Approximate Counting and Sampling

Brief Introduction with a Few Examples

劉爽

上海交通大學致遠學院 2012 級 ACM 班

2013.11.29

FPRAS

FPRAS

Definition ((ϵ, δ)-approximation)

A randomized algorithm gives an (ϵ, δ)-approximation for the value V if the output X satisfies:

$$Pr(|X - V| \leq \epsilon V) \geq 1 - \delta$$

FPRAS

Definition ((ϵ, δ)-approximation)

A randomized algorithm gives an (ϵ, δ)-approximation for the value V if the output X satisfies:

$$Pr(|X - V| \leq \epsilon V) \geq 1 - \delta$$

Definition (FPRAS)

A fully polynomial randomized approximation scheme (FPRAS) for a problem is a randomized algorithm for which, give an input X and any parameters $0 < \epsilon, \delta < 1$, the algorithm outputs an (ϵ, δ) approximation to $V(x)$ in $1/\epsilon$, $ln\delta^{-1}$ and the size of the input n.

Monte Carlo Method

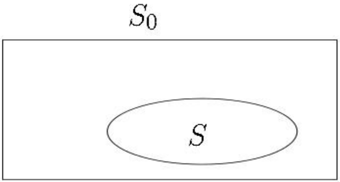
Monte Carlo Method

Definition (Monte Carlo Method)

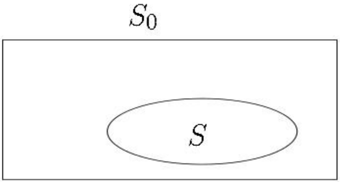
Use an efficient Process to generate a sequence of independent and identically distributed random samples with $\mathbb{E}[X_i] = V$. Get enough samples for an (ϵ,δ)-approximation for V.

Dart Throwing

Dart Throwing

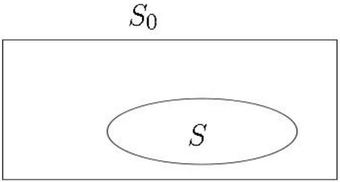


Dart Throwing



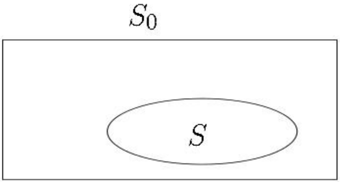
- Suppose we want to estimate $|S|$, we find a set $S_0 \supseteq S$, the size of which $|S_0|$ is known, and it is easy to pick a (near) random member of S_0

Dart Throwing



- Suppose we want to estimate $|S|$, we find a set $S_0 \supseteq S$, the size of which $|S_0|$ is known, and it is easy to pick a (near) random member of S_0
- To estimate $|S|$ we choose random points from S_0 and estimate $\frac{|S|}{|S_0|}$ by the proportion of samples that are in S.

Dart Throwing



- Suppose we want to estimate $|S|$, we find a set $S_0 \supseteq S$, the size of which $|S_0|$ is known, and it is easy to pick a (near) random member of S_0
- To estimate $|S|$ we choose random points from S_0 and estimate $\frac{|S|}{|S_0|}$ by the proportion of samples that are in S.
- How large should the ratio $\frac{|S|}{|S_0|}$ be?

Dart Throwing

Dart Throwing

- We generate M random variables X_i and take $\mu = \frac{\sum X_i}{M}$ as our estimation

Dart Throwing

- We generate M random variables X_i and take $\mu = \frac{\sum X_i}{M}$ as our estimation
- denote $V = |S|$ and $V_0 = |S_0|$, since $\mathbb{E}(X_i) = V$, we have $\mathbb{E}(\mu) = V$

Dart Throwing

- We generate M random variables X_i and take $\mu = \frac{\sum X_i}{M}$ as our estimation
- denote $V = |S|$ and $V_0 = |S_0|$, since $\mathbb{E}(X_i) = V$, we have $\mathbb{E}(\mu) = V$
- If we use Chebyshev inequality, we have

$$\Pr(|\mu - V| \geq \epsilon V) \leq \frac{Var(X)}{M\epsilon^2 V^2} \leq \delta$$

$$M \geq \frac{Var(X)}{V^2} \frac{1}{\epsilon^2 \delta}$$

Dart Throwing

- We generate M random variables X_i and take $\mu = \frac{\sum X_i}{M}$ as our estimation
- denote $V = |S|$ and $V_0 = |S_0|$, since $\mathbb{E}(X_i) = V$, we have $\mathbb{E}(\mu) = V$
- If we use Chebyshev inequality, we have

