Sampling & Counting *for* Big Data

南京大学
尹一通

2019年全国理论计算机科学学术年会
2019年8月3日于兰州大学

Sampling *vs* Counting

[Jerrum-Valiant-Vazirani '86]: for all *self-reducible* problems

$\left\{ \begin{matrix} \text{exact} \\ \text{approx} \end{matrix} \right\}$ sampling

$X = (X_1, X_2, \ldots, X_n)$

$X \sim \Omega$

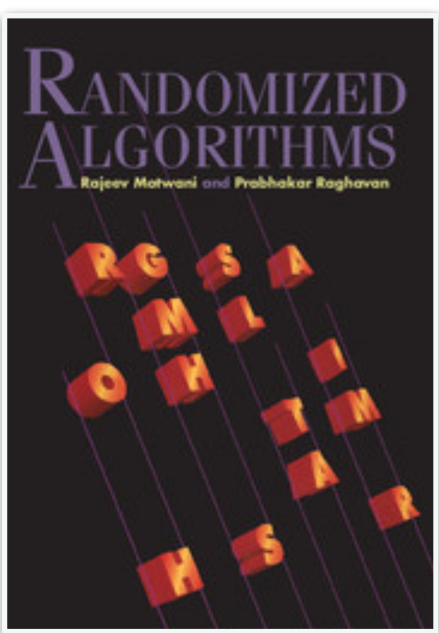
⟺ **Poly-Time Turing Machine**

approx counting

$\mathrm{vol}(\Omega)$

approx inference

$\Pr[X_i = \cdot \mid X_S = \sigma]$



RANDOM GENERATION OF COMBINATORIAL STRUCTURES FROM A UNIFORM DISTRIBUTION

Mark R. JERRUM
Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom

Leslie G. VALIANT *
Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, U.S.A.

Vijay V. VAZIRANI **
Computer Science Department, Cornell University, Ithaca, NY 14853, U.S.A.

MCMC Sampling

Markov chain for sampling $X = (X_1, X_2, \ldots, X_n) \sim \mu$

- Gibbs sampling (Glauber dynamics, heat-bath)

  > pick a random i;
  > resample $X_i \sim \mu_v(\cdot \mid N(v))$;

  [Glauber, '63]
  [Geman, Geman, '84]

- Metropolis-Hastings algorithm

  > pick a random i;
  > propose a random c;
  > $X_i = c$ w.p. $\propto \mu(X')/\mu(X)$;

  [Metropolis *et al*, '53]
  [Hastings, '84]

- Analysis: coupling methods

  [Aldous, '83] [Jerrum, '95] [Bubley, Dyer '97]

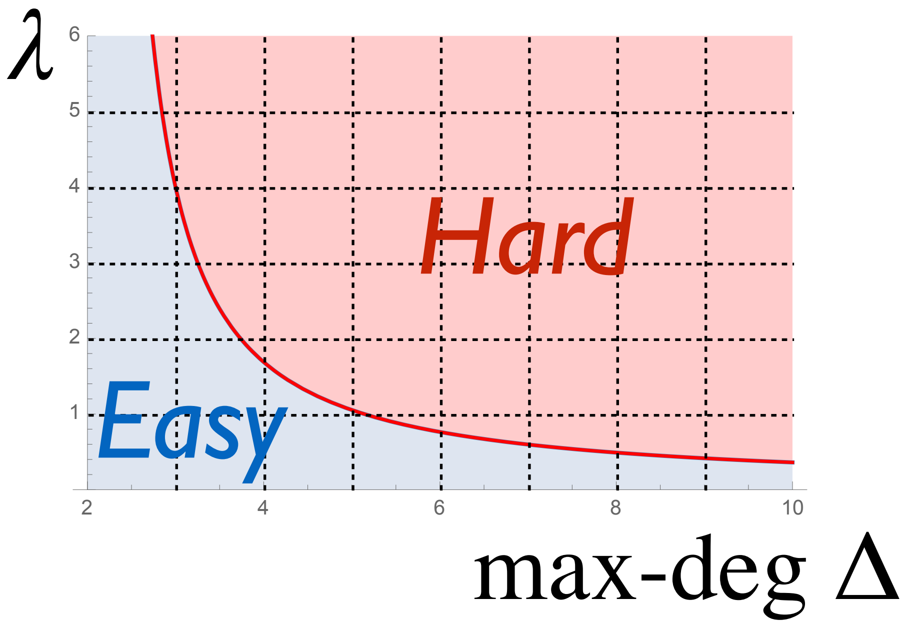
  may give $O(n \log n)$ upper bound for *mixing time*

Computational Phase Transition

hardcore model: graph $G(V,E)$, max-degree Δ, fugacity $\lambda>0$

approx sample independent set I in G w.p. $\propto \lambda^{|I|}$

$$\lambda_c(\Delta) = \frac{(\Delta-1)^{(\Delta-1)}}{(\Delta-2)^{\Delta}}$$



- [Weitz, **STOC**'06]: If $\lambda<\lambda_c$, $n^{O(\log \Delta)}$ time.
- [Sly, **FOCS**'10 best paper]: If $\lambda>\lambda_c$, **NP**-hard even for Δ=O(1).

[Efthymiou, Hayes, Štefankovič, Vigoda, Y., **FOCS**'16]:
If $\lambda<\lambda_c$, $O(n \log n)$ mixing time.
If Δ is large enough, and there is no small cycle.

A phase transition occurs at λ_c.

Sampling *vs* Counting

[Jerrum-Valiant-Vazirani '86]: for all *self-reducible* problems

{exact / approx} sampling

$X = (X_1, X_2, \ldots, X_n)$

$X \sim \Omega$

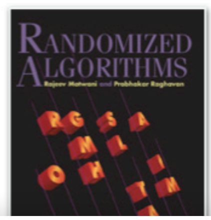
⟺ Poly-Time Turing Machine

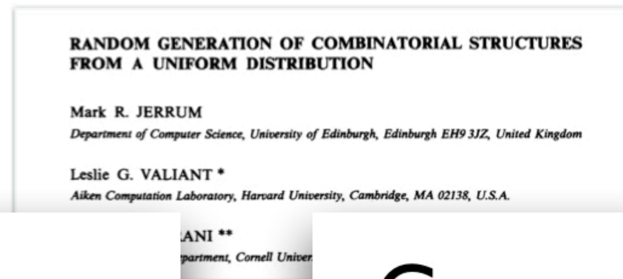
approx counting

$\mathrm{vol}(\Omega)$

approx inference

$\Pr[X_i = \cdot \mid X_S = \sigma]$





RANDOM GENERATION OF COMBINATORIAL STRUCTURES FROM A UNIFORM DISTRIBUTION

Mark R. JERRUM
Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom

Leslie G. VALIANT *
Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, U.S.A.

...ANI **
...partment, Cornell Univer...

MCMC Sampling

Markov chain for sampling $X = (X_1, X_2, \ldots, X_n) \sim \mu$

- Gibbs sampling (Glauber dynamics, heat-bath)

  pick a random i;
  resample $X_i \sim \mu_v(\cdot \mid N(v))$;

  [Glauber, '63]
  [Geman, Geman, '84]

- Metropolis-Hastings algorithm

  pick a random i;
  propose a random c;
  $X_i = c$ w.p. $\propto \mu(X')/\mu(X)$;

  [Metropolis *et al*, '53]
  [Hastings, '84]

- Analysis: coupling methods

  [Aldous, '83] [Jerrum, '95] [Bubley, Dyer '97]

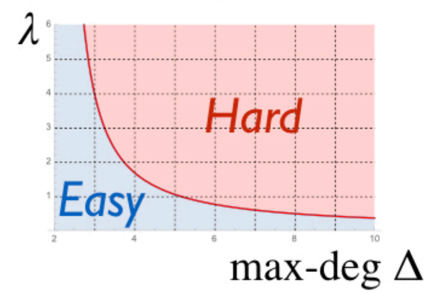
  may give $O(n \log n)$ upper bound for *mixing time*

Computational Phase Transition

hardcore model: graph $G(V,E)$, max-degree Δ, fugacity $\lambda>0$

approx sample independent set I in G w.p. $\propto \lambda^{|I|}$

$$\lambda_c(\Delta) = \frac{(\Delta-1)^{(\Delta-1)}}{(\Delta-2)^{\Delta}}$$



- [Weitz, STOC'06]: If $\lambda<\lambda_c$, $n^{O(\log \Delta)}$ time.
- [Sly, FOCS'10 best paper]: If $\lambda>\lambda_c$, **NP**-hard even for $\Delta=O(1)$.

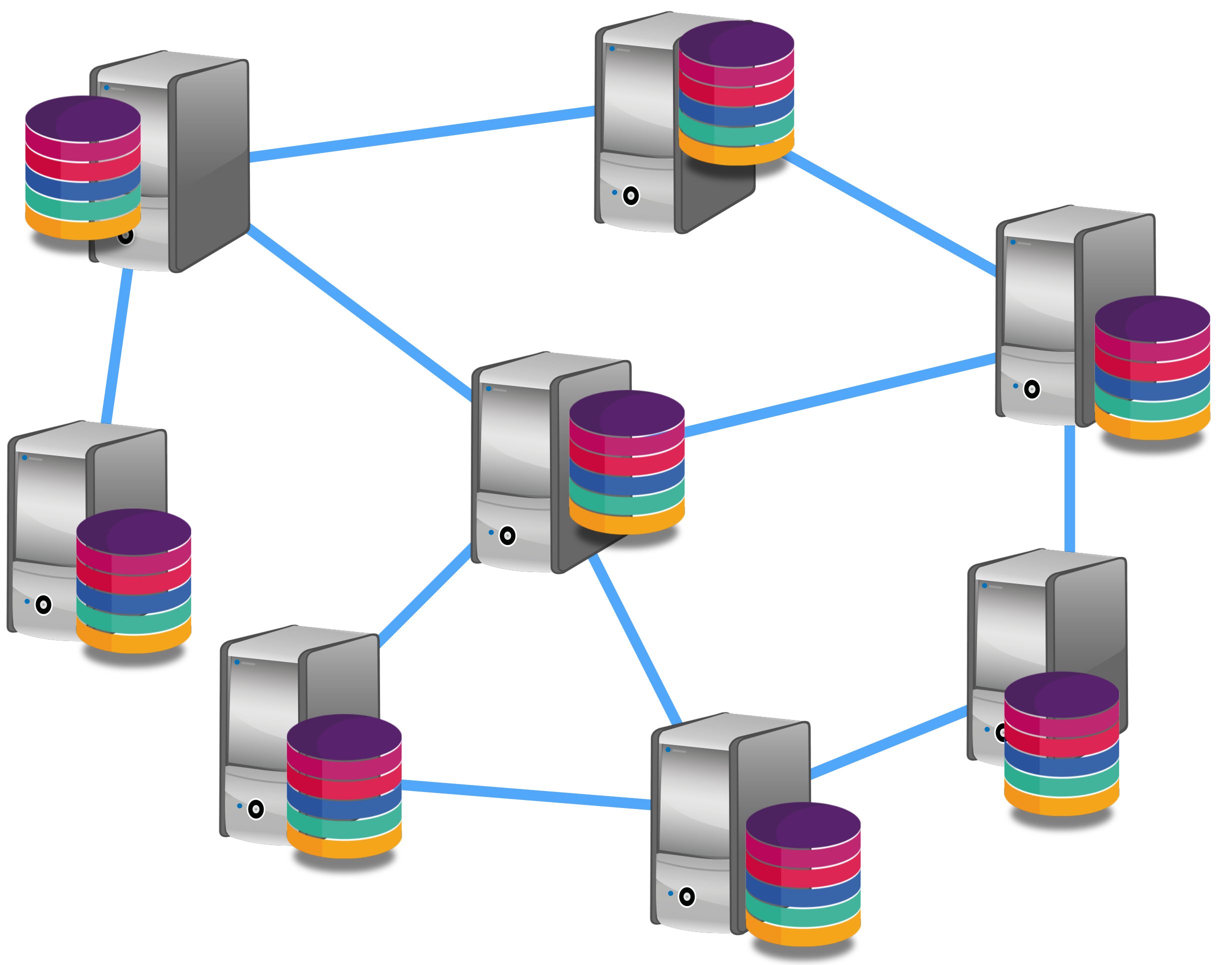
[Efthymiou, Hayes, Štefankovič, Vigoda, Y., FOCS'16]:
If $\lambda<\lambda_c$, $O(n \log n)$ mixing time.
If Δ is large enough, and there is no small cycle.

