Sampling in High-Dimensional Space in Network Environment

(from a *Theory of Computing* Point of View)
Sampling in High-Dimensional Space

Given an $n$-dimensional joint distribution $\mu$, draw a sample $X = (X_1, X_2, \ldots, X_n) \sim \mu$.

• One of the earliest computer programs (nuclear Monte Carlo simulations on ENIAC)
Sampling in High-Dimensional Space

- One of the earliest computer programs (nuclear Monte Carlo simulations on ENIAC)
- Crucial for today’s computational tasks:
  - Probabilistic inference: guessing possible values of $X_i$ given values of $X_S$
  - Optimization via sampling: finding $x$ with max $\mu(x)$ by drawing $X \sim \mu$
  - High-dimensional integration: calculating $\int_{\mathbb{R}^n} \mu(x) \, dx$ or estimating volumes
  - Statistical physics: simulating interacting particle systems
  - Approximate counting: e.g. estimating Network Reliability

Given an $n$-dimensional joint distribution $\mu$, draw a sample $X = (X_1, X_2, \ldots, X_n) \sim \mu$. 
Gibbs Distribution

- High-dimensional distribution $\mu$ described by local constraints:
  - $n$ variables on finite discrete domain $\Omega$
  - a set $\mathcal{C}$ of local constraints $(f, S)$ with scope $S \subseteq [n]$ and $f : \Omega^S \rightarrow [0,1]$

\[
\forall x \in \Omega^n: \quad \mu(x) \propto \prod_{(f,S) \in \mathcal{C}} f(x_S)
\]

- For hard constraints $f : \Omega^S \rightarrow \{0,1\}$, the $\mu$ is uniform distribution over constraint satisfaction solutions

- Examples of Gibbs distributions: graphical model, Bayesian network, Boltzmann machine, Markov random field, factor graph, spin system, weighted CSP, …
Gibbs Sampler
(a.k.a. Glauber dynamics, heat-bath dynamics) [Glauber 1963]

the Markov chain maintains an \( x \in \Omega^n \); at each step:

- pick an \( v \in \{1,2,\ldots,n\} \) uniformly at random;
- update \( x_v \) randomly according to the marginal distribution \( \mu_v(\cdot \mid x_{N(v)}) \);

Random walk in \( \Omega^n \)
Gibbs Sampler  
(*a.k.a. Glauber dynamics, heat-bath dynamics*)  
*[Glauber 1963]*

- The Gibbs sampler converges (*mixes*) to $\mu$.
  - Many other Markov chains converge to $\mu$, e.g. the Metropolis algorithm  
    *[Metropolis 1953]*
- The convergence rate (*mixing time*) depends on properties of $\mu$.
  
  \[
  T_{\text{mix}} = \max_{x} \min \left\{ t \geq 1 \mid \| P^t(x, \cdot) - \mu \|_1 \leq 1/e \right\}
  \]
- What makes a family of distributions easy/hard to sample? New algorithms?
Outline

• Computational Phase Transition of Sampling
  • Phase transition of probabilistic graphical models
  • Phase transition of sampling constraint satisfaction solutions (a.k.a. a sampling Lovász local lemma)

• Network Algorithms for Gibbs Sampling
  • Parallel/Distributed/Dynamic sampling algorithms

• Application: Network Reliability Estimation
Computational Phase Transition of Sampling
Computational Phase Transition

- Gibbs distribution:
  \[ \forall x \in \Omega^n: \quad \mu(x) \propto \prod_{(f,S) \in \mathcal{C}} f(x_S) \]

- locally constrained random variables ↔ locally interacting particle states

- Continuous change of strength of local interaction → sharp transition of global state
  (state of matter / computational complexity)
Hardcore Model

- Given a graph \( G(V, E) \) and a parameter \( \lambda > 0 \):
  \[ \forall \text{ independent set } I \subseteq V \text{ of } G: \]
  \[ \mu(I) \propto \lambda^{|I|} \]

