Dynamic Sampling from Graphical Models

Weiming Feng
Nanjing University

Joint work with: Nisheeth K. Vishnoi (Yale University)
Yitong Yin (Nanjing University)

STOC 2019
Phoenix, AZ.
Graphical Model

- Hyper graph $H = (V, E)$
  - $V$: vertices
  - $E \subseteq 2^V$: hyper edges.
- Vertex: variable with domain $Q$.
- Hyper edge: constraint on its variables.
- Weight functions (factors): $\Phi = (\phi_v)_{v \in V} \cup (\phi_e)_{e \in E}$
  - each variable $\phi_v: Q \to \mathbb{R}_{\geq 0}$;
  - each constraint $\phi_e: Q^e \to \mathbb{R}_{\geq 0}$.
- Each configuration $\sigma \in Q^V$: its weight
  \[ w(\sigma) = \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e). \]
Graphical Model

Instance $I = (V, E, Q, \Phi)$

- $V$: variables
- $E$: constraints
- $Q$: domain
- $\Phi = (\phi_v)_{v \in V} \cup (\phi_e)_{e \in E}$: weight functions (factors)

Gibbs distribution $\mu$ over $Q^V$:

\[
\forall \sigma \in Q^V: \quad \mu(\sigma) \propto w(\sigma) = \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e)
\]
Hardcore Model

- **Graph** \( G = (V, E) \)

\[ I(G) = \{ \text{independent sets in } G \}. \]

- **Fugacity** of vertex \( v \in V \): \( \lambda_v \in \mathbb{R}_{\geq 0} \).

- **Weight** of independent set \( S \in I(G) \):

\[ w(S) = \prod_{v \in S} \lambda_v. \]

- **Hardcore model**: distribution \( \mu \) over \( I(G) \), each \( S \in I(G) \):

\[ \mu(S) \propto w(S). \]

**Graph** \( G \)

\[ \lambda_4 \quad \lambda_1 \]
\[ \lambda_3 \quad \lambda_2 \]

**Weight**:

- \( 1 \)
- \( \lambda_1 \)
- \( \lambda_2 \)
- \( \lambda_3 \)
- \( \lambda_4 \)
- \( \lambda_1 \lambda_3 \)

**Product of vertex fugacities**
Ising Model

- Graph $G = (V,E)$.
- **Inverse temperature** of edge $e \in E$: $\beta_e \in \mathbb{R}_{\geq 0}$.
- **Spin state** of vertex $v \in V$: $\{-1, +1\}$.
- **Weight** of configuration $\sigma \in \{-1, +1\}^V$

$$w(\sigma) = \prod_{e=\{u,v\}\in E} \exp(\beta_e \sigma_u \sigma_v).$$

- **Ising model**: distribution $\mu$ over $\{-1, +1\}^V$:

$$\mu(\sigma) \propto w(\sigma).$$

**Weight** = $\exp(-\beta_1) \exp(\beta_2) \exp(\beta_3) \exp(-\beta_4) \exp(-\beta_5)$. 

**product of pairwise interactions**
Graphical Model

• **Machine Learning**
  representation, inference, learning;

• **Statistical Physics**
  Ising model, hardcore model;

• **Theoretical Computer Science**
  sampling, counting.

---

**Sampling from Graphical Model**

- **Input:** a graphical model $\mathcal{I}$;
- **Output:** a sample $X \sim \mu_{\mathcal{I}}$. 

---

application: image denoising
Dynamic Sampling Problem

- Graphical model $\mathcal{I} = (V, E, Q, \Phi)$

$$
\mu_j(\sigma) \propto \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e).
$$

- Random sample: $X \sim \mu_j$.

- Updates of graphical model $\mathcal{I} \rightarrow \mathcal{I}'$
  - add/delete constraints;
  - change weight functions.

**Question:** Can we modify $X$ to $X' \sim \mu_{j'}$ with a small incremental cost? 

random sample for updated graphical model
Update is represented by a pair \((D, \Phi_D)\)

- \(D \subseteq V \cup 2^V\): updated variables & updated constraints;
- \(\Phi_D = (\phi_a)_{a \in D}\): new weight functions.

**Dynamic Sampling from Graphical Model**

- **Input:** a graphical model \(J = (V, E, Q, \Phi)\); a sample \(X \sim \mu_J\) an update \((D, \Phi_D)\) that modifies \(J\) to \(J'\);
- **Output:** a sample \(X' \sim \mu_{J'}\).