A phase transition occurs at λ_c.

Big Data?

Sampling and Inference for *Big Data*

- Sampling from a *joint distribution* (specified by a *probabilistic graphical model*).
- Inferring according to a *probabilistic graphical model*.
- The data (*probabilistic graphical model*) is BIG.



✓ • Parallel/distributed algorithms for sampling?

- PTIME $\Longrightarrow$ Polylog(n) *rounds*

✓ • For parallel/distributed computing: sampling ≡ approx counting/inference?

- PTIME $\Longrightarrow$ Polylog(n) *rounds*

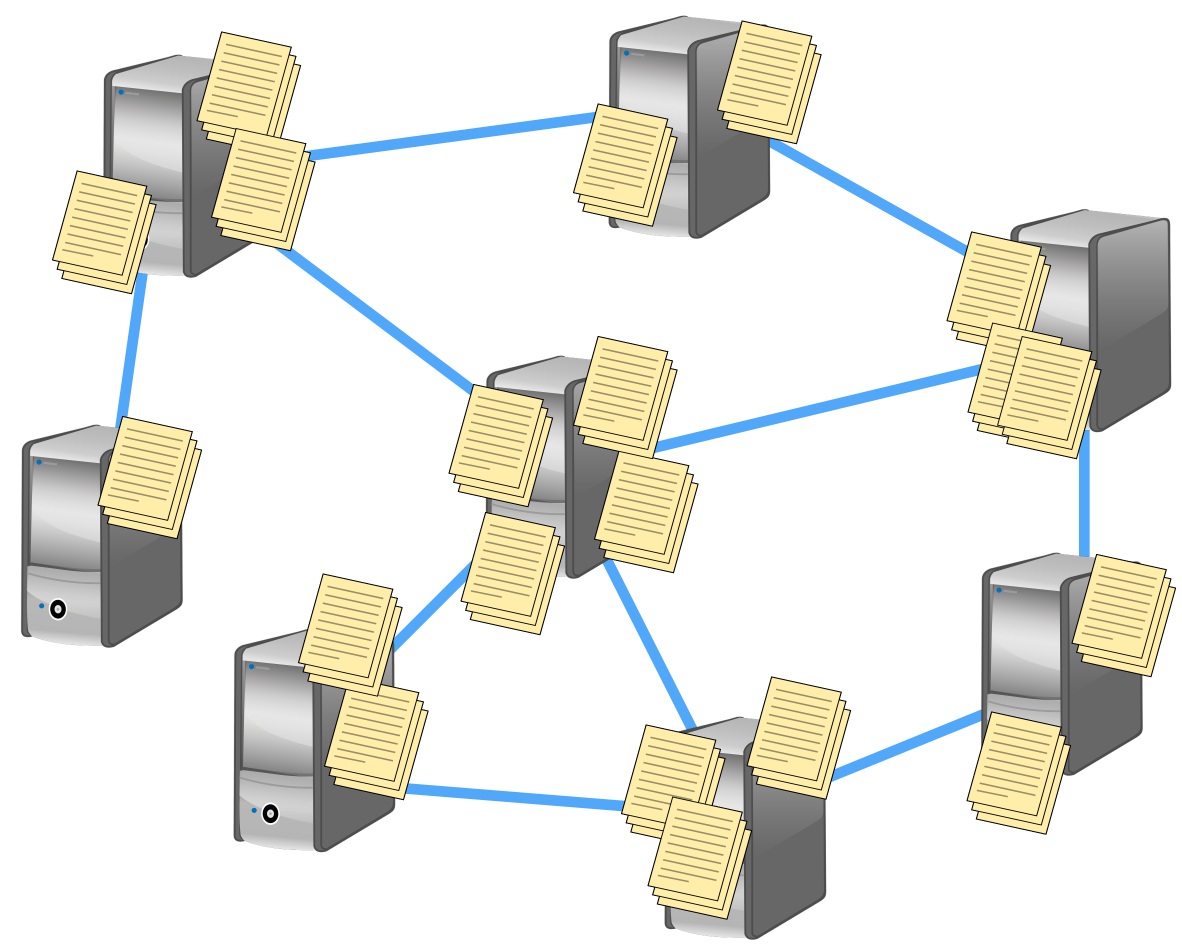
✓ • Dynamic sampling algorithms?

- PTIME $\Longrightarrow$ Polylog(n) *incremental* cost

Local Computation

> *"What can be computed locally?"*
> [Noar, Stockmeyer, STOC'93, SICOMP'95]

the *LOCAL* model [Linial '87]:



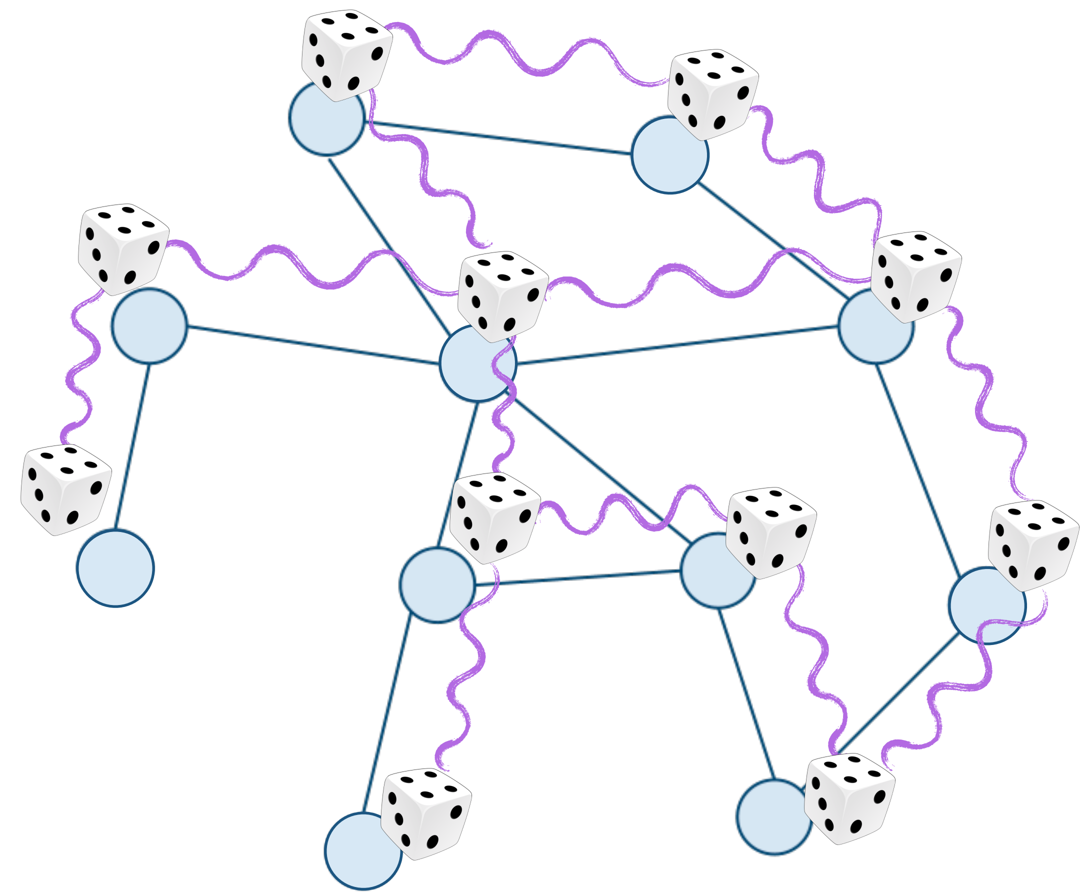
- Communications are synchronized.
- In each round: unlimited local computation and communication with neighbors.
- Complexity: # of rounds to terminate in the worst case.

- In t rounds: each node can collect information up to distance t.

PLOCAL: $t = \mathrm{polylog}(n)$

"What can be *sampled* locally?"

- Joint distribution defined by local constraints:
  - Markov random field
  - Graphical model
- Sample a random solution from the joint distribution:
  - distributed algorithms (in the $\mathcal{LOCAL}$ model)



network $G(V,E)$

Q: "What locally definable joint distributions are locally sample-able?"

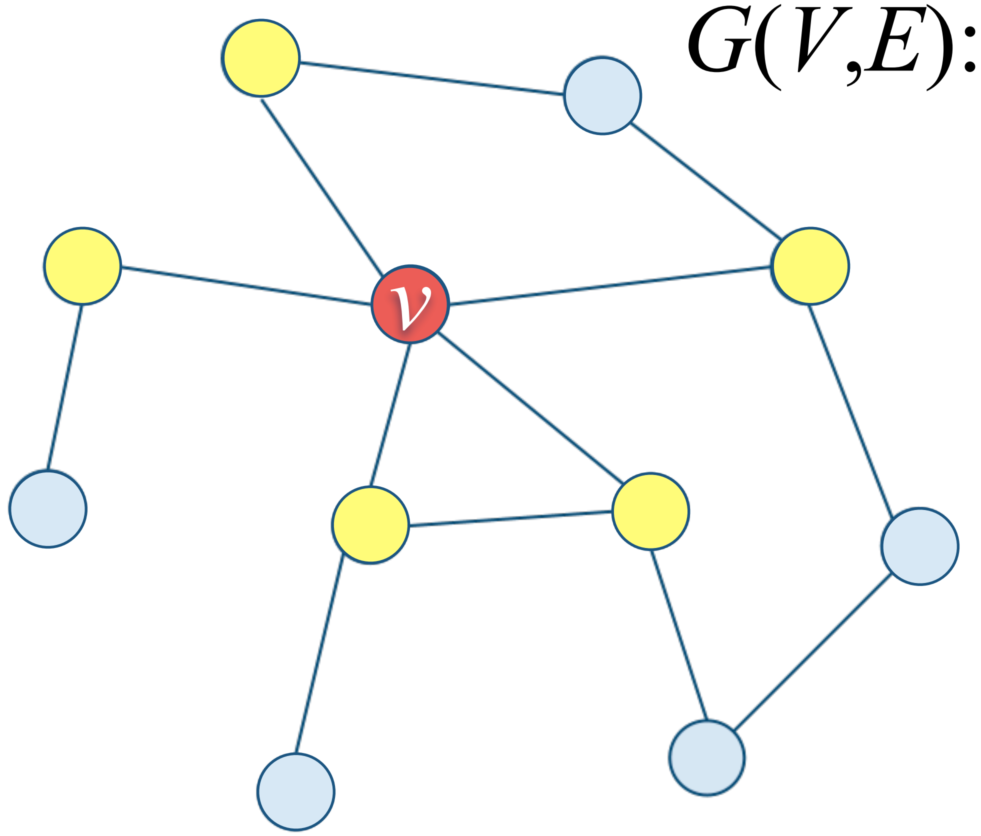
MCMC Sampling

Classic MCMC sampling:

Markov chain $X_t \to X_{t+1}$:

pick a uniform random vertex v;

update $X(v)$ conditioning on $X(N(v))$;



$O(n \log n)$ time when mixing

Parallelization:

- Chromatic scheduler [folklore] [Gonzalez *et al.*, AISTAT'11]: Vertices in the same color class are updated in parallel.
  - $O(\Delta \log n)$ mixing time (Δ is max degree)
- "Hogwild!" [Niu, Recht, Ré, Wright, NIPS'11][De Sa, Olukotun, Ré, ICML'16]: All vertices are updated in parallel, ignoring concurrency issues.
  - Wrong distribution!

