Sampling symmetric Gibbs distributions on the sparse random graph and hypergraph

Charis Efthymiou
University of Warwick

Algorithms, Dynamics, and Information Flow in Networks

Algorithm Dynamics Information Flow June, 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)

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

Remarks

- for $q = 2$ we have the Ising model
- for $\beta = -\infty$ we have the Colouring 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
  - monochromatic edges are not allowed!

Efficient Sampling Problem

Efficient Sampling Problem

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

Efficient Sampling Problem

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

Status of the problem . . .

Efficient Sampling Problem

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

Status of the problem . . .

- worst-case the problem is *computationally hard*

Efficient Sampling Problem

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

Status of the problem ...

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

Efficient Sampling Problem

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

Status of the problem ...

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

Efficient Sampling Problem

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

Status of the problem ...

- worst-case the problem is *computationally hard*
- generate efficiently $\boldsymbol{\sigma}$ which is distributed "*close*" to μ
  - approximate sampling
- *there are cases* where even approximate sampling is hard

Efficient Sampling Problem

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

Status of the problem ...

- worst-case the problem is *computationally hard*
- generate efficiently $\boldsymbol{\sigma}$ which is distributed "*close*" to μ
  - approximate sampling
- *there are cases* where even approximate sampling is hard
- the *range of parameters* of the problem in which we can get "*good*" approximations

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 $\boldsymbol{G}(n, m)$

The sparse random graph

$\boldsymbol{G}(n, m)$ 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 $m = \Theta(n)$

Two levels of randomness

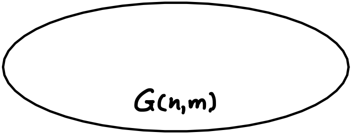
The case of $G(n, m)$

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 $m = \Theta(n)$

Two levels of randomness



Level 1

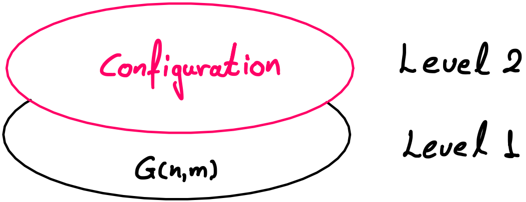
The case of $G(n, m)$

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 $m = \Theta(n)$

Two levels of randomness



Sampling Problem on $G(n, m)$

Sampling Problem on $G(n, m)$

- focus on approximate sampling

Sampling Problem on $G(n, m)$

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

Sampling Problem on $G(n, m)$

- focus on approximate sampling
- use concepts from **physics** for better algorithms
- intuition from the **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
    - Glauber dynamics, Metropolis Process

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
  - Glauber dynamics, Metropolis Process
- Message Passing Algorithms

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
    - Glauber dynamics, Metropolis Process
- Message Passing Algorithms
    - Belief Propagation

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
  - Glauber dynamics, Metropolis Process
- Message Passing Algorithms
  - Belief Propagation
- Other approaches

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
  - Glauber dynamics, Metropolis Process
- Message Passing Algorithms
  - Belief Propagation
- Other approaches
  - Many of those you listen in here in the summer-school.

Popular approaches to sampling problem

- Markov Chain Monte Carlo method
  - Glauber dynamics, Metropolis Process
- Message Passing Algorithms
  - Belief Propagation
- Other approaches
  - Many of those you listen in here in the summer-school.

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

What is the idea?

The sampling algorithm

What is the idea?

The sampling algorithm

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

What is the idea?

The sampling algorithm

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

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

What is 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 $\{v_i, u_i\}$

What is 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**

What is 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

What is 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** $\boldsymbol{\sigma}_i$

What is the idea?

The sampling algorithm

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

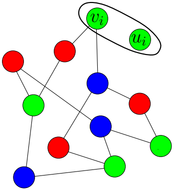
$G_0, G_1, \ldots, G_r = G$

$-$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** $\boldsymbol{\sigma}_i$



What is the idea?

The sampling algorithm

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

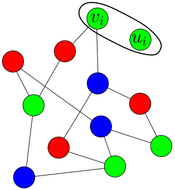
$G_0, G_1, \ldots, G_r = G$

$-$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`

What is 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 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: $\boldsymbol{\sigma}_r$

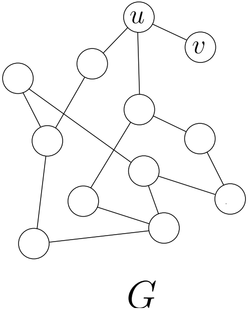
Example from the past

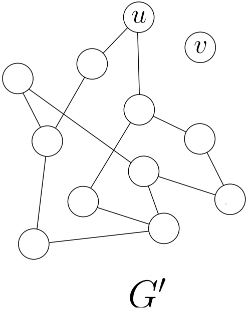
Example from the past

Example with the Colouring Model

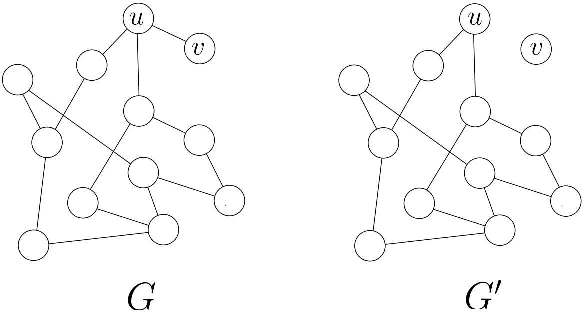
Observation

Observation



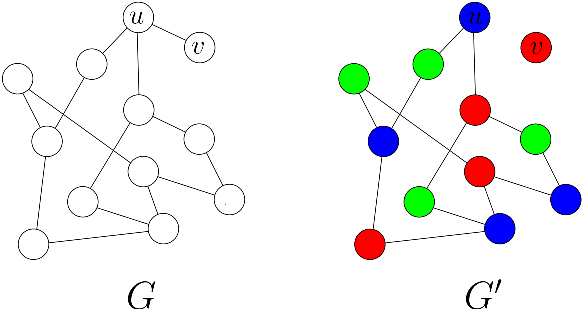


Observation



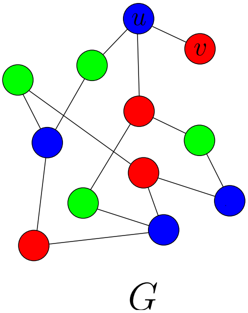
A random colouring of G can be seen as a random colouring of the **simpler** G' conditional that v, u receive **different colours**.

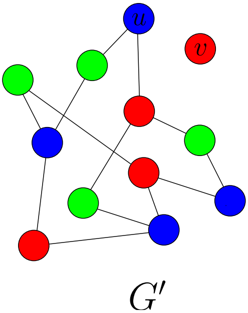
Observation



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.

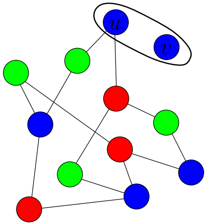
Output: **random** q-colouring of G, conditional u, v are assigned **different** colours.

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.

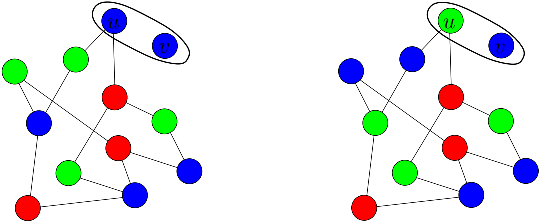


Aim at ....

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.

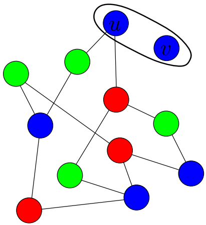


Aim at ....

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.

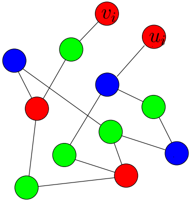


Be careful...

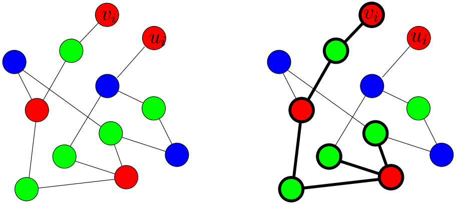
We can not change the colours of the vertices **arbitrarily**.

How does `Update` look like?

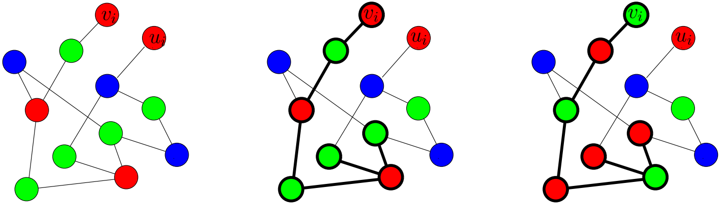
How does `Update` look like?



How does `Update` look like?

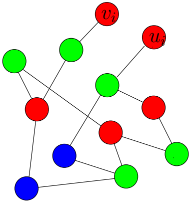


How does `Update` look like?

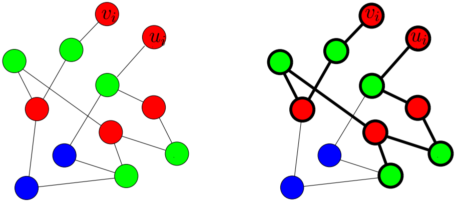


... why approximate sampling?

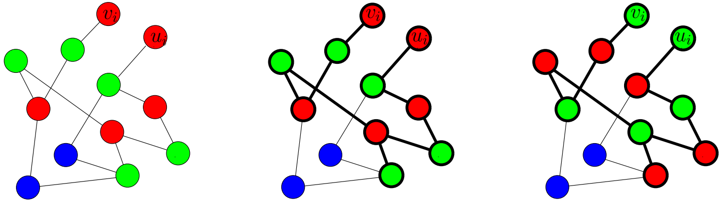
... why approximate sampling?



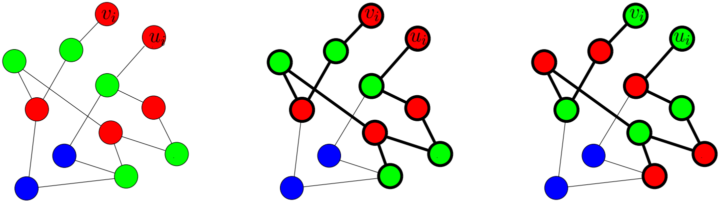
… why approximate sampling?



... why approximate sampling?



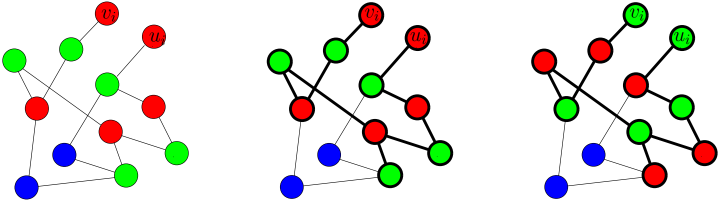
... why approximate sampling?



