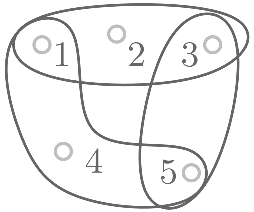
A Perfect Sampler for Hypergraph Independent Sets

Guoliang Qiu, Yanheng Wang, Chihao Zhang 06.07.2022

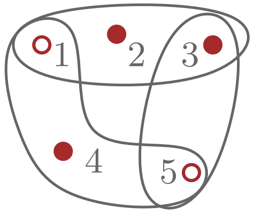
A Perfect Sampler for Hypergraph Independent Sets

vertex set $[n]$



A Perfect Sampler for Hypergraph Independent Sets

vertex set $[n]$

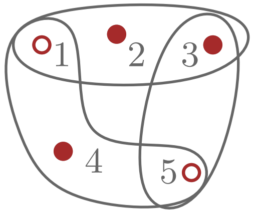


$\sigma : [n] \to \{\bullet, \circ\}$ such that no hyperedge is fully-$\bullet$

A Perfect Sampler for Hypergraph Independent Sets

outputs an independent set uniformly at random

vertex set $[n]$



$\sigma : [n] \to \{\bullet, \circ\}$ such that no hyperedge is fully-$\bullet$

Motivation and Results

For k-uniform d-regular hypergraphs,

- $d > 5 \cdot 2^{k/2}$: no poly-time approximate sampler assuming RP $\neq$ NP [BGGGS19]
- $d \leq c \cdot 2^{k/2}$: poly-time approximate sampler via Glauber dynamics [HSZ19]

Motivation and Results

For k-uniform d-regular hypergraphs,

- $d > 5 \cdot 2^{k/2}$: no poly-time approximate sampler assuming RP $\neq$ NP [BGGGS19]
- $d \leq c \cdot 2^{k/2}$: poly-time approximate sampler via Glauber dynamics [HSZ19]

Questions:

- non-uniform, non-regular?
- perfect (instead of approximate) samples?
- simpler analysis?

Motivation and Results

For k-uniform d-regular hypergraphs,

- $d > 5 \cdot 2^{k/2}$: no poly-time approximate sampler assuming RP $\neq$ NP [BGGGS19]
- $d \leq c \cdot 2^{k/2}$: poly-time approximate sampler via Glauber dynamics [HSZ19]

Questions:

- non-uniform, non-regular?
- perfect (instead of approximate) samples?
- simpler analysis?

Result: A sampler that outputs a *perfectly* uniform independent set a.s. Its expected running time is polynomial, assuming

- either a LLL condition for general hypergraphs;
- or $d \leq c \cdot 2^{k/2}$ for k-uniform d-regular hypergraphs.

Motivation and Results

For k-uniform d-regular hypergraphs,

- $d > 5 \cdot 2^{k/2}$: no poly-time approximate sampler assuming RP $\neq$ NP [BGGGS19]
- $d \leq c \cdot 2^{k/2}$: poly-time approximate sampler via Glauber dynamics [HSZ19]

Questions:

- non-uniform, non-regular?
- perfect (instead of approximate) samples? ⟶ "coupling from the past" [PW96]
- simpler analysis?

Result: A sampler that outputs a *perfectly* uniform independent set a.s. Its expected running time is polynomial, assuming

- either a LLL condition for general hypergraphs;
- or $d \leq c \cdot 2^{k/2}$ for k-uniform d-regular hypergraphs.

Motivation and Results

For k-uniform d-regular hypergraphs,

- $d > 5 \cdot 2^{k/2}$: no poly-time approximate sampler assuming RP $\neq$ NP [BGGGS19]
- $d \leq c \cdot 2^{k/2}$: poly-time approximate sampler via Glauber dynamics [HSZ19]

Questions:

- non-uniform, non-regular?
- perfect (instead of approximate) samples? ⟶ "coupling from the past" [PW96]
- simpler analysis? ⟶ systematic scan + witness structure for analysis [HSW21]

Result: A sampler that outputs a *perfectly* uniform independent set a.s. Its expected running time is polynomial, assuming

- either a LLL condition for general hypergraphs;
- or $d \leq c \cdot 2^{k/2}$ for k-uniform d-regular hypergraphs.