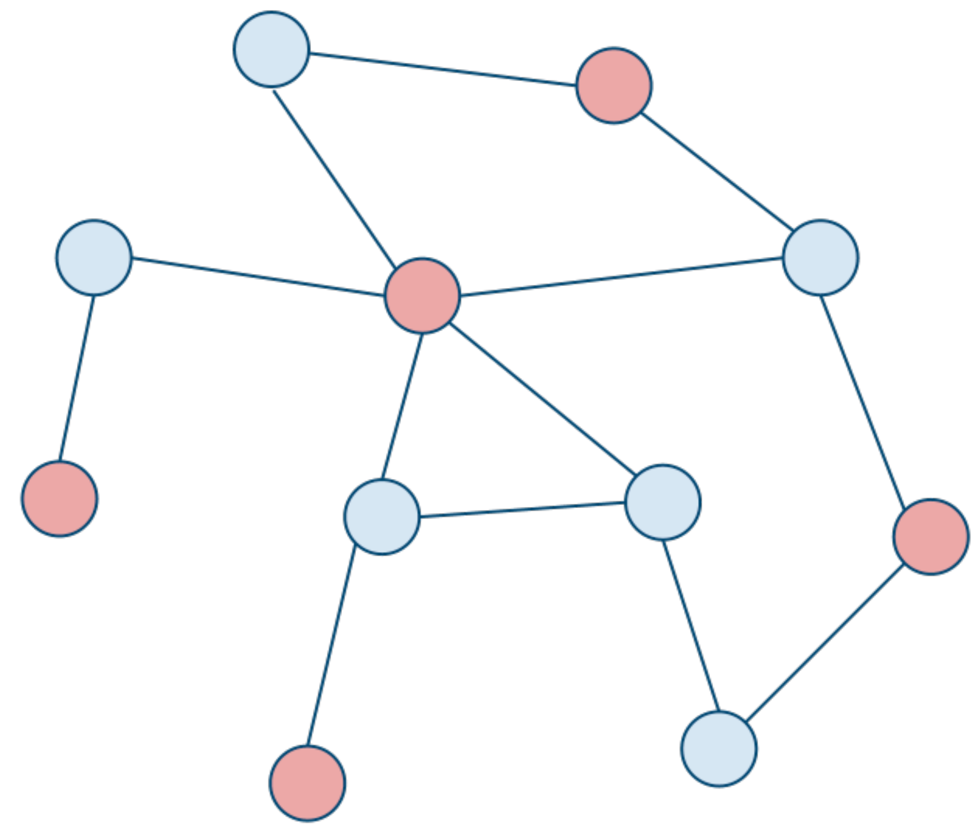
Crossing the Chromatic # Barrier

Sequential $O(n \log n)$ → **Parallel** $O(\Delta \log n)$

parallel speedup $= \theta(n / \Delta)$

Δ = max-degree

χ = chromatic no.



Do not update adjacent vertices simultaneously.

⇒ It takes $\geq \chi$ steps to update all vertices at least once.

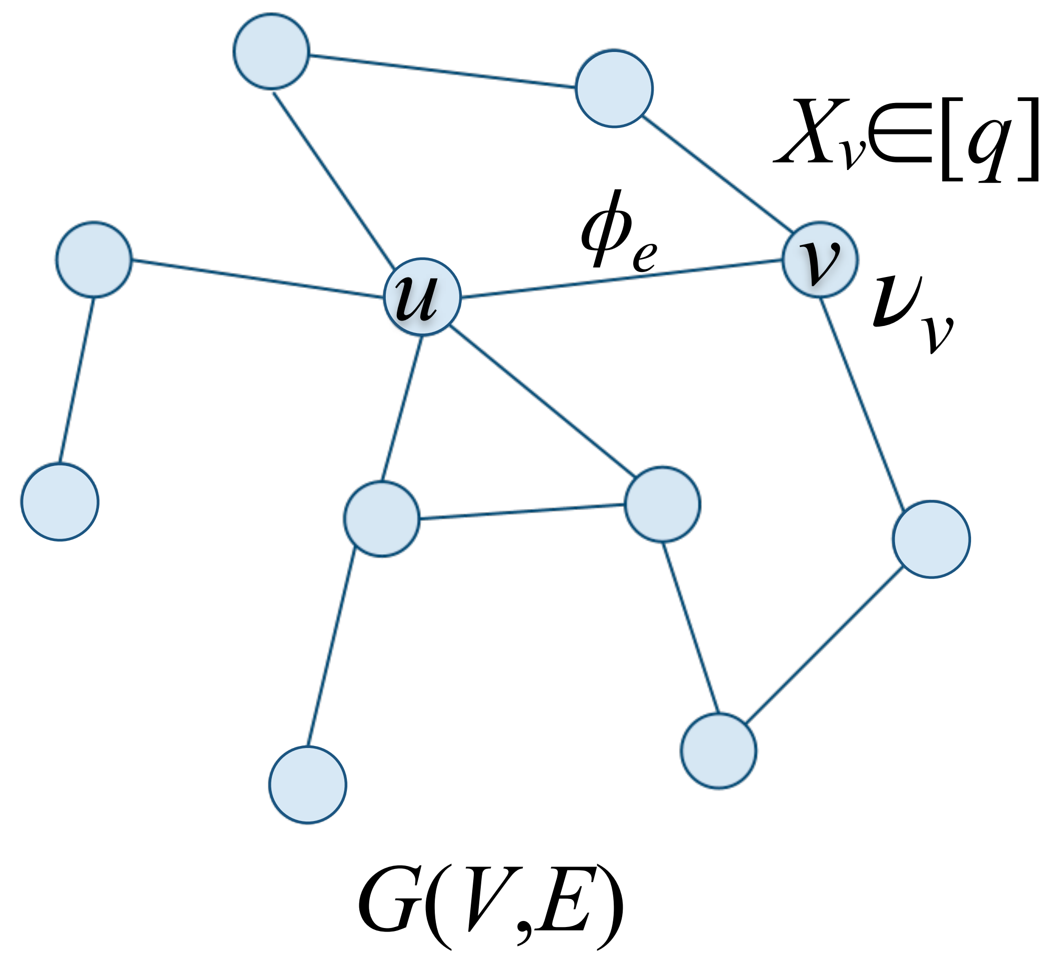
Q: "How to update all variables simultaneously and still converge to the correct distribution?"

Markov Random Fields
(MRF)

$$\forall \sigma \in [q]^V: \quad \mu(\sigma) \propto \prod_{v \in V} \nu_v(\sigma_v) \prod_{e=(u,v)\in E} \phi_e(\sigma_u, \sigma_v)$$

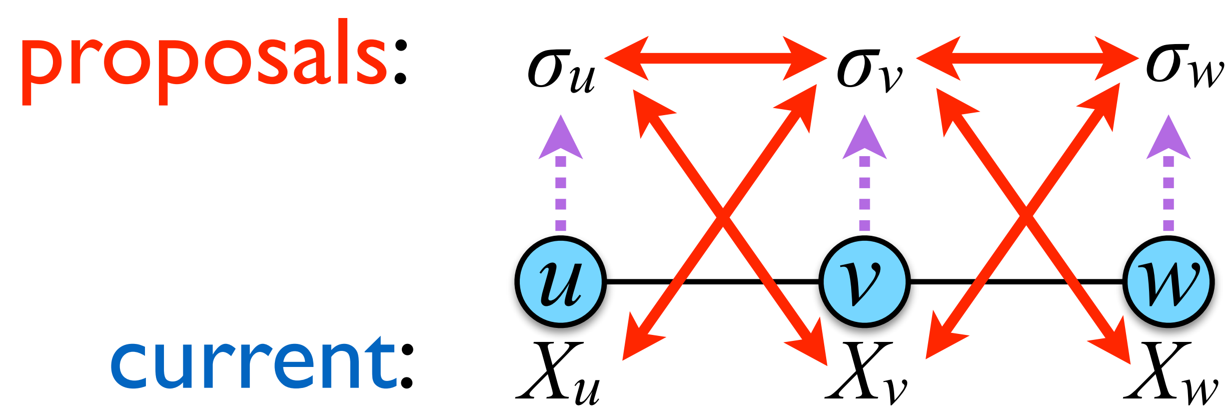
- Each vertex $v \in V$: a variable over domain $[q]$ with distribution ν_v
- Each edge $e=(u,v) \in E$: a symmetric binary constraint:

$$\phi_e : [q] \times [q] \to [0,1]$$



The *Local-Metropolis* Algorithm

[Feng, Sun, Y., *What can be sample locally*? **PODC**'17]



Markov chain $X_t \to X_{t+1}$:

each vertex $v \in V$ independently proposes a random $\sigma_v \sim \nu_v$;

each edge $e=(u,v)$ passes its check independently with prob:

$$\phi_e(X_u, \sigma_v) \cdot \phi_e(\sigma_u, X_v) \cdot \phi_e(\sigma_u, \sigma_v);$$

each vertex $v \in V$ update X_v to σ_v if *all its edges pass checks*;

- ***Local-Metropolis*** converges to the correct distribution μ.

The *Local-Metropolis* Algorithm

[Feng, Sun, Y., *What can be sample locally*? **PODC**'17]

each vertex $v \in V$ independently proposes a random $\sigma_v \sim \nu_v$;

each edge $e=(u,v)$ passes its check independently with prob:

$$\phi_e(X_u, \sigma_v) \cdot \phi_e(\sigma_u, X_v) \cdot \phi_e(\sigma_u, \sigma_v);$$

each vertex $v \in V$ update X_v to σ_v if *all its edges pass checks*;

- ***Local-Metropolis*** converges to the correct distribution μ.

MRF: $$\mu(\sigma) \propto \prod_{v \in V} \nu_v(\sigma_v) \prod_{e=(u,v) \in E} \phi_e(\sigma_u, \sigma_v)$$

- under coupling condition for *Metropolis-Hastings*:
  - *Metropolis-Hastings*: $O(n \log n)$ time
  - (lazy) ***Local-Metropolis***: $O(\log n)$ time

Lower Bounds

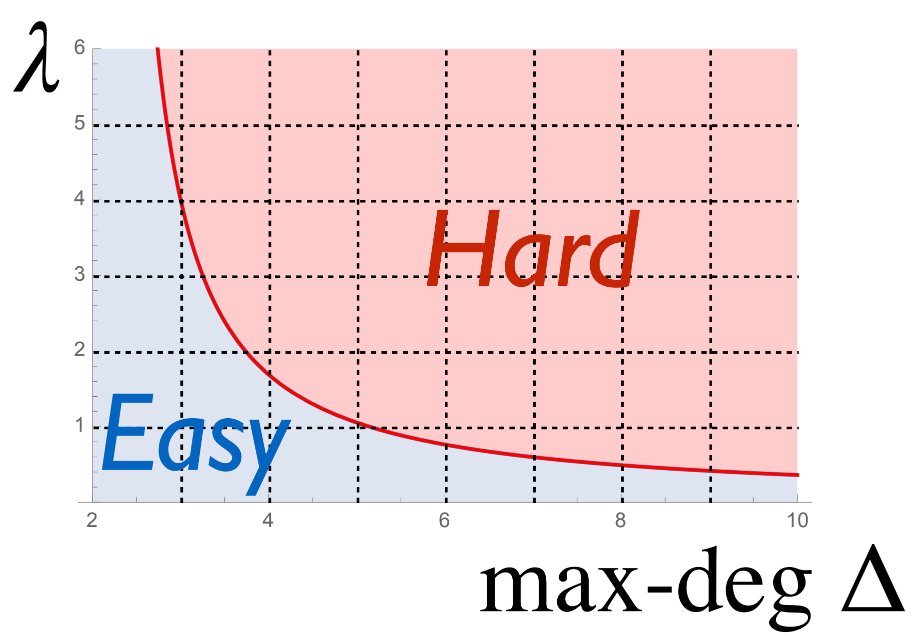
[Feng, Sun, Y., *What can be sample locally*? **PODC**'17]

Approx sampling from any MRF requires $\Omega(\log n)$ rounds.

- for sampling: $O(\log n)$ is the new criteria of "*local*"

If $\lambda > \lambda_c$, sampling from hardcore model requires $\Omega(diam)$ rounds.

$$\lambda_c(\Delta) = \frac{(\Delta-1)^{(\Delta-1)}}{(\Delta-2)^{\Delta}}$$



strong separation: *sampling* vs other local computation tasks

- Independent set is trivial to construct locally (e.g. ∅).
- The lower bound holds not because of the locality of information, but because of the locality of correlation.