Failure

When both v_i and u_i change colour `Update` fails

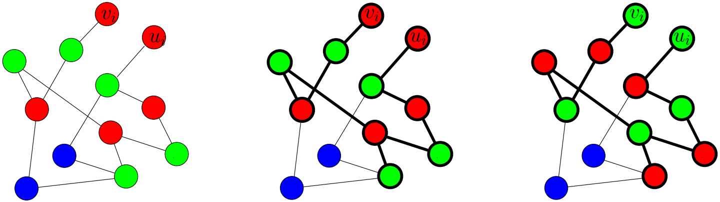
… why approximate sampling?



Failure Vs Approximation

The failures imply that `Update` is an *approximation algorithm*

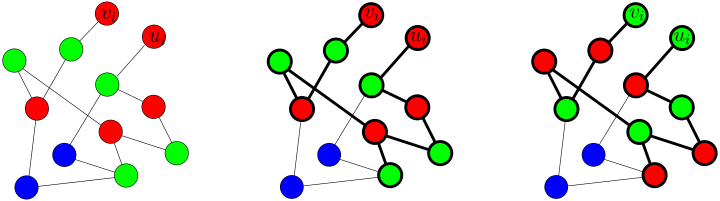
... why approximate sampling?



ℓ_1-error for `Update`

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

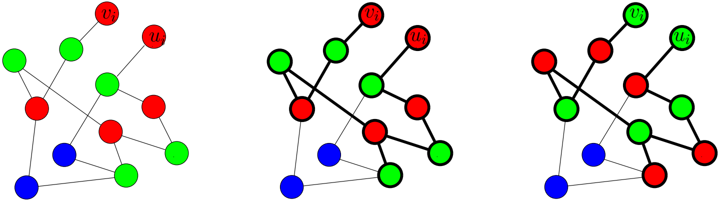
… why approximate sampling?



Approximate Sampler

The sampling algorithm that uses `Update` is approximation, too

... why approximate sampling?



Approximate Sampler

The sampling algorithm that uses `Update` is approximation, too

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

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

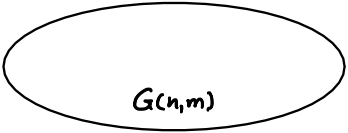
- The idea was proposed in [Efthymiou 2012]
    - specific to graph colourings
    - further improved in [Efthymiou 2016]
        - we need $q > d + 1$

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 $G(n, m)$

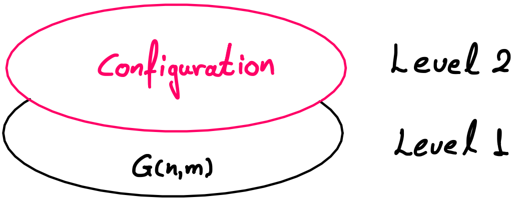
Some hints for the analysis

Some hints for the analysis

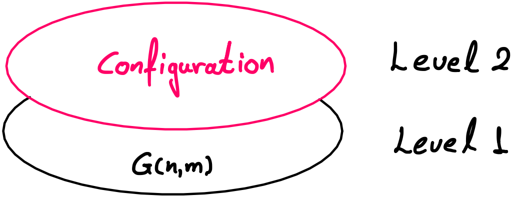


Level 1

Some hints for the analysis

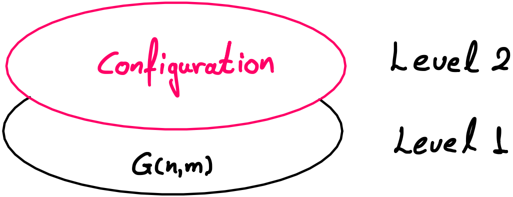


Some hints for the analysis



Objective

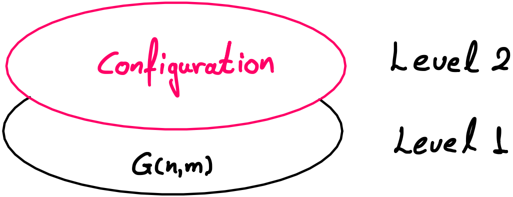
Some hints for the analysis



Objective

- pick at random two distant vertices, e.g., say u and v

Some hints for the analysis



Objective

- pick at random two distant vertices, e.g., say u and v
- with "very large probability" there is no Kempe chain that includes both u and v

The probability of 2-coloured paths (I)

The probability of 2-coloured paths (I)

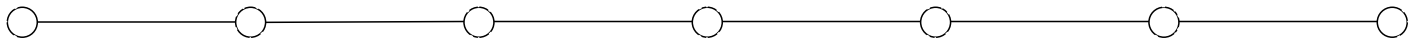
Graph first

The probability of 2-coloured paths (I)

Graph first

The probability of 2-coloured paths (I)

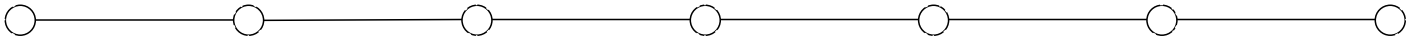
Graph first



The probability of 2-coloured paths (I)

Graph first

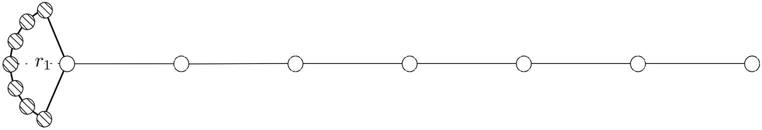
- reveal a neighborhood around each vertex in a BFS manner



The probability of 2-coloured paths (I)

Graph first

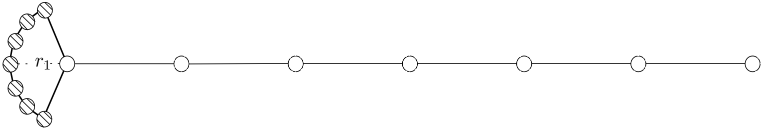
- reveal a neighborhood around each vertex in a BFS manner



The probability of 2-coloured paths (I)

Graph first

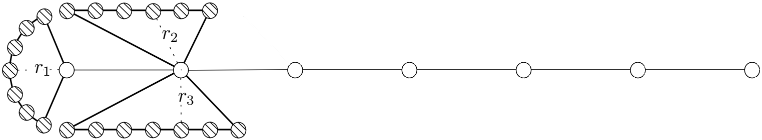
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

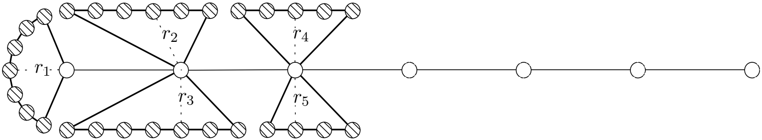
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

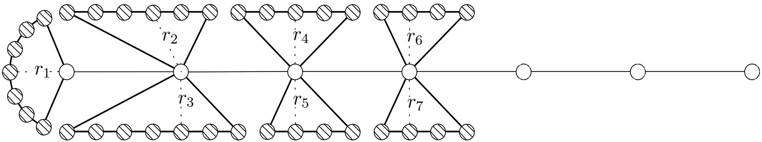
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

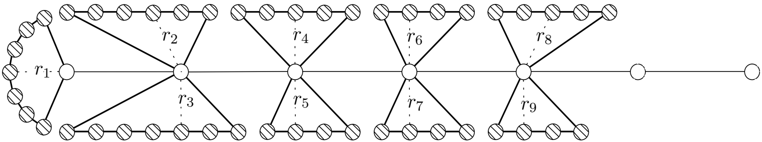
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

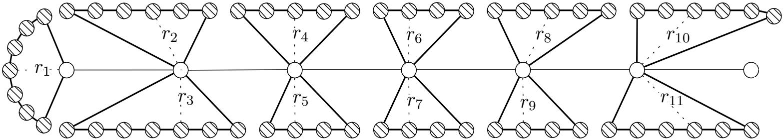
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

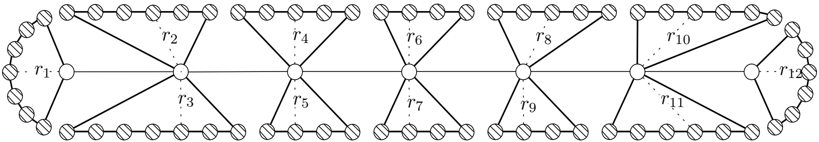
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

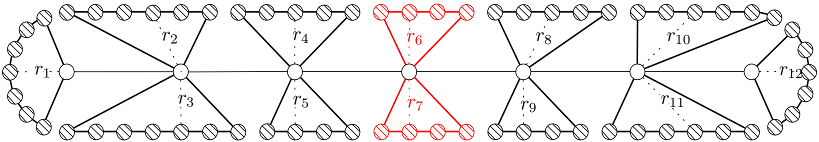
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

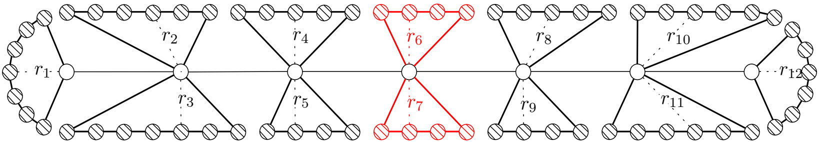
- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'



The probability of 2-coloured paths (I)

Graph first

- reveal a neighborhood around each vertex in a BFS manner
  - constant number of neighbours
  - everything is within fixed radius r'
- "most of the times"
  - the neighbourhood is a tree of height at most r'
  - maximum degree $< (1+\epsilon)d$
  - the neighbourhood does not intersect with others



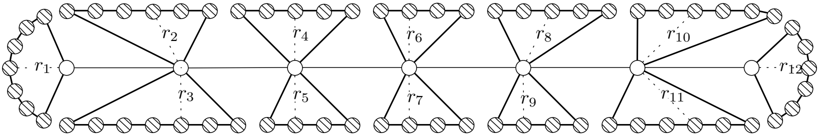
The probability of 2-coloured paths (II)

The probability of 2-coloured paths (II)

The random colouring part

The probability of 2-coloured paths (II)

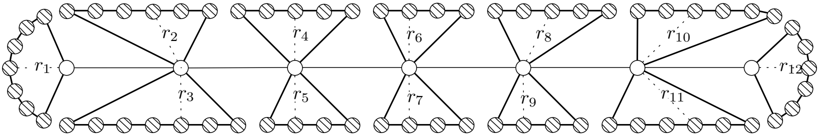
The random colouring part



The probability of 2-coloured paths (II)

The random colouring part

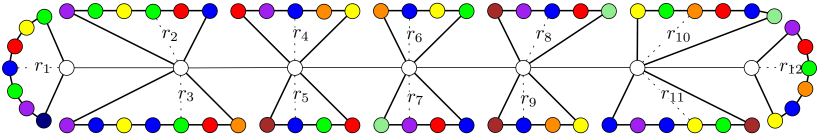
- we consider a worst case boundary condition



The probability of 2-coloured paths (II)

