Approximate Counting and Markov Chain Monte Carlo

A Randomized Approach

Arindam Pal

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

March 18, 2011
April 8, 2011

Agenda

- Introduction to Monte Carlo Method
- Approximating the Value of π
- The Complexity Classes $\#P$ and $\#P$-complete
- Fully Polynomial Randomized Approximation Scheme
- DNF Counting Problem
- Introduction to Markov Chains
- Random Walks on Graphs
- Markov Chain Monte Carlo Method
- Counting the Number of KNAPSACK Solutions
- Rapidly Mixing Markov Chains
- Mixing Time, Canonical Paths and Conductance

The Monte Carlo Method

- Uses random sampling to estimate the value of a quantity.
- Primarily used in numerical analysis and simulation.
- Widely used in Physical sciences, Engineering, Computational Biology and Statistics among many others.
- A classic example is to estimate the value of π.
- Other examples include evaluating a definite integral (area under a curve), solving optimization problems using random walks.
- Simulated annealing and the Metropolis algorithm are good heuristics for approximating the global optimum of a given function in a large search space.

Algorithms for Approximate Counting Problems

- DNF counting problem. (Karp and Luby)
- Network reliability. (Karger)
- Counting the number of Knapsack solutions. (Dyer, Frieze, Kannan, Kapoor, Perkovic and Vazirani)
- Approximating the Permanent. (Jerrum and Sinclair)
- Estimating the volume of a convex body. (Kannan, Lovasz and Simonovits)

Approximating the Value of π

- Let $S = \{(x, y) \in \mathbb{R}^2 : |x| \leq 1, |y| \leq 1\}$ be the 2×2 square centered at $(0, 0)$.
- Let $C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ be the unit circle centered at $(0, 0)$.
- Choose a point $(x, y) \in S$ uniformly at random. This is equivalent to choosing x and y independently from a uniform distribution on the interval $[-1, 1]$.
- Define a random variable,

$$Z_i = \begin{cases} 1 & \text{if } (x, y) \in C \text{ in the } i^{th} \text{ iteration,} \\ 0 & \text{otherwise.} \end{cases}$$

- Clearly, $\mathrm{E}[Z_i] = \Pr[Z_i = 1] = \frac{\Delta(C)}{\Delta(S)} = \frac{\pi}{4}$.

Analysis

- Suppose we do this experiment m times independently.
- Let $Z = \sum_{i=1}^{m} Z_i$.
- $\mathrm{E}[Z] = \sum_{i=1}^{m} \mathrm{E}[Z_i] = \frac{m\pi}{4}$.
- Let $X = \left(\frac{4}{m}\right) Z$. Then, $\mathrm{E}[X] = \left(\frac{4}{m}\right) \mathrm{E}[Z] = \frac{4}{m} \cdot \frac{m\pi}{4} = \pi$.
- Applying Chernoff bound,

$$\begin{aligned}
\Pr[|X - \pi| \geq \varepsilon\pi] &= \Pr\left[\left|Z - \frac{m\pi}{4}\right| \geq \frac{\varepsilon m\pi}{4}\right] \\
&= \Pr[|Z - \mathrm{E}[Z]| \geq \varepsilon \mathrm{E}[Z]] \\
&\leq 2e^{-m\pi\varepsilon^2/12}.
\end{aligned}$$

- Since the error probability decreases exponentially with the number of trials m, we can get a very good approximation to π with high probability.

The complexity classes $\#P$ and $\#P$-complete

Definition

A problem $\Pi \in \#P$ if there is a nondeterministic polynomial time Turing machine that for any instance I of Π has a number of accepting computations exactly equal to the number of distinct solutions to I. Π is $\#P$-complete if any problem $\Gamma \in \#P$ can be reduced to Π by a polynomial time Turing reduction relating the cardinalities of solution sets.

Importance of $\#P$-completeness

- The class $\#P$ consists of all counting problems associated with the decision problems in NP.
- There are *easy* problems in $\#P$, like counting the number of spanning trees of a graph, which can be solved in polynomial time using *Kirchhoff's Matrix Tree Theorem*.
- If a $\#P$-complete problem can be solved in polynomial time, then $P = NP$.
- There is no known efficient deterministic approximation algorithm for any $\#P$-complete problem.
- But there are efficient randomized approximation algorithms for many $\#P$-complete problems.