$$\Pr(|\mu - V| \geq \epsilon V) \leq \frac{Var(X)}{M\epsilon^2 V^2} \leq \delta$$

$$M \geq \frac{Var(X)}{V^2} \frac{1}{\epsilon^2 \delta}$$

- we need $\frac{Var(X)}{V^2}$ to be $poly(n)$

Dart Throwing

- We generate M random variables X_i and take $\mu = \frac{\sum X_i}{M}$ as our estimation
- denote $V = |S|$ and $V_0 = |S_0|$, since $\mathbb{E}(X_i) = V$, we have $\mathbb{E}(\mu) = V$
- If we use Chebyshev inequality, we have

$$\Pr(|\mu - V| \geq \epsilon V) \leq \frac{Var(X)}{M\epsilon^2 V^2} \leq \delta$$

$$M \geq \frac{Var(X)}{V^2} \frac{1}{\epsilon^2 \delta}$$

- we need $\frac{Var(X)}{V^2}$ to be $poly(n)$
- Since $Var(X) \leq V_0^2$, We only need $\frac{V}{V_0} = \frac{1}{poly(n)}$

Dart Throwing

Dart Throwing

- If we estimate $\frac{V}{V_0}$ instead and also suppose X_i takes value from $\{0,1\}$, we can use Chernoff bound, we have

$$\Pr(|\mu - \frac{V}{V_0}| \geq \epsilon \frac{V}{V_0}) \leq 2e^{-\frac{\epsilon^2 MV}{3V_0}} \leq \delta$$

$$M \geq \frac{3ln\frac{2}{\delta}}{\epsilon^2}\frac{V_0}{V}$$

Dart Throwing

- If we estimate $\frac{V}{V_0}$ instead and also suppose X_i takes value from $\{0, 1\}$, we can use Chernoff bound, we have

$$\Pr(|\mu - \frac{V}{V_0}| \geq \epsilon \frac{V}{V_0}) \leq 2e^{-\frac{\epsilon^2 MV}{3V_0}} \leq \delta$$

$$M \geq \frac{3ln\frac{2}{\delta}}{\epsilon^2} \frac{V_0}{V}$$

we need $\frac{V}{V_0}$ to be $\frac{1}{poly(n)}$

#DNF-SAT

#DNF-SAT

Definition (DNF)

In boolean logic, a disjunctive normal form (DNF) is a standardization (or normalization) of a logical formula which is a disjunction of conjunctive clauses

$$(A \wedge \neg B \wedge \neg C) \vee (\wedge D \wedge E \wedge F) \vee (\neg A \wedge F)$$

#DNF-SAT

#DNF-SAT

- Suppose there are n variables and k be the number of clauses

#DNF-SAT

- Suppose there are n variables and k be the number of clauses
- A first approach, choose S_0 be the all 2^n assignments.

#DNF-SAT

- Suppose there are n variables and k be the number of clauses
- A first approach, choose S_0 be the all 2^n assignments.
- $\frac{|S|}{2^n}$ should be $\frac{1}{poly(n)}$, but we don't know the size of S

#DNF-SAT

#DNF-SAT

- A better approach, construct a matrix

	assignment 1	assignment 2	...			assignment 2^n-1	assignment 2^n
clause 1	1	0	0	0	1	0	0
clause 2	0	1	0	0	0	0	0
clause 3	1	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	1	1	0	0	0	1
...	0	0	0	0	1	0	0
clause k-1	0	0	0	0	0	0	0
clause k	0	0	0	0	0	0	0

#DNF-SAT

- A better approach, construct a matrix

	assignment 1	assignment 2	...			assignment 2^n-1	assignment 2^n
clause 1	1	0	0	0	1	0	0
clause 2	0	1	0	0	0	0	0
clause 3	1	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	1	1	0	0	0	1
...	0	0	0	0	1	0	0
clause k-1	0	0	0	0	0	0	0
clause k	0	0	0	0	0	0	0

- $C_{ij} = 1$ if clause i can be satisfied by assignment j, 0 otherwise.