The random colouring part

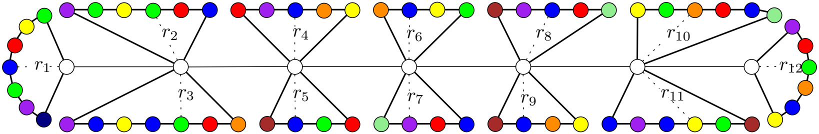
- we consider a worst case boundary condition



The probability of 2-coloured paths (II)

The random colouring part

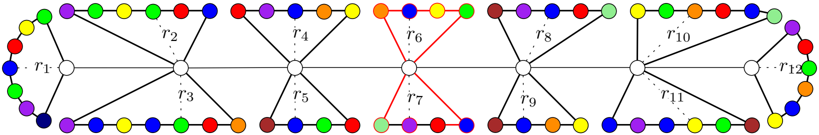
- we consider a worst case boundary condition
- it suffices to have that the colouring of the vertex in the path does not depend too much on the condition



The probability of 2-coloured paths (II)

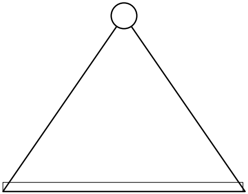
The random colouring part

- we consider a worst case boundary condition
- it suffices to have that the colouring of the vertex in the path does not depend too much on the condition

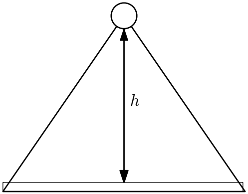


Gibbs Tree Uniqueness

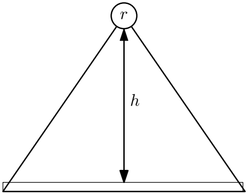
Gibbs Tree Uniqueness



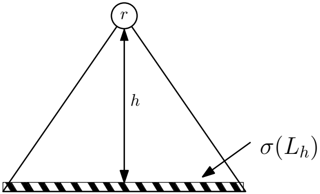
Gibbs Tree Uniqueness



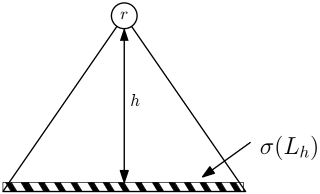
Gibbs Tree Uniqueness



Gibbs Tree Uniqueness

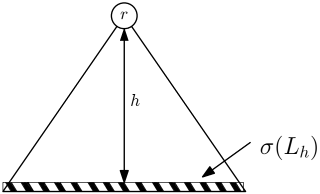


Gibbs Tree Uniqueness



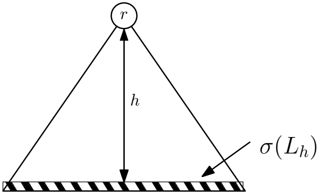
$$||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

Gibbs Tree Uniqueness



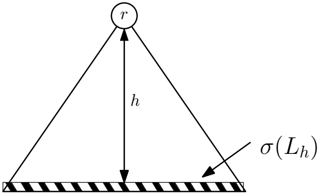
$$\lim_{h\to\infty} ||\mu(\cdot)-\mu(\cdot \mid \sigma(L_h))||_{\{r\}}$$

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



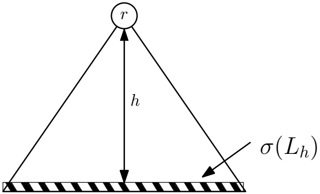
$$\lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}} = \begin{cases} 0 \\ \delta > 0 \end{cases}$$

Uniqueness condition

$\forall \sigma(L_h)$ we have that

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

Gibbs Tree Uniqueness



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

Uniqueness condition

$\forall \sigma(L_h)$ we have that

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

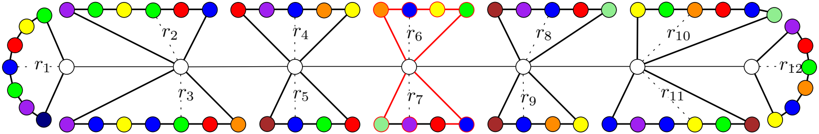
Suppose that the tree is R-ary. How many colour do we need for uniqueness?

Back to the analysis

Back to the analysis

Trees around the path

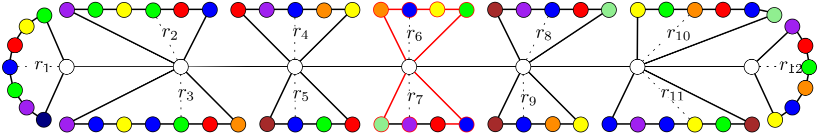
- the neighbourhood is a tree of height at most r
- maximum degree $< (1+\epsilon)d$



Back to the analysis

Trees around the path

- the neighbourhood is a tree of height at most r
- maximum degree $< (1 + \epsilon)d$



Conclusion

We need a number of colours q that guarantees Gibbs uniqueness in the $(1 + \epsilon)d$-ary tree.

Result

Theorem

Let $\epsilon > 0$ be a fixed number, let d be sufficiently large number and fixed $k \geq (1+\epsilon)d$.
Consider $\mathbf{G} = G(n, d/n)$ and let μ the uniform distribution over the k-colouring of $\mathbf{G}$. Let $\hat{\mu}$ be the distribution of the colouring that is returned by our algorithm on input $\mathbf{G}$.
Let $c = \frac{\epsilon}{80(1+\epsilon/4)\log d}$, with probability at least $1 - n^{-c}$ over the input instances $\mathbf{G}$ it holds that

$$||\mu - \hat{\mu}|| = O\left(n^{-c}\right).$$

What about other distributions?

What about other distributions?

Question

Suppose I have Gibbs Uniques for general μ on the d-ary tree, can I use a similar approach for sampling on $G(n, m)$?

What about other distributions?

Question

Suppose I have Gibbs Uniques for general μ on the d-ary tree, can I use a similar approach for sampling on $G(n, m)$?

Remarks

What about other distributions?

Question

Suppose I have Gibbs Uniques for general μ on the d-ary tree, can I use a similar approach for sampling on $G(n, m)$?

Remarks

- `Update` is specific to the distributions we are sampling from

What about other distributions?

Question

Suppose I have Gibbs Uniques for general μ on the d-ary tree, can I use a similar approach for sampling on $G(n, m)$?

Remarks

- `Update` is specific to the distributions we are sampling from
- For many distributions uniqueness is not established
  - there are only conjectures

What about other distributions?

Question

Suppose I have Gibbs Uniques for general μ on the d-ary tree, can I use a similar approach for sampling on $G(n, m)$?

Remarks

- `Update` is specific to the distributions we are sampling from
- For many distributions uniqueness is not established
  - there are only conjectures
- For the hypergraph $\mathbf{H}_k(n, m)$, uniqueness is be too restrictive,
  - go beyond uniqueness

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 previous approach

The previous approach

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: $\boldsymbol{\sigma}_r$

The `Update` problem

The `Update` problem

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

The `Update` problem

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

Setting ...

The `Update` problem

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

The `Update` problem

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 $G' = G \cup \{e\}$

The `Update` problem

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**

The `Update` problem

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.

The `Update` problem

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

The `Update` problem

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

Setting ...

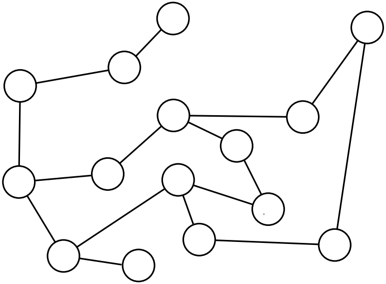
- symmetric Gibbs distribution
  - ... e.g. antiferromagnetic Ising, or Potts
- two graphs G and G' such that $G' = G \cup \{e\}$
  - ... assume that both are of **high girth**
- Gibbs distributions μ and μ' on G and G', resp.
- configuration $\boldsymbol{\sigma}$ distributed as in μ

Objective

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

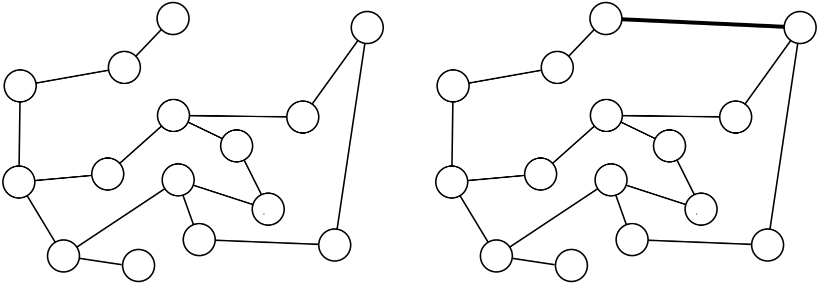
Coupling-Based Solution

Coupling-Based Solution



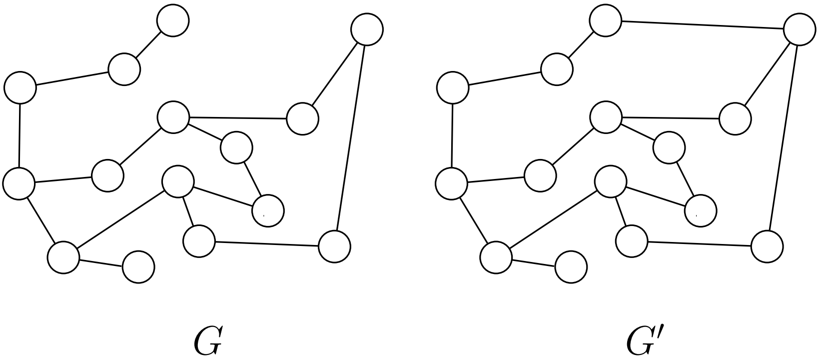
G

Coupling-Based Solution

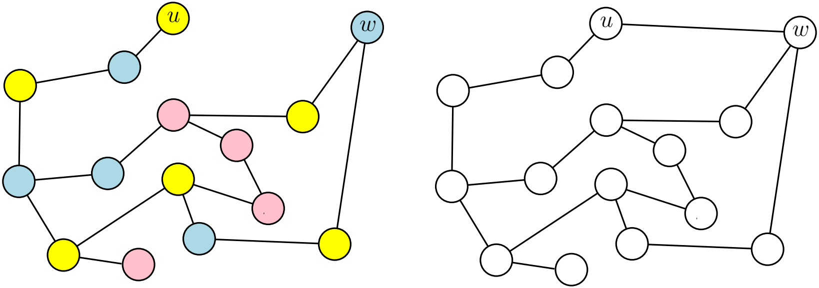


G G'