Kirchhoff's Matrix Tree Theorem

Theorem

The number of spanning trees $\tau(G)$ of a graph G on n vertices is the absolute value of the determinant of any $(n-1) \times (n-1)$ submatrix (cofactor) of the Laplacian matrix $L(G)$. Equivalently, if $\lambda_1, \ldots, \lambda_{n-1}$ are the non-zero eigenvalues of $L(G)$, then

$$\tau(G) = \frac{1}{n} \prod_{i=1}^{n-1} \lambda_i.$$

- $L(G) = \Delta(G) - A(G)$, where $\Delta(G)$ and $A(G)$ are the degree matrix and the adjacency matrix of G respectively.
- Using this, one can prove Cayley's formula: $\tau(K_n) = n^{n-2}$.

Fully Polynomial Randomized Approximation Scheme (FPRAS)

Definition

An (ε, δ)-Fully Polynomial Randomized Approximation Scheme (FPRAS) for a problem is a randomized algorithm A which, given an input x and two parameters ε, δ $(0 < \varepsilon, \delta < 1)$, outputs a value $A(x)$ corresponding to the actual value $V(x)$ such that

$$\Pr[|A(x) - V(x)| \leq \varepsilon V(x)] \geq 1 - \delta,$$

in time polynomial in $|x|$, $\frac{1}{\varepsilon}$ and $\log(\frac{1}{\delta})$.

DNF Counting Problem

- F is a boolean formula in disjunctive normal form (DNF) over n boolean variables $x_1, \ldots, x_n$.
- $F = C_1 \vee \ldots \vee C_t$ is a disjunction of clauses C_i.
- Each clause $C_i = l_1 \wedge \cdots \wedge l_{r_i}$ of r_i literals.
- Each literal l_j is either a variable x_k or it's complement $\overline{x_k}$.
- $\#F$ is the number of distinct satisfying assignments of F.
- Our goal is to compute $\#F$. Note that $0 \leq \#F \leq 2^n$.
- Computing $\#F$ is $\#P$-complete.
- Design an (ε, δ)-FPRAS with running time polynomial in $n, t, \frac{1}{\varepsilon}$ and $\log(\frac{1}{\delta})$.

A Naive Random Sampling Algorithm

1. $X \leftarrow 0$.
2. for $k = 1, \ldots, m$
3. for $i = 1, \ldots, n$
4. set $x_i \leftarrow 1$ with probability $\frac{1}{2}$.
5. end for
6. if this random assignment satisfies F
7. $X \leftarrow X + 1$.
8. end for
9. return $Y \leftarrow \left(\frac{X}{m}\right) 2^n$.

Analysis of the Naive Algorithm

- Define a random variable,

$$X_k = \begin{cases} 1 & \text{if } F \text{ is satisfied in the } k^{th} \text{ iteration,} \\ 0 & \text{otherwise.} \end{cases}$$

- X_k-s are independent $0-1$ random variables.
- $X = \sum_{k=1}^{m} X_k$.
- $\mathrm{E}[X_k] = \Pr[X_k = 1] = \frac{\#F}{2^n}$.

$$\begin{aligned} \mathrm{E}[Y] &= \left(\frac{2^n}{m}\right) \mathrm{E}[X] \\ &= \left(\frac{2^n}{m}\right) \sum_{k=1}^{m} \mathrm{E}[X_k] \\ &= \left(\frac{2^n}{m}\right) \left(\frac{m}{2^n}\right) \#F \\ &= \#F. \end{aligned}$$

Problem with the Naive Algorithm

- X_k is a Bernoulli distribution with parameter $p = \frac{\#F}{2^n}$.
- X is a binomial distribution with parameters m and p.
- By Chernoff bound,

$$\begin{aligned} & \Pr[(1-\varepsilon)\#F \leq Y \leq (1+\varepsilon)\#F] \\ = \; & \Pr[(1-\varepsilon)p2^n \leq \left(\frac{X}{m}\right)2^n \leq (1+\varepsilon)p2^n] \\ = \; & \Pr[(1-\varepsilon)mp \leq X \leq (1+\varepsilon)mp] \\ \geq \; & 1 - 2e^{-mp\varepsilon^2/4}. \end{aligned}$$

