Advanced Algorithms

南京大学

尹一通
Monte Carlo method

**Universe** $U$, subset $S \subset U$ where $\rho = \frac{|S|}{|U|}$

Assume a *uniform sampler* for elements of $U$.

Estimate $Z = |S|$.

$\varepsilon$-approximation: $\hat{Z}$ is an $\varepsilon$-approximation of $Z = |S|$ if

$$(1 - \varepsilon)Z \leq \hat{Z} \leq (1 + \varepsilon)Z$$

**Montecarlo Method:**

sample $X_1, X_2, \ldots, X_N$ uniformly and independently from $U$

$$Y_i = \begin{cases} 1 & X_i \in S \\ 0 & X_i \notin S \end{cases} \quad \hat{Z} = \frac{|U|}{N} \sum_{i=1}^{N} Y_i$$
Universe \( U \), subset \( S \subseteq U \) where \( \rho = \frac{|S|}{|U|} \)

Estimate \( Z = |S| \).

\( \epsilon \)-approximation: \( (1 - \epsilon)Z \leq \hat{Z} \leq (1 + \epsilon)Z \)

Sample \( X_1, X_2, \ldots, X_N \) uniformly and independently from \( U \)

\[ Y_i = \begin{cases} 1 & X_i \in S \\ 0 & X_i \notin S \end{cases} \]

\[ \hat{Z} = \frac{|U|}{N} \sum_{i=1}^{N} Y_i \]

**Estimator Theorem:**

\[ N \geq \frac{4}{\epsilon^2 \rho} \ln \frac{2}{\delta} = \Theta \left( \frac{1}{\epsilon^2 \rho} \ln \frac{1}{\delta} \right) \quad \Rightarrow \quad \Pr[\hat{Z} \text{ is } \epsilon\text{-approx of } |S|] \geq 1 - \delta \]

Let \( Y = \sum_{i=1}^{N} Y_i \) then \( \mathbb{E}[Y] = \rho N \)

\[ \Pr[\hat{Z} \geq (1 + \epsilon)|S|] = \Pr[Y \geq (1 + \epsilon)\rho N] = \Pr[Y \geq (1 + \epsilon)\mathbb{E}[Y]] \]
**Chernoff Bound**

Independent \( X_1, X_2, \ldots, X_n \in \{0, 1\} \)

\[
X = \sum_{i=1}^{n} X_i \quad \mathbb{E}[X] = \mu
\]

\[0 < \delta \leq 1:\]

\[
\Pr[X \geq (1 + \delta)\mu] < \exp\left(-\frac{\mu\delta^2}{3}\right)
\]

\[
\Pr[X \leq (1 - \delta)\mu] < \exp\left(-\frac{\mu\delta^2}{2}\right)
\]

\[t \geq 2e\mu:\]

\[
\Pr[X \geq t] \leq 2^{-t}
\]
Universe $U$, subset $S \subseteq U$ where $\rho = |S|/|U|$

Estimate $Z = |S|$.

$\varepsilon$-approximation: $(1 - \varepsilon)Z \leq \hat{Z} \leq (1 + \varepsilon)Z$

Sample $X_1, X_2, \ldots, X_N$ uniformly and independently from $U$

$$Y_i = \begin{cases} 1 & X_i \in S \\ 0 & X_i \notin S \end{cases} \quad \hat{Z} = \frac{|U|}{N} \sum_{i=1}^{N} Y_i$$

Estimator Theorem:

$$N \geq \frac{4}{\varepsilon^2 \rho} \ln \frac{2}{\delta} = \Theta \left( \frac{1}{\varepsilon^2 \rho} \ln \frac{1}{\delta} \right) \quad \Pr[\hat{Z} \text{ is } \varepsilon\text{-approx of } |S|] \geq 1 - \delta$$

Let $Y = \sum_{i=1}^{N} Y_i$ \quad $\Pr[\hat{Z} \geq (1 + \varepsilon)|S|] = \Pr[Y \geq (1 + \varepsilon)\mathbb{E}[Y]]$

Chernoff: $\leq e^{-\frac{\varepsilon^2 \rho N}{3}}$

Similarly: $\Pr[\hat{Z} \leq (1 - \varepsilon)|S|] \leq e^{-\frac{\varepsilon^2 \rho N}{2}}$
Universe $U$, subset $S \subset U$ where $\rho = |S|/|U|$

Estimate $Z = |S|$.