Coupling-Based Solution

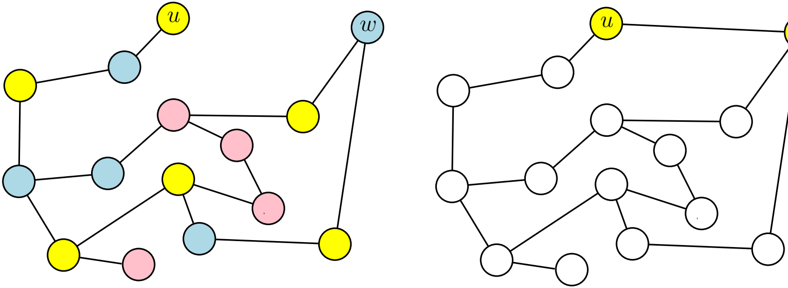


Coupling-Based Solution



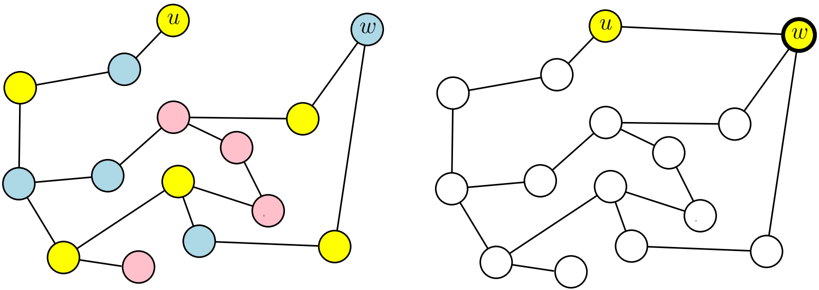
(G, σ) G'

Coupling-Based Solution



(G, σ) $\qquad$ G'

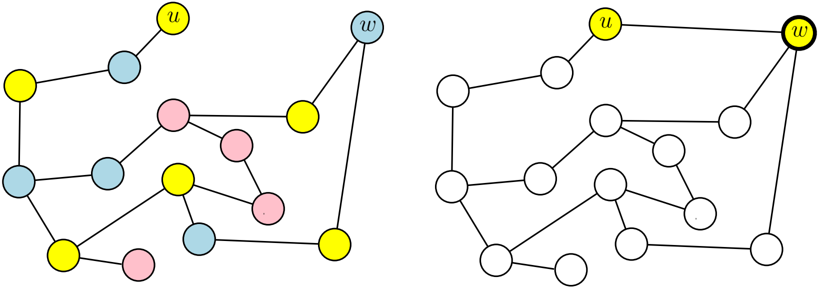
Coupling-Based Solution



(G, σ) $\quad$ G'

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

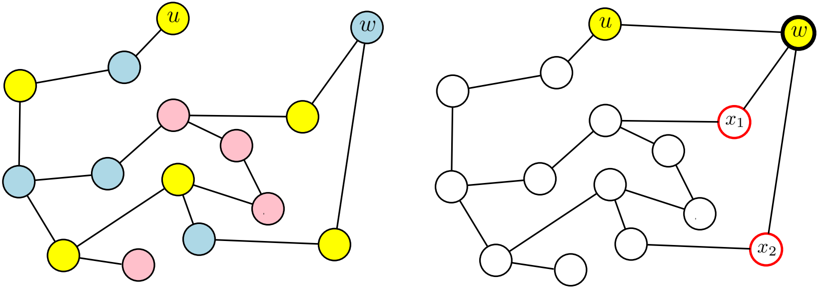
Coupling-Based Solution



(G, σ) $\qquad$ G'

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

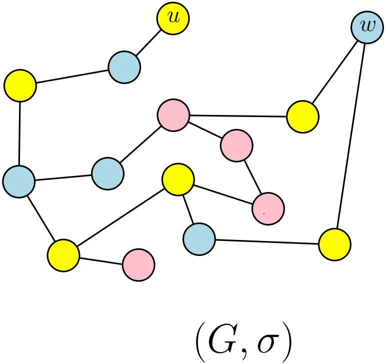
Coupling-Based Solution

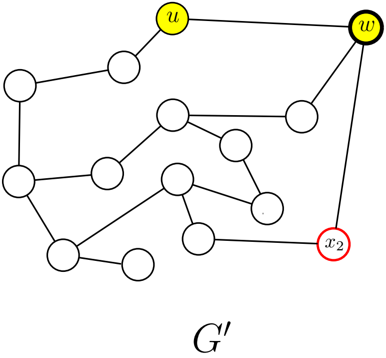


(G, σ) G'

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

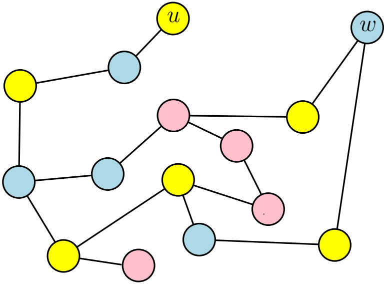
Coupling-Based Solution



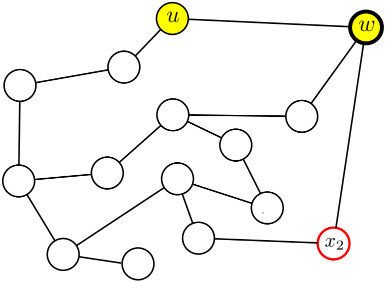


pick x_2 and decide $\tau(x_2)$ such that $\tau(x_2) \in \{\texttt{blue}, \texttt{yellow}\}$

Coupling-Based Solution



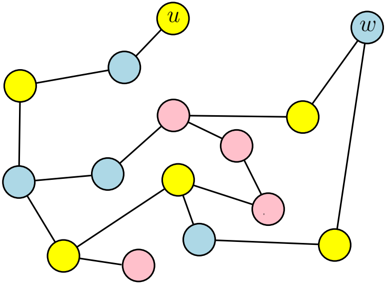
(G, σ)



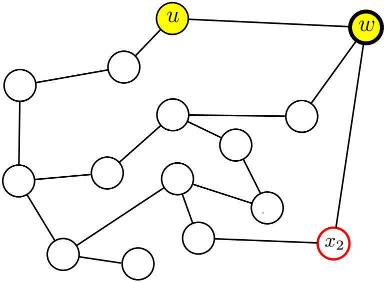
G'

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

Coupling-Based Solution



(G, σ)

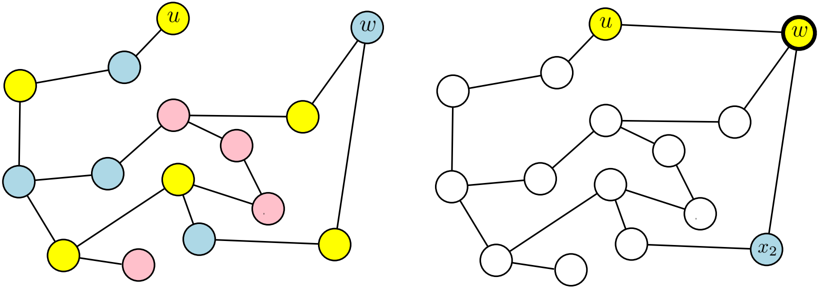


G'

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

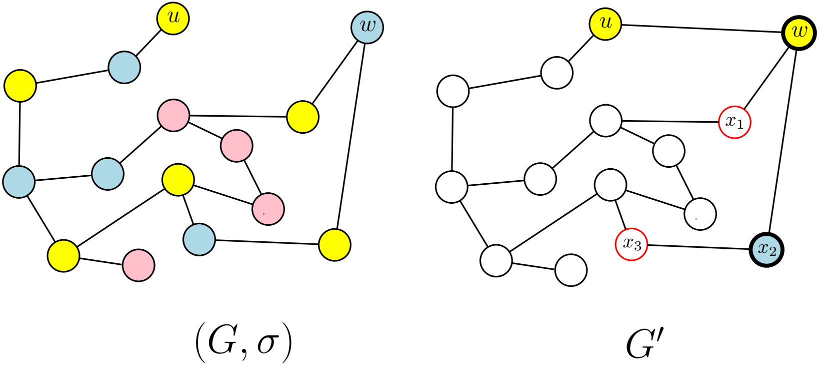


(G, σ) $\quad$ G'

maximal coupling

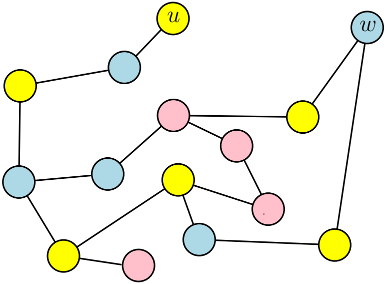
$$\Pr[\tau(x_2) = \texttt{blue}] = \max\left\{0, 1 - \frac{\mu'_{x_2}(\sigma(x_2) \mid \tau(\{u,w\}))}{\mu_{x_2}(\sigma(x_2) \mid \sigma(\{u,w\}))}\right\}.$$

Coupling-Based Solution

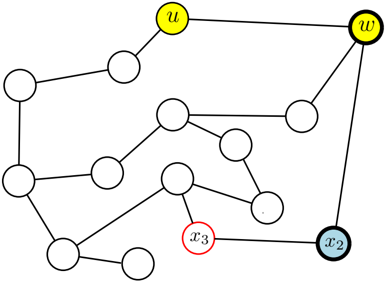


look for vertices z next to the disagreements such that $\sigma(z) \in \{\texttt{blue}, \texttt{yellow}\}$

Coupling-Based Solution



(G, σ)

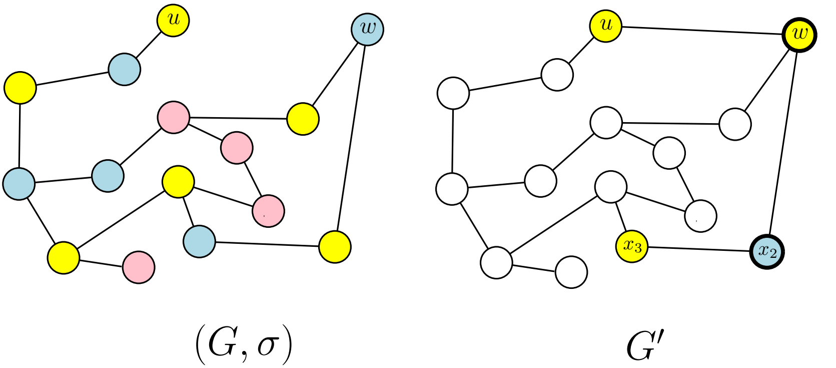


G'

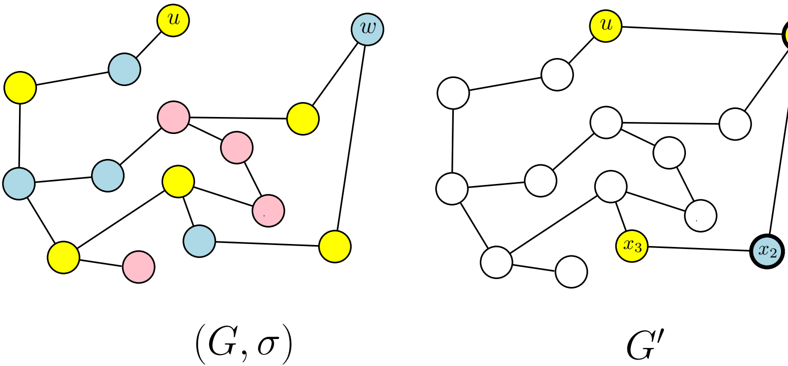
choose x_3 and repeat as before ...