- For this to be at least $1-\delta$, we must have,

$$m \geq \frac{4}{p\varepsilon^2}\ln\frac{2}{\delta} = \frac{4\cdot 2^n}{\#F\cdot\varepsilon^2}\ln\frac{2}{\delta}.$$

- If $\#F$ is sub-exponential (e.g. a polynomial) in n, this is an exponential-time algorithm and hence is not an FPRAS.

Towards an FPRAS for DNF Counting

- $F = C_1 \vee \ldots \vee C_t$.
- If $|C_i| = r_i$, there are 2^{n-r_i} satisfying assignments for C_i.
- Let SC_i be the set of assignments satisfying C_i.
- Let $U = \{(i, a) : 1 \leq i \leq t \text{ and } a \in SC_i\}$.
- $|U| = \sum_{i=1}^{t} |SC_i|$ can be computed efficiently.
- We want to estimate $\#F = |\bigcup_{i=1}^{t} SC_i|$.
- Note that, $\#F \leq |U|$, since an assignment can satisfy more than one clause and so can appear in more than one pair in U.

Sampling More Carefully

- The idea is to define a set $S \subseteq U$ with $|S| = \#F$.
- $S = \{(i, a) : 1 \leq i \leq t \text{ and } a \in SC_i, \text{ but } a \notin SC_j \text{ for } j < i\}$.
- We can estimate $|S|$ by estimating the ratio $\frac{|S|}{|U|}$, since we know $|U|$.
- Sample uniformly at random from U and count how many times they are in S.
- S is *relatively dense* in U, since each assignment can satisfy at most t different clauses, $\frac{|S|}{|U|} \geq \frac{1}{t}$.

How to Sample Uniformly at Random from U?

- First choose i with probability $\frac{|SC_i|}{|U|}$.
- Then choose an assignment $a \in SC_i$ uniformly at random.
- This can be done by setting each literal $l \notin SC_i$ to 1 with probability $\frac{1}{2}$ (independently and uniformly at random).

$$\begin{aligned}\Pr[(i,a) \text{ is chosen}] &= \Pr[i \text{ is chosen}] \cdot \Pr[a \text{ is chosen} | i \text{ is chosen}] \\ &= \frac{|SC_i|}{|U|} \cdot \frac{1}{|SC_i|} \\ &= \frac{1}{|U|}.\end{aligned}$$

- This method samples uniformly at random from U.

An FPRAS for DNF Counting

1. $X \leftarrow 0$.
2. for $k = 1, \ldots, m$
3. with probability $\frac{|SC_i|}{|U|}$, choose uniformly at random, an
4. assignment $a \in SC_i$.
5. if $a \notin SC_j$ for all $j < i$
6. $X \leftarrow X + 1$.
7. end for
8. return $Y \leftarrow \left(\frac{X}{m}\right)|U|$.

Analysis of the FPRAS

- A similar analysis shows that

$$m \geq \frac{4t}{\varepsilon^2} \ln \frac{2}{\delta}.$$

- Since the running time is polynomial in $n, t, \frac{1}{\varepsilon}$ and $\log(\frac{1}{\delta})$, this is an (ε, δ)-FPRAS.

Introduction to Markov Chains

Definition

A finite *Markov chain* $\mathfrak{M}$ is a discrete-time stochastic process defined over a finite set of n states Ω and an $n \times n$ matrix P of *transition probabilities*. If X_t is the state of $\mathfrak{M}$ at time t, then the *memorylessness property* states that,

$$\begin{aligned}&\Pr[X_{t+1} = j | X_0 = i_0, \ldots, X_{t-1} = i_{t-1}, X_t = i] \\ &= \Pr[X_{t+1} = j | X_t = i] = P_{ij}.\end{aligned}$$

- $\mathfrak{M}$ makes state transitions at discrete time steps $t = 1, 2, \ldots$
- P has one row and one column for each state in Ω.
- The entry P_{ij} is the probability that the next state will be j, given that the current state is i.
- For all $i, j \in \Omega, 0 \leq P_{ij} \leq 1$ and $\sum_{j=1}^{n} P_{ij} = 1$.

