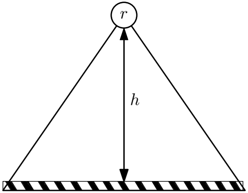
Gibbs Tree Uniqueness



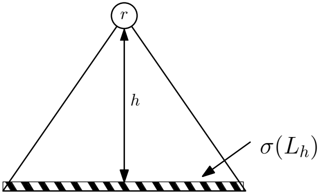
Gibbs Tree Uniqueness



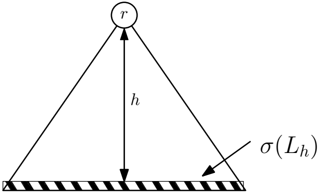
Gibbs Tree Uniqueness



Gibbs Tree Uniqueness

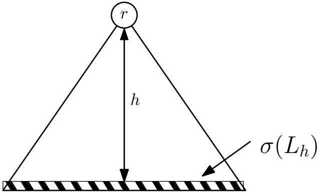


Gibbs Tree Uniqueness



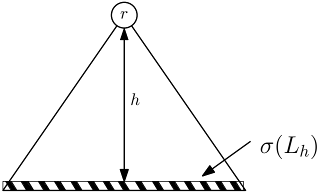
$$||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

Gibbs Tree Uniqueness



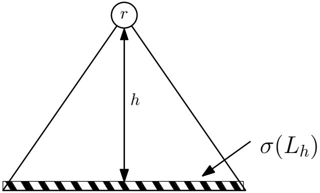
$$\lim_{h \to \infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

Gibbs Tree Uniqueness



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

Gibbs Tree Uniqueness



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

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

Use different condition...

Use different condition...

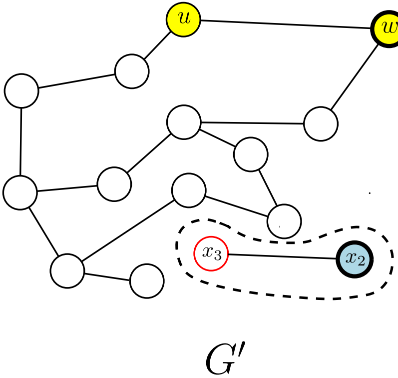
- for many distributions here uniqueness is **not established**
  - there are only conjectures

Use different condition...

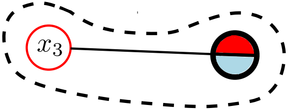
- for many distributions here uniqueness is **not established**
  - there are only conjectures
- for hypergraphs uniqueness is **too restrictive** a condition
  - go beyond uniqueness

Influence Condition

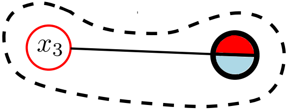
Influence Condition



Influence Condition



Influence Condition



$$\max_{\eta,\theta} ||\mathfrak{m}_{x_3}(\cdot \mid \eta) - \mathfrak{m}_{x_3}(\cdot \mid \theta)|| < 1/d$$

Reconsider the order of randomness

Reconsider the order of randomness

Uniform Model

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

Reconsider the order of randomness

Uniform Model

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

Teacher-Student Model

Reconsider the order of randomness

Uniform Model

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

Teacher-Student Model

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

Reconsider the order of randomness

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^*

Reconsider the order of randomness

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`

Reconsider the order of randomness

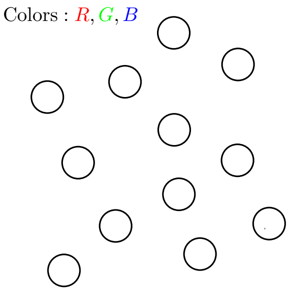
Uniform Model

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



Reconsider the order of randomness

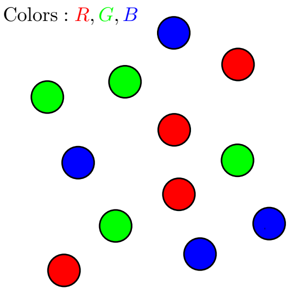
Uniform Model

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



Reconsider the order of randomness

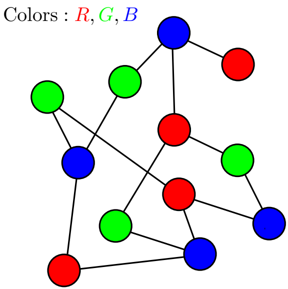
Uniform Model

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

Teacher-Student Model

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

Planting Colourings



Reconsider the order of randomness

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`

Reconsider the order of randomness

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

Reconsider the order of randomness

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

Reconsider the order of randomness

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
- with Influence Cond., the failure probability is **very small**

Reconsider the order of randomness

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
- with Influence Cond., the failure probability is **very small**
- ... this implies small failure probability for the "real process"

Reconsider the order of randomness

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
- with Influence Cond., the failure probability is **very small**
- ... this implies small failure probability for the "real process"
- the above can be true if **contiguity holds**

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**.

Condition 2

Contiguity between the Gibbs distribution and the corresponding Teacher Student model

Remarks

Remarks

- Influence Cond. is more restrictive than Contiguity
  - Contiguity holds up to "Replica Symmetry Breaking"

Remarks

- Influence Cond. is more restrictive than Contiguity
    - Contiguity holds up to "Replica Symmetry Breaking"
- For graphs, Influence Cond. coincides with the (conjectured) Gibbs Uniqueness

Remarks

- Influence Cond. is more restrictive than Contiguity
  - Contiguity holds up to "Replica Symmetry Breaking"
- For graphs, Influence Cond. coincides with the (conjectured) Gibbs Uniqueness
- For hyper-graphs, Influence Cond. gets us *beyond uniqueness*
  - This gets us to "non-reconstruction" region

Concluding Remarks

Concluding Remarks

- Presented a novel approximate sampling algorithm

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$
  - **any** fixed expected degree $d > 1$

Concluding Remarks

- Presented a novel approximate sampling algorithm
  - underlying graph is $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$
  - **any** fixed expected degree $d > 1$
  - works for **any** symmetric Gibbs distribution

Concluding Remarks

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

Concluding Remarks

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

Concluding Remarks

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

Concluding Remarks

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

Concluding Remarks

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

Concluding Remarks

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

The end

Thank you!