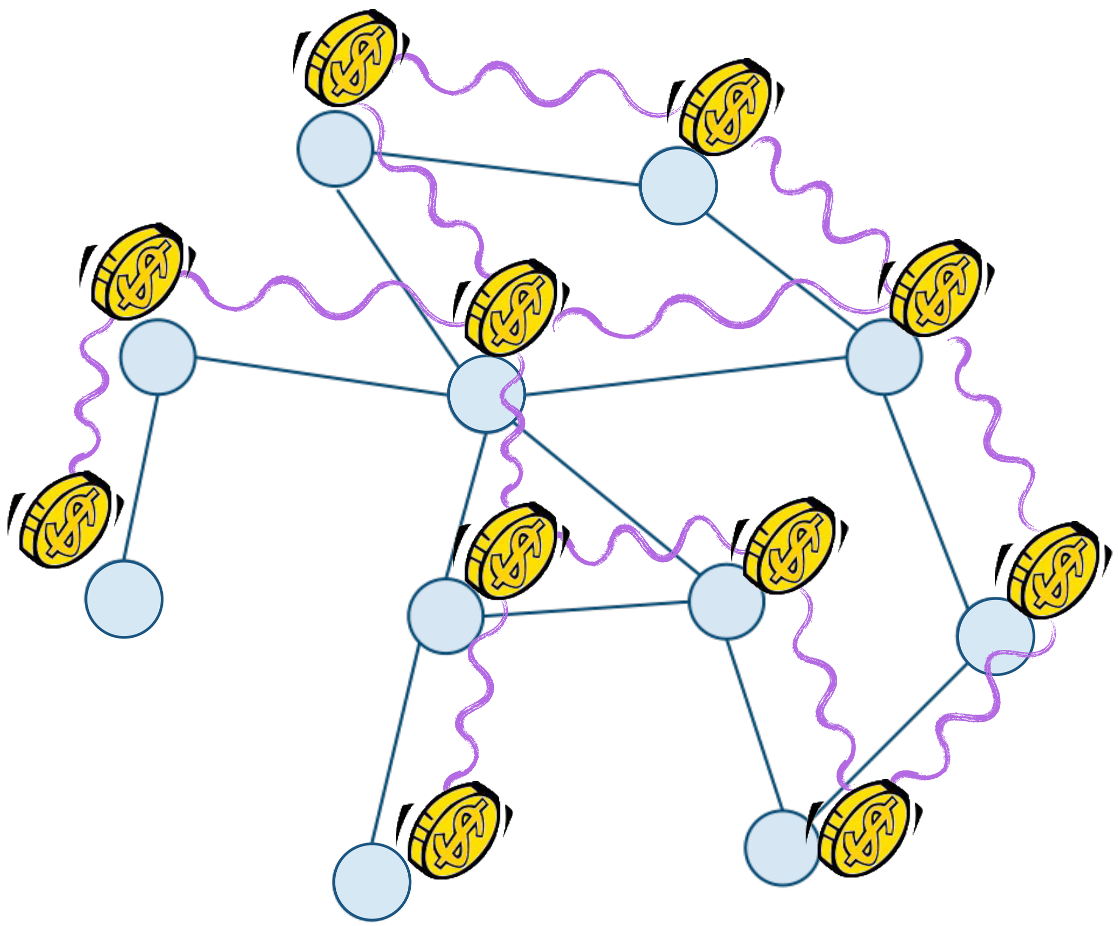
- ✓ Parallel/distributed algorithms for sampling?
  - PTIME $\Longrightarrow$ Polylog(n) *rounds*
- ✓ For parallel/distributed computing: sampling ≡ approx counting/inference?
  - PTIME $\Longrightarrow$ Polylog(n) *rounds*
- ✓ Dynamic sampling algorithms?
  - PTIME $\Longrightarrow$ Polylog(n) *incremental* cost

Example: Sample Independent Set

(hardcore model)

μ: distribution of independent sets I in G $\propto \lambda^{|I|}$

- $\boldsymbol{Y} \in \{0,1\}^V$ indicates an independent set
- Each $v \in V$ returns a $Y_v \in \{0,1\}$, such that $\boldsymbol{Y} = (Y_v)_{v \in V} \sim \mu$
- Or: $d_{\mathrm{TV}}(\boldsymbol{Y}, \mu) < 1/\mathrm{poly}(n)$



network $G(V,E)$

Inference (Local Counting)

μ: distribution of independent sets I in $G \propto \lambda^{|I|}$

μ_v^σ: marginal distribution at v conditioning on $\sigma \in \{0,1\}^S$.

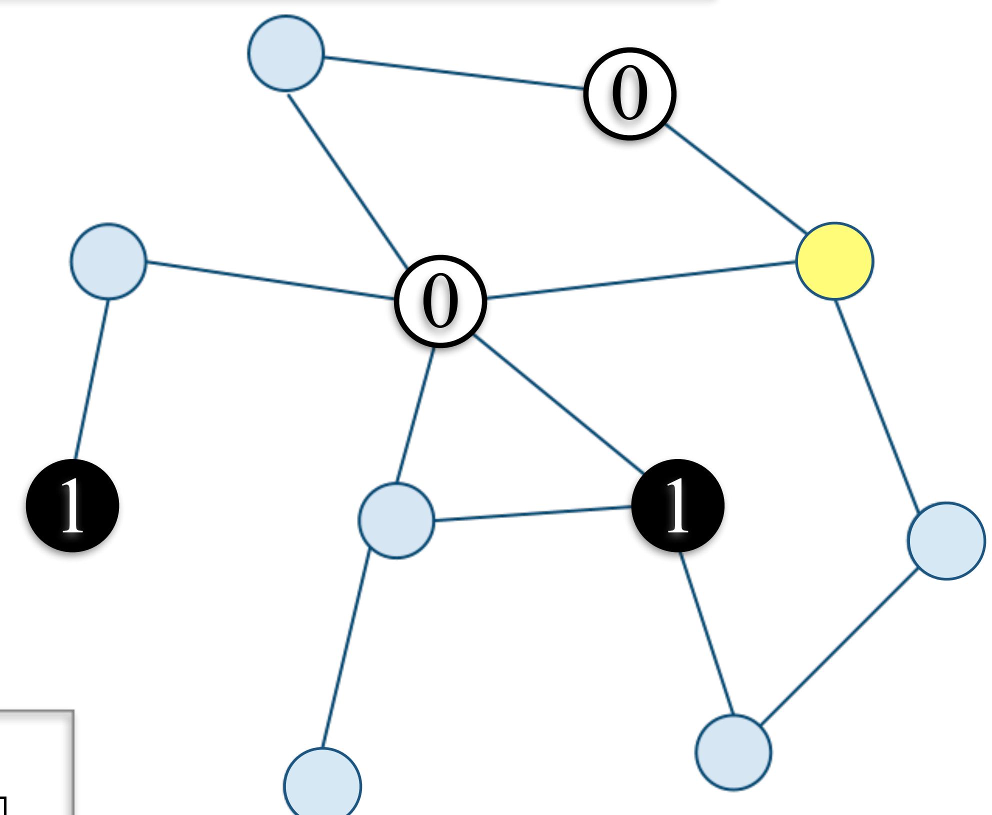
$$\forall y \in \{0,1\}: \quad \mu_v^\sigma(y) = \Pr_{\boldsymbol{Y}\sim\mu}[Y_v = y \mid Y_S = \sigma]$$

- Each $v \in S$ receives σ_v as input.
- Each $v \in V$ returns a marginal distribution $\hat{\mu}_v^\sigma$ such that:

$$d_{\mathrm{TV}}(\hat{\mu}_v^\sigma, \mu_v^\sigma) \le \frac{1}{\mathrm{poly}(n)}$$

$$\frac{1}{Z} = \mu(\emptyset) = \prod_{i=1}^{n} \Pr_{\boldsymbol{Y}\sim\mu}[Y_{v_i} = 0 \mid \forall j < i : Y_{v_j} = 0]$$

Z: partition function (counting)



network $G(V,E)$

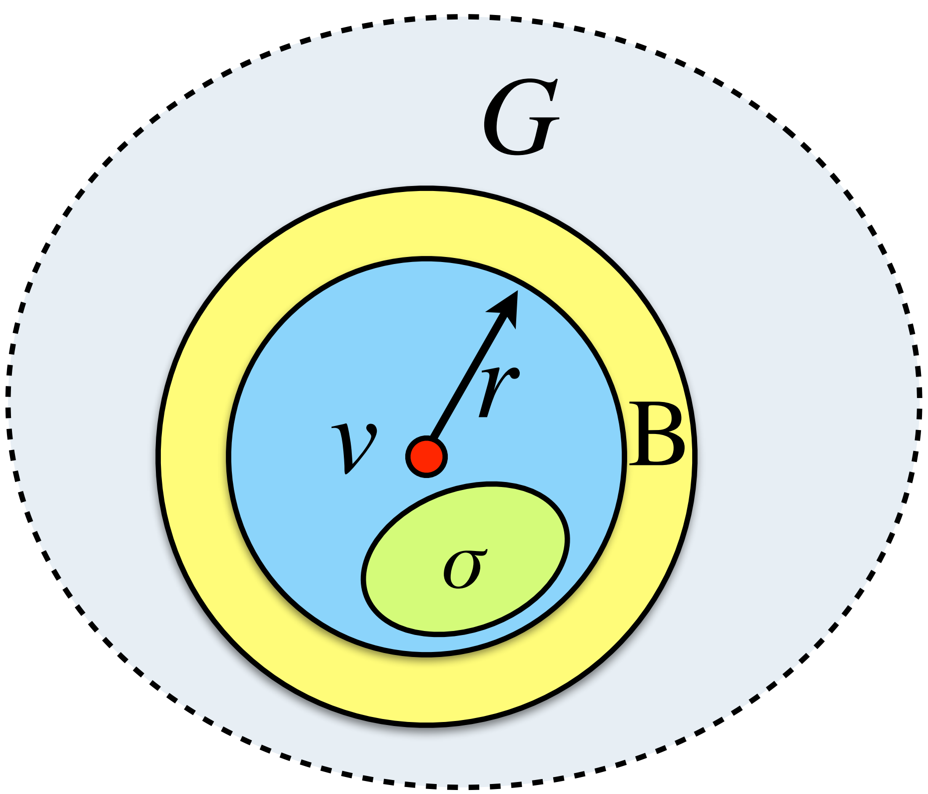
Decay of Correlation

μ_v^σ: marginal distribution at v conditioning on $\sigma \in \{0,1\}^S$.