Classification of States

- For $n \geq 0$, the n-step transition probability is defined as $P_{ij}^n = \Pr[X_{t+n} = j | X_t = i]$.
- State j is *accessible* from state i ($i \to j$), if for some $n \geq 0, P_{ij}^n > 0$. If $i \to j$ and $j \to i$, then we say that i and j *communicate* ($i \leftrightarrow j$). This is an *equivalence relation*.
- A Markov chain is *irreducible* if all states belong to one communicating class. This happens if and only if its graph representation is *strongly connected*.
- Let r_{ij}^t be the probability that starting at state i, the first transition to state j occurs at time t. More precisely,

$$r_{ij}^t = \Pr[X_t = j, X_s \neq j \text{ for } 1 \leq s \leq t-1 | X_0 = i].$$

Classification of States continued...

- A state i is *recurrent* (*persistent*) if $f_{ii} = \sum_{t=1}^{\infty} r_{ii}^t = 1$. Else, it is *transient*. A Markov chain is *recurrent* if every state is recurrent.
- The expected time to first reach j, after starting at i is $h_{ij} = \sum_{t=1}^{\infty} t \cdot r_{ij}^t$. This is called the *hitting time*.
- A recurrent state i is *positive recurrent* (*non-null persistent*) if $h_{ii} < \infty$. Otherwise, it is *null recurrent* (*null persistent*).
- Note that $f_{ii} = 1$ does not mean that $h_{ii} < \infty$.
- A state i is called *periodic*, if there exists an integer $T > 1$ such that $P_{ii}^n = \Pr[X_{t+n} = i | X_t = i] = 0$, unless $T|n$. T is called the *period* of i. Otherwise, i is called *aperiodic*. A Markov chain is called *aperiodic* if all its states are aperiodic.

Ergodic Markov Chains and Stationary Distributions

- An aperiodic and positive recurrent state is called an *ergodic* state. A Markov chain is called *ergodic* if all its states are ergodic.
- A probability distribution π is called a *stationary distribution* for $\mathfrak{M}$ with transition matrix P, if $\pi P = \pi$.

Fundamental Theorem of Markov Chains

Any finite, irreducible and aperiodic Markov chain is ergodic and has a unique stationary distribution π.

Random Walks on Graphs

- Let $G = (V, E)$ be a connected, non-bipartite, undirected graph, where $|V| = n$ and $|E| = m$.
- This induces a Markov chain $\mathfrak{M}_G$.
- The states of $\mathfrak{M}_G$ are the vertices of G.
- For any two vertices $u, v \in V$,

$$P_{uv} = \begin{cases} \frac{1}{d(u)} & \text{if } (u, v) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

- $\mathfrak{M}_G$ is ergodic with stationary distribution π given by,

$$\pi_v = \frac{d(v)}{2m}.$$

Counting the Number of KNAPSACK Solutions

- Let $a = (a_0, \ldots, a_{n-1}) \in \mathbb{N}^n$ be an n-dimensional integer vector and let $b \in \mathbb{N}$ be any integer.
- Given an inequality $a \cdot x = \sum_{i=0}^{n-1} a_i x_i \leq b$, where $x \in \{0,1\}^n$.
- Compute the number N of such vectors x.
- Suppose $a_0, \ldots, a_{n-1}$ are the sizes of n items that can be packed into a knapsack of size b.
- N is the number of combinations of items that can be fitted into the knapsack.
- We have to count the number of KNAPSACK solutions.
- This problem is $\#P$-complete.
- Note that $\sum_{i=0}^{n-1} a_i > b$. Else, $\sum_{i=0}^{n-1} a_i x_i \leq \sum_{i=0}^{n-1} a_i \leq b$, and the number of solutions is 2^n.