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

Coupling-Based Solution

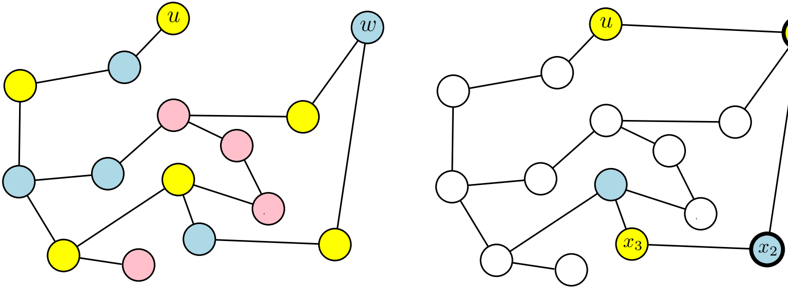


Coupling-Based Solution



repeat in the same way for the rest of the vertices

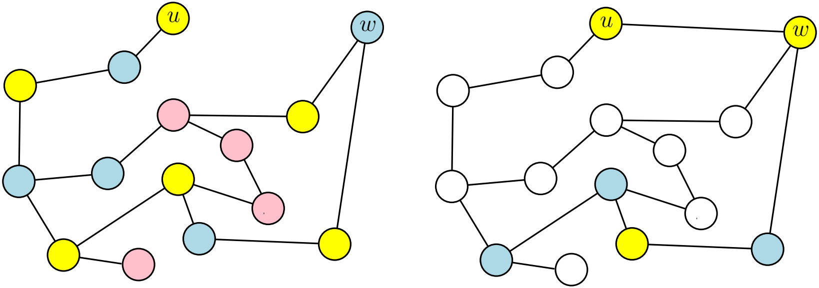
Coupling-Based Solution



(G, σ) $\quad$ G'

repeat in the same way for the rest of the vertices

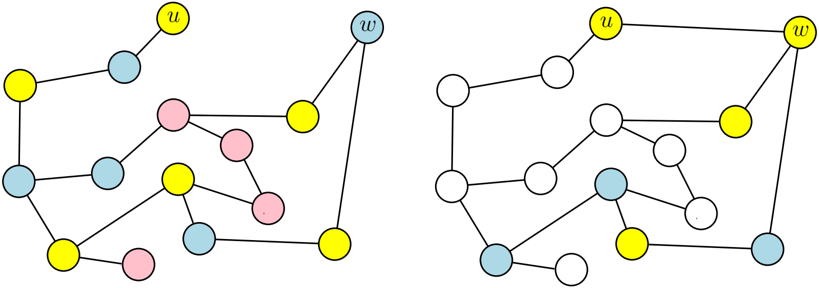
Coupling-Based Solution



(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

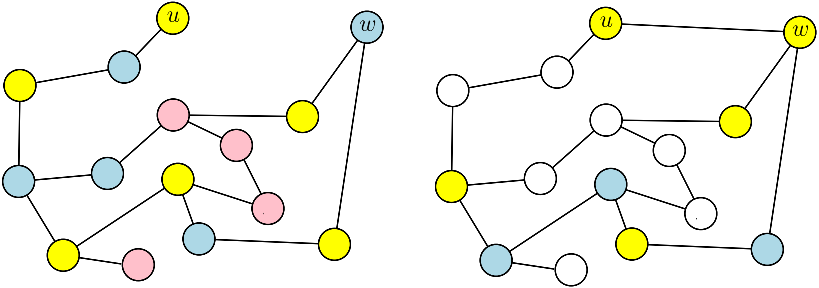
Coupling-Based Solution



(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

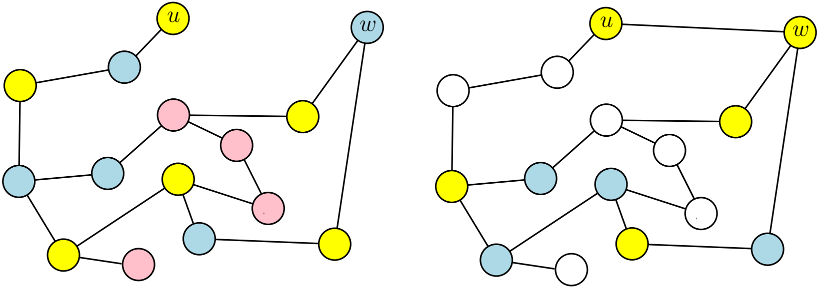
Coupling-Based Solution



(G, σ) $\qquad\qquad$ G'

repeat in the same way for the rest of the vertices

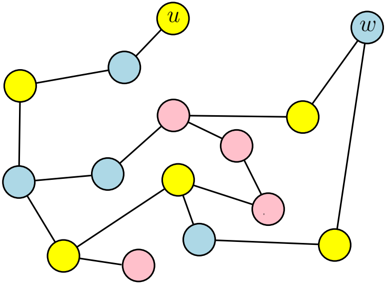
Coupling-Based Solution



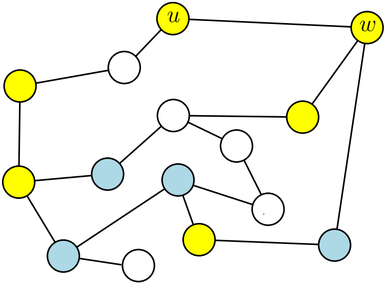
(G, σ) $\qquad$ G'

repeat in the same way for the rest of the vertices

Coupling-Based Solution



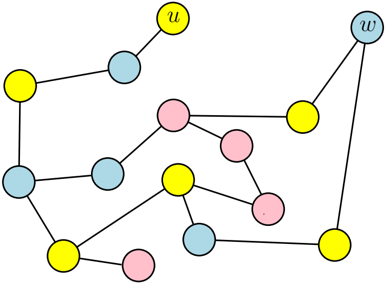
(G, σ)



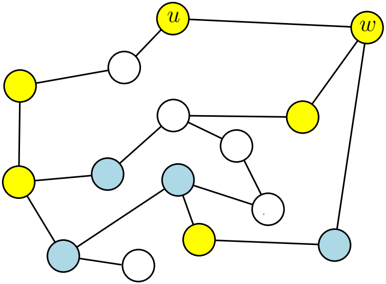
G'

repeat in the same way for the rest of the vertices

Coupling-Based Solution



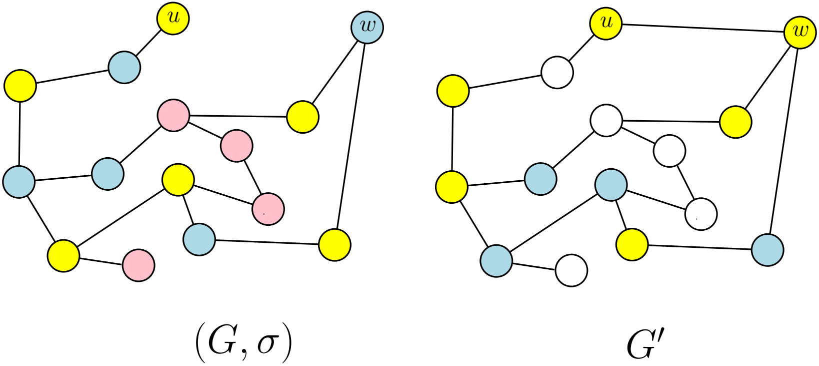
(G, σ)



G'

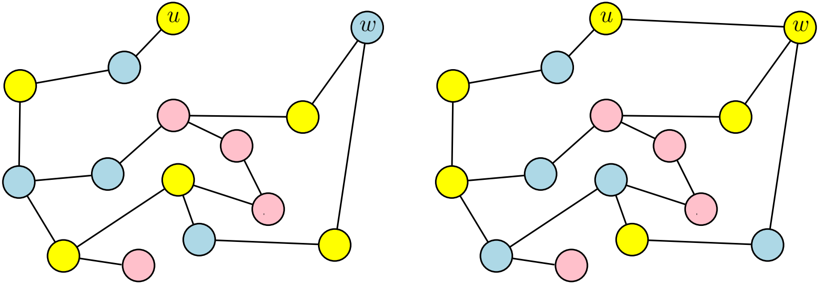
disagreement cannot propagate any more

Coupling-Based Solution



the remaining vertices keep the initial assignments.

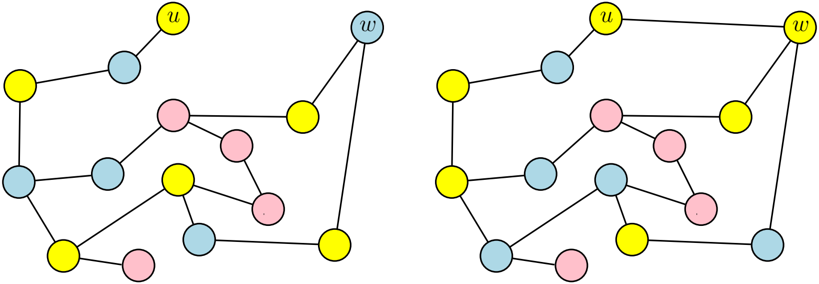
Coupling-Based Solution



(G, σ) (G', τ)

the remaining vertices keep the initial assignments.

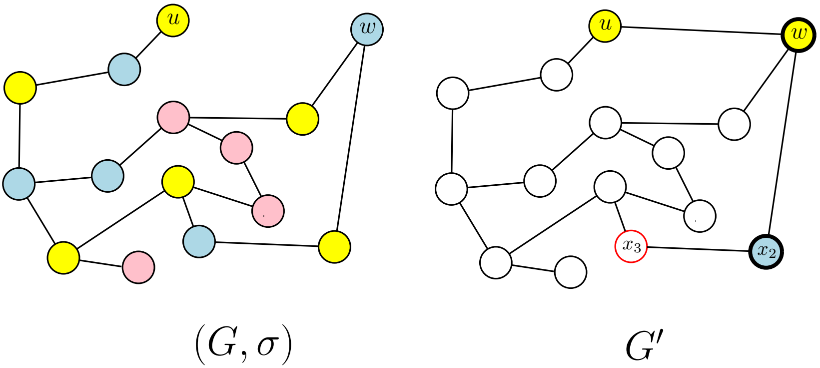
Coupling-Based Solution



(G, σ) (G', τ)

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

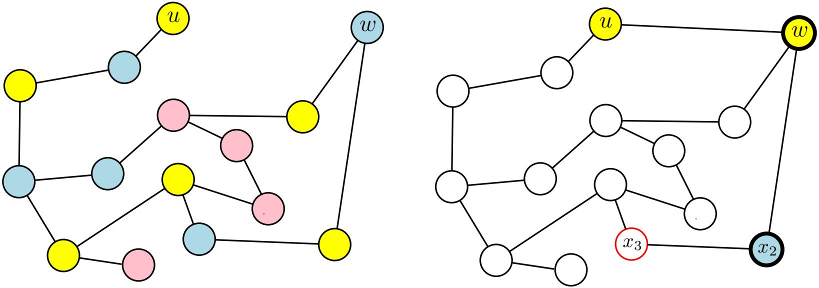
Coupling-Based Solution



The catch . . .