The Systematic Scan

Update schedule & rule:

time t

$v_t =$

The Systematic Scan

Update schedule & rule:

The Systematic Scan

Update schedule & rule:

$\longrightarrow$ time t

$v_t = 1 \quad 2$

The Systematic Scan

Update schedule & rule:

$\longrightarrow$ time t

$v_t = 1 \quad 2 \quad \cdots \quad n$

The Systematic Scan

Update schedule & rule:

$$v_t = 1 \quad 2 \quad \cdots \quad n \quad 1 \longrightarrow \text{time } t$$

The Systematic Scan

Update schedule & rule:

$\longrightarrow$ time t

$v_t = 1 \quad 2 \quad \cdots \quad n \quad 1 \quad 2$

The Systematic Scan

Update schedule & rule:

$$v_t = 1 \quad 2 \quad \cdots \quad n \quad 1 \quad 2 \quad \cdots \quad n \longrightarrow \text{time } t$$

The Systematic Scan

Update schedule & rule:

$$v_t = 1 \quad 2 \quad \cdots \quad n \quad 1 \quad 2 \quad \cdots \quad n \quad \longrightarrow \text{time } t$$

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

The Systematic Scan

Update schedule & rule:

$v_t = \boxed{1 \quad 2 \quad \cdots \quad n}\boxed{1 \quad 2 \quad \cdots \quad n} \longrightarrow$ time t

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

Convergence:
Group every n steps into one block.

The Systematic Scan

Update schedule & rule:

$$v_t = \boxed{1 \quad 2 \quad \cdots \quad n}\boxed{1 \quad 2 \quad \cdots \quad n} \longrightarrow \text{time } t$$

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

Convergence:
Group every n steps into one block.

Coupling from the Past [PW96]

For each independent set σ, denote by (X_t^σ) the Markov chain starting from σ.

The Systematic Scan

Update schedule & rule:

$v_t = $ | 1 2 ⋯ n | 1 2 ⋯ n | → time t

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

Convergence:
Group every n steps into one block.

Coupling from the Past [PW96]

For each independent set σ, denote by (X_t^σ) the Markov chain starting from σ.

Grand coupling: Run these chains in parallel, sharing the same random coin R_t.

The Systematic Scan

Update schedule & rule:

$v_t = $ | $1 \quad 2 \quad \cdots \quad n$ | $1 \quad 2 \quad \cdots \quad n$ | → time t

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

Convergence:
Group every n steps into one block.

Coupling from the Past [PW96]

For each independent set σ, denote by (X_t^σ) the Markov chain starting from σ.

Grand coupling: Run these chains in parallel, sharing the same random coin R_t.

Coalescence by time T:
$X_T^\sigma = X_T^\pi$ for all $\sigma, \pi \in \Omega$.

The Systematic Scan

Update schedule & rule:

$v_t = \boxed{1 \quad 2 \quad \cdots \quad n}\boxed{1 \quad 2 \quad \cdots \quad n} \longrightarrow$ time t

At time t,

- Flip a fair coin $R_t \in \{\bullet, \circ\}$;
- Colour v_t by R_t if allowed.

Convergence:
Group every n steps into one block.

Coupling from the Past [PW96]

For each independent set σ, denote by (X_t^σ) the Markov chain starting from σ.

Grand coupling: Run these chains in parallel, sharing the same random coin R_t.

Coalescence by time T:
$X_T^\sigma = X_T^\pi$ for all $\sigma, \pi \in \Omega$.

CFTP transformation: If we can design a routine that detects coalescence, then we can turn it into a perfect sampler!

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^{\sigma} \neq X_T^{\pi}$), then ...