A Naive Random Sampling Algorithm

1. $X \leftarrow 0$.
2. for $k = 1, \ldots, m$
3. for $i = 0, \ldots, n-1$
4. set $x_i \leftarrow 1$ with probability $\frac{1}{2}$.
5. end for
6. if this random assignment satisfies $a \cdot x \leq b$
7. $X \leftarrow X + 1$.
8. end for
9. return $Y \leftarrow \left(\frac{X}{m}\right) 2^n$.

Problem with this Approach

- Take $a = (1, \ldots, 1)$ and $b = \frac{n}{3}$.
- The expected number of trials before the event $a \cdot x \leq b$ occurs for the first time is exponential in n.
- A sequence of trials of "reasonable" length will typically yield a mean close to 0, even though the actual number of Knapsack solutions may be exponentially large.
- The variance of the estimator is too large for it to be of any practical value.

Analysis of the Naive Algorithm

- Define a random variable,

$$X_k = \begin{cases} 1 & \text{if } a \cdot x \leq b \text{ in the } k^{th} \text{ iteration,} \\ 0 & \text{otherwise.} \end{cases}$$

- X_k-s are independent $0-1$ random variables.
- $X = \sum_{k=1}^{m} X_k$.
- $\mathrm{E}[X_k] = \Pr[X_k = 1] = \frac{N}{2^n}$.

$$\begin{aligned} \mathrm{E}[Y] &= \left(\frac{2^n}{m}\right) \mathrm{E}[X] \\ &= \left(\frac{2^n}{m}\right) \sum_{k=1}^{m} \mathrm{E}[X_k] \\ &= \left(\frac{2^n}{m}\right) \left(\frac{m}{2^n}\right) N \\ &= N. \end{aligned}$$

Problem with the Naive Algorithm

- X_k is a Bernoulli distribution with parameter $p = \frac{N}{2^n}$.
- X is a binomial distribution with parameters m and p.
- By Chernoff bound,

$$\begin{aligned} & \Pr[(1-\varepsilon)N \leq Y \leq (1+\varepsilon)N] \\ = \; & \Pr[(1-\varepsilon)p2^n \leq \left(\frac{X}{m}\right) 2^n \leq (1+\varepsilon)p2^n] \\ = \; & \Pr[(1-\varepsilon)mp \leq X \leq (1+\varepsilon)mp] \\ \geq \; & 1 - 2e^{-mp\varepsilon^2/4}. \end{aligned}$$

- For this to be at least $1-\delta$, we must have,

$$m \geq \frac{4}{p\varepsilon^2} \ln \frac{2}{\delta} = \frac{4 \cdot 2^n}{N \cdot \varepsilon^2} \ln \frac{2}{\delta}.$$

- If N is sub-exponential (e.g. a polynomial) in n, this is an exponential-time algorithm and hence is not an FPRAS.

A Markov Chain Monte Carlo Algorithm

Consider the Markov chain $\mathfrak{M}_{\text{KNAPSACK}}$ with state space $\Omega = \{x \in \{0,1\}^n : a \cdot x \leq b\}$ and transitions from a state $x = (x_0, \ldots, x_{n-1}) \in \Omega$ to another state $y = (y_0, \ldots, y_{n-1}) \in \Omega$ is defined by the following rules:

State transition rules for $\mathfrak{M}_{\text{KNAPSACK}}$

1. set $y = x$ with probability $\frac{1}{2}$.
2. select i uniformly at random from the range $0 \leq i \leq n-1$.
3. let $y' = (x_0, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_{n-1})$.
4. if $a \cdot y' \leq b$
5. $\quad y = y'$.
6. else
7. $\quad y = x$.

Properties of $\mathfrak{M}_{\text{KNAPSACK}}$

- $\mathfrak{M}_{\text{KNAPSACK}}$ may be interpreted as a random walk (with stationary moves) on the n-dimensional boolean hypercube with vertex set $\{0,1\}^n$, truncated by the hyperplane $a \cdot x \leq b$.
- It is ergodic, since all pairs of states intercommunicate via the state $(0,\ldots,0)$, and the presence of loops ensures aperiodicity.
- It can be easily checked that the stationary distribution is uniform over Ω.
- Starting in state $(0,\ldots,0)$, simulate $\mathfrak{M}_{\text{KNAPSACK}}$ for sufficiently many steps until the distribution over states is "close" to the uniform distribution.
- Return the current state as the result.
- This gives a procedure for sampling KNAPSACK solutions "almost" uniformly at random.