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

Coupling-Based Solution

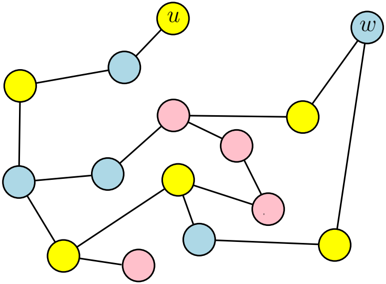


(G, σ) $\qquad$ G'

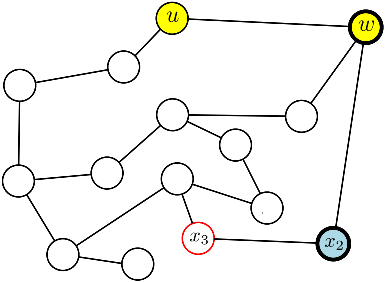
The catch . . .

we don't know how to compute $\mu_{x_3}(\sigma(x_3) \mid \sigma(\{u, w, x_2\}))$ efficiently

Coupling-Based Solution



(G, σ)



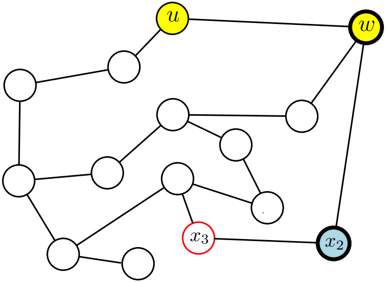
G'

The idea ...

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

Some Intuition . . .

Some Intuition . . .

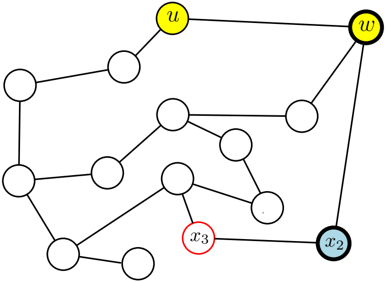


G'

Observation . . .

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

Some Intuition . . .

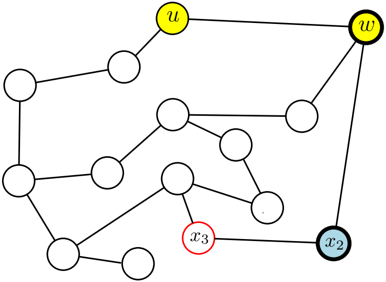


G'

However . . .

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

Some Intuition . . .

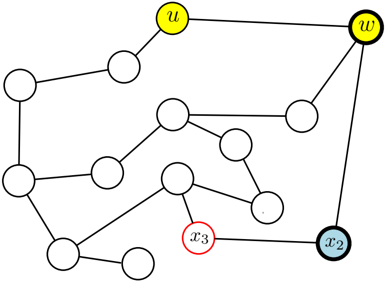


G'

Choosing the appropriate parameters . . .

essentially only one vertex influences the marginal

Some Intuition . . .

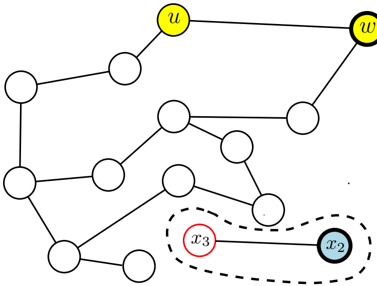


G'

Compute marginal but . . .

ignore the influence on x_3 from u and w

Some Intuition . . .

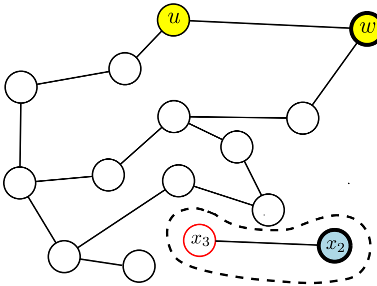


G'

Effectively

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

Some Intuition . . .



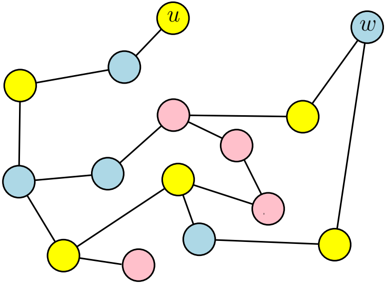
G'

Remark

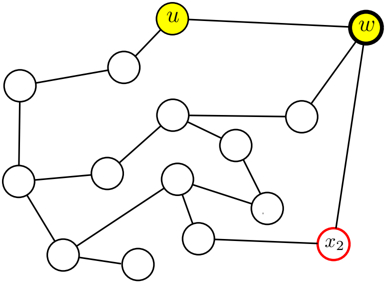
we can compute the "simplified" marginal at x_3 in $O(1)$ steps

To sum up . . .

To sum up . . .



(G, σ)

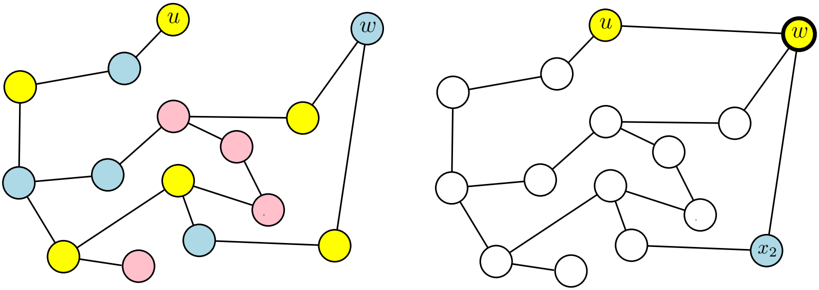


G'

"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 . . .

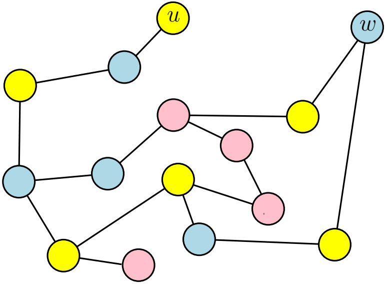


(G, σ) $\qquad$ G'

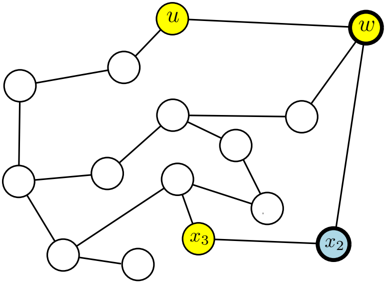
"maximal coupling"

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

To sum up . . .



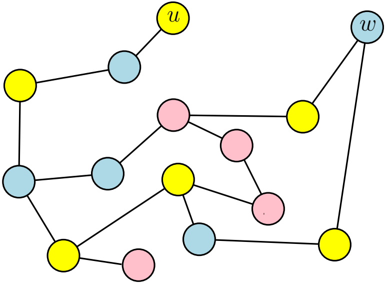
(G, σ)



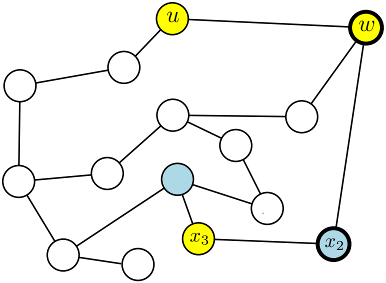
G'

repeat in the same way for the rest of the vertices

To sum up ...



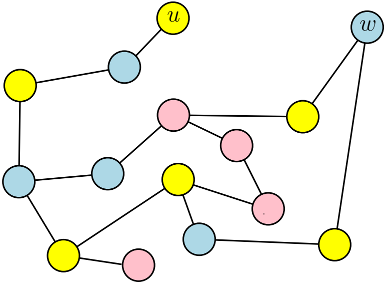
(G, σ)



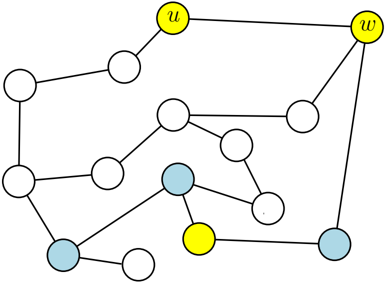
G'

repeat in the same way for the rest of the vertices

To sum up . . .



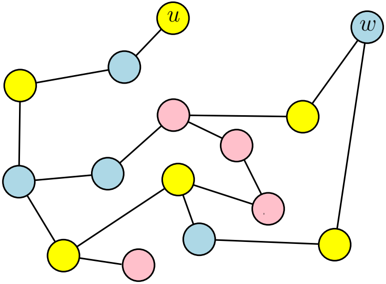
(G, σ)



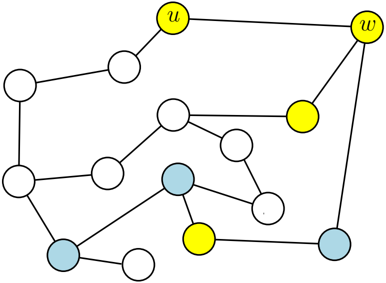
G'

repeat in the same way for the rest of the vertices

To sum up . . .



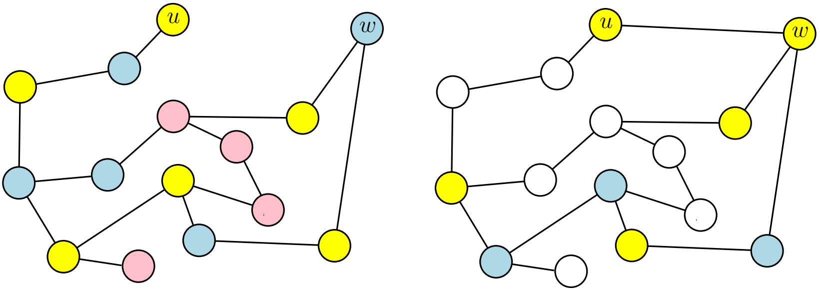
(G, σ)



G'

repeat in the same way for the rest of the vertices

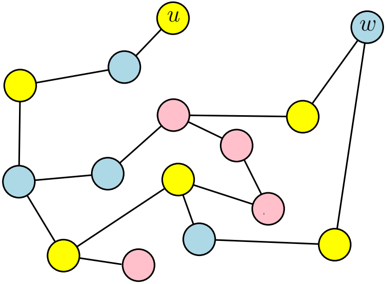
To sum up . . .



