**University of Chicago** **Autumn 2003**

**CS37101-1 Markov Chain Monte Carlo Methods**

Lecture 1: September 30, 2003

Relationship between Counting and Sampling

*Eric Vigoda*

# 1 Introduction

This course focuses on Markov chain based algorithms for randomly sampling from probability distributions defined on a large state space. Typically we are interested in generating a sample in time polynomial of the logarithm of the size of the state space. In other words, we want to sample from the distribution after exploring only an exponentially small portion of it.

We'll begin with two classical examples to illustrate the type of sampling problem we are interested in. These two examples will arise several times during the course.

The main topic of this lecture is to show the intimate relationship between random sampling and approximate counting. One consequence is that an efficient algorithm for random sampling yields an efficient randomized approximation algorithm to an associated counting problem. Our running examples will clarify the type of sampling and counting problems.

## 1.1 Ising Model

The following is a description of the (ferromagnetic) *Ising model* (with no external field). Let $G = (V, E)$ be the $d$-dimensional integer lattice with side length $L$. More formally

$$V = \{1, \ldots, L\}^d, \quad E = \left\{ \{(u_1, u_2, \ldots, u_d), (v_1, v_2, \ldots, v_d)\} \mid \sum_{1 \le i \le d} |u_i - v_i| = 1 \right\}.$$

Let $\Omega = \{+1, -1\}^V$, i.e., each vertex of $G$ can be in one of the two states, called spins.

The Hamiltonian of a state $\sigma \in \Omega$ is

$$H(\sigma) = \frac{1}{2} \sum_{\{u,v\} \in E} (1 - \sigma(u)\sigma(v)).$$

The Hamiltonian measures the energy of the state $\sigma$. Let $\beta = 1/kT$ where $T \geq 0$ is the temperature and $k$ is the Boltzmann constant. The model is described by the Gibbs (or Boltzmann-Gibbs) probability distribution $\mu$ on $\Omega$ where a state $\sigma \in \Omega$ occurs with probability,

$$\mu(\sigma) = \frac{\exp(-\beta H(\sigma))}{Z},$$

where

$$Z = \sum_{\sigma \in \Omega} \exp(-\beta H(\sigma))$$

is the appropriate normalizing factor, known as the *partition function*. Observe that the states with lower energy are more probable. Thus, neighboring spins prefer to align their spins.

Simulating the system requires sampling from the Gibbs distribution. During the course we will look at various results on simulating the Ising model. We will see in this lecture that sampling from the Gibbs distribution is closely related to computing the partition function.

## 1.2 Permanent

**Definition 1.** *For a $n \times n$ matrix $A$, define its* permanent *as*

$$\mathrm{per}(A) = \sum_{\sigma \in S_n} \prod_i a_{i,\sigma(i)},$$

*where $S_n$ denotes the set of permutations of $\{0, 1, \ldots, n-1\}$.*

Notice that this quantity is simply the determinant without the alternating sign.

If $A$ is a 0/1 matrix, then the permanent equals the number of perfect matchings in the bipartite graph $G$ with incidence matrix $A$. To be precise, the bipartite graph has $2n$ vertices where each vertex is associated with a specific row or column of $A$. The vertex for row $i$ and the vertex for column $j$ have an edge iff $a_{i,j} = 1$. A permutation corresponds to a pairing of row and column vertices. If all of the edges in this pairing exist, and therefore defines a perfect matching, then the permutation contributes 1 to $\mathrm{per}(A)$, and contributes 0 if any edge in the pairing does not exist in $G$.

We will look at results on estimating the permanent. These results will rely on an efficient method for generating a perfect matching uniformly at random.

# 2 Chernoff bounds

Throughout the course we will make use of Chernoff inequalities. There are many references which give a nice introduction to these topics, e.g., see [2, 4, 1].

**Theorem 2 (Chernoff).** *Let $X_1, \ldots, X_m$ be independent, identically distributed $\{0,1\}$-random variables where $p = \mathrm{E}(\ X_i\ )$. For all $\epsilon \leq 3/2$,*

$$\Pr\left(\ \left|\sum X_i - pn\right| > \epsilon pn\ \right) \leq 2\exp(-\epsilon^2 pn/3).$$

This is a simplified version of slightly stronger bounds, with more complicated expressions on the right-hand side.