Product Estimators

- Let V be the set of elements we want to enumerate. The size of V is typically exponentially large in terms of the natural size k of the problem. Suppose, we can find a chain of subsets $V_0 \subset V_1 \subset \cdots \subset V_m = V$ such that for each i,
  - $|V_0|$ is known (usually $|V_0| = 1$).
  - $\frac{|V_{i+1}|}{|V_i|}$ is polynomially bounded in k.
  - m is polynomially bounded.
  - There is a polynomial-time oracle to generate a random element uniformly distributed over V_i, for each $i : 1 \leq i \leq m$.
- Then we can estimate the ratios $\frac{|V_{i+1}|}{|V_i|}$ by generating a polynomial number of elements of V_{i+1} and counting how often we hit V_i. The product of these estimates and of $|V_0|$ gives an estimate for $|V|$.
- This scheme typically results in a FPRAS.

From Sampling to Estimation of $|\Omega|$

- We keep the vector a fixed, but allow the bound b to vary.
- Let $\mathfrak{M}_{\text{KNAPSACK}}(b)$ and $\Omega(b)$ be the Markov chain and its state space as functions of b.
- Assume without loss of generality that $a_0 \leq a_1 \leq \ldots \leq a_{n-1}$.
- Define $b_0 = 0$ and $b_i = \min\{b, \sum_{j=0}^{i-1} a_j\}$, for $1 \leq i \leq n$.
- Note that $b_n = b$, since $\sum_{j=0}^{n-1} a_j > b$.
- It is easy to see that $|\Omega(b_{i-1})| \leq |\Omega(b_i)| \leq (n+1)|\Omega(b_{i-1})|$, for $1 \leq i \leq n$, since any element of $\Omega(b_i)$ can be converted to an element of $\Omega(b_{i-1})$ by changing the rightmost 1 to a 0. Conversely, from any element of $\Omega(b_{i-1})$ we can get at most $n+1$ elements of $\Omega(b_i)$.

Estimating the Value of $|\Omega|$

$$
\begin{aligned}
|\Omega(b)| &= |\Omega(b_n)| \\
&= |\Omega(b_0)| \prod_{i=1}^{n} \frac{|\Omega(b_i)|}{|\Omega(b_{i-1})|} \\
&= \prod_{i=1}^{n} \frac{1}{\rho_i}.
\end{aligned}
$$

- Note that $|\Omega(b_0)| = 1$.
- $\rho_i = \frac{|\Omega(b_{i-1})|}{|\Omega(b_i)|} \geq \frac{1}{n+1}$.
- ρ_i can be estimated by sampling almost uniformly from $\Omega(b_i)$ using the Markov chain $\mathfrak{M}_{\text{KNAPSACK}}(b_i)$, and computing the fraction of the samples that are in $\Omega(b_{i-1})$.

Analysis

- Consider the random variable X_i associated with a single run of the Markov chain $\mathfrak{M}_{\text{KNAPSACK}}(b_i)$.

$$X_i = \begin{cases} 1 & \text{if the final state is a member of } \Omega(b_{i-1}), \\ 0 & \text{otherwise.} \end{cases}$$

- If we were able to simulate $\mathfrak{M}_{\text{KNAPSACK}}(b_i)$ to infinity, then we have $\mathrm{E}[X_i] = \rho_i$.
- However, we must terminate the simulation at some point, thereby introducing a small bias.
- We will ignore this and assume that $\mathrm{E}[X_i] = \rho_i$ and $\mathrm{Var}[X_i] = \rho_i(1 - \rho_i)$.

Analysis continued...