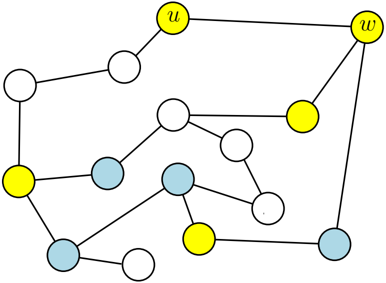
(G, σ) $\qquad\qquad$ G'

repeat in the same way for the rest of the vertices

To sum up . . .



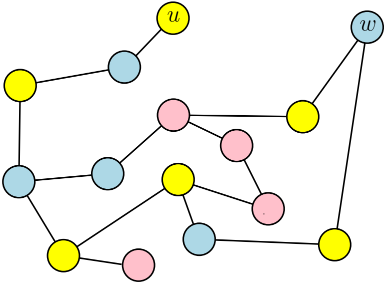
(G, σ)



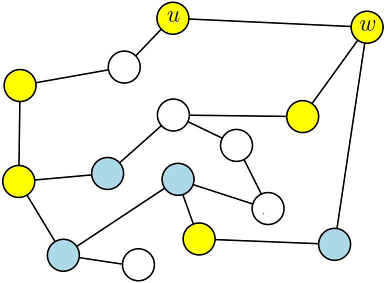
G'

repeat in the same way for the rest of the vertices

To sum up . . .



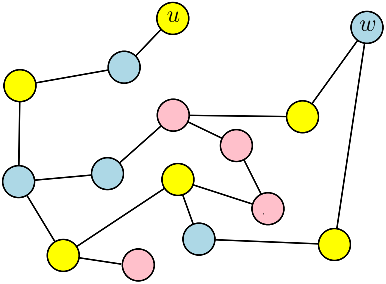
(G, σ)



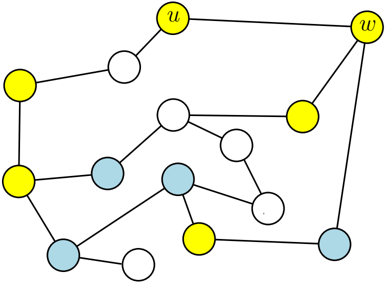
G'

repeat in the same way for the rest of the vertices

To sum up . . .



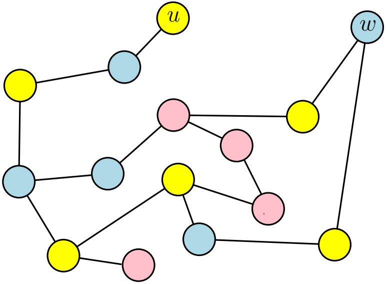
(G, σ)



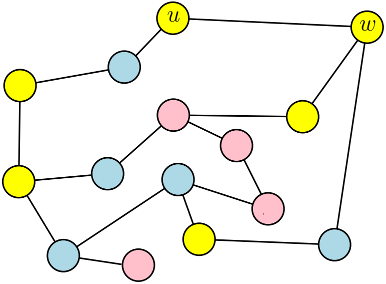
G'

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

To sum up . . .

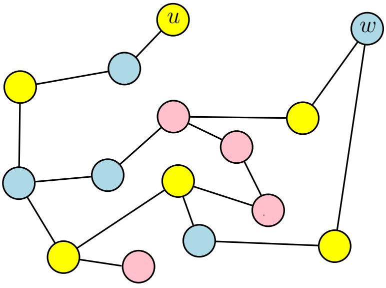


(G, σ)

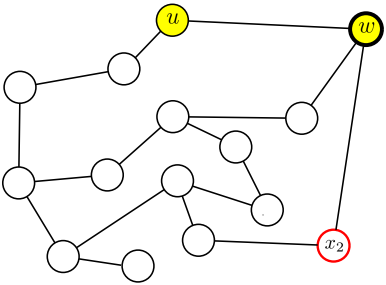


(G', τ)

To sum up . . .



(G, σ)

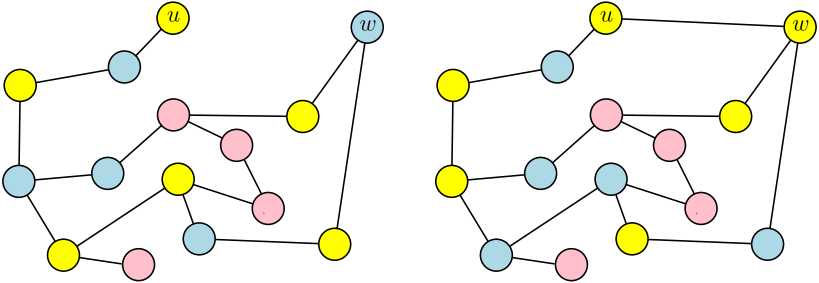


G'

. . . another catch

disagreements should not cover *all* the vertices of a cycle in G'

To sum up . . .



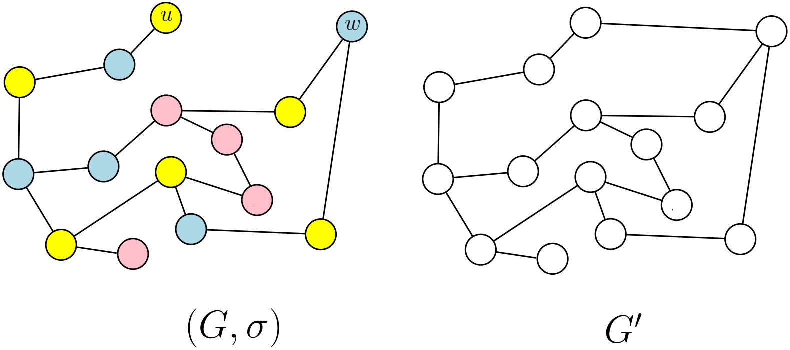
(G, σ) (G', τ)

Otherwise . . .

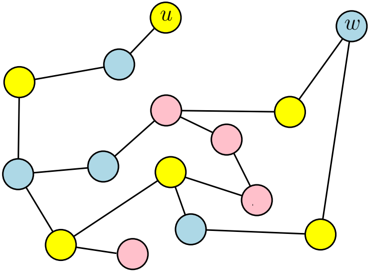
we have a **failure**!

Failure Vs Approximation

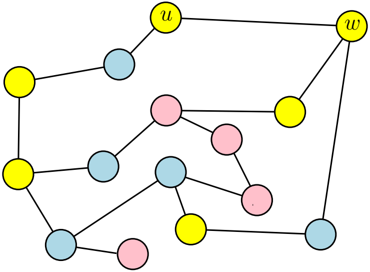
Failure Vs Approximation



Failure Vs Approximation

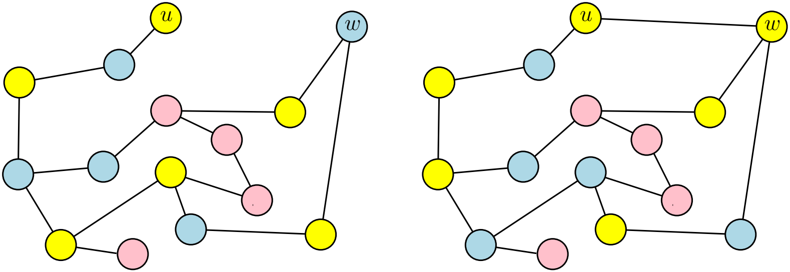


(G, σ)



(G', τ)

Failure Vs Approximation



(G, σ) (G', τ)

ℓ_1-error for `Update`

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

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: $\boldsymbol{\sigma}_r$

Approximate Sampler

The sampling algorithm that uses `Update` is approximation, too

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

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

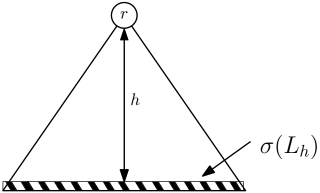
For which parameters of the Gibbs distribution on $G(n, m)$ do we get good approximations?

The parameters

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

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

Won't use Uniqueness!



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

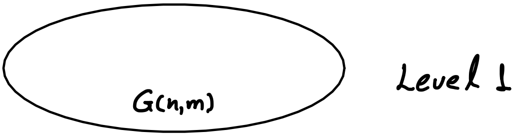
Uniqueness condition

$\forall \sigma(L_h)$ we have that

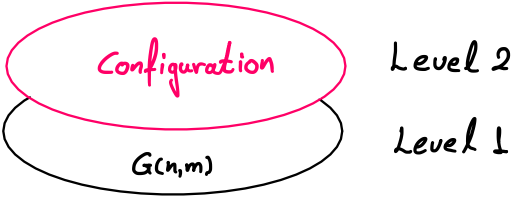
$$\lim_{h\to\infty} ||\mu(\cdot) - \mu(\cdot \mid \sigma(L_h))||_{\{r\}} = 0.$$

Back to the Two Levels

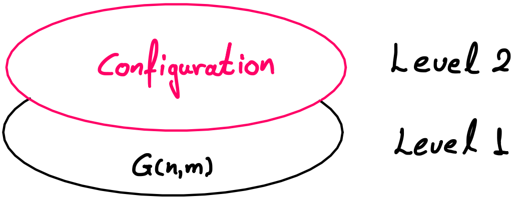
Back to the Two Levels



Back to the Two Levels

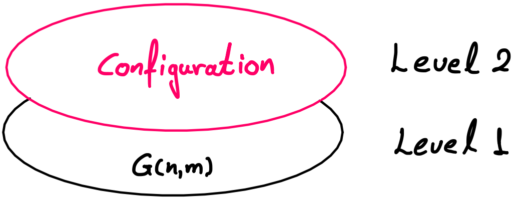


Back to the Two Levels



Objective

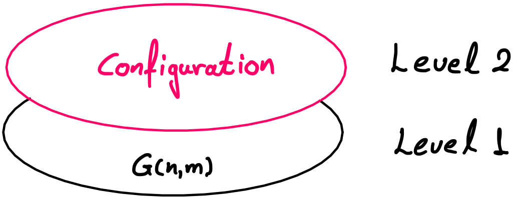
Back to the Two Levels



Objective

- pick at random two distant vertices, e.g., say u and v

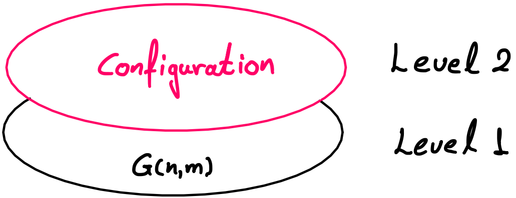
Back to the Two Levels



Objective

- pick at random two distant vertices, e.g., say u and v
- apply the `Update`

Back to the Two Levels

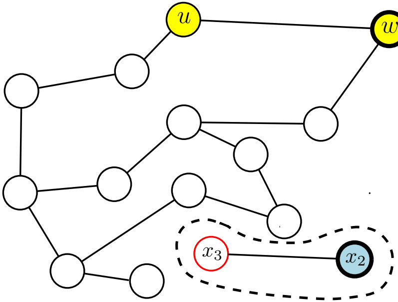


Objective

- pick at random two distant vertices, e.g., say u and v
- apply the `Update`
- with "very large probability" the update is local.