#DNF-SAT

- A better approach, construct a matrix

	assignment 1	assignment 2	...			assignment 2^n-1	assignment 2^n
clause 1	1	0	0	0	1	0	0
clause 2	0	1	0	0	0	0	0
clause 3	1	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	1	1	0	0	0	1
...	0	0	0	0	1	0	0
clause k-1	0	0	0	0	0	0	0
clause k	0	0	0	0	0	0	0

- $C_{ij} = 1$ if clause i can be satisfied by assignment j, 0 otherwise.
- $S_0 = \{$number of ones in the matrix$\}$
  $S = \{$columns contain at least one$\}$
- In other words
  $S = \{$the uppermost one in each column$\}$

#DNF-SAT

#DNF-SAT

- First we show $\frac{|S|}{|S0|}$ is not too small

$$\frac{|S|}{|S_0|} \geq \frac{1}{k}$$

#DNF-SAT

- First we show $\frac{|S|}{|S0|}$ is not too small

$$\frac{|S|}{|S_0|} \geq \frac{1}{k}$$

- Then we show elements in S_0 can be generated uniformly at random

#DNF-SAT

#DNF-SAT

- Suppose there are R_i ones in row i, then we have $R_i = 2^{n-d_i}$, where d_i is the number of variables in the i_{th} clause. Let $R = \sum R_i = |S_0|$.

#DNF-SAT

- Suppose there are R_i ones in row i, then we have $R_i = 2^{n-d_i}$, where d_i is the number of variables in the i_{th} clause. Let $R = \sum R_i = |S_0|$.
- We choose row i with probability $\frac{R_i}{R}$, and then determine each of the variable not in clause i uniformly at random, thus get the 2^{n-d_i} ones in column i uniformly at random.

#DNF-SAT

- Suppose there are R_i ones in row i, then we have $R_i = 2^{n-d_i}$, where d_i is the number of variables in the i_{th} clause. Let $R = \sum R_i = |S_0|$.
- We choose row i with probability $\frac{R_i}{R}$, and then determine each of the variable not in clause i uniformly at random, thus get the 2^{n-d_i} ones in column i uniformly at random.
- Finally, if the chosen one is the uppermost one in its column, we let X_i be 1, otherwise be 0.

#DNF-SAT

#DNF-SAT

- If we repeat M times, our estimation thus is $\frac{\sum X_i}{M}|S_0|$

#DNF-SAT

- If we repeat M times, our estimation thus is $\frac{\sum X_i}{M}|S_0|$
- Give ϵ and δ, Let $M = \frac{3kln\frac{2}{\delta}}{\epsilon^2}$, this algorithm gives an (ϵ, δ)-approximation, thus is a FPRAS.

FPAUS

FPAUS

Definition (Variation Distance)

The variation distance between two probability distributions π and π' on a countable state space S is given by

$$\|\pi - \pi'\| = \frac{1}{2} \sum_{x \in S} |\pi(x) - \pi'(x)|$$

FPAUS

Definition (Variation Distance)

The variation distance between two probability distributions π and π' on a countable state space S is given by

$$\|\pi - \pi'\| = \frac{1}{2} \sum_{x \in S} |\pi(x) - \pi'(x)|$$

Definition (FPAUS)

An almost uniform sampler (AUS) is a randomized algorithm that takes as input of size n and a tolerance δ, and produces a random event $F \in \Omega(x)$, such that the probability distribution of F is within variation distance δ of the desired distribution on $\Omega(x)$. A fully polynomial almost uniformly sampler is an AUS runs in poly-time in n and $ln\delta^{-1}$

Markov chain Monte Carlo Method

When the samples cannot be sampled "directly", we often use the following method

Definition (MCMC)

The Markov chain Monte Carlo (MCMC) Method runs as follows: Define an ergodic Markov Chain $\mathcal{M}$ with states the elements of the sample space, $\mathcal{M}$ must converge to the requered distribution fast enough (as a FPAUS). From any starting state and after a sufficient number of steps r the distribution of X_r will be close to the stationary distribution, and we use as almost independent samples $X_r, X_{2r}, X_{3r} \dots$

K-Colouring

K-Colouring

- Let G $=$ (V, E) be a graph of maximum degree Δ. We want to uniformly at rondom sample proper k-colourings of G (use k colours to paint the vertices so that any adjacent vertex have different colours). Here we will assume that $k > 2\Delta$.

K-Colouring

K-Colouring

Lemma

For a finite space Ω and neighborhood structure $\{N(x)|x \in \Omega|\}$. Let $N = max_{x\in\Omega}|N(x)|$.Let $M \geq N$. If the following MC is irreducible, aperiodic then the stationary distribution is the uniform distribution.

$$P_{x,y} = \begin{cases} \frac{1}{M} & \text{if } x \neq y \text{ and } y \in N(x), \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x), \\ 1 - \frac{N(x)}{M} & \text{if } x = y, \end{cases}$$

K-Colouring

K-Colouring

- Let's define a Markov chain on $\Omega(k, G)$.