- Suppose we perform $t = 17n^2\epsilon^{-2}$ trials, and let $\overline{X}_i$ be the sample mean.

$$\begin{aligned}\frac{\mathrm{Var}[\overline{X}_i]}{\mathrm{E}[\overline{X}_i]^2} &= \frac{\frac{\rho_i(1-\rho_i)}{t}}{\rho_i^2} \\ &= \frac{1-\rho_i}{t\rho_i} \\ &\leq \frac{n}{t}, \text{since } \rho_i \geq \frac{1}{n+1} \\ &= \frac{\epsilon^2}{17n}.\end{aligned}$$

- Suppose this process is repeated for each ρ_i.

Analysis continued...

- Let $Z = \prod_{i=1}^{n} \overline{X}_i$. Note that $\overline{X}_i$ are independent.
- Hence, $\mathrm{E}[Z] = \prod_{i=1}^{n} \mathrm{E}[\overline{X}_i] = \prod_{i=1}^{n} \rho_i = \frac{1}{|\Omega(b)|}$.

$$\begin{aligned}
\frac{\mathrm{Var}[Z]}{\mathrm{E}[Z]^2} &= \frac{\mathrm{E}[Z^2] - \mathrm{E}[Z]^2}{\mathrm{E}[Z]^2} \\
&= \frac{\mathrm{E}[\prod_{i=1}^{n} \overline{X}_i^2]}{(\mathrm{E}[\prod_{i=1}^{n} \overline{X}_i])^2} - 1 \\
&= \frac{\prod_{i=1}^{n} \mathrm{E}[\overline{X}_i^2]}{(\prod_{i=1}^{n} \mathrm{E}[\overline{X}_i])^2} - 1 \\
&= \prod_{i=1}^{n} \left(\frac{\mathrm{E}[\overline{X}_i]^2 + \mathrm{Var}[\overline{X}_i]}{\mathrm{E}[\overline{X}_i]^2} \right) - 1 \\
&= \prod_{i=1}^{n} \left(1 + \frac{\mathrm{Var}[\overline{X}_i]}{\mathrm{E}[\overline{X}_i]^2} \right) - 1.
\end{aligned}$$

Analysis continued...

$$\frac{\mathrm{Var}[Z]}{\mathrm{E}[Z]^2} \leq \left(1 + \frac{\epsilon^2}{17n}\right)^n - 1$$
$$\leq \frac{\epsilon^2}{16}.$$

- By Chebyshev's inequality, we conclude that

$$\Pr\left[\left(1 - \frac{\epsilon}{2}\right)\frac{1}{|\Omega(b)|} \leq Z \leq \left(1 + \frac{\epsilon}{2}\right)\frac{1}{|\Omega(b)|}\right] \geq \frac{3}{4}.$$

- The number of trials (Markov chain simulations) used is $nt = 17n^3\epsilon^{-2}$, which is polynomial in n and $\frac{1}{\epsilon}$.
- This is an FPRAS for the number of KNAPSACK solutions, provided that $\mathfrak{M}_{\text{KNAPSACK}}$ is *rapidly mixing*.

Rapidly Mixing Markov Chains

- A Markov chain is *rapidly mixing*, if it converges to the stationary distribution after a polynomial number of steps in n.
- This is a non-trivial condition for $\mathfrak{M}_{\text{KNAPSACK}}$, since the size of the state space Ω is exponential in n.
- It is not known if $\mathfrak{M}_{\text{KNAPSACK}}$ is rapidly mixing.
- Whether there exists an FPRAS of any kind for the KNAPSACK counting problem is still unresolved.
- There are some recent advances in this area.
- Three techniques for bounding the *mixing time* of a Markov chain are *coupling*, *canonical paths* and *conductance*.