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \; X_T^\sigma(v) \neq X_T^\pi(v)$

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \; X_T^\sigma(v) \neq X_T^\pi(v)$
- What happened at T', the last time we updated v?

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \ X_T^\sigma(v) \neq X_T^\pi(v)$
- What happened at T', the last time we updated v?
- $R_{T'} = \bullet$ *and* $\exists C' \ni v$:almost full in $X_{T'}^\sigma$ but not so in $X_{T'}^\pi$

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \; X_T^\sigma(v) \neq X_T^\pi(v)$
- What happened at T', the last time we updated v?
- $R_{T'} = \bullet$ *and* $\exists C' \ni v$:almost full in $X_{T'}^\sigma$ but not so in $X_{T'}^\pi$

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \; X_T^\sigma(v) \neq X_T^\pi(v)$
- What happened at T', the last time we updated v?
- $R_{T'} = \bullet$ *and* $\exists C' \ni v$:almost full in $X_{T'}^\sigma$ but not so in $X_{T'}^\pi$

- $\exists v' : \; X_{T'}^\sigma(v') \neq X_{T'}^\pi(v')$
- What happened at T'', the last time we updated v'?
- $R_{T''} = \bullet$ *and* $\exists C'' \ni v'$: almost full in $X_{T''}^\sigma$ but not so in $X_{T''}^\pi$

Idea of Coalescence Detection

If *not coalesce* by time T (that is $\exists \sigma, \pi : X_T^\sigma \neq X_T^\pi$), then ...

- $\exists v : \; X_T^\sigma(v) \neq X_T^\pi(v)$
- What happened at T', the last time we updated v?
- $R_{T'} = \bullet$ *and* $\exists C' \ni v$:almost full in $X_{T'}^\sigma$ but not so in $X_{T'}^\pi$

- $\exists v' : \; X_{T'}^\sigma(v') \neq X_{T'}^\pi(v')$
- What happened at T'', the last time we updated v'?
- $R_{T''} = \bullet$ *and* $\exists C'' \ni v'$: almost full in $X_{T''}^\sigma$ but not so in $X_{T''}^\pi$

...

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

foreach time $t = 1, \ldots, T$:

if $v_t \in C$ *then* create a node $e_{C,t}$

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

foreach time $t = 1, \ldots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:
  foreach time $t = 1, \dots, T$:
    if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$

t	1	2	3	4	5	6	7	$\cdots$
v_t	1	2	3	4	1	2	3	$\cdots$

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

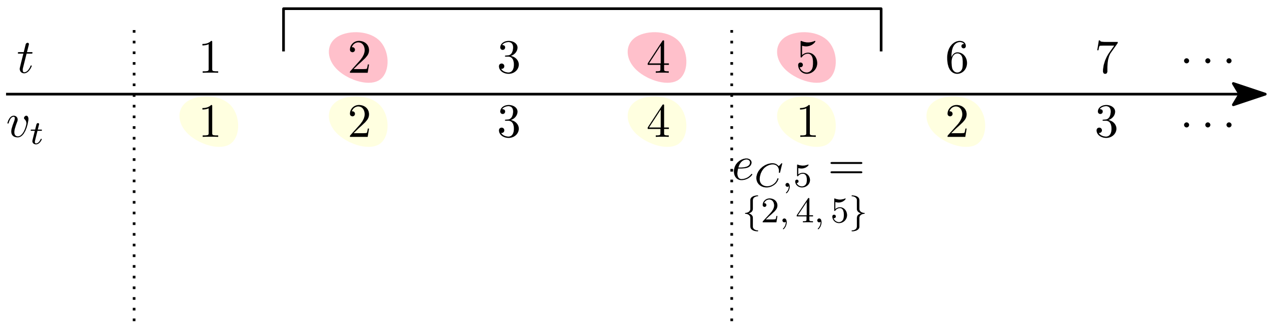
foreach time $t = 1, \dots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

