Sampling and Counting Hypergraph Colourings

Poster Presenter: Weiming Feng[1]

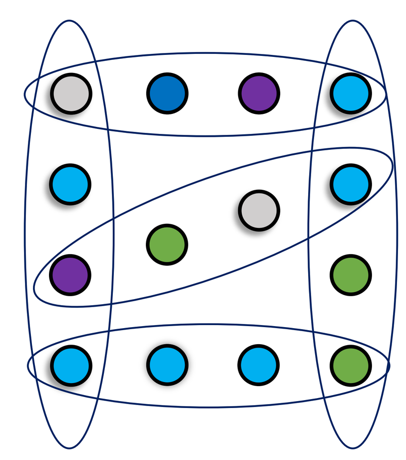
Based on joint works with Heng Guo[1], Kun He[2], Jiaheng Wang[1], Chunyang Wang[3] and Yitong Yin[3]

1 University of Edinburgh 2 Renmin University of China 3 Nanjing University

The Problem

Instance: colour set $[q] = \{1,2,\ldots,q\}$ and a hypergraph graph $H = (V, E)$

- number of vertices $n = |V|$;
- each edge contains k vertices;
- each vertex belongs to at most Δ edges.



Colouring: $X \in [q]^V$ s.t. no edge is monochromatic

Total number of colourings: Z

Uniform distribution over all colourings: μ

Construction: find an arbitrary colouring

Sampling: draw approximate sample X s.t. $\|X-\mu\|_{TV} \le \epsilon$

Randomised approximate counting: output $\hat{Z}$ s.t. $\Pr[(1-\epsilon)Z \le \hat{Z} \le (1+\epsilon)Z] \ge 2/3$

Deterministic approximate counting: output $\hat{Z}$ s.t. $(1-\epsilon)Z \le \hat{Z} \le (1+\epsilon)Z$

Our Results and Related Works

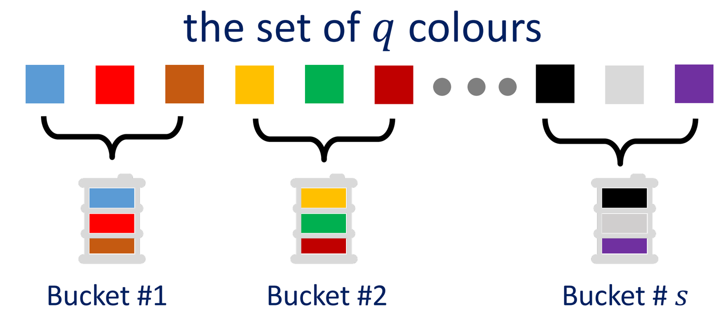
Problem	Work	Condition	Running Time
Construction	Moser Tardos 2009	$q \gtrsim \Delta^{1/k}$	$\mathrm{poly}(\Delta k)n$
Sampling Randomised Counting	F. He, Yin 2021 Jain, Pham, Vuong 2021	$q \gtrsim \Delta^{3/k}$	$\mathrm{poly}(\Delta k)\tilde{O}(n^{1.001})$
Deterministic Counting	Moitra2016 Guo, Liao, Lu, Zhang 2027 Jain, Pham, Vuong 2021	$q \gtrsim \Delta^{7/k}$	$n^{\mathrm{poly}(\Delta k)}$
Deterministic Counting	He, Yin, Wang 2022 F., Guo, Wang, Wang, Yin 2022	$q \gtrsim \Delta^{3/k}$	$n^{\mathrm{poly}(\Delta k)}$
Hardness for Sampling and Counting	Galanis, Guo, Wang 2022	$q \lesssim \Delta^{2/k}$	-

Technical challenges for sampling and approximate counting

- MCMC cannot be used directly as solution space is *disconnected* [Frizez,Melsted 2009].
- Correlation decay method [Weitz06] can not be used directly as *strong spatial mixing fails*.

Projection Technique for Sampling

the set of q colours



the set of s buckets

Projection **from colours to buckets**

$$h: [q] \to [s]$$

- number of buckets $s \approx q^{2/3}$
- for each bucket $b \in [s]$

$$|h^{-1}(c)| = \left\lfloor \frac{q}{s} \right\rfloor \ or \ \left\lceil \frac{q}{s} \right\rceil$$

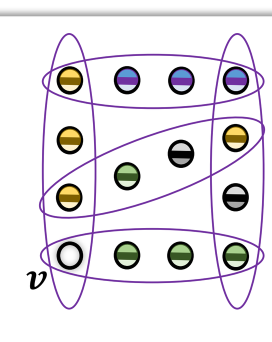
Projected distribution π over the configurations of the buckets $[s]^V$

$$h(X) = \left(h(X_v)\right)_{v \in V} \sim \pi, \qquad where\ X \sim \mu$$

- different colourings X, X' may be projected to the same state $h(X) = h(X')$;
  - **projection compresses space of colourings**
- π is not a Gibbs distribution (distribution defined by local interactions).