# 3 Approximation Algorithms

## 3.1 Definitions

We can view a general counting problem (e.g., computing the permanent or computing the partition function of the Ising model) as computing a function $f : \Sigma^* \to \mathbb{N}$, where $\Sigma$ is a finite alphabet used to encode problem instances (e.g., the input matrix we'd like to compute the permanent of).

Our goal is a *fully polynomial randomized approximation scheme*, known as an *FPRAS*. Given an input $x \in \Sigma^*$, error parameter $\epsilon > 0$ and confidence parameter $0 < \delta < 1$, our goal is to compute $OUT$ such that

$$\Pr(\ (1-\epsilon)f(x) \leq OUT \leq (1+\epsilon)f(x)\ ) \geq 1 - \delta,$$

in time polynomial in $|x|, \epsilon^{-1}$ and $\log(1/\delta)$.

It suffices to achieve the above with $\delta = 1/4$. The following algorithm then boosts the error probability to arbitrary $\delta$. Run $k = 16\log(2/\delta)$ trials with error probability $1/4$, obtaining outputs $y_1, \ldots, y_k$. Let $m$ be the median of these $k$ values. The value $m$ achieves the desired error probability. To see this, let

$$X_i = \begin{cases} 1 & \text{if } y_i \in (1 \pm \epsilon)f(x) \\ 0 & \text{otherwise} \end{cases}$$

Note, $\mathrm{E}\left(\sum X_i\right) \geq \frac{3}{4}k$. Then,

$$\begin{aligned}
\Pr\left(\, m \notin (1 \pm \epsilon) f(x) \,\right) &\leq \Pr\left(\sum X_i < k/2\right) \\
&\leq \Pr\left(\, |\sum X_i - E(\sum X_i)| > k/4 \,\right) \\
&\leq 2e^{-k^2/16k} \\
&\leq \delta,
\end{aligned}$$

where the penultimate inequality follows by Chernoff's inequality.

For sampling problems, we aim for a *fully polynomial almost uniform sampler (FPAUS)*. Given an instance $x \in \Sigma^*$, a sampling problem is looking to output from a distribution (perhaps the uniform distribution or the Gibbs distribution) over the set of solutions to $x$. Let $\pi$ denote the desired distribution. We will settle for an approximation to $\pi$.

For distributions $\mu, \pi$ on $\Omega$, the *total variation distance* between $\mu$ and $\pi$ (which is one-half the $L_1$ distance), is given by

$$d_{TV}(\mu, \pi) = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \pi(x)| = \max_{A \subseteq \Omega} \mu(A) - \pi(A).$$

Our goal is an algorithm which generates solutions from some distribution $\mu$ such that

$$d_{TV}(\mu, \pi) \leq \delta,$$

in time polynomial in the input size $|x|$ and $\log(1/\delta)$.

## 3.2 Equivalences

The notions of counting and sampling are closely related. The following table summarizes the implications. An arrow indicates that if you can do the tail, then you can do the head.

$$\begin{array}{ccc}
\text{Exact Counter} & \Longrightarrow & \text{Exact Sampling} \\
\Downarrow & & \Downarrow \\
\text{Approximate Counter (FPRAS)} & \Longleftrightarrow & \text{Approximate Sampling (FPAUS)}
\end{array}$$

These implications are for self-reducible problems (see [3]). We won't define self-reducibility, instead we will present a specific example which clearly demonstrates the notion. Our running example will be matchings (not necessarily perfect) of a graph. Let $G = (V, E)$ be a graph, and let $\mathcal{M}(G)$ be the set of matchings of $G$.

We'll prove some of the implications: exact counting implies an exact sampler, and an exact sampler implies an approximate counter.