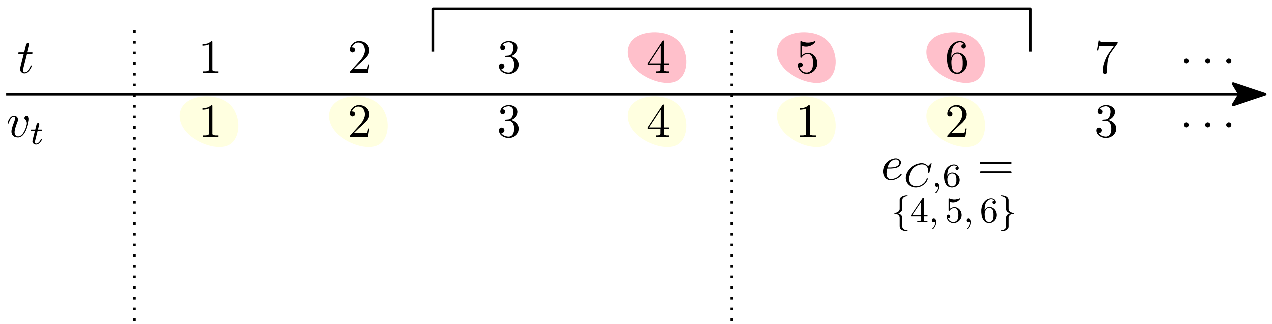
foreach time $t = 1, \ldots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

foreach time $t = 1, \dots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$

t	1	2	3	4	5	6	7	$\cdots$
v_t	1	2	3	4	1	2	3	$\cdots$
$e_{C,t}$	$\{1\}$	$\{1,2\}$		$\{1,2,4\}$	$\{2,4,5\}$	$\{4,5,6\}$		
$e_{D,t}$		$\{2\}$	$\{2,3\}$			$\{3,6\}$	$\{6,7\}$	

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

  foreach time $t = 1, \ldots, T$:

    if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

Arcs. $e_{C',t'} \leftarrow e_{C,t}$ iff $t' < t$ and $(e_{C',t'} \cap e_{C,t}) \neq \emptyset$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$

t	1	2	3	4	5	6	7	$\cdots$
v_t	1	2	3	4	1	2	3	$\cdots$
$e_{C,t}$	$\{1\}$	$\{1,2\}$		$\{1,2,4\}$	$\{2,4,5\}$	$\{4,5,6\}$		
$e_{D,t}$		$\{2\}$	$\{2,3\}$			$\{3,6\}$	$\{6,7\}$	

Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

  foreach time $t = 1, \ldots, T$:

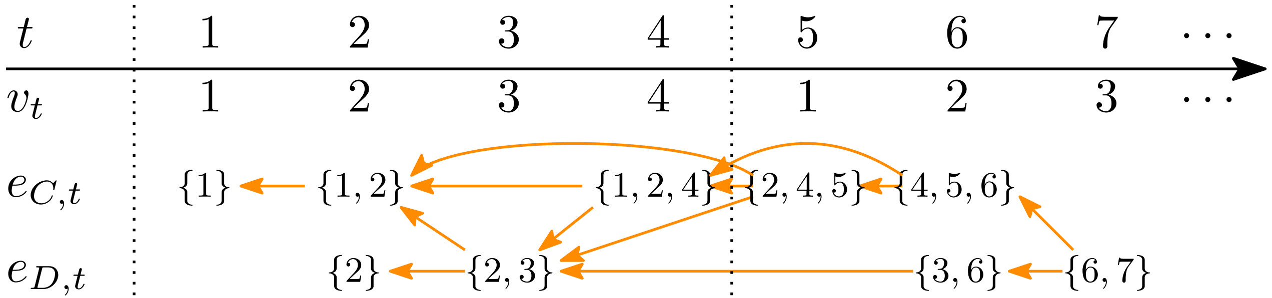
    if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