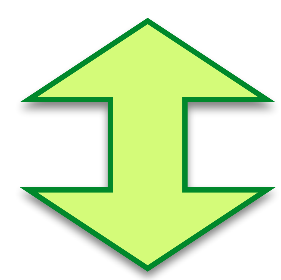
strong spatial mixing (SSM):

$\forall$ boundary condition $B \in \{0,1\}^{r\text{-sphere}(v)}$:

$$d_{\mathrm{TV}}(\mu_v^\sigma, \mu_v^{\sigma,B}) \le \mathrm{poly}(n) \cdot \exp(-\Omega(r))$$



SSM (iff $\lambda \le \lambda_c$ when μ is the hardcore model)

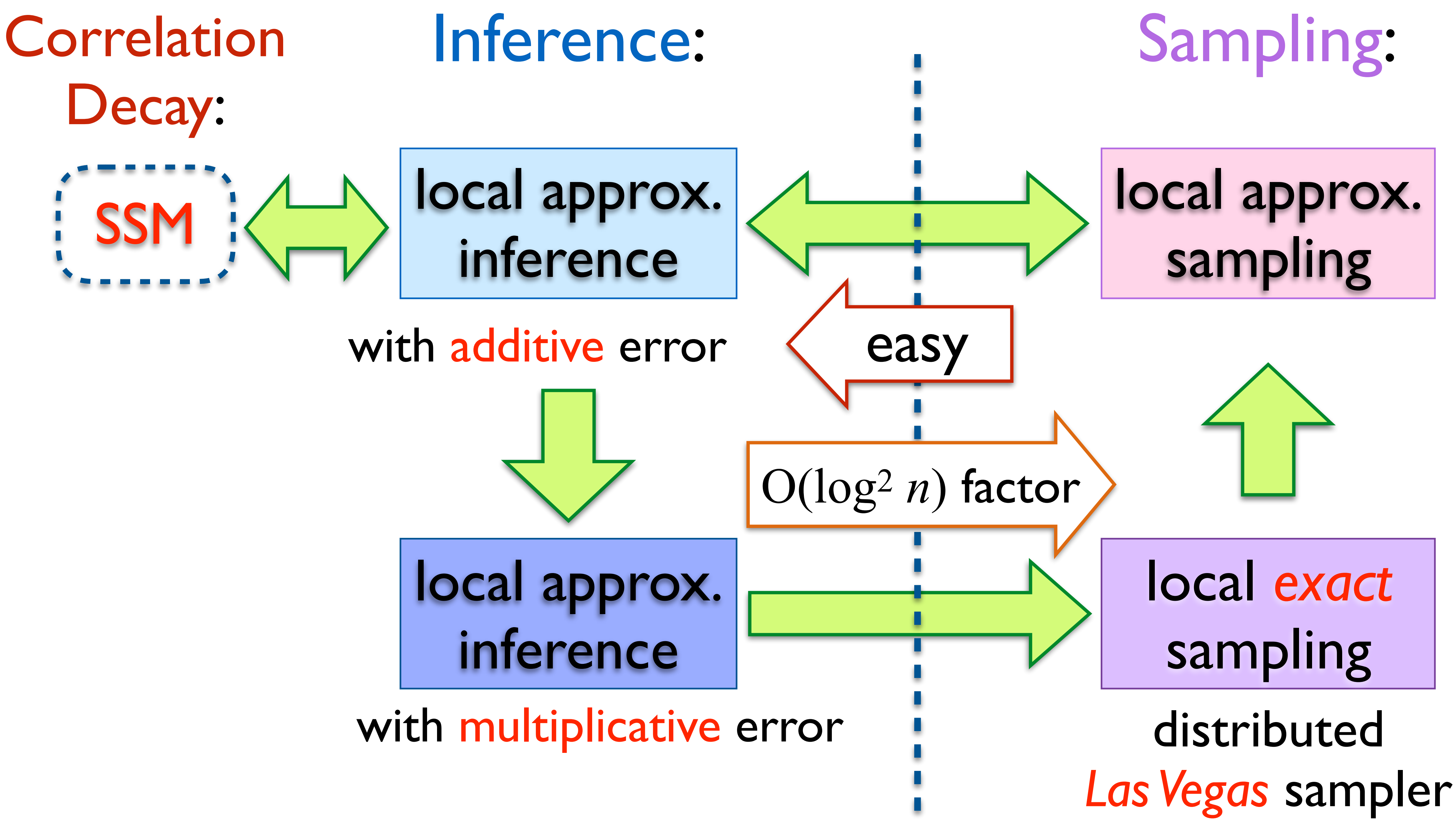


approx. inference is solvable in O(log n) rounds in the $\mathcal{LOCAL}$ model

Locality of Counting & Sampling

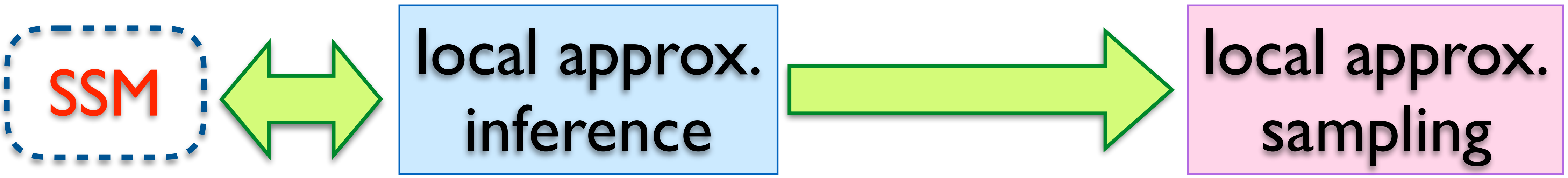
[Feng, Y., **PODC'18**]

For all *self-reducible* graphical models:



Locality of Sampling

Correlation Decay: | **Inference:** | **Sampling:**



Inference: each v can compute a $\hat{\mu}_v^\sigma$ within $O(\log n)$-ball s.t. $d_{\mathrm{TV}}\left(\hat{\mu}_v^\sigma, \mu_v^\sigma\right) \le \frac{1}{\mathrm{poly}(n)}$

Sampling: return a random $\boldsymbol{Y} = (Y_v)_{v \in V}$ whose distribution $\hat{\mu} \approx \mu$, $d_{\mathrm{TV}}\left(\hat{\mu}, \mu\right) \le \frac{1}{\mathrm{poly}(n)}$

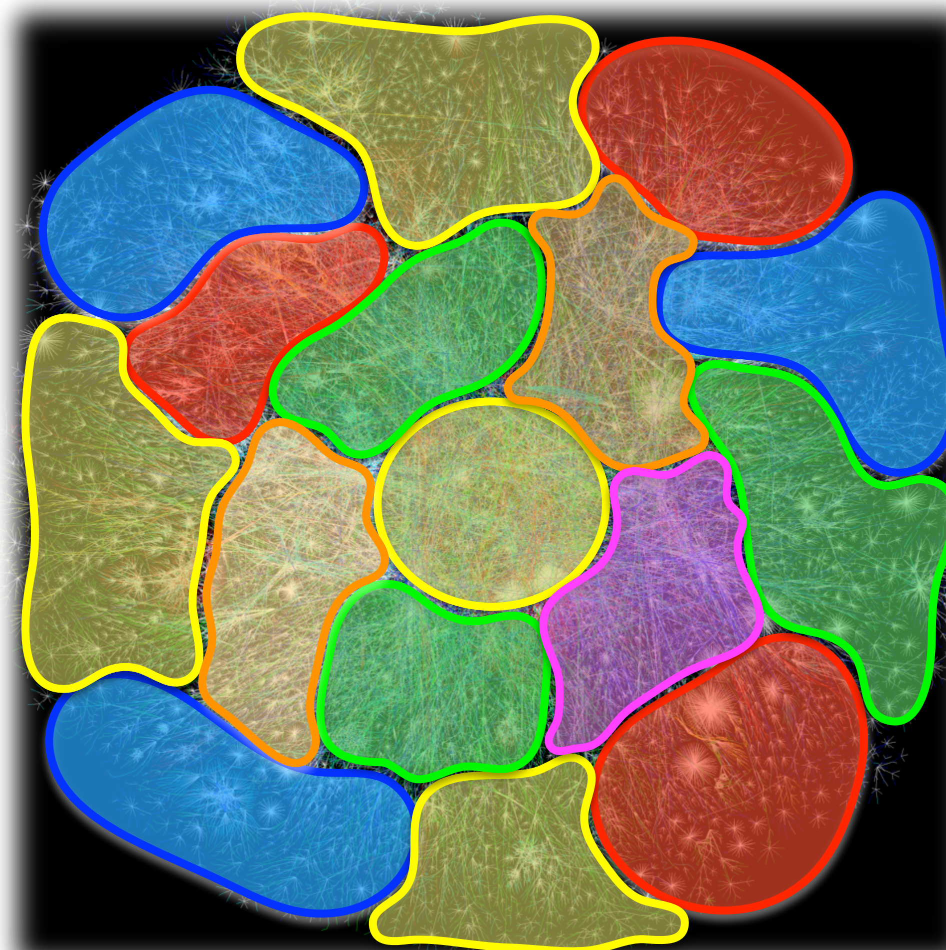
sequential $O(\log n)$-local procedure:

- scan vertices in V in an arbitrary order $v_1, v_2, \ldots, v_n$
- for $i=1,2,\ldots,n$: sample Y_{v_i} according to $\hat{\mu}_{v_i}^{Y_{v_1},\ldots,Y_{v_{i-1}}}$

Network Decomposition

(C,D) -network-decomposition of G:

- classifies vertices into clusters;
- assign each cluster a color in $[C]$;
- each cluster has diameter $\leq D$;
- clusters are properly colored.



$(C,D)^r$-ND: (C,D)-ND of G^r

Given a $(C,D)^r$- ND:

sequential r-local procedure: $r = O(\log n)$

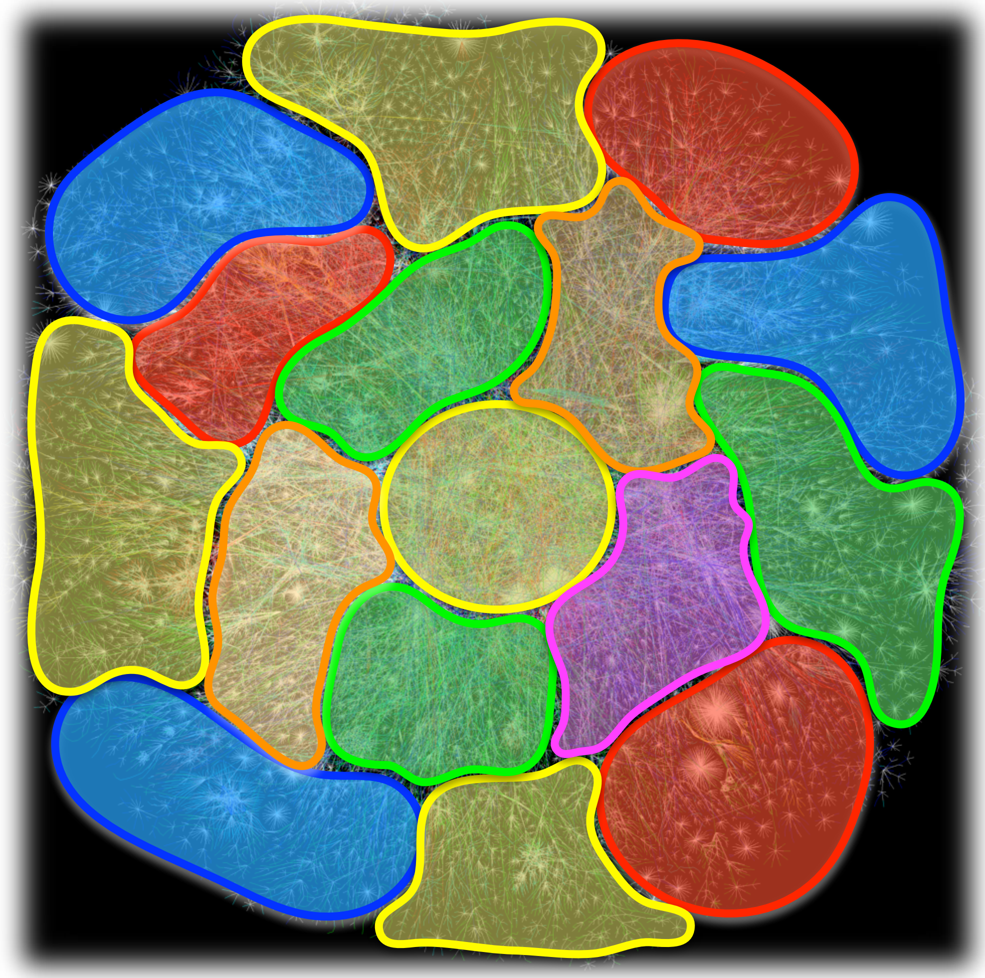
- scan vertices in V in an arbitrary order $v_1, v_2, \ldots, v_n$
- for i=1,2, …, n: sample Y_{v_i} according to $\hat{\mu}_{v_i}^{Y_{v_1},\ldots,Y_{v_{i-1}}}$

can be simulated in $O(CDr)$ rounds in $\mathcal{LOCAL}$ model

Network Decomposition

(C,D)-network-decomposition of G:

- classifies vertices into clusters;
- assign each cluster a color in $[C]$;
- each cluster has diameter $\leq D$;
- clusters are properly colored.