**Offline adversary:** update is independent with the input sample \(X\).
Motivations

• Online learning with dynamic or streaming data

• Dynamic graphical models
  • Video: a sequence of closely related images.

• Approximate counting [Jerrum, Valiant, Vazirani, 1986]
  • Graph $G = (V, E)$, count $\#\{\text{independent sets of } G\}$.
  • Self reduction: a sequence of graphs $G_0, G_1, ..., G_{|E|}$:
Static Sampling
• **Input:** a graphical model \( J \);
• **Output:** a sample \( X \sim \mu_J \).

Well studied

Dynamic Sampling
• **Input:** a graphical model \( J \);
  a sample \( X \sim \mu_J \)
  a update \( (D, \Phi_D) \)
• **Output:** a sample \( X' \sim \mu_{J'} \).

Lacking studies

Algorithms for static sampling
• Markov chain Monte Carlo (MCMC)
  • Metropolis Hastings [Metropolis 1953]
  • Glauber Dynamics [Glauber 1963]
• Coupling from the past (CFTP) [Propp and Wilson 1996]

Not suitable for dynamic sampling, per se.
• **can not use** the input sample \( X \);
• rerunning sampling algorithm on \( J' \) is **wasteful**.
Dynamic Sampling Problem

- **Input:** a graphical model $I$; a sample $X \sim \mu_I$; an update $(D, \Phi_D)$
- **Output:** a sample $X' \sim \mu_{I'}$

**New Algorithm**

- **Fast**
  a broad class of graphical models $\mathbb{E}[\text{running time}] = O(|D|)$

- **Exact Sampling**
  $X$ follows precisely distribution $\mu_{I'}$

- **Las Vegas**
  algorithm knows when to stop

- **Distributed / Parallel**
  each step uses only local information
Graphical Model

Instance $\mathcal{I} = (V, E, Q, \Phi)$

- $V$: variables
- $E$: constraints
- $Q$: domain
- $\Phi = (\phi_v)_{v \in V} \cup (\phi_e)_{e \in E}$: weight functions (factors)

Gibbs distribution $\mu$ over $Q^V$:

$$\forall \sigma \in Q^V: \quad \mu(\sigma) \propto w(\sigma) = \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e)$$
Rejection Sampling

Graphical model $\mathcal{I} = (V, E, Q, \Phi)$ with Gibbs distribution

$$\mu_{\mathcal{I}}(\sigma) \propto \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e).$$

**Assumption: normalized weighted functions**

- each $\phi_v: Q \rightarrow [0,1]$ is a **distribution** over $Q$: $\sum_{c \in Q} \phi_v(c) = 1$;
- each $\phi_e: Q^e \rightarrow [0,1]$. 
Rejection Sampling

- Each $v \in V$ samples $X_v \sim \phi_v$ ind.;
- Each $e \in E$ becomes accepted ind. w.p. $\phi_e(X_e)$; o.w. $e$ becomes rejected.
- **Accept** $X = (X_v)_{v \in V}$ if all $e \in E$ are accepted;
- **Reject** $X$ if otherwise.

$$\Pr[X = \sigma \land X \text{ is accepted}] = \prod_{v \in V} \phi_v(\sigma_v) \prod_{e \in E} \phi_e(\sigma_e).$$

*generate $X = \sigma$ all $e \in E$ are accepted

$$\Pr[\text{all } e \in E \text{ are accepted}] = \exp(-\Omega(|E|)).$$

Rejection Sampling is **Correct but Slow.**
**Question**

Can we obtain an *efficient* rejection sampling algorithm?

- Fast
- Dynamic
- Distributed / Parallel

This problem was **partially solved** by Partial Rejection Sampling (PRS) [Guo, Jerrum, Liu, 2017].

- **Boolean** weight function $\phi_e \rightarrow \{0,1\}$
- **Not known** to be dynamic
- **Not** distributed / parallel
Our Contribution

Dynamic Sampling Problem

• **Input:** a graphical model $\mathcal{I}$; a sample $X \sim \mu_\mathcal{I}$; a update $(D, \Phi_D)$

• **Output:** a sample $X' \sim \mu_{\mathcal{I}'}. $

New Algorithm

Efficient Rejection Sampling

• Fast
• Dynamic
• Distributed/Parallel
Rejection Sampling

- Each $v \in V$ samples $X_v \sim \phi_v$ ind.;
- Each $e \in E$ becomes accepted ind. w.p. $\phi_e(X_e)$; o.w. $e$ becomes rejected
- Accept $X = (X_v)_{v \in V}$ if all $e \in E$ are accepted;
- Reject $X$ if otherwise.