Arcs. $e_{C',t'} \leftarrow e_{C,t}$ iff $t' < t$ and $(e_{C',t'} \cap e_{C,t}) \neq \emptyset$

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

foreach time $t = 1, \ldots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

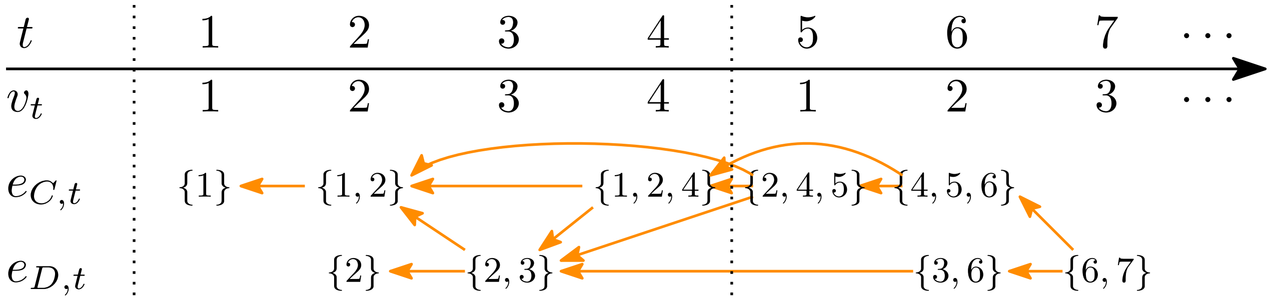
Arcs. $e_{C',t'} \leftarrow e_{C,t}$ iff $t' < t$ and $(e_{C',t'} \cap e_{C,t}) \neq \emptyset$

Lemma.
If no coalescence occur by time T, then there exists an *induced* path $P = (e_\ell \leftarrow \cdots \leftarrow e_1)$ of length $\ell \geq T/n$ in the witness graph such that $R_t = \bullet$ for all $t \in \bigcup_{i=1}^{\ell} e_i$.

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:

foreach time $t = 1, \dots, T$:

if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

Arcs. $e_{C',t'} \leftarrow e_{C,t}$ iff $t' < t$ and $(e_{C',t'} \cap e_{C,t}) \neq \emptyset$

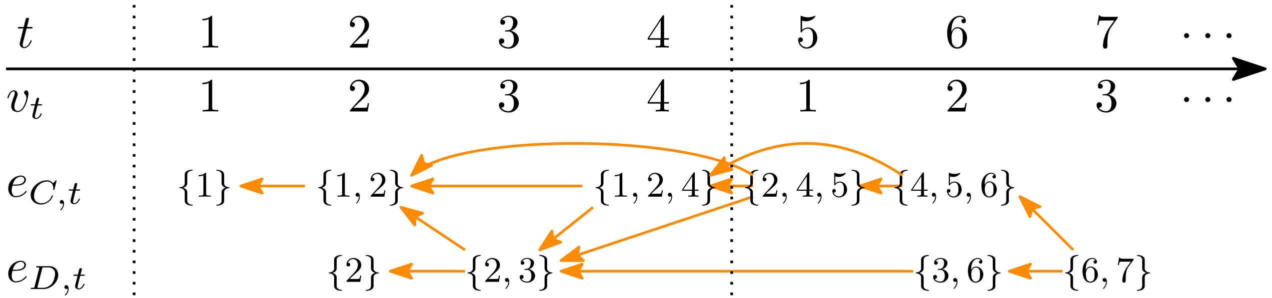
Lemma.
If no coalescence occur by time T, then there exists an *induced* path $P = (e_\ell \leftarrow \cdots \leftarrow e_1)$ of length $\ell \geq T/n$ in the witness graph such that $R_t = \bullet$ for all $t \in \bigcup_{i=1}^{\ell} e_i$.

bad event B_P; $\mathbb{P}(B_P)$ is very low!