Run the *Systematic Scan* on π to draw an approximate sample $Y \in [s]^V$

> Start from a uniform random $Y \in [s]^V$
> **For** each t from 1 to $T = O(n \log n)$
> - Pick $v \in V$ with label $t \bmod n$
> - Resample $Y_v \sim \pi_v(\cdot \mid Y_{S \setminus v})$
>
> **Return** Y



Draw sample $X \sim \mu$ *conditional on* $h(X) = Y$

Properties of the above sampling algorithm

- Systematic scan on π is rapid mixing
  the projection makes ***a substantial compression,*** so the projected space is ***well-connected***.
- The algorithm can be implemented efficiently: fast sampling for *conditional distributions* of π
  the projection ***does not compress too much,*** so π is "similar" to a Gibbs distribution

The Local Uniformity Property

If $q \gtrsim \Delta^{3/k}$, the projected distribution π satisfies for all $v \in V$, all $\sigma \in [s]^{V-v}$

$$\forall b \in [s], \qquad \pi_v(b \mid \sigma) \in \left(1 \pm O\left(\frac{1}{s}\right)\right)\frac{|h^{-1}(b)|}{q} \approx \left(1 \pm O\left(\frac{1}{s}\right)\right)\frac{1}{s}.$$

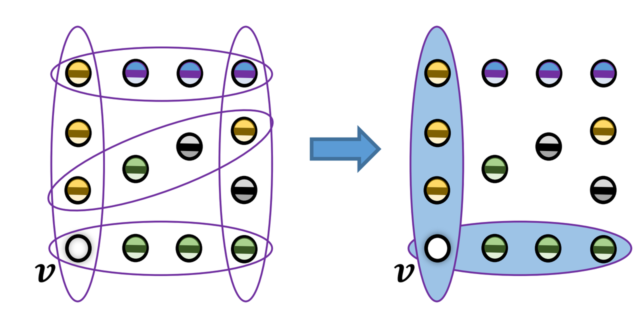
Intuition: π is ***"similar"*** to a ***product distribution***

local uniformity $\xrightarrow{\text{information percolation [Jain, Pham, Vuong, 2021]}}$ rapid mixing of systematic scan

Sampling from the conditional distribution $\pi_v(\cdot \mid \sigma)$, where $v \in V$ and $\sigma \in [s]^{V-v}$

- sample $X \sim \mu$ s.t. $h(X_u) = \sigma_u$ for $u \ne v$ (X is a uniform list colouring)
- return $Y_v = h(X_v)$

Observation: for any $e \in E$, if there exists $u, v \in e$ s.t. $\sigma_u \ne \sigma_v$, then e can be *removed*



Only sample list colouring in the ***component*** containing v

local uniformity property

each σ_u is an almost uniformly at random from $[s]$

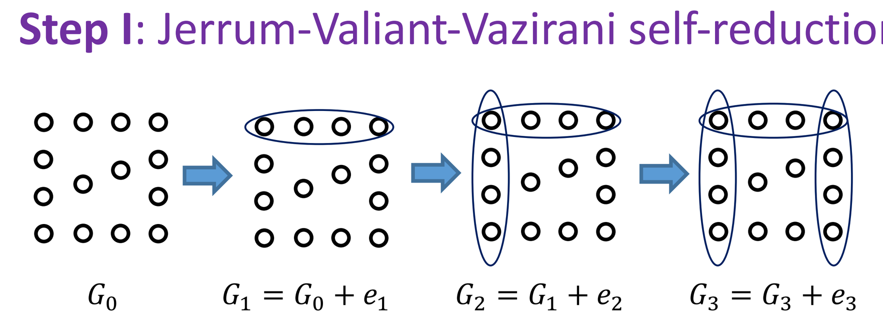
$$\Pr[e \text{ is removed}] \approx s\left(\frac{1}{s}\right)^k = O\left(\frac{1}{d^2}\right)$$

with high probability, size of component is $O(\log n)$

sample list colouring via naïve rejection sampling

Derandomisation Techinique for Counting

Step I: Jerrum-Valiant-Vazirani self-reduction



$Z_0 = q^n$ and approx. each $\frac{Z_{i+1}}{Z_i}$, where

$$\frac{Z_{i+1}}{Z_i} = \Pr_{\mu_i}[e_{i+1} \text{ is } \textbf{\textit{not}} \text{ monochromatic}]$$

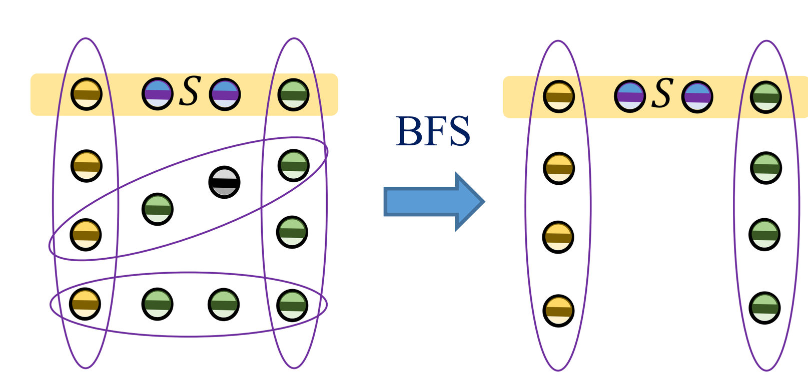
Remark: $e_{i+1} \notin G_i$

Abstract problem: given $G = (V, E)$, $[q]$ and $S \subseteq V$ with $|S| = O(1)$, approx. distribution μ_S.