$\mathbf{\epsilon}$-approximation: $(1 - \epsilon)Z \leq \hat{Z} \leq (1 + \epsilon)Z$

Sample $X_1, X_2, \ldots, X_N$ uniformly and independently from $U$

$$Y_i = \begin{cases} 1 & X_i \in S \\ 0 & X_i \notin S \end{cases} \quad \hat{Z} = \frac{|U|}{N} \sum_{i=1}^{N} Y_i$$

**Estimator Theorem:**

$$N \geq \frac{4}{\epsilon^2 \rho} \ln \frac{2}{\delta} = \Theta \left( \frac{1}{\epsilon^2 \rho} \ln \frac{1}{\delta} \right) \quad \Pr[\hat{Z} \text{ is } \mathbf{\epsilon}\text{-approx of } |S|] \geq 1 - \delta$$

$$\Pr[\hat{Z} \text{ is NOT an } \mathbf{\epsilon}\text{-approx of } |S|] \leq 2e^{-\frac{\epsilon^2 \rho n}{3}} = \delta$$
Counting DNF Solutions

**Input:** a DNF formula $\phi$

**Output:** the number of satisfying solutions to $\phi$.

- $n$ Boolean variables: $x_1, x_2, \ldots, x_n \in \{\text{true, false}\}$
- disjunctive normal form:
  \[
  \text{DNF} \quad \phi = C_1 \lor C_2 \lor \cdots \lor C_m
  \]
- $m$ clauses: $C_1, C_2, \cdots, C_m$
- each clause $C_i = l_{i_1} \land l_{i_2} \land \cdots \land l_{i_k}$
- each literal: $l_j \in \{x_r, \neg x_r\}$ for some $r$

Counting DNF is $\#P$-hard!
Counting DNF Solutions

**Input:** a DNF formula $\phi : \{T, F\}^n \to \{T, F\}$

**Output:** $Z = |\phi^{-1}(T)|$

**FPRAS (Fully Polynomial-time Randomized Approximation Scheme):**

$\forall \varepsilon: \text{return a } \hat{Z} \text{ in time } \text{Poly }(|\phi|, 1/\varepsilon)$

$$\Pr[(1 - \varepsilon)Z \leq \hat{Z} \leq (1 + \varepsilon)Z] \geq 2/3$$

[Chernoff]

$\forall \varepsilon, \delta: \text{return a } \hat{Z} \text{ in time } \text{Poly }(|\phi|, 1/\varepsilon, \log(1/\delta))$

$$\Pr[(1 - \varepsilon)Z \leq \hat{Z} \leq (1 + \varepsilon)Z] \geq 1 - \delta$$
Counting DNF Solutions

Input: a DNF formula $\phi : \{T, F\}^n \rightarrow \{T, F\}$

Output: $Z = |\phi^{-1}(T)|$

Problem: $\rho = |S| / |U|$ can be exponentially small

example: $\phi = C$ where $|C| = \Omega(n)$

the DNF is one big clause
Union of Sets

**Input**: a DNF formula \( \phi : \{T, F\}^n \rightarrow \{T, F\} \)

**Output**: \( Z = |\phi^{-1}(T)| \)

**DNF** \( \phi = C_1 \lor C_2 \lor \cdots \lor C_m \)

- each clause \( C_i = \ell_{i_1} \land \ell_{i_2} \land \cdots \land \ell_{i_k} \)

\[ S = \phi^{-1}(T) = S_1 \cup S_2 \cup \cdots \cup S_m \]

\[ S_i = C_i^{-1}(T) \text{ set of assignments satisfying clause } i \]

**Input** (implicitly): \( m \) sets \( S_1, S_2, \ldots, S_m \)

Estimate the size of \( S = \bigcup_{i=1}^{m} S_i \)
**Input (implicitly):** \( m \) sets \( S_1, S_2, \ldots, S_m \)

Estimate the size of \( S = \bigcup_{i=1}^{m} S_i \)

Multiset union

\[
U = S_1 \uplus S_2 \uplus \cdots \uplus S_m = \{ (x, i) \mid x \in S_i \}
\]

\[
\overline{S} = \{ (x, i^*) \mid x \in S_{i^*} \text{ and } x \in S_i \Rightarrow i^* \leq i \}
\]

\[
|\overline{S}| = |S| \quad \overline{S} \subseteq U \quad \rho = \frac{|\overline{S}|}{|U|} \geq \frac{1}{m}
\]

