Towards derandomising Markov chain Monte Carlo

Heng Guo (University of Edinburgh)

Based on joint works with Weiming Feng, Jiaheng Wang (Edinburgh), Chunyang Wang, Yitong Yin (Nanjing)

Warwick Theory Day, Dec 12, 2022

Deterministic vs randomised counting

Estimating the volume of a convex body:

- No polynomial-time *deterministic* approximation algorithm using membership queries only; (Elekes 1986, Bárány and Füredi 1987)
- Efficient *randomised* approximation algorithm does exist! (Dyer, Frieze, and Kannan 1991)

However, Weitz (2006) gave an FPTAS for the hardcore model up to the tree uniqueness threshold, whose randomised counterparts are not known until very recently (Anari, Liu, and Oveis Gharan, 2020).

Since then, deterministic counting algorithms are catching up in many fronts.

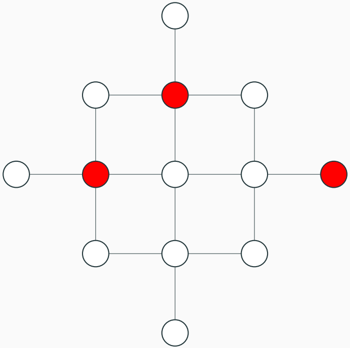
The hardcore model

The Gibbs distribution for the hardcore model:

for an independent set I, $\mu(I) = \frac{\lambda^{|I|}}{Z}$, where $Z = \sum_{I \in \mathcal{I}} \lambda^{|I|}$

We often want to approximate Z, or equivalently, sample from μ.

Standard Glauber dynamics converges to μ.



Glauber dynamics

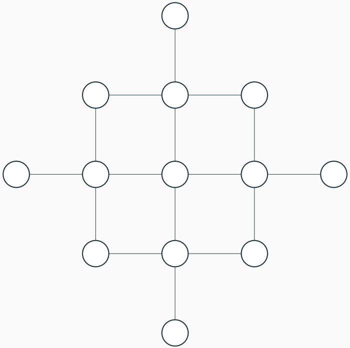
Systematic scan Glauber dyanmics:

Pick the next vertex v, resample its state conditioned on its neighbours

For the resampling step, draw uniform $r \sim [0, 1]$:

- if one of its neighbour is occupied,
  make v unoccupied regardless of r;
- if none of its neighbour is occupied,
  make v unoccupied if $r \leqslant \frac{1}{1+\lambda}$; occupied otherwise.

In either case, v is unoccupied if $r \leqslant \frac{1}{1+\lambda}$.



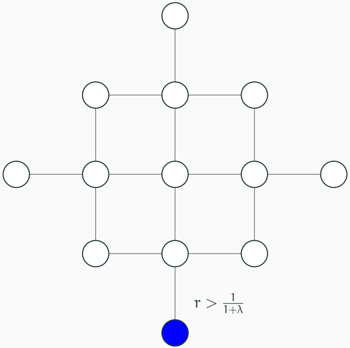
Systematic scan Glauber dyanmics:

Pick the next vertex v, resample its state conditioned on its neighbours

For the resampling step, draw uniform $r \sim [0, 1]$:

- if one of its neighbour is occupied, make v unoccupied regardless of r;
- if none of its neighbour is occupied, make v unoccupied if $r \leqslant \frac{1}{1+\lambda}$; occupied otherwise.

In either case, v is unoccupied if $r \leqslant \frac{1}{1+\lambda}$.



Glauber dynamics

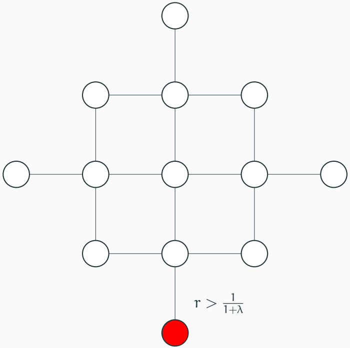
Systematic scan Glauber dyanmics:

Pick the next vertex v, resample its state conditioned on its neighbours

For the resampling step, draw uniform $r \sim [0, 1]$:

- if one of its neighbour is occupied,
  make v unoccupied regardless of r;
- if none of its neighbour is occupied,
  make v unoccupied if $r \leqslant \frac{1}{1+\lambda}$; occupied otherwise.

In either case, v is unoccupied if $r \leqslant \frac{1}{1+\lambda}$.



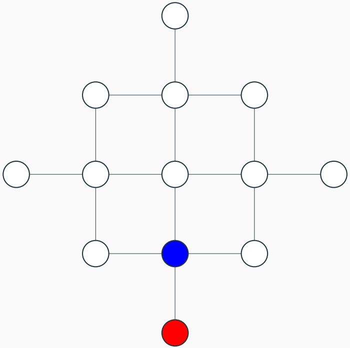
Systematic scan Glauber dyanmics:

Pick the next vertex v, resample its state conditioned on its neighbours

For the resampling step, draw uniform $r \sim [0, 1]$:

- if one of its neighbour is occupied, make v unoccupied regardless of r;
- if none of its neighbour is occupied, make v unoccupied if $r \leqslant \frac{1}{1+\lambda}$; occupied otherwise.

In either case, v is unoccupied if $r \leqslant \frac{1}{1+\lambda}$.



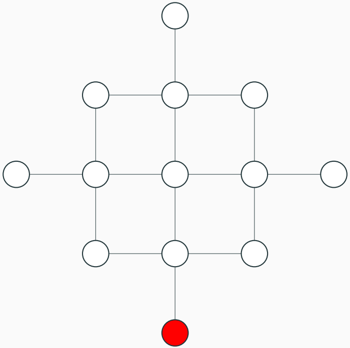
Systematic scan Glauber dyanmics:

Pick the next vertex v, resample its state conditioned on its neighbours

For the resampling step, draw uniform $r \sim [0, 1]$:

- if one of its neighbour is occupied, make v unoccupied regardless of r;
- if none of its neighbour is occupied, make v unoccupied if $r \leqslant \frac{1}{1+\lambda}$; occupied otherwise.

In either case, v is unoccupied if $r \leqslant \frac{1}{1+\lambda}$.



Standard self-reduction (Jerrum, Valiant, and Vazirani, 1986)

$$\frac{1}{Z} = \frac{Z(\sigma_{v_1} = 0)}{Z} \cdot \frac{Z(\sigma_{v_1} = 0, \sigma_{v_2} = 0)}{Z(\sigma_{v_1} = 0)} \cdot \dots \cdot \frac{Z(\wedge_{i=1}^{n} \sigma_{v_i} = 0)}{Z(\wedge_{i=1}^{n-1} \sigma_{v_i} = 0)}$$

Each term $\frac{Z(\wedge_{i=1}^{j} \sigma_{v_i}=0)}{Z(\wedge_{i=1}^{j-1} \sigma_{v_i}=0)}$ is the marginal probability of v_j where $\forall i < j$, v_i is pinned to 0.
Equivalently, we can remove v_i for all $i < j$ from G and consider the marginal of v_j.

It suffices to approximate these marginals within $\frac{\varepsilon}{n}$ to get an ε-approximation to Z.