Step II: Sampling from μ_S via sampling from marginal distributions of π

Input: $S \subseteq V$ and the ***access*** to a random sample $Y \sim \pi$ (query $v \in V$ and return $Y_v \in [s]$)

Output: a random sample $X_S \in [q]^S$ from μ_S



Use ***BFS*** to find components Λ s.t.

- $S \subseteq \Lambda$
- each component in Λ is monochromatic in Y

the BFS only reveal ***local value*** Y_v *around* S

Use brute-force algorithm on the list colouring in $G[\Lambda]$ with colour lists $h^{-1}(Y_\Lambda)$ to sample μ_S

Step III: providing local access to huge random object via coupling towards the past

Systematic Scan on π

For $t = -\infty$ to 0

- Pick the vertex v with label $t \bmod n$
- Sample a random value $r_t \sim \pi_{LB}$
- If $r_t \ne \perp$, then let $Y_v \leftarrow r_t$
- If $r_t = \perp$, then
  - Compute $\pi_v^{Y_{V-v}}$ by a local BFS
  - $Y_v \leftarrow p_t \sim \pi_v^{\mathrm{pad}, Y_{V-v}}$

Local Uniformity: $\forall \sigma \in [s]^{V-v}, c \in [s]$,

$$\pi_v^\sigma(c) \ge p_{LB} \approx (1 - O(1/s))1/s$$

$$\forall c \in [s], \pi_{LB}(c) = p_{LB} \text{ and } \pi_{LB}(\perp) = 1 - sp_{LB}$$

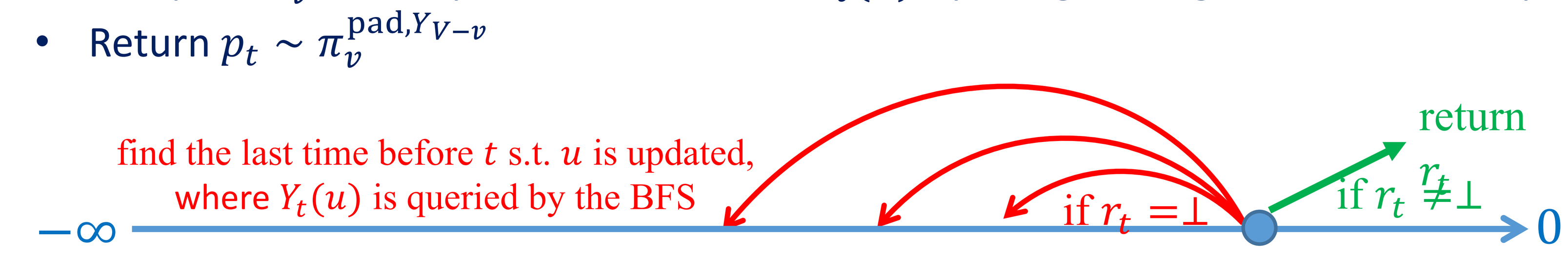
(guess the value from local uniformity)

$$\forall c \in [s], \qquad \pi_v^{\mathrm{pad}, Y_{V-v}}(c) = \frac{\pi_v^{Y_{V-v}}(c) - p_{LB}}{1 - sp_{LB}}$$

(sample from padding distribution if guess fails)

Coupling towards the past for sampling π_v

- Let $(Y_t)_{-\infty}^0$ be the systematic scan on π and $Y_0 \sim \pi$
- Find the last time $t < 0$ s.t. v is picked
- Reveal the value of $r_t \sim \pi_{LB}$
- If $r_t \ne \perp$, then return r_t
- If $r_t \ne \perp$, then
  - Compute $\pi_v^{Y_t(V-v)}$ by a local BFS, access $Y_t(u)$ by using this algorithm recursively
  - Return $p_t \sim \pi_v^{\mathrm{pad}, Y_{V-v}}$



If $q \gtrsim \Delta^{3/k}$, with probability at least $1 - 1/\mathrm{poly}(n)$

the algorithm sample r_t, p_t for $\mathrm{poly}(\Delta k) \log n$ times, and the running time is $n^{\mathrm{poly}(\Delta k)}$

Step IV: brute-force derandomisation by enumerating all possible values of r_t and p_t