- Critical threshold (for phase transition of hardcore gas with fugacity \( \lambda \) on \( \Delta \)-degree Bethe lattice):
  \[ \lambda_c(\Delta) \triangleq \frac{(\Delta - 1)^{\Delta - 1}}{(\Delta - 2)^{\Delta}} \approx \frac{e}{\Delta - 2} \]

- \( \lambda > \lambda_c(\Delta) \implies \) sampling is \textbf{NP-hard} [Sly, FOCS 2010 best paper]

- \( \lambda < \lambda_c(\Delta): \)
  
  Gibbs sampler
  
  \[ n^{O(\log \Delta)} \rightarrow n^{f(\lambda)} \rightarrow \Delta^{O(\Delta^2)} n \log n \rightarrow O(n^2 \log n) \rightarrow O(n \log n) \text{ optimal} \]

  for all Gibbs distributions with pairwise repulsive constraints on Boolean variables (anti-ferromagnetic two-state spin systems)

Gibbs sampler:
- maintain an independent set \( I \subseteq V: \)
  - pick an \( v \in V \) uniformly at random;
  - if \( I \cup \{v\} \) is independent set then
    \[ I \leftarrow \begin{cases} 
    I \cup \{v\} & \text{with prob. } \frac{\lambda}{1 + \lambda} \\
    I \setminus \{v\} & \text{with prob. } \frac{1}{1 + \lambda} 
    \end{cases} \]

- strong spatial mixing (SSM)
- high-dimensional expander (HDX)
- local-to-global argument
- modified log-Sobolev inequality
- field dynamics
  
  ... ...
Constraint Satisfaction Solutions

• For **hard constraints** (Boolean decisions) \( f : \Omega^S \rightarrow \{0,1\} \)

\[
\forall x \in \Omega^n: \quad \mu(x) \propto \prod_{(f,S) \in \mathcal{C}} f(x_S)
\]

\( \mu \) is the uniform distribution over all constraint satisfaction solutions

• Example: \( k \)-SAT with variable degree \( d \)

\[
\text{CNF formula } (x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_4) \land (x_3 \lor \neg x_4 \lor \neg x_5)
\]

• **Barrier**: classic sampling algorithms rely on *connectivity* of solution space

```
Satisfying solution exists when \( k \gtrsim \log d \)
```

(Lovász local lemma)

**Sampling?**
Overcome the Connectivity Barrier

Projected Markov chain:
- Properly construct a subset $U \subseteq V$ of variables;
- Sample $x_U \sim \mu_U$ by simulating Gibbs sampler on $\mu_U$;
- Extend $x_U$ to a satisfying solution $x \sim \mu$;

**Idea:** project onto lower dimension to improve connectivity

- **Efficiently construct a “good” subspace $U \subseteq V$:**
  - Gibbs sampler for $\mu_U$ is fast-convergent (the subspace is well-connected) and efficient to implement
  - it is efficient to extend a random partial solution $x_U \sim \mu_U$ to a uniform satisfying solution $x \sim \mu$

- **Fast sampler in near-linear time (under Lovász local lemma like condition):**
  - **SAT** [Feng, Guo, Y., Zhang, STOC 2020]
  - **CSP with atomic constraints** [Feng, He, Y., STOC 2021]
  - **general CSP (constraint satisfaction problem)** [He, Wang, Y., FOCS 2022]
Network Algorithms for Gibbs Sampling
Distributed Gibbs Sampling

- Generate high-dimensional sample in a network:
  - Each node $v \in \{1, 2, \ldots, n\}$ generates a random $X_v$
  - Altogether it follows the correct joint distribution

$$X = (X_1, X_2, \ldots, X_n) \sim \mu$$
Distributed Gibbs Sampling

Gibbs sampler for $\mu$:

- maintain an $x \in \Omega^n$, at each step:
  - pick a random $v \in \{1, 2, \ldots, n\}$;
  - update $x_v$ according to $\mu_v(\cdot | x_{N(v)})$;

• Classic sampling algorithms are intrinsically **sequential**.

• **Barrier for parallelization**: update of variable depends on neighbors’ states
  - concurrent updates of adjacent variables $\implies$ fault
  - correct parallelization: $O(\Delta)$ overhead!

• Is it possible to **correctly** parallelize the Markov chain with **linear speedup**?