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Witness Directed Graph Up to T

Nodes. *foreach* hyperedge C:
  foreach time $t = 1, \dots, T$:
    if $v_t \in C$ *then* create a node $e_{C,t} := \{\tau \in (t-n, t] : v_\tau \in C\}$

Arcs. $e_{C',t'} \leftarrow e_{C,t}$ iff $t' < t$ and $(e_{C',t'} \cap e_{C,t}) \neq \emptyset$

Lemma.

union bound

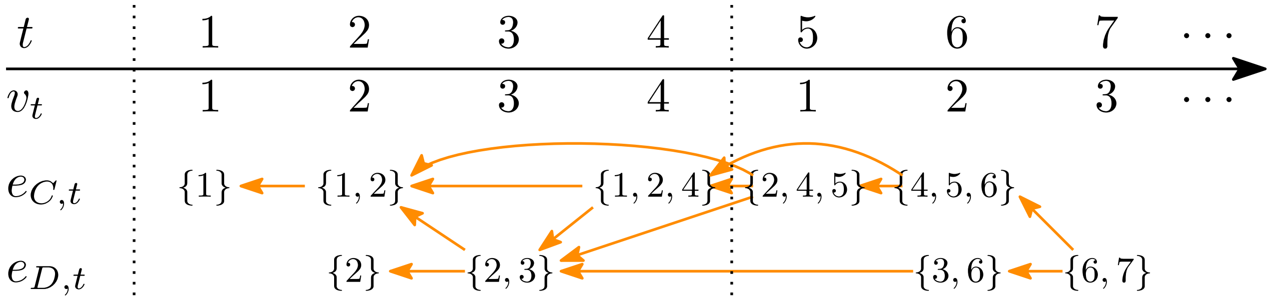
If no coalescence occur by time T, then there exists an *induced* path $P = (e_\ell \leftarrow \cdots \leftarrow e_1)$ of length $\ell \geq T/n$ in the witness graph such that $R_t = \bullet$ for all $t \in \bigcup_{i=1}^{\ell} e_i$.

bad event B_P; $\mathbb{P}(B_P)$ is very low!

$n = 4$

$C := \{1, 2, 4\}$

$D := \{2, 3\}$



Analysis

$$\begin{array}{ccccccc} C_\ell & C_{\ell-1} & & C_4 & C_3 & C_2 & C_1 \end{array}$$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

Analysis

$$\begin{array}{cccccccc} C_\ell & C_{\ell-1} & & C_4 & C_3 & C_2 & C_1 \end{array}$$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

consider odd indices: the sets are disjoint!

Analysis

$C_\ell \quad C_{\ell-1} \quad C_4 \quad C_3 \quad C_2 \quad C_1$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

consider odd indices: the sets are disjoint!

$\mathbb{P}(B_P) \leq \prod_{i=1}^{\ell/2} 2^{-|C_{2i-1}|}$

Analysis

$$\begin{array}{ccccccccccccccc} & C_\ell & & C_{\ell-1} & & & & C_4 & & C_3 & & C_2 & & C_1 & \end{array}$$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

consider odd indices: the sets are disjoint!

$\mathbb{P}(B_P) \leq \prod_{i=1}^{\ell/2} 2^{-|C_{2i-1}|}$

Assumption: Constant $\varepsilon \in (0,1)$ and function $C \mapsto f(C) \in (0,1)$ satisfy

$$2^{-|C|} \leq (1-\varepsilon)\frac{f(C)}{2|C|} \prod_{C' \sim C} (1 - f(C')), \quad \forall C.$$

Analysis

$C_\ell \quad C_{\ell-1} \quad C_4 \quad C_3 \quad C_2 \quad C_1$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

consider odd indices: the sets are disjoint!