**Montecarlo Method:**

Sample uniform \((x,i) \in U\); (\( \propto N = \Theta(\frac{m}{\varepsilon^2} \ln \frac{1}{\delta}) \) times independently)

for each sample \((x,i)\): check if \((x,i) \in \overline{S}\); (and apply estimator Thm)
**Input** (implicitly): \( m \) sets \( S_1, S_2, \ldots, S_m \)

For every \( i \in \{1, 2, \ldots, m\} \), assume it is efficient:
- to compute \(|S_i|\); (sizes of individual sets)
- to sample uniform \( x \in S_i \); (uniform sampler)
- to check if \( x \in S_i \); (membership oracle)

Estimate the size of \( S = \bigcup_{i=1}^{m} S_i \)

**Coverage Algorithm** (Karp-Luby 1983):

Let \( U = \{(x, i) \mid x \in S_i\} \) and \( \bar{S} = \{(x, i^*) \mid x \in S_{i^*} \text{ and } x \in S_i \Rightarrow i^* \leq i\} \)

so \( |\bar{S}| = |S| , \bar{S} \subseteq U \) and \( \rho = |\bar{S}|/|U| \geq 1/m \).

Sample uniform \((x,i) \in U\); (\( \times N = \Theta\left(\frac{m}{\varepsilon^2} \ln \frac{1}{\delta}\right) \) times independently)
- sample \( i \in \{1, 2, \ldots, m\} \) proportional to \(|S_i|\);
- sample uniform \( x \in S_i \);

for each sample \((x,i)\): check if \((x,i) \in \bar{S}\); (and apply estimator Thm)
- check if \( x \in S_i \) and \( \forall j < i, x \notin S_j \);
**Input** (implicitly): $m$ sets $S_1, S_2, \ldots, S_m$

For every $i \in \{1, 2, \ldots, m\}$, assume it is efficient:
- to compute $|S_i|$; (sizes of individual sets)
- to sample uniform $x \in S_i$; (uniform sampler)
- to check if $x \in S_i$; (membership oracle)

Estimate the size of $S = \bigcup_{i=1}^{m} S_i$

$$
\text{DNF } \phi = C_1 \lor C_2 \lor \cdots \lor C_m
$$

- each clause $C_i = l_{i1} \land l_{i2} \land \cdots \land l_{ik}$

$$
S_i = C_i^{-1}(T) \text{ set of assignments satisfying clause } i
$$

$$
S = \phi^{-1}(T) = S_1 \cup S_2 \cup \cdots \cup S_m
$$

- size of each $S_i$; ✓
- uniform sampler for each $S_i$; ✓
- membership oracle for each $S_i$; ✓
Constraint Satisfaction Problem

- **variables**: $X = \{x_1, x_2, \ldots, x_n\}$
- **domain**: $\Omega$, usually $\Omega = [q]$ for a finite $q$
- **constraints**: $C = (\psi, S)$ where $\psi: \Omega^k \rightarrow \{0,1\}$ and scope $S \subseteq X$ is a subset of $k$ variables
- **CSP instance $I$**: a set of constraints defined on $X$
- **assignment**: $\sigma \in \Omega^X$ assigns values to variables
- **a constraint $C = (\psi, S)$ is satisfied** if $\psi(\sigma_S) = 1$
- **CSP solution**: an assignment $\sigma$ is a solution to a CSP instance if it satisfies all constraints.

**Input**: a CSP instance $I$.

**Output**: the number of CSP solutions.
Counting CSP

**Input:** a CSP instance $I$.

**Output:** the number of CSP solutions.

**Examples:**

- **Counting independent sets:** number of independent sets in a graph.
- **Counting matchings:** number of matchings in a graph.
- **Counting graph colorings:** number of proper $q$-colorings of a graph.
- **$\#SAT$:** number of satisfying assignments of a CNF.

They are all $\#P$-hard!
Approximate Counting

**Input:** a CSP instance $I$.

Estimate $Z = \text{number of CSP solutions}$.