Counting to sampling reduction

Standard self-reduction (Jerrum, Valiant, and Vazirani, 1986)

$$\frac{1}{Z} = \frac{Z(\sigma_{v_1} = 0)}{Z} \cdot \frac{Z(\sigma_{v_1} = 0, \sigma_{v_2} = 0)}{Z(\sigma_{v_1} = 0)} \cdot \dots \cdot \frac{Z(\wedge_{i=1}^{n} \sigma_{v_i} = 0)}{Z(\wedge_{i=1}^{n-1} \sigma_{v_i} = 0)}$$

Each term $\frac{Z(\wedge_{i=1}^{j} \sigma_{v_i}=0)}{Z(\wedge_{i=1}^{j-1} \sigma_{v_i}=0)}$ is the marginal probability of v_j where $\forall i < j$, v_i is pinned to 0.

Equivalently, we can remove v_i for all $i < j$ from G and consider the marginal of v_j.

It suffices to approximate these marginals within $\frac{\varepsilon}{n}$ to get an ε-approximation to Z.

While the whole Glauber dynamics requires a lot of time / randomness to simulate, can we draw from for the marginal distribution more efficiently?

Standard self-reduction (Jerrum, Valiant, and Vazirani, 1986)

$$\frac{1}{Z} = \frac{Z(\sigma_{v_1} = 0)}{Z} \cdot \frac{Z(\sigma_{v_1} = 0, \sigma_{v_2} = 0)}{Z(\sigma_{v_1} = 0)} \cdot \cdots \cdot \frac{Z(\wedge_{i=1}^{n} \sigma_{v_i} = 0)}{Z(\wedge_{i=1}^{n-1} \sigma_{v_i} = 0)}$$

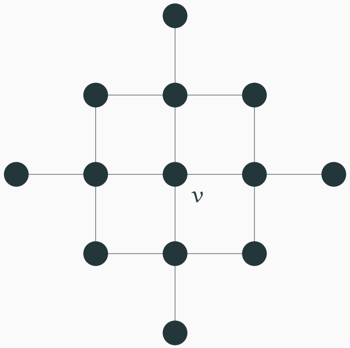
Each term $\frac{Z(\wedge_{i=1}^{j} \sigma_{v_i}=0)}{Z(\wedge_{i=1}^{j-1} \sigma_{v_i}=0)}$ is the marginal probability of v_j where $\forall i < j, v_i$ is pinned to 0.
Equivalently, we can remove v_i for all $i < j$ from G and consider the marginal of v_j.
It suffices to approximate these marginals within $\frac{\varepsilon}{n}$ to get an ε-approximation to Z.

While the whole Glauber dynamics requires a lot of time / randomness to simulate, can we draw from for the marginal distribution more efficiently?

For example, instead of $O(n \log n)$, can we use $O(\log n)$ time / random variables for each vertex?

An alternative sampling algorithm

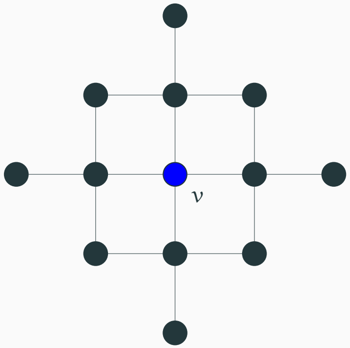
Target distribution: $\mu(I) \propto \lambda^{|I|}$



An alternative sampling algorithm

Target distribution: $\mu(I) \propto \lambda^{|I|}$

Imagine that we have a stationary Glauber dynamics.
To know the value of v, we just need its last update.

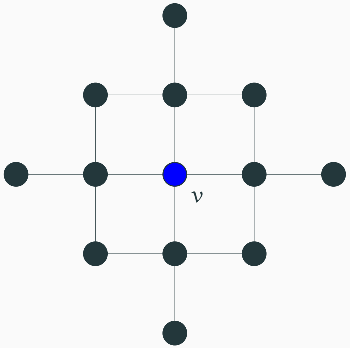


Target distribution: $\mu(I) \propto \lambda^{|I|}$

Imagine that we have a stationary Glauber dynamics.
To know the value of v, we just need its last update.

Draw a uniform $r_0 \sim [0, 1]$ first, then

- if $r_0 \leqslant \frac{1}{1+\lambda}$, v is unoccupied;
- otherwise, $r_0 > \frac{1}{1+\lambda}$, we need its neighbours' states.



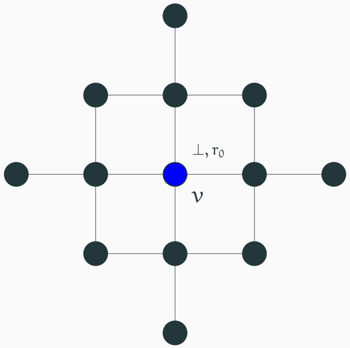
An alternative sampling algorithm

Target distribution: $\mu(I) \propto \lambda^{|I|}$

Imagine that we have a stationary Glauber dynamics.
To know the value of v, we just need its last update.

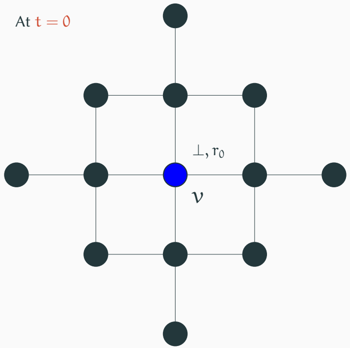
Draw a uniform $r_0 \sim [0, 1]$ first, then

- if $r_0 \leqslant \frac{1}{1+\lambda}$, v is unoccupied;
- otherwise, $r_0 > \frac{1}{1+\lambda}$, we need its neighbours' states.



An alternative sampling algorithm

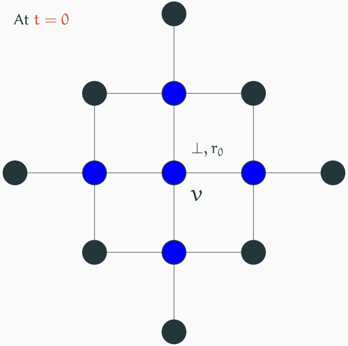
To know the neighbours, again, we perform their last update.



An alternative sampling algorithm

To know the neighbours, again, we perform their last update.

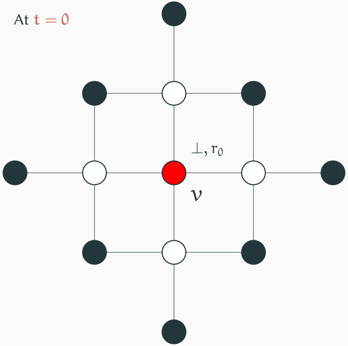
Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.



To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

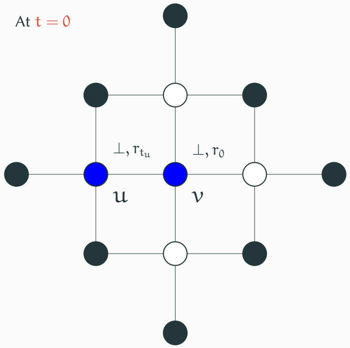
- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.



To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.

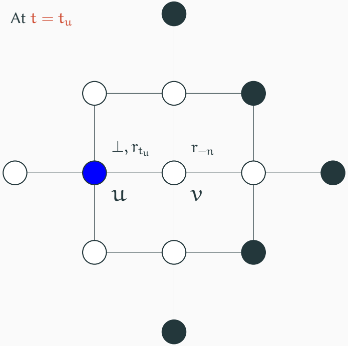


To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.