K-Colouring

- Let's define a Markov chain on $\Omega(k, G)$.
- **Step 0** X_0 is any k-colouring of G
- **Step t** Choose v uniformly at random from V and i uniformly at random from 1 to k, try to re-colour v in X_{t-1} with colour i. If succeed, X_t should be the re-coloured graph, otherwise, $X_t = X_{t-1}$.

K-Colouring

- Let's define a Markov chain on $\Omega(k, G)$.
- **Step 0** X_0 is any k-colouring of G
- **Step t** Choose v uniformly at random from V and i uniformly at random from 1 to k, try to re-colour v in X_{t-1} with colour i. If succeed, X_t should be the re-coloured graph, otherwise, $X_t = X_{t-1}$.
- This chain obviously satisfies of the requirements of the previous lemma.

K-Colouring

K-Colouring

Definition (Mixing Time)

Let p_x^t be the distribution of the state of a Markov Chain starting at x after t steps and π be the desired stationary distribution. We define

$$\Delta_X(t) = \|p_x^t - \pi\|$$

We define $\tau_X(\epsilon) = min\{t|\Delta_X(t) \leq \epsilon\}$ and the mixing time $\tau(\epsilon) = max_{X \in S}\tau_X(\epsilon)$.
A chain is called rapidly mixing if $\tau(\epsilon)$ is polynomial in $\frac{1}{\epsilon}$ and the size of the problem n.

Jerrum showed that if $k > 2\Delta(G)$ then the mixing time of the chain defined previously is $O(kn\log n)$

From good sampling to approximate counting

From good sampling to approximate counting

Now we want to estimate the number of k-colourings of a graph.

Theorem

Suppose we have a AUS for the k-colouring of a graph, which works for graphs G with maximum degree bounded by Δ and suppose that the sampler has time complexity $T(n, \delta)$. Then we may construct an FPRAS for the number of k-colouring of a graph, with degree bound Δ and time complexity

$$O(\frac{m^2}{\epsilon^2} T(n, \frac{\epsilon}{6m}))$$

where m is the number of edges in G and ϵ the specified error bound.

From good sampling to approximate counting

From good sampling to approximate counting

Proof outline.

Let $\Omega(k, G)$ denote the set of proper k-colourings of G. Let $E = \{e_1, e_2, \ldots, e_m\}$ and let $G_i = (V, \{e_1, e_2, \ldots, e_i\})$. Then

$$|\Omega(k, G)| = |\Omega(k, G_0)| \prod_{i=1}^{m} \frac{|\Omega(k, G_i)|}{|\Omega(k, G_{i-1}|}$$

Since $|\Omega(k, G_0)| = k^{|V|}$, we only need to estimate the ratios

$$\rho_i = \frac{|\Omega(k, G_i)|}{|\Omega(k, G_{i-1})|}$$

Then we can use MCMC to choose near random members of $\Omega(k, G_{i-1})$ and seeing what proportion are also in $\Omega(k, G_i)$. Since $k > 2\Delta$, $\rho_i > \frac{1}{2}$, which prevents the failure of the sampling. □

From approximate counting to good sampling

Let's take the number of independent sets as our example (which is quite similar to k-colouring)

From approximate counting to good sampling

Let's take the number of independent sets as our example (which is quite similar to k-colouring)

Theorem

Suppose we have an FPRAS APPROXCOUNT(G, ϵ, δ) for the number of independent sets of a graph $G = (V, E)$ and suppose that APPROXCOUNT(G, ϵ, δ) has time complexity $T(n, \epsilon, \delta)$. Then we can construct a AUS U_{GEN} which has expectd time complexity

$$O(T(n, O(\frac{1}{n}), O(\frac{\delta}{n})))$$

From approximate counting to good sampling

From approximate counting to good sampling

Proof Idea.

Construct an algorithm as follows: There is a function called U_{GENX}, we call it repeatly untill we get an sample. The function U_{GENX} has an argument ϕ roughly means the probablity of failure.

$v = \max V$ and X is the set of neighbours of v in G.
$G_1 = G - v - X$ and $G_2 = G - v$
$N_1 = \text{APPROXCOUNT}(G_1, \epsilon_1, \delta_1)$ and $N_2 = \text{APPROXCOUNT}(G_2, \epsilon_1, \delta_1)$

$$\textbf{Output } I = \begin{cases} v + \text{UGENX}\left(G_1, \epsilon_1, \phi\frac{N_1+N_2}{N_1}\right) & \text{probability } \frac{N_1}{N_1+N_2} \\ \text{UGENX}\left(G_2, \epsilon_1, \phi\frac{N_1+N_2}{N_2}\right) & \text{probability } \frac{N_2}{N_1+N_2} \end{cases}$$

□

Thank You