**FPRAS** (Fully Polynomial-time Randomized Approximation Scheme):

$\forall \varepsilon, \delta$: return a $\widehat{Z}$ in time $\text{Poly}(|I|, 1/\varepsilon, \log(1/\delta))$

$$\Pr[(1 - \varepsilon)Z \leq \widehat{Z} \leq (1 + \varepsilon)Z] \geq 1 - \delta$$

**Computationally equivalent tasks** (under some conditions):

- **(approximate) counting**: estimate # of CSP solutions;
- **(almost) sampling**: sample a uniform random CSP solution;
- **(approximate) inference**: estimate the probability of a variable being a value in a CSP solution;
**Example: Independent Sets**

**Input**: an undirected graph $G(V,E)$.

Estimate the number of independent sets in $G$.

$Z(G) = \# \text{ of independent sets in } G$

suppose $V = \{v_1, v_2, \ldots, v_n\}$

let $V_i = \{v_1, v_2, \ldots, v_i\}$

$G_i = G[V_i]$ subgraph induced by $\{v_1, \ldots, v_i\}$

so $G_n = G$, $G_0 = \emptyset$ empty graph

$$Z(G) = \frac{Z(G_n)}{Z(G_{n-1})} \times \frac{Z(G_{n-1})}{Z(G_{n-2})} \times \cdots \times \frac{Z(G_1)}{Z(G_0)} = 1$$

$$\frac{Z(G_i)}{Z(G_{i-1})} = \frac{1}{\Pr[v_i \notin I]} \quad \text{for uniform random independent set } I \text{ in } G_i$$
Example: Independent Sets

**Input:** an undirected graph $G(V,E)$.

Estimate the number of independent sets in $G$.

$$Z(G) = \# \text{ of independent sets in } G$$

suppose $V = \{v_1, v_2, \ldots, v_n\}$ let $G_i = G[\{v_1, v_2, \ldots, v_i\}]$

$$Z(G) = \prod_{i=1}^{n} \frac{1}{\Pr_{I \text{ in } G_i}[v_i \notin I]}$$

for uniform random independent set $I$ in $G_i$

uniform sampler for independent sets in each $G_i$

$\Pr_{I \text{ in } G_i}[v_i \notin I]$ is not too small

FPRAS for $Z(G)$
Example: Independent Sets

**Input:** an undirected graph $G(V,E)$.
Estimate the number of independent sets in $G$.

$Z(G) = \# \text{ of independent sets in } G$

suppose $V = \{v_1, v_2, \ldots, v_n\}$  let $G_i = G[\{v_1, v_2, \ldots, v_i\}]$

$$Z(G) = \prod_{i=1}^{n} \frac{1}{\Pr_{I \text{ in } G_i} [v_i \notin I]}$$

if we have:

- uniform sampler for independent sets in each $G_i$

- $\Pr_{I \text{ in } G_i} [v_i \notin I] \geq \frac{1}{2}$

for uniform random independent set $I$ in $G_i$

FPRAS for $Z(G)$
**Input:** a CSP instance $I$.

Estimate $Z(I)$ the number of CSP solutions.

- **variables:** $V = \{x_1, x_2, \ldots, x_n\}$
- **domain:** $\Omega = [q]$ for a finite $q$

$X = (X_1, \ldots, X_n)$: uniform random CSP solution of $I$

$\sigma \in [q]^V$: a CSP solution of $I$

**uniform distribution:**

$$
\Pr[X = \sigma] = \frac{1}{Z(I)}
$$

**chain rule:**

$$
\Pr[X = \sigma] = \prod_{i=1}^{n} \Pr[X_i = \sigma_i \mid \forall j < i, X_j = \sigma_j]
$$

$$
Z(I) = \frac{1}{\prod_{i=1}^{n} \Pr[X_i = \sigma_i \mid \forall j < i, X_j = \sigma_j]}
$$
**Input:** a CSP instance $I$.

Estimate $Z(I)$ the number of CSP solutions.

$X = (X_1, \ldots, X_n)$: uniform random CSP solution of $I$

$\sigma \in [q]^V$: a CSP solution of $I$

$$Z(I) = \frac{1}{\prod_{i=1}^{n} \Pr[X_i = \sigma_i \mid \forall j < i, X_j = \sigma_j]}$$

for $i = 1, 2, \ldots, n$:

for every value $x \in [q]$:

estimate marginal probability $p_i = \Pr[X_i = x \mid \forall j < i, X_j = \sigma_j]$ by sampling uniform CSP solution $X$ conditioning on $X_1 = \sigma_1, X_2 = \sigma_2, \ldots, X_{i-1} = \sigma_{i-1}$; (from estimator Thm)

let $\sigma_i = $ the value $x$ with the largest estimated $\hat{p}_i$;

return $\hat{Z} = 1/(\prod_{i=1}^{n} \hat{p}_i)$
Input: a CSP instance $I$.
Estimate $Z(I)$ the number of CSP solutions.