Suppose all neighbours of u at t_u are all unoccupied.
Then as $r_{t_u} > \frac{1}{1+\lambda}$, u was occupied.



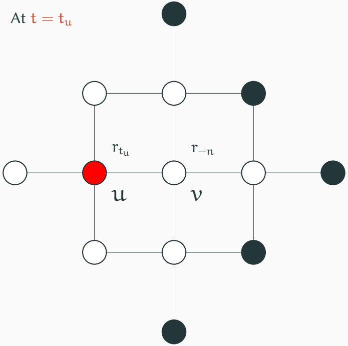
An alternative sampling algorithm

To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.

Suppose all neighbours of u at t_u are all unoccupied.
Then as $r_{t_u} > \frac{1}{1+\lambda}$, u was occupied.



An alternative sampling algorithm

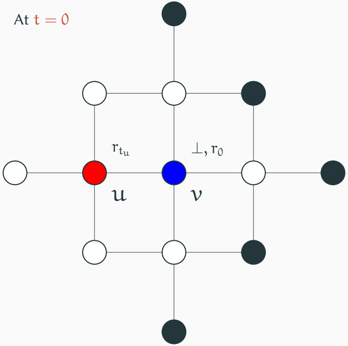
To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.

Suppose all neighbours of u at t_u are all unoccupied.
Then as $r_{t_u} > \frac{1}{1+\lambda}$, u was occupied.

Back to time 0, we deduce that v has to be unoccupied.



An alternative sampling algorithm

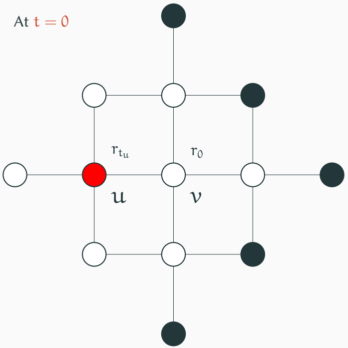
To know the neighbours, again, we perform their last update.

Draw uniform $r_{t_u} \sim [0, 1]$ where t_u is the time of last updating u, and decide accordingly.

- If all neighbours are not occupied, v must be occupied as $r_0 > \frac{1}{1+\lambda}$.
- Otherwise, say some neighbour u is $\perp$, we recursively resolve u.

Suppose all neighbours of u at t_u are all unoccupied.
Then as $r_{t_u} > \frac{1}{1+\lambda}$, u was occupied.

Back to time 0, we deduce that v has to be unoccupied.



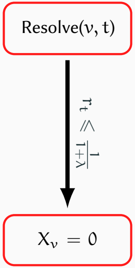
Coupling towards the past

When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before

Resolve(v, t)

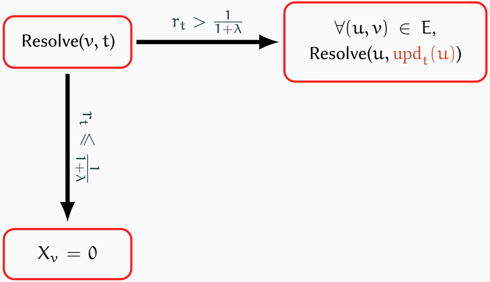
Coupling towards the past

When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before



Coupling towards the past

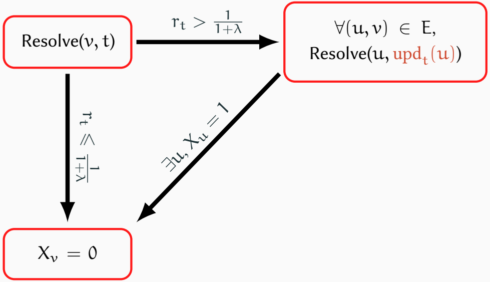
When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before



$\mathrm{upd}_t(u)$ is the last update time of u before t

Coupling towards the past

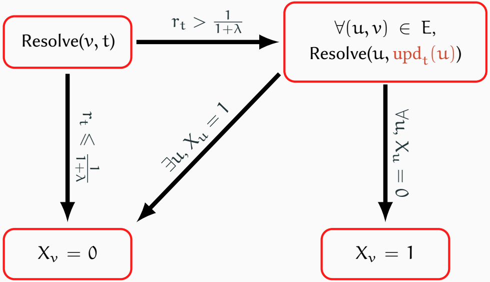
When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before



$\mathrm{upd}_t(u)$ is the last update time of u before t

Coupling towards the past

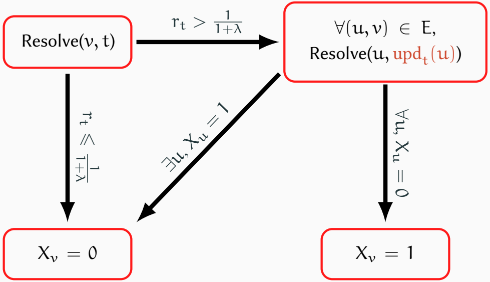
When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before



$\text{upd}_t(u)$ is the last update time of u before t

Coupling towards the past

When resolving (v, t), first check if $X_t(v)$ is known, or if r_t has been drawn before



$\text{upd}_t(u)$ is the last update time of u before t

This can be viewed as either

- a coupling with the stationary process, or
- a grand coupling (using the same r_t) for all possible starting X_0.

This grand coupling is very similar to Coupling From The Past by Wilson and Propp (1996).

Truncation

Running CTTP till it terminates yields a perfect sample.

Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\text{ Truncation }]$$

In a typical application (such as $\lambda < \frac{1}{\Delta - 1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.

v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
r_0	r_{-1}	r_{-2}	r_{-3}	r_{-4}	r_{-5}	r_{-6}	r_{-7}
r_{-7}	r_{-8}	r_{-9}	r_{-10}	r_{-11}	r_{-12}	r_{-13}	r_{-14}
r_{-14}	r_{-15}	r_{-16}	r_{-17}	r_{-18}	r_{-19}	r_{-20}	r_{-21}
r_{-21}	r_{-22}	r_{-23}	r_{-24}	r_{-25}	r_{-26}	r_{-27}	r_{-28}
r_{-28}	r_{-29}	r_{-30}	r_{-31}	r_{-32}	r_{-33}	r_{-34}	r_{-35}
r_{-35}	r_{-36}	r_{-37}	r_{-38}	r_{-39}	r_{-40}	r_{-41}	r_{-42}
r_{-42}	r_{-43}	r_{-44}	r_{-45}	r_{-46}	r_{-47}	r_{-48}	r_{-49}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Truncation

Running CTTP till it terminates yields a perfect sample.

Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\,\text{Truncation}\,]$$

In a typical application (such as $\lambda < \frac{1}{\Delta-1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.

v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
r_0	r_{-1}	r_{-2}	r_{-3}	r_{-4}	r_{-5}	r_{-6}	r_{-7}
r_{-7}	r_{-8}	r_{-9}	r_{-10}	r_{-11}	r_{-12}	r_{-13}	r_{-14}
r_{-14}	r_{-15}	r_{-16}	r_{-17}	r_{-18}	r_{-19}	r_{-20}	r_{-21}
r_{-21}	r_{-22}	r_{-23}	r_{-24}	r_{-25}	r_{-26}	r_{-27}	r_{-28}
r_{-28}	r_{-29}	r_{-30}	r_{-31}	r_{-32}	r_{-33}	r_{-34}	r_{-35}
r_{-35}	r_{-36}	r_{-37}	r_{-38}	r_{-39}	r_{-40}	r_{-41}	r_{-42}
r_{-42}	r_{-43}	r_{-44}	r_{-45}	r_{-46}	r_{-47}	r_{-48}	r_{-49}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Truncation

Running CTTP till it terminates yields a perfect sample.