The sample $X$ is rejected

Set of Bad Variables

$$\mathcal{R} = \bigcup_{e \in E: e \text{ is rejected}} e,$$

$\mathcal{R}$: variables in rejected constraints
A “Natural” Resampling Algorithm

• each $v \in \mathcal{R}$ resamples $X_v \sim \phi_v$ ind.;
  o.w. $e$ becomes rejected
• each $e \in \text{ICD} (\mathcal{R})$ becomes accepted ind. w.p. $\phi_e (X_e)$;
• construct new $\mathcal{R} \leftarrow \bigcup_{e \in E : e \text{ is rejected}} e$.

ICD($\mathcal{R}$): constraints incident to $\mathcal{R}$

$\text{ICD} (\mathcal{R}) = \{ e \in E \mid e \cap \mathcal{R} \neq \emptyset \}$. 

$\mathcal{R}$
A “Natural” Resampling Algorithm

- each $v \in \mathcal{R}$ resamples $X_v \sim \phi_v$ ind.;  
- each $e \in \text{ICD}(\mathcal{R})$ becomes accepted ind. w.p. $\phi_e(X_e)$;  
- construct new $\mathcal{R} \leftarrow \bigcup_{e \in E : e \text{ is rejected}} e$.

\[\text{While}(\mathcal{R} \neq \emptyset)\]
\[\quad \text{Update } (X, \mathcal{R}) \text{ by “Natural” Resampling Algorithm}\]
\[\text{Output } X.\]

- Similar to Morse-Tardos for LLL. [Morse, Tardos, 2009]
- The output $X$ does NOT follow the Gibbs distribution $\mu$. [Harris, Srinivasan, 2016], [Guo, Jerrum, Liu, 2017]
A “Natural” Resampling Algorithm

- each $v \in \mathcal{R}$ resamples $X_v \sim \phi_v$ ind.;
- each $e \in \text{ICD}(\mathcal{R})$ becomes accepted ind. w.p. $\phi_e(X_e)$;
- construct new $\mathcal{R} \leftarrow \bigcup_{e \in E: e \text{ is rejected}} e$.

$\text{ICD}(\mathcal{R})$: constraints incident to set $\mathcal{R}$

- internal constraints
  \[ E(\mathcal{R}) = \{ e \in E \mid e \subseteq \mathcal{R} \}; \]
- boundary constraints
  \[ \delta(\mathcal{R}) = \{ e \in E \setminus E(\mathcal{R}) \mid e \cap \mathcal{R} \neq \emptyset \}. \]
A “Natural” Resampling Algorithm

• each $v \in R$ resamples $X_v \sim \phi_v$ ind.;
• each $e \in ICD(R)$ becomes accepted ind. w.p. $\phi_e(X_e)$;
• construct new $R \leftarrow \bigcup_{e \in E : e \text{ is rejected}} e$.

Our Algorithm: Local-Resample($X, R$)

• each $v \in R$ resamples $X_v \sim \phi_v$ ind.;
• each $e \in E(R)$ becomes accepted ind. w.p. $\phi_e(X_e)$;
• each $e \in \delta(R)$ becomes accepted ind. with a modified probability;
• construct new $R \leftarrow \bigcup_{e \in E : e \text{ is rejected}} e$.
• return $(X, R)$;
Our Algorithm: Local-Resample($X, \mathcal{R}$)

1. each $v \in \mathcal{R}$ resamples $X_v \sim \phi_v$ ind.;
2. each $e \in E(\mathcal{R})$ becomes accepted ind. w.p. $\phi_e(X_e)$;
3. each $e \in \delta(\mathcal{R})$ becomes accepted ind. w.p. $C_e \cdot \frac{\phi_e(X_e)}{\phi_e(X_e^{\text{old}})} \leq 1$;
4. construct new $\mathcal{R} \leftarrow \bigcup_{e \in E : e \text{ is rejected}} e$.
5. return $(X, \mathcal{R})$;

$X^{\text{old}} \in Q^V$ is the old $X$ before the resampling in step ①

Normalization Factor $C_e = C_e(X^{\text{old}}_{\mathcal{R}})$:

$C_e = \min_{y \in Q^e : y \cap \mathcal{R} = X^{\text{old}}_{e \cap \mathcal{R}}} \phi_e(y)$.

While($\mathcal{R} \neq \emptyset$)

$(X, \mathcal{R}) \leftarrow$ Local-Resample($X, \mathcal{R}$)

Output $X$.  