$(C,D)^r$-ND: (C,D)-ND of G^r

$(O(\log n), O(\log n))^r$-ND can be constructed in $O(r \log^2 n)$ rounds *w.h.p.*

[Ghaffari, Kuhn, Maus, STOC'17]:

r-local SLOCAL algorithm:
$\forall$ ordering $\pi=(v_1, v_2, \ldots, v_n)$,
returns random vector $Y^{(\pi)}$

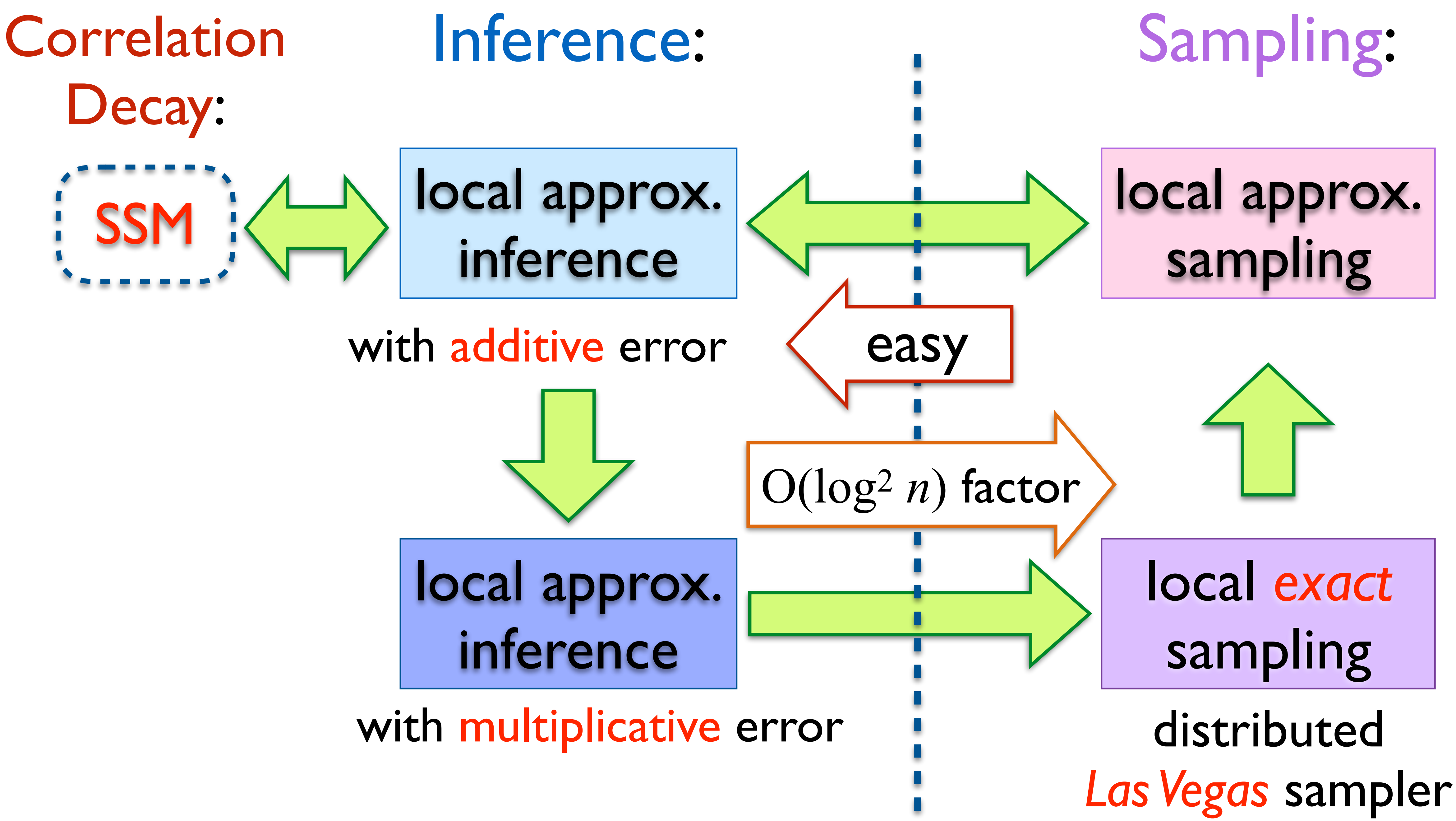
ND ⇒

$O(r\log^2 n)$-round LOCAL alg.:
returns *w.h.p.* the $Y^{(\pi)}$
for some ordering π

Locality of Counting & Sampling

[Feng, Y., **PODC'18**]

For all *self-reducible* graphical models:



Boosting Local Inference

SSM → local approx. inference

each v computes a $\hat{\mu}_v^\sigma$ within r-ball

additive error:

$$d_{\mathrm{TV}}\left(\hat{\mu}_v^\sigma, \mu_v^\sigma\right) \le \frac{1}{\mathrm{poly}(n)}$$

multiplicative error:

$$\frac{\hat{\mu}_v^\sigma(0)}{\mu_v^\sigma(0)}, \frac{\hat{\mu}_v^\sigma(1)}{\mu_v^\sigma(1)} \in \left[\mathrm{e}^{-1/\mathrm{poly}(n)}, \mathrm{e}^{1/\mathrm{poly}(n)}\right]$$

SSM —*local self-reduction*→ both are achievable with $r = O(\log n)$

boosted sequential r-local sampler: $r = O(\log n)$

- scan vertices in V in an arbitrary order $v_1, v_2, \ldots, v_n$
- for i=1,2, …, n: sample Y_{v_i} according to $\hat{\mu}_{v_i}^{Y_{v_1},\ldots,Y_{v_{i-1}}}$

multiplicative error: $\forall \sigma \in \{0,1\}^V : \ \mathrm{e}^{-1/n^2} \le \frac{\hat{\mu}(\sigma)}{\mu(\sigma)} \le \mathrm{e}^{1/n^2}$

SLOCAL JVV

Scan vertices in V in an arbitrary order $v_1, v_2, \ldots, v_n$:

pass 1: sample $\boldsymbol{Y} \in \{0,1\}^V$ by *boosted* sequential r-local sampler $\hat{\mu}$;

$$\forall \sigma \in [q]^V : \quad e^{-1/n^2} \le \frac{\hat{\mu}(\sigma)}{\mu(\sigma)} \le e^{1/n^2}$$

$r = O(\log n)$

pass 1': construct a sequence of ind. sets $\emptyset = \boldsymbol{Y}_0, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n = \boldsymbol{Y}$;

s.t. $\forall\ 0 \le i \le n$:
- $\boldsymbol{Y}_i$ agrees with $\boldsymbol{Y}$ over $v_1, \ldots, v_i$
- $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_{i-1}$ differ only at v_i

v_i samples $F_{v_i} \in \{0,1\}$ independently with $\Pr[F_{v_i} = 0] = q_{v_i}$

where $q_{v_i} = \dfrac{\hat{\mu}(\boldsymbol{Y}_{i-1})}{\hat{\mu}(\boldsymbol{Y}_i)} \cdot e^{-3/n^2} \in [e^{-5/n^2}, 1]$

$O(\log n)$-local to compute

Each $v \in V$ returns:

- $Y_v \in \{0,1\}$ to indicate the ind. set;
- $F_v \in \{0,1\}$ indicate failure at v.

Scan vertices in V in an arbitrary order $v_1, v_2, \ldots, v_n$:

pass 1: sample $\boldsymbol{Y} \in \{0,1\}^V$ by *boosted* sequential r-local sampler $\hat{\mu}$;

$$\forall \sigma \in [q]^V: \quad e^{-1/n^2} \le \frac{\hat{\mu}(\sigma)}{\mu(\sigma)} \le e^{1/n^2}$$

$r = O(\log n)$

pass 1': construct a sequence of ind. sets $\emptyset = \boldsymbol{Y}_0, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n = \boldsymbol{Y}$;

s.t. $\forall\ 0 \le i \le n$:
- $\boldsymbol{Y}_i$ agrees with $\boldsymbol{Y}$ over $v_1, \ldots, v_i$
- $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_{i-1}$ differ only at v_i

v_i samples $F_{v_i} \in \{0,1\}$ independently with $\Pr[F_{v_i} = 0] = q_{v_i}$

where $q_{v_i} = \dfrac{\hat{\mu}(\boldsymbol{Y}_{i-1})}{\hat{\mu}(\boldsymbol{Y}_i)} \cdot e^{-3/n^2} \in [e^{-5/n^2}, 1]$

$\forall \sigma \in \{0,1\}^V$:

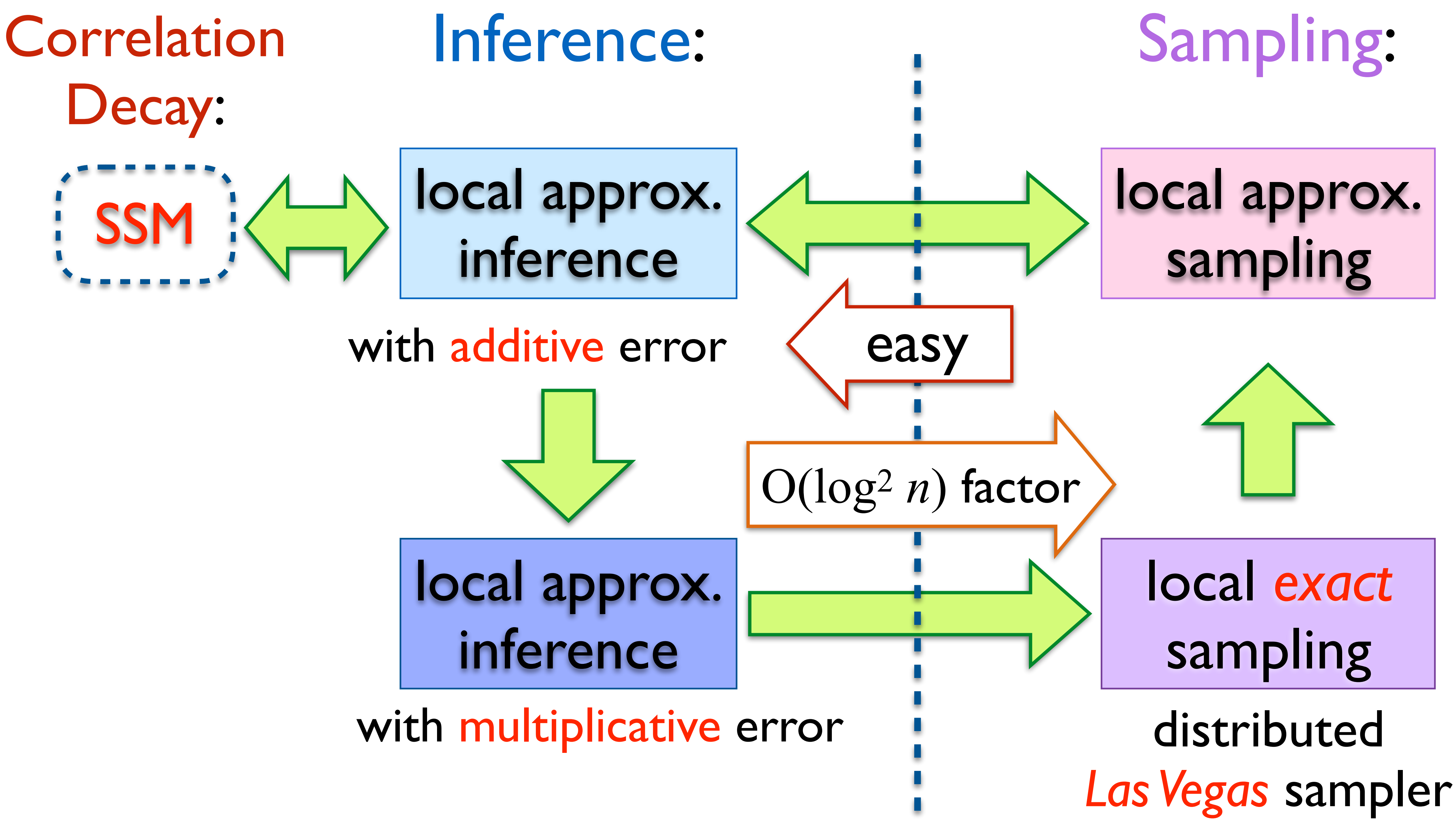
$$\Pr[\boldsymbol{Y} = \sigma \wedge \forall i : F_{v_i} = 0] = \hat{\mu}(\sigma) \prod_{i=1}^{n} q_{v_i} = \hat{\mu}(\sigma) \prod_{i=1}^{n} \left(\frac{\hat{\mu}(\boldsymbol{Y}_{i-1})}{\hat{\mu}(\boldsymbol{Y}_i)} \cdot e^{-3/n^2} \right) \Bigg|_{\boldsymbol{Y}_n = \boldsymbol{Y} = \sigma}$$

$$= \hat{\mu}(\sigma) \cdot \frac{\hat{\mu}(\emptyset)}{\hat{\mu}(\sigma)} \cdot e^{-\frac{3}{n}} \quad \propto \begin{cases} \lambda^{\|\sigma\|_1} & \sigma \text{ is ind. set} \\ 0 & \text{otherwise} \end{cases}$$

Locality of Counting & Sampling

[Feng, Y., **PODC'18**]

For all *self-reducible* graphical models:



Local Exact Sampler

hardcore model: distribution of independent sets $I \propto \lambda^{|I|}$

[Feng, Y., **PODC**'18]:

If $\lambda < \lambda_c(\Delta) = \frac{(\Delta-1)^{\Delta-1}}{(\Delta-2)^{\Delta}}$:

- strong spatial mixing holds [Weitz '06];
- $\exists$ $O(\log^3 n)$-round distributed Las Vegas sampler.