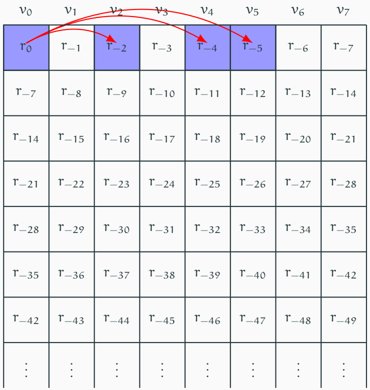
Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\text{ Truncation }]$$

In a typical application (such as $\lambda < \frac{1}{\Delta-1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log\frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.



Truncation

Running CTTP till it terminates yields a perfect sample.

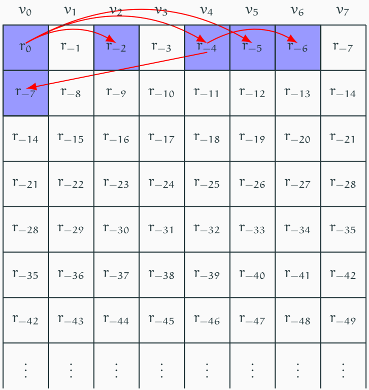
Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\text{ Truncation }]$$

In a typical application (such as $\lambda < \frac{1}{\Delta - 1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.



Truncation

Running CTTP till it terminates yields a perfect sample.

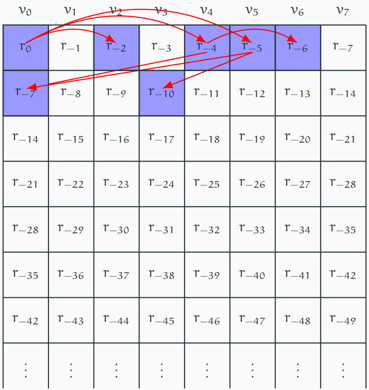
Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\,\text{Truncation}\,]$$

In a typical application (such as $\lambda < \frac{1}{\Delta - 1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.



Running CTTP till it terminates yields a perfect sample.

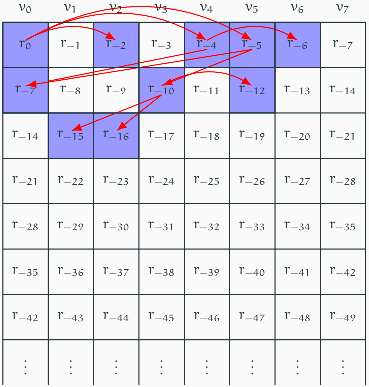
Truncate it if $\geqslant T$ random variables are revealed.

$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\text{ Truncation }]$$

In a typical application (such as $\lambda < \frac{1}{\Delta - 1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.



Truncation

Running CTTP till it terminates yields a perfect sample.

Truncate it if $\geqslant T$ random variables are revealed.

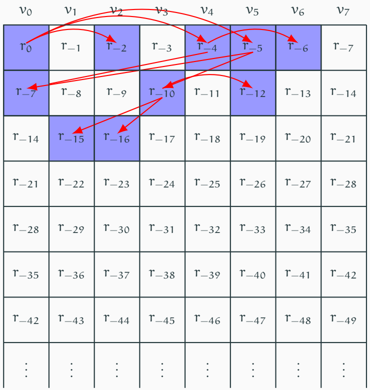
$$d_{TV}\left(\mu_v, \mu_{alg}\right) \leqslant \Pr[\text{ Truncation }]$$

In a typical application (such as $\lambda < \frac{1}{\Delta-1}$ for hardcore),

$$\Pr[t_{run} \geqslant T] \leqslant \exp(-O(T))$$

Thus, taking $T = O(\log \frac{n}{\varepsilon})$ yields $\frac{\varepsilon}{n}$ error.

By enumerating all possible $\exp(T) = \left(\frac{n}{\varepsilon}\right)^C$ random choices, we can deterministically estimate the marginal probability with $\frac{\varepsilon}{n}$ error.



Anand–Jerrum (2022) algorithm

Our algorithm is inspired by the algorithm of Anand and Jerrum (2022):

- recursive marginal sampler
- designed for spin systems on infinite graphs
- constant expected running time with exponential tail bounds
- uses strong spatial mixing

The main difference is that in Anand–Jerrum, once a vertex is fixed, it has to stay fixed in all future recursive calls.

What are these algorithms good for?

Pros

- Approximate samples from the marginal distribution in $O(\log n)$ time
- Can be used to perfectly sample a full configuration in linear expected running time
- Deterministic approximation algorithm

Cons

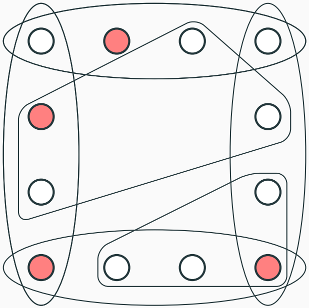
- Weaker bounds for spin systems

  For hardcore models in bounded degree graphs, CTTP works if $\lambda \leqslant \frac{1}{\Delta-1}$, smaller than the critical $\lambda_c(\Delta) \approx \frac{e}{\Delta}$ (Weitz, 2006).

Applications

Hypergraph independent sets

Let $H = (V, E)$ be a hypergraph.
$S \subseteq V$ is *independent* if there is no $e \in E$ such that $e \subseteq S$.



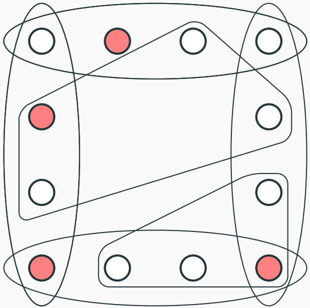
Hypergraph independent sets

Let $H = (V, E)$ be a hypergraph.
$S \subseteq V$ is *independent* if there is no $e \in E$ such that $e \subseteq S$.

Running CTTP for HIS is almost the same as for the hard-core model.

To update v, we need to find a "boundary" of v, conditioned on which the value of v is independent from the rest.

There is a $1/2$ lower bound for "unoccupy".



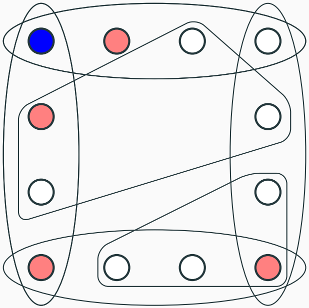
Hypergraph independent sets

Let $H = (V, E)$ be a hypergraph.
$S \subseteq V$ is *independent* if there is no $e \in E$ such that $e \subseteq S$.

Running CTTP for HIS is almost the same as for the hard-core model.

To update v, we need to find a "boundary" of v, conditioned on which the value of v is independent from the rest.

There is a $1/2$ lower bound for "unoccupy".