Generate high-dimensional samples in a network:
Distributed Gibbs Sampling

Gibbs sampler for $\mu$:

- maintain an $x \in \Omega^n$, at each step:
  - pick a random $v \in \{1,2,\ldots,n\}$;
  - update $x_v$ according to $\mu_v(\cdot | x_{N(v)})$;

- Is it possible to **correctly** parallelize the Markov chain with **linear speedup**?
  - a parallel chain called *LocalMetropolis* [Feng, Sun, Y., PODC 2017]
  - sampling by *network decomposition* [Feng, Y., PODC 2018]
  - parallelize *Metropolis algorithm* [Feng, Hayes, Y., SODA 2021]
An Idealized Parallel “Sampling Algorithm”

Continuous-time Gibbs sampler for \( \mu \):
- each \( \nu \in \{1,2,\ldots,n\} \) holds a Poisson clock;
- when the clock at \( \nu \) rings:
  - update \( x_\nu \) according to \( \mu_\nu( \cdot \mid x_{N(\nu)}) \);

\( O(T) \) continuous-time duration \( \iff \) \( O(nT) \) discrete-time steps

- This is the original definition of Gibbs sampler [Glauber 1963].

- An idealized (continuous-time with atomic update operation) process that models the evolution of physical world.

- Simulate this idealized process on computer network with no overhead?
Parallelize the Gibbs Sampler

**Continuous-time** Gibbs sampler for $\mu$: [Glauber 1963]

Each $\nu \in \{1,2,\ldots,n\}$ holds a Poisson clock; when the clock at $\nu$ rings:
- update $x_\nu$ according to $\mu_\nu(\cdot \mid x_{N(\nu)})$;

**Ideas:**
- Construct a **dynamical system** whose **fixpoint** corresponds to the correct evolution of the chain.
- Simulate this dynamical system by a **locally-iterative message passing algorithm** on the network.
- A **universal coupling** of random bits used in different iterations to ensure fast **stabilization** to fixpoint.

- A much weakened **Dobrushin’s condition** (which is almost always satisfied) 
  $\implies$ **faithful parallel simulation of Gibbs sampler with linear speedup** [Liu, Y., STOC 2022]

(all single-site dynamics)
Dynamic Sampling

Dynamic Sampling problem: for a dynamically changing graphical model $\mu \rightarrow \mu'$

- Sampling/inference tasks on dynamically changing data:
  - Online data, data streams, network environment, etc.
- Dynamically changing graphical models generated in:
  - Locally-iterative algorithms for learning.
  - Self-reduction procedure in approximate counting.

- **Algorithmic Lipschitz**: transform $X \sim \mu$ to $X' \sim \mu'$ with cost proportional to $\text{diff}(\mu, \mu')$
Dynamic Sampling

Dynamic Sampling problem: for a dynamically changing graphical model $\mu \rightarrow \mu'$

$X \sim \mu$ \hspace{1cm} $X' \sim \mu'$

• A dynamic sampling algorithm:
  [Feng, Vishnoi, Y., STOC 2019]
  • correct and efficient on dynamic data
  • parallel, distributed, communication-efficient
  • Las Vegas algorithm for perfect sampling

• Based on Partial Rejection Sampling
  [Guo, Jerrum, Liu, STOC 2017]
  • very different from Markov chains (random walks).
Application: Network Reliability Estimation
Network Reliability

[Valiant 1979]

- Given an undirected graph (a network) $G(V, E)$, and parameters $\vec{p} \in [0,1]^E$:
  - each edge $e \in E$ fails independently with prob. $p_e$
  - let $G(\vec{p})$ denote the resulting network
- (all-terminal) network reliability:
  
  $R_{\vec{p}}(G) = \sum_{C \subseteq E \text{ that connects } V} \prod_{e \in C} (1 - p_e) \prod_{e \not\in C} p_e$

the probability that $G(\vec{p})$ is connected

enumerating all connected subgraphs
Computational Complexity of Counting

- Let $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$ be a square matrix.