**Lemma 3.** *Given an algorithm which exactly computes the number of matchings of an arbitrary graph $G = (V, E)$ in time polynomial in $|V|$, we can then construct an algorithm which outputs a (uniformly) random matching of an arbitrary graph $G = (V, E)$ in time polynomial in $|V|$.*

*Proof.* Choose an arbitrary $e = (u, v) \in E$. Let $G_1 = (V, E \setminus e)$, and let $G_2$ denote the induced subgraph on $V \setminus \{u, v\}$. For a matching $M$ of $G$, either $e \notin M$ and $M$ is also a matching of $G_1$, or $e \in M$ and $M \setminus e$ is a matching of $G_2$. Since the reverse implication also holds, we have

$$|\mathcal{M}(G)| = |\mathcal{M}(G_1)| + |\mathcal{M}(G_2)|.$$

Let $R$ denote a random matching from $\mathcal{M}(G)$. Thus,

$$\Pr(\ e \in R\ ) = \frac{|\mathcal{M}(G_2)|}{|\mathcal{M}(G_1)| + |\mathcal{M}(G_2)|}.$$

Therefore, we can recursively construct $R$ by considering one edge at a time. □

We now look at the reverse direction, given an exact sampling algorithm only results in an approximate counter.

**Lemma 4.** *Given an algorithm which for an arbitrary graph $G = (V, E)$ generates a random matching in time polynomial in $|V|$, then we can construct an FPRAS for estimating $|\mathcal{M}(G)|$.*

*Proof.* Arbitrarily order the edges as $E = \{e_1, e_2, \ldots, e_m\}$. Let $G_0 = G$ denote the input graph, and let $G_i = (V, E_{i-1} \setminus e_i), i = 1, \ldots, m$. We can write the number of matchings of $G$ as a telescoping product:

$$|\mathcal{M}(G)| = \frac{|\mathcal{M}(G_0)|}{|\mathcal{M}(G_1)|}\frac{|\mathcal{M}(G_1)|}{|\mathcal{M}(G_2)|}\cdots\frac{|\mathcal{M}(G_{m-1})|}{|\mathcal{M}(G_m)|}|\mathcal{M}(G_m)|.$$

Note, the final term is trivial since $G_m$ is the empty graph. Each term in the telescoping product can be accurately estimated using the exact sampler. Let

$$p_i = \frac{|\mathcal{M}(G_{i+1})|}{|\mathcal{M}(G_i)|}.$$

Then,

$$|\mathcal{M}(G)| = \prod_i \frac{1}{p_i}.$$

Since $\mathcal{M}(G_{i+1}) \subseteq \mathcal{M}(G_i)$ we have $p_i \leq 1$. This also gives a simple way to estimate $p_i$, just generate random matchings from $G_i$ and count the fraction which are also matchings of $G_{i+1}$. The number of samples needed to accurately estimate $p_i$ depends on the range of $p_i$.

Observe,

$$|\mathcal{M}(G_i) \setminus \mathcal{M}(G_{i+1})| \leq |\mathcal{M}(G_{i+1})|,$$

and

$$\mathcal{M}(G_i) \cap \mathcal{M}(G_{i+1}) \subseteq \mathcal{M}(G_{i+1}).$$

These two observations imply $p_i \geq 1/2$. Thus, we will need very few samples to closely estimate $p_i$.

Draw $s$ random samples from $\mathcal{M}(G_i)$. Let $q_i$ denote the number of samples in $\mathcal{M}(G_{i+1})$.

By Chernoff's inequality,

$$\Pr\left( |p_i - q_i| > \epsilon/m \right) < \delta/m,$$

for

$$s = O\left(\frac{\log(2m/\delta)}{(\epsilon/3m)^2}\right).$$

Letting

$$OUT = \prod_i \frac{1}{q_i},$$

we have

$$\Pr\left( OUT \notin (1 \pm \epsilon)|\mathcal{M}(G)| \right) < \delta.$$

□

The implication from an approximate sampling algorithm to an approximate counting algorithm is the same approach as above with the various error probabilities incorporated in.

# References

[1] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], New York, second edition, 2000. With an appendix on the life and work of Paul Erdős.

[2] S. Janson, T. Łuczak, and A. Rucinski. *Random graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[3] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43(2-3):169–188, 1986.

[4] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, Cambridge, 1995.