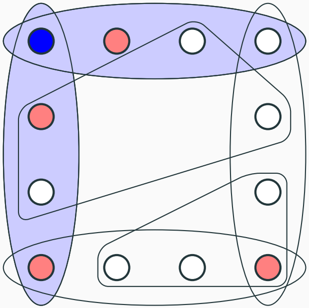
Hypergraph independent sets

Let $H = (V, E)$ be a hypergraph.
$S \subseteq V$ is *independent* if there is no $e \in E$ such that $e \subseteq S$.

Running CTTP for HIS is almost the same as for the hard-core model.

To update v, we need to find a "boundary" of v, conditioned on which the value of v is independent from the rest.

There is a $1/2$ lower bound for "unoccupy".



Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

CTTP for HIS

Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

Theorem

Let $k \geqslant 2$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{\sqrt{8ek^2}} \cdot 2^{\frac{k}{2}}$. There is an FPTAS for the number of independent sets in k-uniform hypergraphs with maximum degree Δ.

Bezáková, Galanis, Goldberg, G., and Štefankovič (2019): $\Delta \geqslant 5 \cdot 2^{\frac{k}{2}}$, **NP**-hard

Hermon, Sly, and Zhang (2019): $\Delta \leqslant c2^{\frac{k}{2}}$, randomised algorithm

Qiu, Wang, and Zhang (2022): $\Delta \leqslant \frac{c}{k} \cdot 2^{\frac{k}{2}}$, perfect sampler

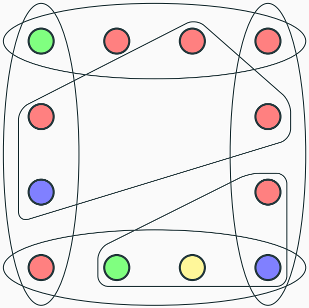
He, Wang, and Yin (2023): $\Delta \lesssim 2^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008, 2006); Jain, Pham, and Vuong (2021)

Let $H = (V, E)$ be a hypergraph.

A colouring $\sigma : V \to [q]$ is *proper* if no $e \in E$ is monochromatic.

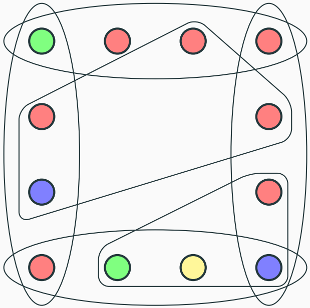


Hypergraph colourings

Let $H = (V, E)$ be a hypergraph.
A colouring $\sigma : V \to [q]$ is *proper* if no $e \in E$ is monochromatic.

The Markov chain runs on a projected state space.
(Feng, G., Yin, and Zhang, 2021; Feng, He, and Yin, 2021)

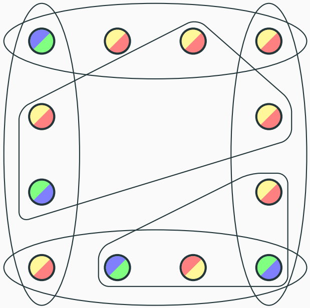
Instead of assigning colours, we divide q colours into s "buckets".
(Eventually we pick $s = q^{2/3}$.)



Let $H = (V, E)$ be a hypergraph.
A colouring $\sigma : V \to [q]$ is *proper* if no $e \in E$ is monochromatic.

The Markov chain runs on a projected state space.
(Feng, G., Yin, and Zhang, 2021; Feng, He, and Yin, 2021)

Instead of assigning colours, we divide q colours into s "buckets".
(Eventually we pick $s = q^{2/3}$.)



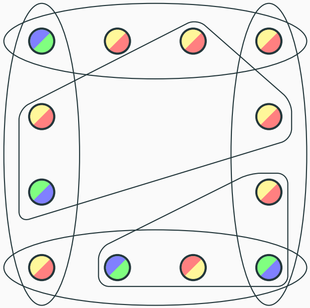
Hypergraph colourings

Let $H = (V, E)$ be a hypergraph.
A colouring $\sigma : V \to [q]$ is *proper* if no $e \in E$ is monochromatic.

The Markov chain runs on a projected state space.
(Feng, G., Yin, and Zhang, 2021; Feng, He, and Yin, 2021)

Instead of assigning colours, we divide q colours into s "buckets".
(Eventually we pick $s = q^{2/3}$.)

The local lemma ensures that with suitable parameters, every vertex's marginal distribution, under an arbitrary conditioning, is close to uniform.



Detour — Lovász local lemma

The original local lemma (Erdős and Lovász 1975) was introduced to show the existence of 3-colourings in hypergraphs.

Let $H = (V, \mathcal{E})$ be the hypergraph, and $\Gamma(e)$ be the set of hyperedges intersecting $e \in \mathcal{E}$. Then $|\Gamma(e)| \leqslant (\Delta - 1)k$.

Theorem (Lovász 1977)

If there exists an assignment $x : \mathcal{E} \to (0,1)$ such that for every $e \in \mathcal{E}$ we have

$$\Pr(e \text{ is monochromatic}) \leqslant x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')), \tag{1}$$

then a proper colouring exists.

Typically we set $x(e) = \frac{1}{k\Delta}$. It gives

$$x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')) \geqslant \frac{1}{k\Delta}\left(1 - \frac{1}{k\Delta}\right)^{k(\Delta-1)} \geqslant \frac{1}{ek\Delta}. \tag{2}$$

Notice that $\Pr(e \text{ is monochromatic}) = \frac{q}{q^k} = \frac{1}{q^{k-1}}$. Thus $\Delta \leqslant \frac{q^{k-1}}{ek}$ suffices.

Detour — Lovász local lemma

The original local lemma (Erdős and Lovász 1975) was introduced to show the existence of 3-colourings in hypergraphs.

Let $H = (V, \mathcal{E})$ be the hypergraph, and $\Gamma(e)$ be the set of hyperedges intersecting $e \in \mathcal{E}$. Then $|\Gamma(e)| \leqslant (\Delta - 1)k$.

Theorem (Lovász 1977)

If there exists an assignment $x : \mathcal{E} \to (0, 1)$ such that for every $e \in \mathcal{E}$ we have

$$\Pr(e \text{ is monochromatic}) \leqslant x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')), \tag{1}$$

then a proper colouring exists.

Typically we set $x(e) = \frac{1}{k\Delta}$. It gives

$$x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')) \geqslant \frac{1}{k\Delta} \left(1 - \frac{1}{k\Delta}\right)^{k(\Delta - 1)} \geqslant \frac{1}{ek\Delta}. \tag{2}$$

Notice that $\Pr(e \text{ is monochromatic}) = \frac{q}{q^k} = \frac{1}{q^{k-1}}$. Thus $\Delta \leqslant \frac{q^{k-1}}{ek}$ suffices.

Detour — Lovász local lemma

The original local lemma (Erdős and Lovász 1975) was introduced to show the existence of 3-colourings in hypergraphs.

Let $H = (V, \mathcal{E})$ be the hypergraph, and $\Gamma(e)$ be the set of hyperedges intersecting $e \in \mathcal{E}$. Then $|\Gamma(e)| \leqslant (\Delta - 1)k$.

Theorem (Lovász 1977)

If there exists an assignment $x : \mathcal{E} \to (0, 1)$ such that for every $e \in \mathcal{E}$ we have

$$\Pr(e \text{ is monochromatic}) \leqslant x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')), \tag{1}$$

then a proper colouring exists.