- **Determinant**: can be computed as fast as matrix multiplication

$$
\sum_{\pi \in S_n} \text{sgn}(\pi) \prod_{i=1}^{n} a_{i,\pi(i)}
$$

- **Permanent**: is $\#P$-complete [Valiant 1979]

$$
\sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)}
$$

solvable in polynomial-time $\implies$ the polynomial hierarchy ($\text{PH}$) collapses $\implies$ $\text{NP} = \text{P}$
Network Reliability

- Given an undirected graph (a network) $G(V, E)$, and a parameter $\vec{p} \in [0, 1]^E$:
  
  (all-terminal) network reliability: $R_{\vec{p}}(G) = \sum_{C \subseteq E \text{ that connects } V} \prod_{e \in C} (1 - p_e) \prod_{e \not\in C} p_e$

- The problem is \#P-complete [Valiant 1979] [Jerrum 1981]:
  - $R_{\vec{p}}(G)$ cannot be evaluated precisely in polynomial time unless $\text{NP} = \text{P}$

- **Approximation** by Monte Carlo method: return an estimation $\hat{R}_{\vec{p}}(G)$

  $\Pr \left[ (1 - \epsilon)R_{\vec{p}}(G) \leq \hat{R}_{\vec{p}}(G) \leq (1 + \epsilon)R_{\vec{p}}(G) \right] \geq 1 - o(1)$
Network Reliability by Sampling

\( G(\vec{p}) \): a subgraph of \( G \) obtained by removing each \( e \in E \) independently with prob. \( p_e \)

- A naïve Monte Carlo estimation of network reliability \( R_{\vec{p}}(G) \):
  
  for \( j = 1,2,\ldots,k \) for a large enough \( k \):
  
  generate a \( G^{(j)} \sim G(\vec{p}) \) by removing each \( e \in E \) independently with prob. \( p_e \);
  
  return \( \frac{1}{k} \sum_{i=1}^{k} 1 \left[ G^{(j)} \text{ is connected} \right] \);

- Requires too many samples \( G^{(j)} \sim G(\vec{p}) \) if \( R_{\vec{p}}(G) \) is close to 0 (unreliable network).

- Monte Carlo method based on self-reduction:
  
  - Drawing samples \( C \sim G(\vec{p}) \) conditioned on \( C \) being connected on \( V \).
Network Reliability by Sampling

\[ G(\vec{p}) \]: a subgraph of \( G \) obtained by removing each \( e \in E \) independently with prob. \( p_e \)

- Monte Carlo method based on **self-reduction**:
  - Drawing samples \( C \sim G(\vec{p}) \) conditioned on \( C \) being **connected** on \( V \).

- **Edge-contraction**:

- **Telescopic product**:
  \[
  R_{\vec{p}}(G) = \frac{R_{\vec{p}}(G_0)}{R_{\vec{p}}(G_1)} \cdot \frac{R_{\vec{p}}(G_1)}{R_{\vec{p}}(G_2)} \cdot \frac{R_{\vec{p}}(G_2)}{R_{\vec{p}}(G_3)} \cdot R_{\vec{p}}(G_3)
  \]

- \( \frac{R_{\vec{p}}(G_i)}{R_{\vec{p}}(G_{i+1})} \) can be estimated by sampling \( C \sim G_{i+1}(\vec{p}) \) | **connected**.
Markov Chain for Connected Subgraphs

\[ G(\vec{p}) : \text{a subgraph of } G \text{ obtained by removing each } e \in E \text{ independently with prob. } p_e \]

- Monte Carlo method based on **self-reduction**:  
  - Drawing samples \( C \sim G(\vec{p}) \) conditioned on \( C \) being connected on \( V \).

- A natural Markov chain (**Gibbs sampler**) for connected subgraphs:
  
  start with \( C_0 = E \); and for each step \( t = 0,1,2 \ldots \):
  
  - pick an edge \( e \in E \) uniformly at random;
  - if \( C_t - \{ e \} \) disconnects \( V \) then \( C_{t+1} = C_t \); otherwise
    
    \[ C_{t+1} = \begin{cases} 
    C_t \cup \{ e \} & \text{with prob. } 1 - p_e \\
    C_t \setminus \{ e \} & \text{with prob. } p_e
    \end{cases} \]

  \( m \): number of edges  
  \( n \): number of vertices

- The chain **mixes** (**converges**) to \( G(\vec{p})|_{\text{connected}} \) in \( O(m^2 \log n) \) steps.