[Feng, Sun, Y., **PODC**'17]:

If $\lambda > \lambda_c$, any approx sampler requires $\Omega(\mathit{diam})$ rounds.

Sampling *vs* Counting

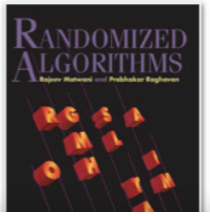
[Jerrum-Valiant-Vazirani '86]: for all *self-reducible* problems

{exact / approx} sampling
$X = (X_1, X_2, \ldots, X_n)$
$X \sim \Omega$

⟺ **Poly-Time Turing Machine**

approx counting
$\mathrm{vol}(\Omega)$

approx inference
$\Pr[X_i = \cdot \mid X_S = \sigma]$



RANDOM GENERATION OF COMBINATORIAL STRUCTURES FROM A UNIFORM DISTRIBUTION

Mark R. JERRUM
Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom

Leslie G. VALIANT *
Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, U.S.A.

...ANI **
...partment, Cornell Univer...

MCMC Sampling

Markov chain for sampling $X = (X_1, X_2, \ldots, X_n) \sim \mu$

- Gibbs sampling (Glauber dynamics, heat-bath)

  pick a random i;
  resample $X_i \sim \mu_v(\cdot \mid N(v))$;

  [Glauber, '63]
  [Geman, Geman, '84]

- Metropolis-Hastings algorithm

  pick a random i;
  propose a random c;
  $X_i = c$ w.p. $\propto \mu(X')/\mu(X)$;

  [Metropolis *et al*, '53]
  [Hastings, '84]

- Analysis: coupling methods

  [Aldous, '83] [Jerrum, '95] [Bubley, Dyer '97]

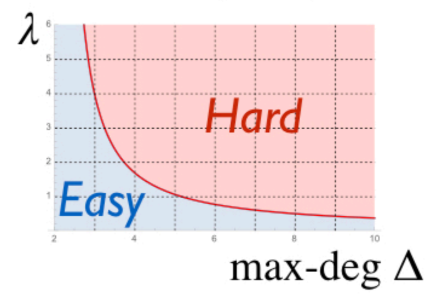
  may give $O(n \log n)$ upper bound for *mixing time*

Computational Phase Transition

hardcore model: graph $G(V,E)$, max-degree Δ, fugacity $\lambda>0$

approx sample independent set I in G w.p. $\propto \lambda^{|I|}$

$$\lambda_c(\Delta) = \frac{(\Delta-1)^{(\Delta-1)}}{(\Delta-2)^{\Delta}}$$



- [Weitz, STOC'06]: If $\lambda<\lambda_c$, $n^{O(\log \Delta)}$ time.
- [Sly, FOCS'10 best paper]: If $\lambda>\lambda_c$, **NP**-hard even for $\Delta=O(1)$.

[Efthymiou, Hayes, Štefankovič, Vigoda, Y., FOCS'16]:
If $\lambda<\lambda_c$, $O(n \log n)$ mixing time.
If Δ is large enough, and there is no small cycle.

A phase transition occurs at λ_c.

Hold for Big Data (local computation)!

Distributed Las Vegas Sampler

Las Vegas (certifiable failure):

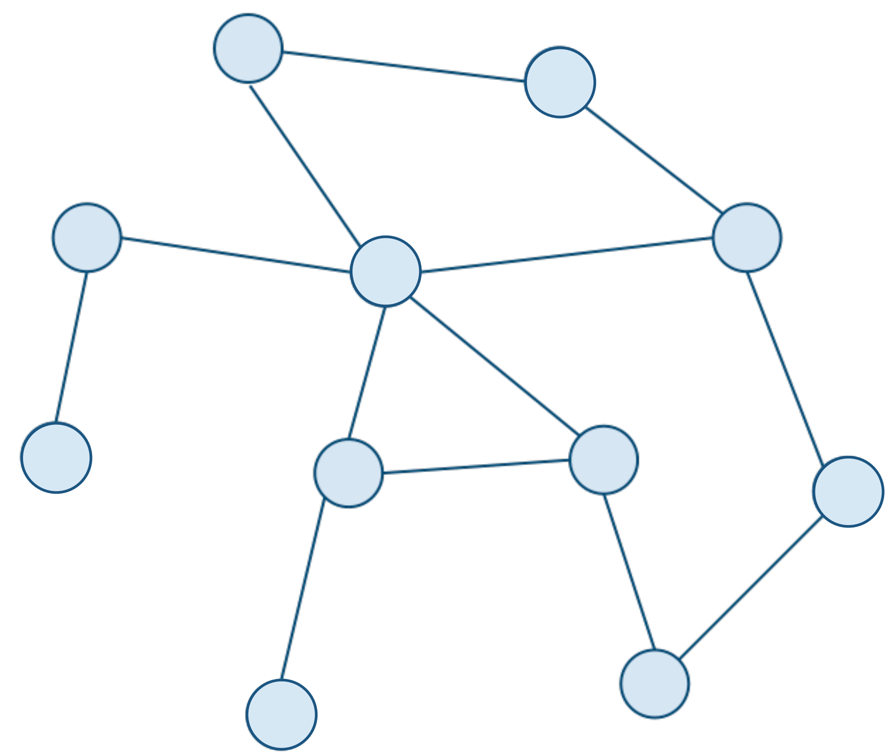
- Each $v \in V$ returns in **fixed** rounds:
  - local output $Y_v \in \{0,1\}$;
  - local failure $F_v \in \{0,1\}$.
- Succeeds w.h.p.: $\sum_{v \in V} \mathbf{E}[F_v] = o(1)$.
- Conditioning on success, $\boldsymbol{Y} \sim \mu$.

?

✓ dynamic sampler

Las Vegas (zero failure):

- Each $v \in V$ returns in **random** rounds:
  - local output $Y_v \in \{0,1\}$.
- Correctness: $\boldsymbol{Y} \sim \mu$.



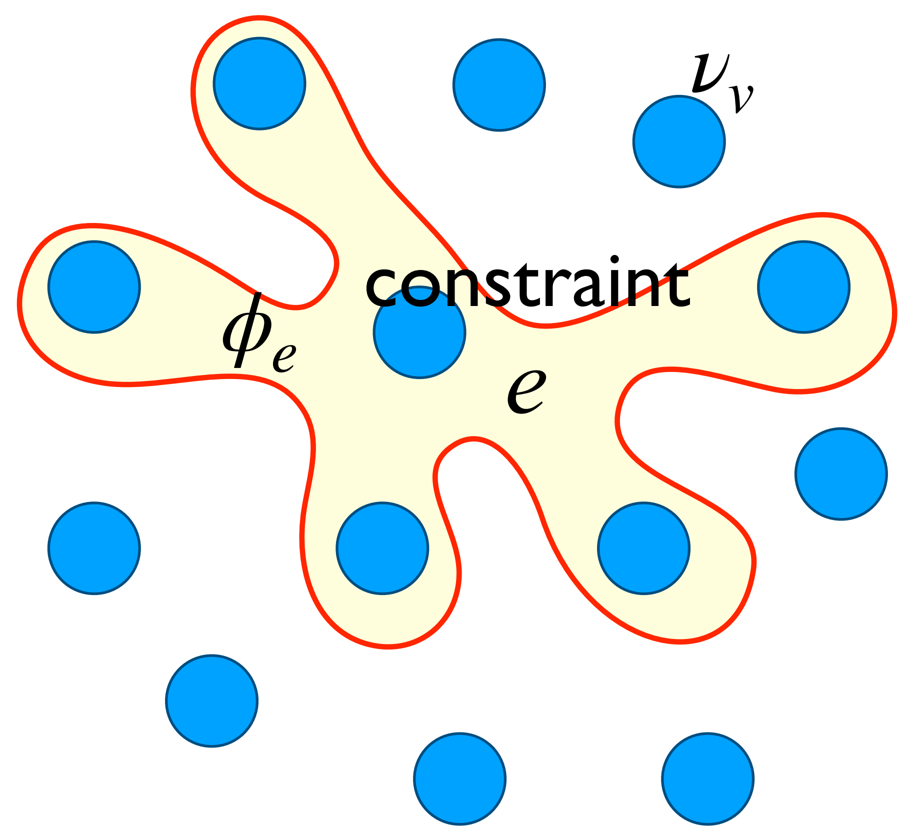
- ✓ Parallel/distributed algorithms for sampling?
  - PTIME $\Longrightarrow$ Polylog(n) *rounds*
- ✓ For parallel/distributed computing: sampling ≡ approx counting/inference?
  - PTIME $\Longrightarrow$ Polylog(n) *rounds*
- ✓ Dynamic sampling algorithms?
  - PTIME $\Longrightarrow$ Polylog(n) *incremental* cost

Graphical Model

$$\forall \sigma \in [q]^V: \quad \mu(\sigma) \propto \prod_{v\in V} \nu_v(\sigma_v) \prod_{e\in E} \phi_e(\sigma_e)$$

- Each $v\in V$: a *variable* with domain $[q]$ following distribution ν_v
- Each $e\in E$ is a set of variables and corresponds to a constraint (factor)

$$\phi_e : [q]^e \to [0,1]$$



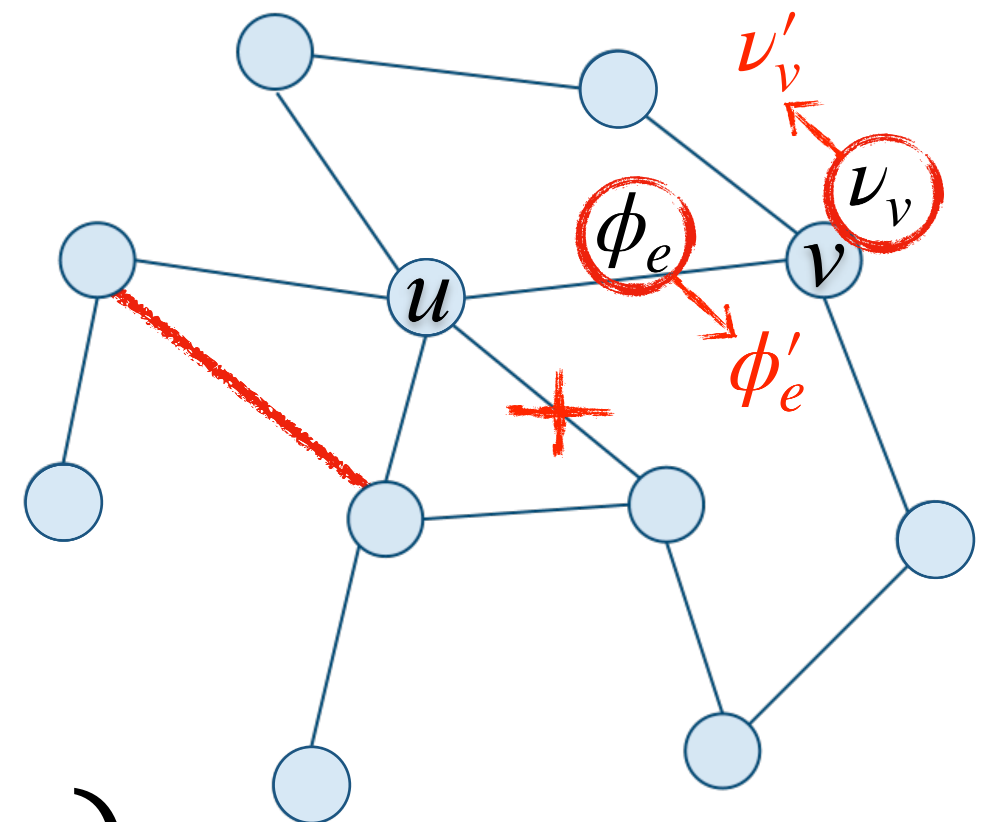
hypergraph (V,E)

Dynamic Sampling

- distribution μ over all $\sigma \in [q]^V$:

$$\mu(\sigma) \propto \prod_{v \in V} \nu_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e)$$

current sample: $X \sim \mu$



dynamic update:

- adding/deleting a constraint e
- changing a function ν_v or ϕ_e
- adding/deleting an independent variable v

new distribution μ'

Question:

Obtain $X' \sim \mu'$ from $X \sim \mu$ with small *incremental* cost.