Correct Distribution

Normalization factor
Dynamic Sampler

Dynamic Sampling from Graphical Model

- **Input:** a graphical model $\mathcal{I}$; a sample $\mathbf{X} \sim \mu_{\mathcal{I}}$; a update $(D, \Phi_D)$ that modifies $\mathcal{I}$ to $\mathcal{I}'$;
- **Output:** a sample $\mathbf{X}' \sim \mu_{\mathcal{I}'}$.

- $D$: updated variables & updated constraints;
- $\text{vbl}(D)$: variables involved by the update:
  - **updated variables:** $D \cap V$;
  - **variables incident to updated constraints:** $\bigcup_{e \in D \cap 2^V} e$.

Graphical models $\mathcal{I}$ and $\mathcal{I}'$ differ only on $\text{vbl}(D)$.

The initial **bad set** $\mathcal{R} = \text{vbl}(D)$.
Dynamic Sampler

• Apply changes \((D, \Phi_{D})\) to current graphical model \(I\).
• \(R \leftarrow \text{vbl}(D)\);
• While\((R \neq \emptyset)\)
  • \((X, R) \leftarrow \text{Local-Resample}(X, R)\);
• Return \(X\);

Theorem: Correctness [This Work]
Upon termination, the dynamic sampler outputs \(X \sim \mu_{j'}\).

A dynamic sampler for general graphical model:
• **Exact sampling**
• **Las Vegas**
• **Distributed / Parallel**
Proof of Correctness

**Dynamic Sampler**

- Apply changes \((D, \Phi_D)\) to current graphical model \(\mathcal{I}\).
- \(\mathcal{R} \leftarrow \text{vbl}(D)\);
- While \((\mathcal{R} \neq \emptyset)\)
  - \((X, \mathcal{R}) \leftarrow \text{Local-Resample}(X, \mathcal{R})\);
- Return \(X\);

The algorithm maintains \((X, \mathcal{R}) \in Q^V \times 2^V\)

- \(\mathcal{R}\): bad set;
- \(V \setminus \mathcal{R}\): good set

\(X_{V \setminus \mathcal{R}}\) follows “correct” distribution.

**Conditional Gibbs property w.r.t. \(\mu\)**

Conditioning on any \(\mathcal{R} \subseteq V\) and any assignment \(\sigma \in Q^\mathcal{R}\) of \(X_{\mathcal{R}}\), the distribution of \(X_{V \setminus \mathcal{R}}\) is \(\mu^\sigma_{V \setminus \mathcal{R}}\).

\(\mu^\sigma_{V \setminus \mathcal{R}}\): marginal distribution of \(\mu\) on \(V \setminus \mathcal{R}\) conditioning on \(\sigma\).
Local-Resample($X, \mathcal{R}$) 

define

**Resampling chain**
- **Markov chain** on $\Omega = Q^V \times 2^V$
- Transition Matrix $P \in \mathbb{R}^{\Omega \times \Omega}$
  
  $P: (X, \mathcal{R}) \rightarrow (X', \mathcal{R}')$

---

**Equilibrium Condition**

If $(X, \mathcal{R})$ satisfies the conditionally Gibbs property w.r.t. $\mu$, then so does $(X', \mathcal{R}')$.

---

**Equation System for Equilibrium Condition**

\[
\forall S, T \subseteq V, \sigma \in Q^V \setminus S \text{ and } \tau \in Q^V \setminus T, \\
\forall y \in Q^V, y_{V \setminus T} = \tau: \sum_{x \in Q^V \atop x_{V \setminus S} = \sigma} \mu_S(x_S) \cdot P((x, S), (y, T)) = C(S, \sigma, T, \tau) \cdot \mu_T(y_T).
\]

Our algorithm is **a solution to** this equation system.
Theorem: Fast Convergence [This Work]

The updated graphical model satisfies $d = O(1)$, $\max_{e \in E'} |e| = O(1)$, and

$$\forall e \in E': \min_x \phi'_e(x) > \sqrt{1 - \frac{1}{d + 1}}$$

where $d$ is the maximum degree of the dependency graph.

The cost of the dynamic sampler is

- $O(\log |D|)$ iterations in expectation;
- $O(|D|)$ resamplings in expectation.

Ising Model: $\forall e \in E$:

$$1 - \exp(-2|\beta_e|) < \frac{1}{4\Delta}$$

Uniqueness Regime: $\forall e \in E$:

$$1 - \exp(-2|\beta_e|) < \frac{2}{\Delta}$$
Theorem: Fast Convergence [This Work]

Hardcore model and Ising model on bounded degree graph s.t.