- **Conjecture**: the chain mixes in \( O(m \log n) \) steps.
Markov Chain for Connected Subgraphs

\[ G(\vec{p}) \]: a subgraph of \( G \) obtained by removing each \( e \in E \) independently with prob. \( p_e \)

- Monte Carlo method based on self-reduction:
  - Drawing samples \( C \sim G(\vec{p}) \) conditioned on \( C \) being connected on \( V \).

- A substantially more complicated Markov chain (matroid basis exchange) for connected subgraphs:
  - Each step (matroid basis exchange) requires \( O(m) \) computation.

- The chain mixes (converges) to \( G(\vec{p})|_{\text{connected}} \) in \( O(m \log n) \) steps
  [Anari, Liu, Oveis Gharan, Vinzant, STOC 2019 best paper]
    - Strongly log-concave distribution and high-dimension expander (HDX)

- Markov chain comparison \( \implies \) the Gibbs sampler mixes in \( O(m^2 \log n) \) steps
Markov Chain for Connected Subgraphs

\[ G(\vec{p}) : \text{a subgraph of } G \text{ obtained by removing each } e \in E \text{ independently with prob. } p_e \]

- Monte Carlo method based on self-reduction:
  - Drawing samples \( C \sim G(\vec{p}) \) conditioned on \( C \) being connected on \( V \).
- The Gibbs sampler converges in \( O(m^2 \log n) \) steps.
- The matroid basis exchange chain converges in \( O(m \log n) \) steps [Anari, Liu, Oveis Gharan, Vinzant, STOC 2019 best paper]
  - Each step (matroid basis exchange) requires \( O(m) \) computation.
- Fastest estimation of network reliability runs in \( O(mn^2 \log n) \) [Guo, He, 2020]
Network Reliability Estimation

- Given an undirected graph (a network) $G(V, E)$, and a parameter $\vec{p} \in [0, 1]^E$:
  
  (all-terminal) network reliability: $R_{\vec{p}}(G) = \sum_{C \subseteq E \text{ that connects } V} \prod_{e \in C} (1 - p_e) \prod_{e \notin C} p_e$

  - the probability that the network remains connected when each edge $e \in E$ fails independently with prob. $p_e$

- Precisely evaluating $R_{\vec{p}}(G)$ is \#P-complete

- Approximation by Monte Carlo method in $\tilde{O}(mn^2)$ time

- Open problems:
  - estimating network reliability in $\tilde{O}(mn)$ time or less
  - network algorithms for network reliability (on going project ...)
Computational Phase Transition of Sampling

Network Algorithms for Gibbs Sampling

Application: Network Reliability Estimation
Thank you!

- [Chen, Feng, Y., Zhang '22]: Optimal mixing for two-state anti-ferromagnetic spin systems. FOCS '22.
- [He, Wang, Y. '22]: Sampling Lovász local lemma for general constraint satisfaction solutions in near-linear time. FOCS '22.
- [Liu, Y. '22]: Simple parallel algorithms for single-site dynamics. STOC '22.
- [Chen, Feng, Y., Zhang '21]: Rapid mixing of Glauber dynamics via spectral independence for all degrees. FOCS '21.
- [Feng, He, Y. '21]: Sampling constraint satisfaction solutions in the local lemma regime. STOC '21.
- [Feng, Hayes, Y. '21]: Distributed Metropolis sampler with optimal parallelism. SODA '21.
- [Feng, Guo, Y., Zhang '20]: Fast sampling and counting $k$-SAT solutions in the local lemma regime. STOC '20. JACM '21.
- [Feng, Vishnoi, Y. '19]: Dynamic sampling from graphical models. STOC '19. SICOMP '21.
- [Feng, Y. '18]: On local distributed sampling and counting. PODC '18.
- [Feng, Sun, Y. '17]: What can be sampled locally? PODC '17.