Dynamic Sampling

Input: a graphical model which defines distribution μ
a sample $X \sim \mu$, and an update changing μ to μ'

Output: a new sample $X' \sim \mu'$

- inference/learning tasks where the graphical model is changing dynamically
  - video processing
  - online learning with dynamic or streaming data
- sampling/inference/learning algorithms which adaptively and locally change the joint distribution
  - stochastic gradient descent
  - approximate counting / self-reduction

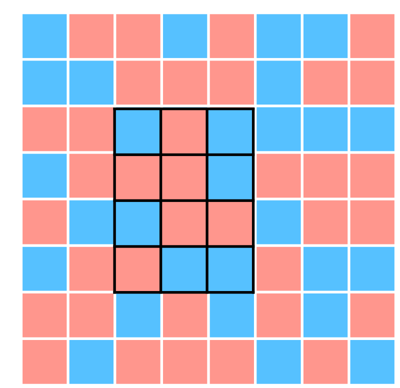
Dynamic Sampling

Input: a graphical model which defines distribution μ
a sample $X \sim \mu$, and an update changing μ to μ'

Output: a new sample $X' \sim \mu'$

Goal:

transform a $X \sim \mu$ to a $X' \sim \mu'$
by local changes



Current sampling techniques are not powerful enough:

- μ could be changed significantly by dynamic updates;
- Monte Carlo sampling does not know when to stop;
- notions such as mixing time give worst-case estimation.

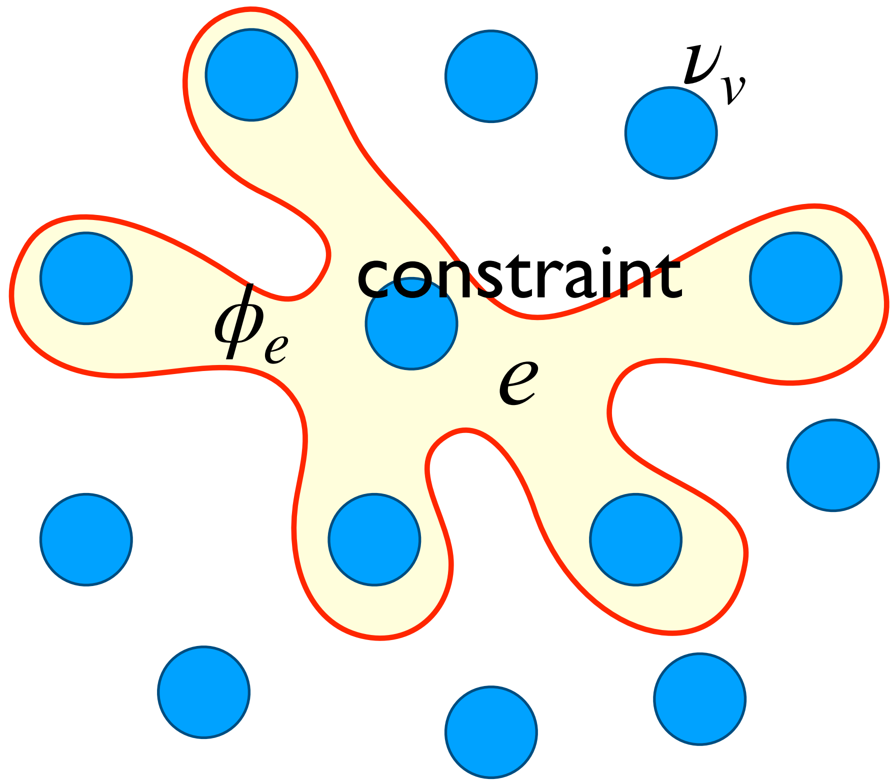
Rejection Sampling

- distribution μ over all $\sigma \in [q]^V$:

$$\mu(\sigma) \propto \prod_{v \in V} \nu_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e)$$

distribution ν_v over $[q]$

$$\phi_e : [q]^e \to [0,1]$$



- each $v \in V$ independently samples $X_v \in [q]$ according to ν_v;
- each $e \in E$ is passed independently with probability $\phi_e(X_e)$;
- X is accepted if all constraints $e \in E$ are passed.

- μ: distribution of X conditioning on accept
- Probability of accept is exponentially small!

For general graphical models:

Question I:
(dynamic sampling)

Given a $X \sim \mu$, when $\mu \to \mu'$
transform X to a $X' \sim \mu'$.

[Feng, Vishnoi, Y., **STOC**'19]

Dynamic Sampler

Upon receiving an update to the graphical model :

- Let R includes the variables affected by the update;
- while $R \neq \emptyset$:
  - $(X, R) \leftarrow \texttt{Resample}(X, R)$;

$\texttt{Resample}(X, R)$:

- each $e \in E^{+}(R)$ computes $\kappa_e = \min_{x_e:\, x_{e\cap R} = X_{e\cap R}} \phi_e(x_e)/\phi_e(X_e)$
- each $v \in R$ resamples $X_v \in [q]$ independently according to ϕ_v;
- each $e \in E^{+}(R)$ is passed independently with prob. $\kappa_e \cdot \phi_e(X_e)$;
  (otherwise e is violated)
- $R \leftarrow \bigcup_{e \in E:\ \text{violated } e} e$;

Question II:
(rejection sampling)

Make rejection sampling great again!

(when part of X is rejected, only resample the rejected part while still being correct)

[Guo, Jerrum, Liu, STOC'17]
for Boolean CSP

Dynamic Sampler

[Feng, Vishnoi, Y., **STOC'19**]

Upon receiving an update to the graphical model :

- Let R includes the variables affected by the update;
- while $R \neq \emptyset$:
  - $(X, R) \leftarrow \texttt{Resample}(X, R)$;

$\texttt{Resample}(X, R)$:

- each $e \in E^{+}(R)$ computes $\kappa_e = \min_{x_e:\, x_{e\cap R} = X_{e\cap R}} \phi_e(x_e)/\phi_e(X_e)$
- each $v \in R$ resamples $X_v \in [q]$ independently according to ϕ_v;
- each $e \in E^{+}(R)$ is passed independently with prob. $\kappa_e \cdot \phi_e(X_e)$;
  (otherwise e is violated)
- $R \leftarrow \bigcup_{e \in E:\ \text{violated}\ e} e$;

Correctness of Sampling

[Feng, Vishnoi, Y., **STOC'19**]

Upon receiving an update to the graphical model :

- Let R includes the variables affected by the update;
- while $R \neq \emptyset$:
  - $(X, R) \leftarrow \texttt{Resample}(X, R)$;

$\texttt{Resample}(X, R)$:

- each $e \in E^{+}(R)$ computes $\kappa_e = \min_{x_e:\, x_{e\cap R} = X_{e\cap R}} \phi_e(x_e)/\phi_e(X_e)$
- each $v \in R$ resamples $X_v \in [q]$ independently according to ϕ_v;
- each $e \in E^{+}(R)$ is passed independently with prob. $\kappa_e \cdot \phi_e(X_e)$; (otherwise e is violated)
- $R \leftarrow \bigcup_{e \in E:\ \text{violated}\ e} e$;

Correctness:

Assuming input sample $X \sim \mu$, upon termination, the dynamic sampler returns a sample from the updated distribution μ'.

Correctness of Sampling

[Feng, Vishnoi, Y., **STOC'19**]

Upon receiving an update to the graphical model :

- Let R includes the variables affected by the update;
- while $R \neq \emptyset$:
  - $(X, R) \leftarrow \texttt{Resample}(X, R)$;

$\texttt{Resample}(X, R)$:

- each $e \in E^{+}(R)$ computes $\kappa_e = \min_{x_e:\, x_{e\cap R} = X_{e \cap R}} \phi_e(x_e)/\phi_e(X_e)$
- each $v \in R$ resamples $X_v \in [q]$ independently according to ϕ_v;
- each $e \in E^{+}(R)$ is passed independently with prob. $\kappa_e \cdot \phi_e(X_e)$; (otherwise e is violated)
- $R \leftarrow \bigcup_{e \in E:\ \text{violated } e} e$;

Conditional Gibbs Property:

A random (X,R) is conditionally Gibbs *w.r.t.* μ if conditioning on any choice of R and X_R, the distribution of the rest $X_{V \setminus S}$, is correct.

Equilibrium:

If (X,R) is conditionally Gibbs *w.r.t.* μ', then so is (X',R').

Fast Convergence

Upon receiving an update to the graphical model :

- Let R includes the variables affected by the update;
- while $R \neq \emptyset$:
  - $(X, R) \leftarrow \texttt{Resample}(X, R)$;

$\texttt{Resample}(X, R)$:

- each $e \in E^{+}(R)$ computes $\kappa_e = \min_{x_e:\, x_{e\cap R} = X_{e\cap R}} \phi_e(x_e)/\phi_e(X_e)$
- each $v \in R$ resamples $X_v \in [q]$ independently according to ϕ_v;
- each $e \in E^{+}(R)$ is passed independently with prob. $\kappa_e \cdot \phi_e(X_e)$; (otherwise e is violated)
- $R \leftarrow \bigcup_{e \in E:\ \text{violated } e} e$;

Sufficient Condition for Fast Convergence:

If for the graphical model with max-edge-degree *d*:

$$\forall e \in E, \quad \min_x \phi_e(x) > \sqrt{1 - \frac{1}{d+1}}$$

then O(1) incremental cost per update in expectation.

- *Las Vegas* (good for simulation)
- *parallel* & *distributed* (good for systems)
- better static sampling algorithm

- Parallel/distributed algorithms for sampling
  - Feng, Sun, Y.: *What can be sampled locally?* **PODC**'17.
  - Feng, Hayes, Y.: *Distributed Sampling Almost-Uniform Graph Coloring with Fewer Colors.* arXiv: 1802.06953.
  - Feng, Hayes, Y.: *Fully-Asynchronous Distributed Metropolis Sampler with Optimal Speedup.* arXiv:1904.00943.
- For parallel/distributed computing: sampling ≡ approx counting/inference
  - Feng, Y.: *On local distributed sampling and counting.* **PODC**'18.
- Dynamic sampling algorithms
  - Feng, Vishnoi, Y.: *Dynamic Sampling from Graphical Models.* **STOC**'19.
  - Feng, He, Sun, Y.: *Dynamic MCMC Sampling.* arXiv:1904.11807.
  - Feng, Guo, Y.: *Perfect sampling from spatial mixing.* arXiv:1907.06033.

Thank you!