$\mathbb{P}(B_P) \leq \prod_{i=1}^{\ell/2} 2^{-|C_{2i-1}|}$

Assumption: Constant $\varepsilon \in (0,1)$ and function $C \mapsto f(C) \in (0,1)$ satisfy

$$2^{-|C|} \leq (1-\varepsilon)\frac{f(C)}{2|C|} \prod_{C' \sim C} (1 - f(C')), \quad \forall C.$$

$\leq (1-\varepsilon)^{\ell/2} \prod_{i=1}^{\ell/2} \frac{f(C_{2i-1})}{2|C_{2i-1}|} \prod_{C' \sim C_{2i-1}} (1 - f(C'))$

reminds us of Galton-Watson branching process

Analysis

$$\begin{array}{ccccccccccccc} & C_\ell & & C_{\ell-1} & & & & C_4 & & C_3 & & C_2 & & C_1 \end{array}$$

Given induced path $P = (e_\ell \leftarrow e_{\ell-1} \leftarrow \cdots \leftarrow e_4 \leftarrow e_3 \leftarrow e_2 \leftarrow e_1)$

consider odd indices: the sets are disjoint!

$\mathbb{P}(B_P) \leq \prod_{i=1}^{\ell/2} 2^{-|C_{2i-1}|}$

Assumption: Constant $\varepsilon \in (0,1)$ and function $C \mapsto f(C) \in (0,1)$ satisfy

$$2^{-|C|} \leq (1-\varepsilon)\frac{f(C)}{2|C|} \prod_{C' \sim C} (1 - f(C')), \quad \forall C.$$

$\leq (1-\varepsilon)^{\ell/2} \prod_{i=1}^{\ell/2} \frac{f(C_{2i-1})}{2|C_{2i-1}|} \prod_{C' \sim C_{2i-1}} (1 - f(C'))$

reminds us of Galton-Watson branching process

Sketch: Map path P to a labelled tree $\mathcal{T}_P$. Then the product corresponds to the probability that $\mathcal{T}_P$ is generated by a suitable G-W process. Hence $\sum_P \text{product}(P) \leq 1$.

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \leq c \cdot \frac{2^{k/2}}{k^{3/2}}$.

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \leq c \cdot \frac{2^{k/2}}{k^{3/2}}$.

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \le c \cdot \frac{2^{k/2}}{k^{3/2}}$.

Refinement: inductive path-extension argument.

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \leq c \cdot \frac{2^{k/2}}{k^{3/2}}$.

Refinement: inductive path-extension argument.

$$\exists P = (e_\ell \leftarrow \cdots \leftarrow e_1) : B_P \quad \Big\{$$

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \le c \cdot \frac{2^{k/2}}{k^{3/2}}$.

Refinement: inductive path-extension argument.

$$\exists P = (e_\ell \leftarrow \cdots \leftarrow e_1) : B_P \quad \begin{cases} \text{(i)} \quad \exists Q = (e_{\ell-1} \leftarrow \cdots \leftarrow e_1) : B_Q \\ \end{cases}$$

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \le c \cdot \frac{2^{k/2}}{k^{3/2}}$.

Refinement: inductive path-extension argument.

$$\exists P = (e_\ell \leftarrow \cdots \leftarrow e_1) : B_P \quad \begin{cases} \text{(i)} & \exists Q = (e_{\ell-1} \leftarrow \cdots \leftarrow e_1) : B_Q \\ \text{(ii)} & \exists e_\ell \text{ validly extends } Q \end{cases}$$

When Hypergraph Is k-Uniform and d-Regular

Take $f(C) := 1/(kd)^2$ for all C, then the constraint becomes $d \le c \cdot \frac{2^{k/2}}{k^{3/2}}$.

Refinement: inductive path-extension argument.

$$\exists P = (e_\ell \leftarrow \cdots \leftarrow e_1) : B_P \quad \begin{cases} \text{(i)} & \exists Q = (e_{\ell-1} \leftarrow \cdots \leftarrow e_1) : B_Q \\ \text{(ii)} & \exists e_\ell \text{ validly extends } Q \end{cases}$$

Key: If #extensions is large, then the “intersection” between e_ℓ and Q is small

thus $\mathbb{P}[\text{(ii)} \mid \text{(i)}]$ is small