Typically we set $x(e) = \frac{1}{k\Delta}$. It gives

$$x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')) \geqslant \frac{1}{k\Delta} \left(1 - \frac{1}{k\Delta}\right)^{k(\Delta-1)} \geqslant \frac{1}{ek\Delta}. \tag{2}$$

Notice that $\Pr(e \text{ is monochromatic}) = \frac{q}{q^k} = \frac{1}{q^{k-1}}$. Thus $\Delta \leqslant \frac{q^{k-1}}{ek}$ suffices.

Detour — Lovász local lemma

The original local lemma (Erdős and Lovász 1975) was introduced to show the existence of 3-colourings in hypergraphs.

Let $H = (V, \mathcal{E})$ be the hypergraph, and $\Gamma(e)$ be the set of hyperedges intersecting $e \in \mathcal{E}$. Then $|\Gamma(e)| \leqslant (\Delta - 1)k$.

Theorem (Lovász 1977)

If there exists an assignment $x : \mathcal{E} \to (0, 1)$ such that for every $e \in \mathcal{E}$ we have

$$\Pr(e \textit{ is monochromatic}) \leqslant x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')), \tag{1}$$

then a proper colouring exists.

Typically we set $x(e) = \frac{1}{k\Delta}$. It gives

$$x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')) \geqslant \frac{1}{k\Delta} \left(1 - \frac{1}{k\Delta}\right)^{k(\Delta-1)} \geqslant \frac{1}{ek\Delta}. \tag{2}$$

Notice that $\Pr(e \text{ is monochromatic}) = \frac{q}{q^k} = \frac{1}{q^{k-1}}$. Thus $\Delta \leqslant \frac{q^{k-1}}{ek}$ suffices.

The original local lemma (Erdős and Lovász 1975) was introduced to show the existence of 3-colourings in hypergraphs.

Let $H = (V, \mathcal{E})$ be the hypergraph, and $\Gamma(e)$ be the set of hyperedges intersecting $e \in \mathcal{E}$. Then $|\Gamma(e)| \leqslant (\Delta - 1)k$.

Theorem (Lovász 1977)

If there exists an assignment $x : \mathcal{E} \to (0, 1)$ such that for every $e \in \mathcal{E}$ we have

$$\Pr(e \textit{ is monochromatic}) \leqslant x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')), \tag{1}$$

then a proper colouring exists.

Typically we set $x(e) = \frac{1}{k\Delta}$. It gives

$$x(e) \prod_{e' \in \Gamma(e)} (1 - x(e')) \geqslant \frac{1}{k\Delta} \left(1 - \frac{1}{k\Delta}\right)^{k(\Delta - 1)} \geqslant \frac{1}{ek\Delta}. \tag{2}$$

Notice that $\Pr(e \text{ is monochromatic}) = \frac{q}{q^k} = \frac{1}{q^{k-1}}$. Thus $\Delta \leqslant \frac{q^{k-1}}{ek}$ suffices.

Local uniformity

Let $\mu(\cdot)$ be the Gibbs (uniform) distribution on all proper colourings,

The local lemma also gives an upper bound for any event under $\mu(\cdot)$.

Theorem (Haeupler, Saha, and Srinivasan 2011)

If the local lemma holds for every $e \in \mathcal{E}$, then for any event B, $\mu(B) \leqslant \Pr(B)\prod_{e\in\Gamma(B)}(1-x(e))^{-1}$.

This implies that buckets are almost uniform, even with arbitrary conditioning. (Recall that $s = q^{2/3}$.)

Lemma (local uniformity)

If $\lfloor q/s \rfloor^k \geqslant 4eqs\Delta k$, then for any $v \in V$, any subset $\Lambda \subseteq V \setminus \{v\}$ and partial configuration $\sigma_\Lambda \in [s]^\Lambda$, it follows that

$$\forall j \in [s], \quad \frac{1}{s}\left(1 - \frac{1}{4s}\right) \leqslant \psi_v^{\sigma_\Lambda}(j) \leqslant \frac{1}{s}\left(1 + \frac{1}{s}\right).$$

Namely, the probability of getting $\perp$ is always small.

From this we need $\Delta \lesssim \left(\frac{q}{s}\right)^k$.

Local uniformity

Let $\mu(\cdot)$ be the Gibbs (uniform) distribution on all proper colourings,

The local lemma also gives an upper bound for any event under $\mu(\cdot)$.

Theorem (Haeupler, Saha, and Srinivasan 2011)

If the local lemma holds for every $e \in \mathcal{E}$*, then for any event* B, $\mu(B) \leqslant \Pr(B) \prod_{e \in \Gamma(B)} (1 - x(e))^{-1}$.

This implies that buckets are almost uniform, even with arbitrary conditioning. (Recall that $s = q^{2/3}$.)

Lemma (local uniformity)

If $\lfloor q/s \rfloor^k \geqslant 4eqs\Delta k$*, then for any* $v \in V$*, any subset* $\Lambda \subseteq V \setminus \{v\}$ *and partial configuration* $\sigma_\Lambda \in [s]^\Lambda$*, it follows that*

$$\forall j \in [s], \quad \frac{1}{s}\left(1 - \frac{1}{4s}\right) \leqslant \psi_v^{\sigma_\Lambda}(j) \leqslant \frac{1}{s}\left(1 + \frac{1}{s}\right).$$

Namely, the probability of getting $\perp$ is always small.

From this we need $\Delta \lesssim \left(\frac{q}{s}\right)^k$.

Complications in HC

There are a few issues with CTTP for hypergraph colourings.

1. We run Glauber dynamics in the projected state space, meaning that the "boundary" of a vertex v needs to adapt to the current configuration.

   We find a boundary such that all crossing hyperedges are non-monochromatic.

2. We cannot do the telescoping product reduction for the marginals. Instead, we consider a sequence of hypergraphs by removing hyperedges one by one.

   Thus we need to sample the marginal distribution of k vertices, instead of one. Some extra care for consistency is required.

3. The above only samples buckets. To get the colours, we condition on the buckets of all vertices within the boundary of the last update, and use brute force to get the colours.

Complications in HC

There are a few issues with CTTP for hypergraph colourings.

1. We run Glauber dynamics in the projected state space, meaning that the "boundary" of a vertex v needs to adapt to the current configuration.

   We find a boundary such that all crossing hyperedges are non-monochromatic.

2. We cannot do the telescoping product reduction for the marginals. Instead, we consider a sequence of hypergraphs by removing hyperedges one by one.

   Thus we need to sample the marginal distribution of k vertices, instead of one. Some extra care for consistency is required.

3. The above only samples buckets. To get the colours, we condition on the buckets of all vertices within the boundary of the last update, and use brute force to get the colours.

Complications in HC

There are a few issues with CTTP for hypergraph colourings.

1. We run Glauber dynamics in the projected state space, meaning that the "boundary" of a vertex v needs to adapt to the current configuration.

   We find a boundary such that all crossing hyperedges are non-monochromatic.

2. We cannot do the telescoping product reduction for the marginals. Instead, we consider a sequence of hypergraphs by removing hyperedges one by one.

   Thus we need to sample the marginal distribution of k vertices, instead of one. Some extra care for consistency is required.

3. The above only samples buckets. To get the colours, we condition on the buckets of all vertices within the boundary of the last update, and use brute force to get the colours.