Variation Distance and Mixing Time

- Let $\mathfrak{M}$ be an ergodic Markov chain on state space Ω with transition probabilities $P : \Omega^2 \mapsto [0, 1]$.
- Denote by $P^t(x, S)$ the distribution of the state $S \subseteq \Omega$ at time t, given that $x \in \Omega$ is the initial state.
- Let π be the stationary distribution of $\mathfrak{M}$.
- The *variation distance* at time t with respect to x is defined as

$$\begin{aligned}\Delta_x(t) &= \max_{S \subseteq \Omega} \left| P^t(x, S) - \pi(S) \right| \\ &= \frac{1}{2} \sum_{y \in \Omega} \left| P^t(x, y) - \pi(y) \right|.\end{aligned}$$

- The *mixing time* of the Markov chain $\mathfrak{M}$ is given by

$$\tau_x(\varepsilon) = \min\{t : \Delta_x(t') \leq \varepsilon \text{ for all } t' \geq t\}.$$

Time-reversible Markov Chains

- A Markov chain $\mathfrak{M}$ is called *time-reversible* if
  - The loop probabilities $P(x,x) \geq \frac{1}{2}$, for all $x \in \Omega$.
  - It satisfies the following *detailed balance* condition:

$$Q(x,y) = \pi(x)P(x,y) = \pi(y)P(y,x), \text{ for all } x,y \in \Omega.$$

- The detailed balance condition is stronger than that required merely for a stationary distribution.
- There are Markov chains with stationary distributions that do not have detailed balance.
- Detailed balance implies that, around any closed cycle of states, there is no net flow of probability.

Canonical Paths

- We think of $\mathfrak{M}$ as an undirected graph $G = (\Omega, E)$, where $E = \{(x, y) \in \Omega^2 : Q(x, y) > 0\}$.
- For each pair $(x, y) \in \Omega^2$, we specify a *canonical path* p_{xy} from x to y in the graph G.
- The canonical path p_{xy} corresponds to a sequence of legal transitions in $\mathfrak{M}$ from the initial state x to the final state y.
- Let $P = \{p_{xy} : x, y \in \Omega\}$ be the set of all canonical paths.
- The *maximum load* (*congestion*) on any edge in P is defined as

$$\rho(P) = \max_{e \in E} \frac{1}{Q(e)} \sum_{p_{xy} \ni e} \pi(x)\pi(y)|p_{xy}|.$$

- Intuitively, we expect a Markov chain to be rapidly mixing, if it contains no *bottlenecks*, i.e., if it admits a choice of paths P for which $\rho(P)$ is not too large.

Relationship between Mixing Time and Congestion

Theorem

Let $\mathfrak{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x,x) \geq \frac{1}{2}$ for all states x. Let P be a set of canonical paths with maximum edge congestion ρ. Then, for any choice of the initial state x, the mixing time of $\mathfrak{M}$ satisfies

$$\tau_x(\varepsilon) \leq \rho \left(\ln \frac{1}{\pi(x)} + \ln \frac{1}{\varepsilon} \right).$$

- Good upper bounds on the congestion ρ translate to good upper bounds on the mixing time $\tau_x(\varepsilon)$.

Conductance

- The *conductance* Φ of a Markov chain $\mathfrak{M}$ is defined as

$$Q(S, \overline{S}) = \sum_{\substack{x \in S, y \in \overline{S} \\ (x,y) \in E}} Q(x, y),$$

$$\Phi(\mathfrak{M}) = \min_{\substack{S \subset \Omega \\ 0 < \pi(S) \leq \frac{1}{2}}} \frac{Q(S, \overline{S})}{\pi(S)}.$$

- The conductance Φ may be viewed as a weighted version of edge expansion of the graph $G = (\Omega, E)$ associated with $\mathfrak{M}$.
- Alternatively, the ratio can be interpreted as the conditional probability that the chain in equilibrium escapes from the subset S of the state space Ω in one step, given that it is initially in S.
- Φ measures the readiness of the chain to escape from any small enough region of Ω to make rapid progress towards equilibrium.

Relationship between Mixing Time and Conductance

Theorem

Let $\mathfrak{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x,x) \geq \frac{1}{2}$ for all states x. Let Φ be the conductance of $\mathfrak{M}$. Then, for any choice of the initial state x, the mixing time of $\mathfrak{M}$ satisfies

$$\tau_x(\varepsilon) \leq \frac{2}{\Phi^2}\left(\ln\frac{1}{\pi(x)} + \ln\frac{1}{\varepsilon}\right).$$

- Good lower bounds on the conductance Φ translate to good upper bounds on the mixing time $\tau_x(\varepsilon)$.