• Hardcore model: $\forall v \in V: \lambda_v \leq \frac{1}{\sqrt{2}\Delta - 1}$.

• Ising model: $\forall e \in E: 1 - \exp(-2|\beta_e|) \leq \frac{1}{2.221\Delta + 1}$,

where $\Delta$ is the maximum degree.

The cost of the dynamic sampler is

• $O(\log |D|)$ iterations in expectation;

• $O(|D|)$ resamplings in expectation;

Uniqueness Regime:

• Hardcore model: $\forall v \in V: \lambda_v < \frac{(\Delta - 1)^{\Delta - 1}}{(\Delta - 2)^\Delta} \approx \frac{e}{\Delta - 2}$. $\lambda_v = O\left(\frac{1}{\Delta}\right)$

• Ising model: $\forall e \in E: 1 - \exp(-2|\beta_e|) < \frac{2}{\Delta}$. $1 - \exp(-2|\beta_e|) = O\left(\frac{1}{\Delta}\right)$
Proof of the Fast Convergence

\[
\mathcal{R}_0 = \text{vbl}(D) \rightarrow \mathcal{R}_1 \rightarrow \mathcal{R}_2 \rightarrow \ldots \rightarrow \mathcal{R}_T = \emptyset
\]

Potential function on bad set \( \mathcal{R}_t \)

\[
H: 2^V \rightarrow \mathbb{Z}_{\geq 0}
\]

Step-wise decay on expectation of \( H(\mathcal{R}_t) \)

\[
\mathbb{E}[H(\mathcal{R}_t)] \leq (1 - \delta)\mathbb{E}[H(\mathcal{R}_{t-1})].
\]
Summary

• **Dynamic sampling problem.**

• **Dynamic sampler** for general graphical models
  
  *Exact Sampling & Las Vegas & Distributed/Parallel.*

• **Equilibrium conditions** for resampling.

Future Work

• **Dynamic MCMC sampling** [Feng, He, Yin, Sun, arXiv:1904.11807]

• Improve the **regimes** for efficient dynamic sampling
  
  *correlation decay ➔ efficient dynamic sampling algorithm.*

• Extend to **continuous distributions & global constraints.**
Dynamic Sampling from Graphical Models

Weiming Feng†, Nicheeth K. Vishnoi‡, Yitong Yin‡
†Nanjing University, China
‡Yale University, USA

Abstract

In this paper, we study the problem of dynamic sampling from graphical models that evolve over time. We introduce a novel algorithm for sampling from such models, which we call the dynamic sampler. The algorithm maintains a random pair of sample assignments, one for each time step, and updates this pair as the model evolves. We prove correctness of the algorithm and demonstrate its efficiency through experiments.

Dynamic Sampling Problem

Given a graphical model and a sequence of graphs, the problem is to sample from the model at each time step, efficiently. We provide an algorithm that solves this problem.

Dynamic Sampler

The algorithm maintains a random pair of sample assignments, one for each time step, and updates this pair as the model evolves. The correctness of the algorithm is established through a series of lemmas and theorems.

Our Results

Theorem: Correctness

The dynamic sampler maintains the correct sample pair (S, T) with probability 1, where S and T are the sample assignments for the current and previous time steps, respectively.

Equilibrium Condition

The dynamic sampler maintains a well-mixed state (R, R', T', S') with probability 1, where R, R', T', and S' are the random pairs of sample assignments at the current and previous time steps, respectively.

Keywords: Graphical Models, Dynamic Sampling, Markov Chain Monte Carlo, Bayesian Networks, Conditional Independence, Time-Varying Graphs, Sampling Algorithms.

Motivation

Given a graphical model G, our goal is to sample from the model efficiently. We introduce a novel algorithm that solves this problem.

Methodology

We introduce a novel algorithm for sampling from time-varying graphical models. The algorithm maintains a random pair of sample assignments, one for each time step, and updates this pair as the model evolves.

Implementation

We implement the dynamic sampler algorithm in Python and demonstrate its efficiency through experiments.

Conclusion

We have introduced a novel algorithm for dynamic sampling from graphical models that evolve over time. The algorithm maintains a random pair of sample assignments, one for each time step, and updates this pair as the model evolves. We prove correctness and efficiency of the algorithm through a series of lemmas and theorems.

Thank You

See you at the poster session #131