Complications in HC

There are a few issues with CTTP for hypergraph colourings.

1. We run Glauber dynamics in the projected state space, meaning that the "boundary" of a vertex v needs to adapt to the current configuration.

   We find a boundary such that all crossing hyperedges are non-monochromatic.

2. We cannot do the telescoping product reduction for the marginals. Instead, we consider a sequence of hypergraphs by removing hyperedges one by one.

   Thus we need to sample the marginal distribution of k vertices, instead of one. Some extra care for consistency is required.

3. The above only samples buckets. To get the colours, we condition on the buckets of all vertices within the boundary of the last update, and use brute force to get the colours.

CTTP for HC

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

CTTP for HC

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

CTTP for HC

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

CTTP for HC

Theorem

Let $k \geqslant 20$, $\Delta \geqslant 2$ and q be three integers satisfying $\Delta \leqslant \left(\frac{q}{64}\right)^{\frac{k-5}{3}}$. There is an FPTAS for the number of proper q-colourings in k-uniform hypergraphs with maximum degree Δ.

Galanis, G., and Wang (2022+): for even q, $\Delta \geqslant 5 \cdot q^{\frac{k}{2}}$, **NP**-hard

Jain, Pham, and Vuong (2021a): $\Delta \lesssim q^{\frac{k}{3}}$, randomised algorithm

He, Sun, and Wu (2021): $\Delta \lesssim q^{\frac{k}{3}}$, perfect sampler

He, Wang, and Yin (2023): $\Delta \lesssim q^{\frac{k}{5}}$, deterministic algorithm

Other previous work:

Bordewich, Dyer, and Karpinski (2008); G., Liao, Lu, and Zhang (2019); Feng, He, and Yin (2021); Jain, Pham, and Vuong (2021b)

A few words about the analysis

Recall that the truncation probability at $T = O(\log n)$ bounds the error in TV distance.

To bound the truncation probability, we consider the extended hypergraph, introduced by He, Sun, and Wu (2021). It creates a copy of each variable every time it is updated.

If truncation happens, then there must be a large connected component in the extended hypergraph, inside which there are a linear fraction of variables getting $\perp$ when they are first resolved. The last event is very unlikely because of local uniformity from the local lemma.

This analysis requires $\Delta \lesssim s^{k/2}$. Recall that local uniformity requires $\Delta \lesssim (q/s)^k$.

Thus, the best we can do is $\Delta \lesssim q^{k/3}$ by choosing $s \approx q^{2/3}$.

This highlights a major difference between CTTP and Anand–Jerrum: in AJ, once a variable is pinned, it will stay pinned for all future recursive calls. Thus, in the analysis above, it only contributes once.

A few words about the analysis

Recall that the truncation probability at $T = O(\log n)$ bounds the error in TV distance.

To bound the truncation probability, we consider the extended hypergraph, introduced by He, Sun, and Wu (2021). It creates a copy of each variable every time it is updated.

If truncation happens, then there must be a large connected component in the extended hypergraph, inside which there are a linear fraction of variables getting $\perp$ when they are first resolved. The last event is very unlikely because of local uniformity from the local lemma.

This analysis requires $\Delta \lesssim s^{k/2}$. Recall that local uniformity requires $\Delta \lesssim (q/s)^k$.

Thus, the best we can do is $\Delta \lesssim q^{k/3}$ by choosing $s \approx q^{2/3}$.

This highlights a major difference between CTTP and Anand–Jerrum: in AJ, once a variable is pinned, it will stay pinned for all future recursive calls. Thus, in the analysis above, it only contributes once.

A few words about the analysis

Recall that the truncation probability at $T = O(\log n)$ bounds the error in TV distance.

To bound the truncation probability, we consider the extended hypergraph, introduced by He, Sun, and Wu (2021). It creates a copy of each variable every time it is updated.

If truncation happens, then there must be a large connected component in the extended hypergraph, inside which there are a linear fraction of variables getting $\perp$ when they are first resolved. The last event is very unlikely because of local uniformity from the local lemma.

This analysis requires $\Delta \lesssim s^{k/2}$. Recall that local uniformity requires $\Delta \lesssim (q/s)^k$.

Thus, the best we can do is $\Delta \lesssim q^{k/3}$ by choosing $s \approx q^{2/3}$.

This highlights a major difference between CTTP and Anand–Jerrum: in AJ, once a variable is pinned, it will stay pinned for all future recursive calls. Thus, in the analysis above, it only contributes once.

A few words about the analysis

Recall that the truncation probability at $T = O(\log n)$ bounds the error in TV distance.

To bound the truncation probability, we consider the extended hypergraph, introduced by He, Sun, and Wu (2021). It creates a copy of each variable every time it is updated.

If truncation happens, then there must be a large connected component in the extended hypergraph, inside which there are a linear fraction of variables getting $\perp$ when they are first resolved. The last event is very unlikely because of local uniformity from the local lemma.

This analysis requires $\Delta \lesssim s^{k/2}$. Recall that local uniformity requires $\Delta \lesssim (q/s)^k$.

Thus, the best we can do is $\Delta \lesssim q^{k/3}$ by choosing $s \approx q^{2/3}$.

This highlights a major difference between CTTP and Anand–Jerrum: in AJ, once a variable is pinned, it will stay pinned for all future recursive calls. Thus, in the analysis above, it only contributes once.

Linear hypergraphs

Linear: $\forall e_1, e_2 \in E, |e_1 \cap e_2| \leqslant 1$

Theorem

For any real $\delta > 0$, let $k \geqslant \frac{25(1+\delta)^2}{\delta^2}$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{100k^3} 2^{k/(1+\delta)}$. There is an FPTAS for the number of independent sets in k-uniform linear hypergraphs with maximum degree Δ.

Theorem

For any read $\delta > 0$, let $k \geqslant \frac{50(1+\delta)^2}{\delta^2}$, $\Delta \geqslant 2$ and q be three integers such that $\Delta \leqslant \left(\frac{q}{50}\right)^{\frac{k-3}{2+\delta}}$. There is an FPTAS for the number of proper q-colourings in k-uniform linear hypergraphs with maximum degree Δ.

These match various bounds for randomised algorithms in the leading order by Hermon, Sly, and Zhang (2019); Qiu, Wang, and Zhang (2022); Feng, G., and Wang (2022).

For linear hypergraph independent sets, no hardness result is known.
For colouring linear hypergraphs, Galanis, G., and Wang (2022+) showed that it is **NP**-hard to find a colouring if $\Delta \geqslant 2kq^k \log q + 2q$.

Linear hypergraphs

Linear: $\forall e_1, e_2 \in E, |e_1 \cap e_2| \leqslant 1$

Theorem

For any real $\delta > 0$, let $k \geqslant \frac{25(1+\delta)^2}{\delta^2}$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{100k^3} 2^{k/(1+\delta)}$. There is an FPTAS for the number of independent sets in k-uniform linear hypergraphs with maximum degree Δ.

Theorem

For any read $\delta > 0$, let $k \geqslant \frac{50(1+\delta)^2}{\delta^2}$, $\Delta \geqslant 2$ and q be three integers such that $\Delta \leqslant \left(\frac{q}{50}\right)^{\frac{k-3}{2+\delta}}$. There is an FPTAS for the number of proper q-colourings in k-uniform linear hypergraphs with maximum degree Δ.