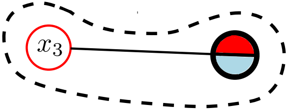
Influence Condition

Influence Condition

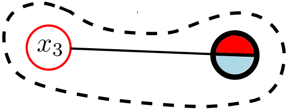


G'

Influence Condition

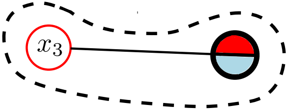


Influence Condition



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

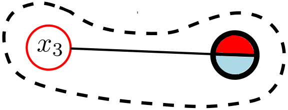
Influence Condition



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

Any further condition?

Influence Condition

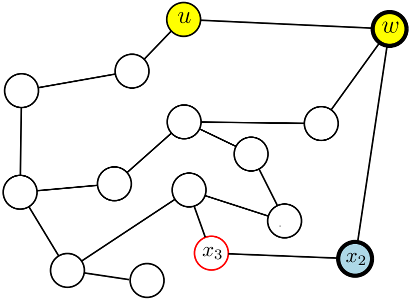


$$\max_{\eta,\theta} ||\mathfrak{m}_{x_3}(\cdot \mid \eta) - \mathfrak{m}_{x_3}(\cdot \mid \theta)|| < 1/d$$

Any further condition? Yes!

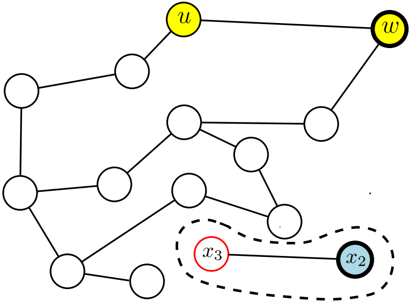
Further condition

Further condition



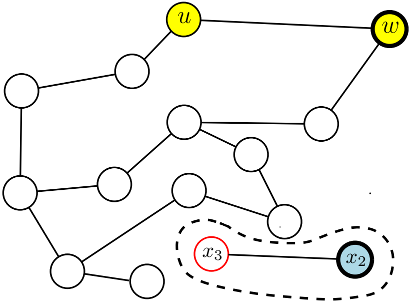
G'

Further condition



G'

Further condition



G'

The new marginal is “good” approximation

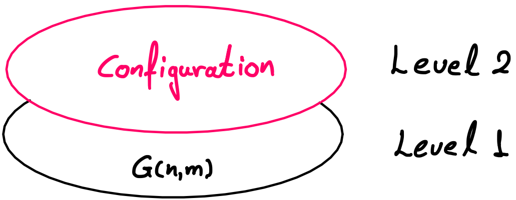
Reversing the Levels of Randomness

Reversing the Levels of Randomness

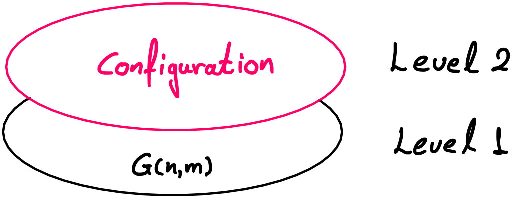
Configuration

Level 2

Reversing the Levels of Randomness



Reversing the Levels of Randomness



Objective

- pick at random two distant vertices, e.g., say u and v
- apply the `Update`
- on influence condition, with "very large probability" the update is local.

Reconsider the order of randomness

Reconsider the order of randomness

Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

Reconsider the order of randomness

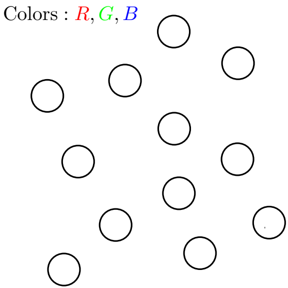
Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

Planting Colourings



Reconsider the order of randomness

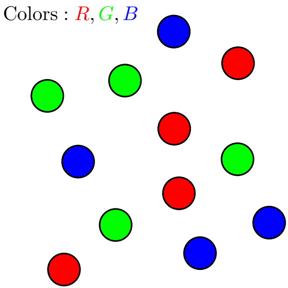
Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

Planting Colourings



Reconsider the order of randomness

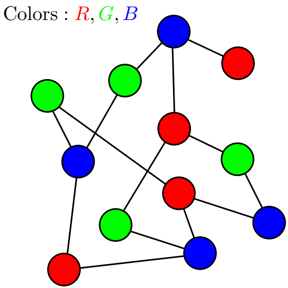
Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

Planting Colourings



Reconstruct the order of randomness

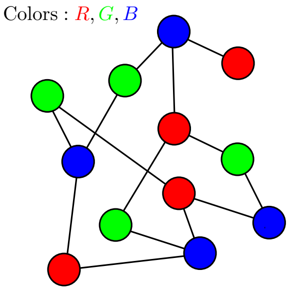
Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

Planting Colourings



Reconsider the order of randomness

Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

`Update` for Teacher-Student

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

Reconsider the order of randomness

Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

`Update` for Teacher-Student

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

- this process is **simpler** to analyse

Reconstruct the order of randomness

Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. apply `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. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

1. generate $\boldsymbol{\sigma}^*$
2. graph G^*
3. apply `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**
- . . . does this imply small failure probability for the "real process"?

Reconsider the order of randomness

Uniform Model

1. generate $G(n, m)$
2. generate $\boldsymbol{\sigma}$
3. apply `Update`

Teacher-Student Model

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

`Update` for Teacher-Student

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

- this process is **simpler** to analyse
- with Influence Cond., the failure probability is **very small**
- ... does this imply 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

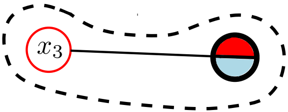
We say that $(\boldsymbol{G}, \boldsymbol{\sigma})$ and $(\boldsymbol{G}^*, \boldsymbol{\sigma}^*)$ are **mutual contiguous** when for any property $\mathcal{A}_n$ we have that

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

Contiguity implies ...

the two distributions have the **same typical properties**.

The conditions for the algorithm



Influence Condition

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

Contiguity

The Gibbs distribution and the corresponding Teacher Student model are contiguous.

Remarks

Remarks

- Results for all aforementioned symmetric Gibbs distributions
  - both on the random graph $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$

Remarks

- Results for all aforementioned symmetric Gibbs distributions
    - both on the random graph $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$
- Influence Cond. is more restrictive than Contiguity
    - Contiguity holds up to "Replica Symmetry Breaking"

Remarks

- Results for all aforementioned symmetric Gibbs distributions
  - both on the random graph $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$
- 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

- Results for all aforementioned symmetric Gibbs distributions
  - both on the random graph $\boldsymbol{G}(n, m)$, or hypergraph $\boldsymbol{H}_k(n, m)$
- 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 the "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 most 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 most 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 most 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 most 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!

Announcement

PhD at CS Warwick!?

Ising Model

Let

$$\beta_{\text{Ising}}(\Delta, k) = \log\left(\frac{\Delta(k-1)+1-2^{k-1}}{\Delta(k-1)+1}\right) \qquad \Delta > \frac{2^{k-1}-1}{k-1}.$$

Theorem

For integer $k \geq 2$ the following holds: For $d > 0$ either

1. *$d > (2^{k-1}-1)/(k-1)$ and $\beta_{\text{Ising}}(d,k) < \beta < 0$, or*
2. *$d \leq (2^{k-1}-1)/(k-1)$ and $\beta < 0$,*

for $\boldsymbol{H} = \boldsymbol{H}(n,m,k)$, where $m = dn/k$, let $\mu = \mu_{\boldsymbol{H}}$ be the antiferromagnetic Ising model on $\boldsymbol{H}$, with inverse temperature β and external field $h = 0$.

There is $c_0 = c_0(k,d,\beta) > 0$, s.t. with probability $1 - o(1)$ over the instances $\boldsymbol{H}$, our algorithm generates a configuration with distribution $\bar{\mu}$ such that

$$||\bar{\mu} - \mu||_{tv} \leq n^{-\frac{c_0}{\log(dk)}}.$$

Potts Model

Let

$$\beta_{\mathrm{Potts}}(\Delta, q, k) = \log\left(\frac{\Delta(k-1)+1-q^{k-1}}{\Delta(k-1)+1}\right) \qquad \Delta > \frac{q^{k-1}+1}{k-1}.$$

Theorem

For $k \geq 2$, let β, q, d satisfy one of

1. $(q^{k-1} - 1)/(k-1) < d$ *and* $\beta_{\mathrm{Potts}}(d, q, k) < \beta < 0$,
2. $(q^{k-1} - 1)/(k-1) > d$ *and* $\beta < 0$, *including* $\beta = -\infty$.

For $\boldsymbol{H} = \boldsymbol{H}(n, m, k)$, where $m = dn/k$, let $\mu = \mu_{\boldsymbol{H}}$ be the q-state antiferromagnetic Potts model on $\boldsymbol{H}$ with inverse temperature β. There exists $c_0 = c_0(k, d, \beta) > 0$ s.t. with probability $1 - o(1)$ over the instances $\boldsymbol{H}$, our algorithm generates a configuration whose distribution $\bar{\mu}$ is such that

$$||\bar{\mu} - \mu||_{tv} \leq n^{-\frac{c_0}{55\log(dk)}}.$$

NAE-k-SAT

Theorem

For $\delta \in (0,1]$, for $k \geq 2$, for any $1/(k-1) \leq d < (1-\delta)\frac{2^{k-1}-1}{k-1}$ and for integer $m = dn/k$, the following is true for our algorithm: Consider $\boldsymbol{F}_k(n,m)$ and let μ be the uniform distribution over the NAE satisfying assignments of $\boldsymbol{F}_k(n,m)$.
With probability $1-o(1)$ over the input instances $\boldsymbol{F}_k(n,m)$, our algorithm generates a configuration whose distribution $\bar{\mu}$ is such that

$$||\bar{\mu} - \mu||_{tv} \leq n^{-\frac{\delta}{55\log(dk)}}.$$

k-spin system

Let

$$F_k(x) = \frac{|e^x - e^{-x}|}{(2^{k-1}-1)e^{-x}+e^x}.$$

Theorem

For $\delta \in (0,1]$, for even integer $k \geq 2$ and $\beta \geq 0$ s.t.

$$\mathbf{E}[F_k(\beta \mathbf{J}_0)] \leq \frac{1-\delta}{d(k-1)}, \qquad \mathbf{J}_0 \sim \mathcal{N}(0,1),$$

consider $\boldsymbol{H} = \boldsymbol{H}(n,m,k)$, where $m = dn/k$, and let μ be the k-spin model on $\boldsymbol{H}$ at inverse temperature β.
With probability $1-o(1)$ over the instances $\boldsymbol{H}$ and the couplings on the edges of $\boldsymbol{H}$, our algorithm generates a configuration whose distribution $\bar{\mu}$ is such that

$$||\bar{\mu} - \mu||_{tv} \leq n^{-\frac{\delta}{55\log(dk)}}.$$