- variables: $V = \{x_1, x_2, \ldots, x_n\}$
- domain: $\Omega = [q]$ for a finite $q$

$\forall S \subseteq V$ and $\forall \sigma \in [q]^V$:

Sampling: uniform CSP solution $X$ conditioning on $X_S = \sigma_S$

Inference: marginal probability
\[
\Pr[X_i = \sigma_i \mid X_S = \sigma_S]
\]

Counting: $Z(I)$
Random Sampling

**Input:** a CSP instance $I$
- on $n$ variables with domain $[q]$;
- Sample a uniform random CSP solution.

- **Independent set:** uniform independent set in a graph.
  (poly-time when max-deg $\leq 5$, NP-hard when max-deg $\leq 6$ or higher)

- **Matching:** uniform matching in a graph.
  (always poly-time)

- **Graph coloring:** uniform proper $q$-coloring of a graph.
  (NP-hard when $q < \text{max-deg}$)
  (Conjecture: poly-time when $q \geq \text{max-deg} + 1$)

- **SAT:** uniform satisfying assignments of a CNF.
  (Conjecture: poly-time when degree $\leq 2^{(1/2-o(1))k}$, NP-hard if otherwise)
Metropolis Algorithm

**Input:** a CSP instance $I$
on $n$ variables with domain $[q]$;
Sample a uniform random CSP solution.

**Metropolis-Hastings Algorithm:**
Initially, start with an arbitrary CSP solution; at each step, the current CSP solution is $\sigma=(\sigma_1, \ldots, \sigma_n)$:

- **(proposal)** pick a variable $i \in [n]$ and value $c \in [q]$ uniformly at random;
- **(filter)** accept the proposal and change $\sigma_i$ to $c$ if it does not violate any constraint;

return the solution $\sigma$ after $T$ steps;
Glauber Dynamics

**Input:** a CSP instance $I$

on $n$ variables with domain $[q]$;

Sample a uniform random CSP solution.

Glauber Dynamics:

Initially, start with an arbitrary CSP solution; at each step, the current CSP solution is $\sigma= (\sigma_1, \ldots, \sigma_n)$:

* pick a variable $i \in [n]$ uniformly at random;
* change value of $\sigma_i$ to a uniform value $c$ among all $\sigma_i$’s available values $c$: changing $\sigma_i$ to $c$ won’t violate any constraint;

return the solution $\sigma$ after $T$ steps;
Initially, start with an arbitrary independent set; at each step:

- (proposal) pick a vertex $v \in V$ and $b \in \{0,1\}$ uniformly at random;
- (filter) change $v$’s state to $b$ if the it gives an independent set;

**Metropolis-Hastings Algorithm:**
Initially, start with an arbitrary independent set; at each step:

- (proposal) pick a vertex $v \in V$ and $b \in \{0,1\}$ uniformly at random;
- (filter) change $v$’s state to $b$ if the it gives an independent set;

$$\sigma \in \{0,1\}^V$$
$$\forall \ uv \in E:$$
NOT $\sigma_u=\sigma_v=1$
Glauber Dynamics:
Initially, start with an arbitrary independent set; at each step:
- pick a uniform vertex \( v \in V \);
- change \( v \)'s state to a uniform random \( b \in \{0,1\} \) if all \( v \)'s neighbors have state 0;
Initially, start with an arbitrary proper $q$-coloring; at each step:

- (proposal) pick a vertex $v \in V$ and color $c \in [q]$ uniformly at random;
- (filter) change $v$'s color to $c$ if it gives a proper coloring;

**Metropolis-Hastings Algorithm:**
Initially, start with an arbitrary proper $q$-coloring; at each step:

- (proposal) pick a vertex $v \in V$ and color $c \in [q]$ uniformly at random;
- (filter) change $v$'s color to $c$ if it gives a proper coloring;
Glauber Dynamics:
Initially, start with an arbitrary proper $q$-coloring; at each step:

- pick a uniform vertex $v \in V$;
- change $v$'s color to a uniform random color $c$ among $v$'s current available colors;