These match various bounds for randomised algorithms in the leading order by Hermon, Sly, and Zhang (2019); Qiu, Wang, and Zhang (2022); Feng, G., and Wang (2022).

For linear hypergraph independent sets, no hardness result is known.
For colouring linear hypergraphs, Galanis, G., and Wang (2022+) showed that it is **NP**-hard to find a colouring if $\Delta \geqslant 2kq^k \log q + 2q$.

Linear hypergraphs

Linear: $\forall e_1, e_2 \in E, |e_1 \cap e_2| \leqslant 1$

Theorem

For any real $\delta > 0$, let $k \geqslant \frac{25(1+\delta)^2}{\delta^2}$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{100k^3} 2^{k/(1+\delta)}$. There is an FPTAS for the number of independent sets in k-uniform linear hypergraphs with maximum degree Δ.

Theorem

For any read $\delta > 0$, let $k \geqslant \frac{50(1+\delta)^2}{\delta^2}$, $\Delta \geqslant 2$ and q be three integers such that $\Delta \leqslant \left(\frac{q}{50}\right)^{\frac{k-3}{2+\delta}}$. There is an FPTAS for the number of proper q-colourings in k-uniform linear hypergraphs with maximum degree Δ.

These match various bounds for randomised algorithms in the leading order by Hermon, Sly, and Zhang (2019); Qiu, Wang, and Zhang (2022); Feng, G., and Wang (2022).

For linear hypergraph independent sets, no hardness result is known.
For colouring linear hypergraphs, Galanis, G., and Wang (2022+) showed that it is **NP**-hard to find a colouring if $\Delta \geqslant 2kq^k \log q + 2q$.

Linear hypergraphs

Linear: $\forall e_1, e_2 \in E, |e_1 \cap e_2| \leqslant 1$

Theorem

For any real $\delta > 0$, let $k \geqslant \frac{25(1+\delta)^2}{\delta^2}$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{100k^3} 2^{k/(1+\delta)}$. There is an FPTAS for the number of independent sets in k-uniform linear hypergraphs with maximum degree Δ.

Theorem

For any read $\delta > 0$, let $k \geqslant \frac{50(1+\delta)^2}{\delta^2}$, $\Delta \geqslant 2$ and q be three integers such that $\Delta \leqslant \left(\frac{q}{50}\right)^{\frac{k-3}{2+\delta}}$. There is an FPTAS for the number of proper q-colourings in k-uniform linear hypergraphs with maximum degree Δ.

These match various bounds for randomised algorithms in the leading order by Hermon, Sly, and Zhang (2019); Qiu, Wang, and Zhang (2022); Feng, G., and Wang (2022).

For linear hypergraph independent sets, no hardness result is known.
For colouring linear hypergraphs, Galanis, G., and Wang (2022+) showed that it is **NP**-hard to find a colouring if $\Delta \geqslant 2kq^k \log q + 2q$.

Linear hypergraphs

Linear: $\forall e_1, e_2 \in E, |e_1 \cap e_2| \leqslant 1$

Theorem

For any real $\delta > 0$, let $k \geqslant \frac{25(1+\delta)^2}{\delta^2}$ and $\Delta \geqslant 2$ be two integers such that $\Delta \leqslant \frac{1}{100k^3} 2^{k/(1+\delta)}$. There is an FPTAS for the number of independent sets in k-uniform linear hypergraphs with maximum degree Δ.

Theorem

For any read $\delta > 0$, let $k \geqslant \frac{50(1+\delta)^2}{\delta^2}$, $\Delta \geqslant 2$ and q be three integers such that $\Delta \leqslant \left(\frac{q}{50}\right)^{\frac{k-3}{2+\delta}}$. There is an FPTAS for the number of proper q-colourings in k-uniform linear hypergraphs with maximum degree Δ.

These match various bounds for randomised algorithms in the leading order by Hermon, Sly, and Zhang (2019); Qiu, Wang, and Zhang (2022); Feng, G., and Wang (2022).

For linear hypergraph independent sets, ~~no hardness result is known.~~
NP-hard if $\Delta \geqslant 2.5 \cdot 2^k$ by Qiu and Wang (2022).

For colouring linear hypergraphs, Galanis, G., and Wang (2022+) showed that it is **NP**-hard to find a colouring if $\Delta \geqslant 2kq^k \log q + 2q$.

Spin systems

With little additional effort, one can show that the algorithm by Anand and Jerrum (2022) obtains approximate marginal samples within $O(\log n)$ time for spin systems with strong spatial mixing in subexponential neighbourhood growth graphs.

This implies various new FPTASes, most notably, for lattices, such as 6-colourings on $\mathbb{Z}^2$.

The main challenge remains:

find a $O(\log n)$-time marginal sampler for the hardcore model or graph colourings under conditions where other methods work.

For q-colouring graphs with degree $\leqslant \Delta$, our method works when $q = \Omega(\Delta^2)$, and yet many rapid mixing or perfect sampling results are known when $q > C\Delta$ for various constant C.

Spin systems

With little additional effort, one can show that the algorithm by Anand and Jerrum (2022) obtains approximate marginal samples within $O(\log n)$ time for spin systems with strong spatial mixing in subexponential neighbourhood growth graphs.

This implies various new FPTASes, most notably, for lattices, such as 6-colourings on $\mathbb{Z}^2$.

The main challenge remains:

find a $O(\log n)$-time marginal sampler for the hardcore model or graph colourings under conditions where other methods work.

For q-colouring graphs with degree $\leqslant \Delta$, our method works when $q = \Omega(\Delta^2)$, and yet many rapid mixing or perfect sampling results are known when $q > C\Delta$ for various constant C.

Open problems

- Hypergraph colourings: $\Delta \lesssim q^{k/2}$?

- Running time:

  we take $T = \text{poly}(\Delta, k, \log q) \log \frac{n}{\varepsilon}$, which leads to $\left(\frac{n}{\varepsilon}\right)^{\text{poly}(\Delta, k, \log q)}$ for FPTAS.

  Does $f(\Delta, k, q) \left(\frac{n}{\varepsilon}\right)^{c}$-time FPTAS exist for a constant c?

- Can we derandomise other chains like the matching chain or the bases-exchange chain?

Open problems

- Hypergraph colourings: $\Delta \lesssim q^{k/2}$?

- Running time:

  we take $T = \text{poly}(\Delta, k, \log q) \log \frac{n}{\varepsilon}$, which leads to $\left(\frac{n}{\varepsilon}\right)^{\text{poly}(\Delta, k, \log q)}$ for FPTAS.

  Does $f(\Delta, k, q) \left(\frac{n}{\varepsilon}\right)^{c}$-time FPTAS exist for a constant c?

- Can we derandomise other chains like the matching chain or the bases-exchange chain?

Open problems

- Hypergraph colourings: $\Delta \lesssim q^{k/2}$?
- Running time:

  we take $T = \text{poly}(\Delta, k, \log q) \log \frac{n}{\varepsilon}$, which leads to $\left(\frac{n}{\varepsilon}\right)^{\text{poly}(\Delta, k, \log q)}$ for FPTAS.

  Does $f(\Delta, k, q) \left(\frac{n}{\varepsilon}\right)^{c}$-time FPTAS exist for a constant c?
- Can we derandomise other chains like the matching chain or the bases-exchange chain?

Thank you!

arXiv:2211.03487