Georgia Tech **Fall 2006**

Markov Chain Monte Carlo Methods

August 31 and September 5, 2006

Relationship between Counting and Sampling

Eric Vigoda

1 Introduction

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.

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 [3, 6, 1].

Theorem 1 (Chernoff). *Let $X_1, \dots, X_m$ be independent, identically distributed $\{0, 1\}$-random variables where $p = \mathrm{E}\left(X_i \right)$. 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 Definitions of FPRAS and FPAUS

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 Σ 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

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

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 δ. 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,

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

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 π denote the desired distribution. We will settle for an approximation to π.

For distributions μ, π on Ω, the *total variation distance* between μ and π (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 μ such that

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

in time polynomial in the input size $|x|$ and $\log(1/\delta)$. We call such a sampler an FPAUS.

4 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.

$$\begin{tikzcd}
\text{Exact Counter} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Exact Sampling} \arrow[d, Rightarrow] \\
\text{Approximate Counter (FPRAS)} \arrow[r, Leftrightarrow] & \text{Approximate Sampling (FPAUS)}
\end{tikzcd}$$

The equivalence between an FPRAS and FPAUS was first proved by Jerrum, Valiant and Vazirani [5]. These implications are for self-reducible problems (see [5]). 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.

5 FPRAS $\leq$ FPAUS

We now prove that approximate counting reduces to approximate sampling.

Lemma 2. *Given an FPAUS for sampling matchings of an arbitrary graph, then we can construct an FPRAS for estimating* $|\mathcal{M}(G)|$.

Proof. Consider $G = (V, E)$ which we want to estimate $|\mathcal{M}(G)|$. Let ϵ denote the desired approximation factor and δ the desired error probability, i.e., we want to find an estimate OUT such that

$$\Pr(\ |OUT - |\mathcal{M}(G)|| \geq \epsilon|\mathcal{M}(G)|\ ) \leq \delta.$$

Arbitrarily order the edges as $E = \{e_1, e_2, \dots, e_m\}$. Let $G_0 = G$ denote the input graph, and let $G_i = (V, E_{i-1} \setminus e_i), i = 1, \dots, m$. 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.

We can not sample from exactly the uniform distribution over $\mathcal{M}(G_i)$, but we can sample from a distribution close to uniform. Set $\eta = \epsilon/12m$, and we will run the FPAUS on G_i so that the samples are with variation distance $\leq \eta$ of the uniform distribution. Draw s random samples (with parameter η) from our FPAUS for $\mathcal{M}(G_i)$. Let q_i denote the number of samples in $\mathcal{M}(G_{i+1})$. Let

$$\mu_i = \mathrm{E}\left(\ q_i \ \right).$$

We have

$$p_i - \eta \leq \mu_i \leq p_i + \eta.$$

Our aim is to estimate p_i within a factor $(1 \pm \epsilon/3m)$ with probability $\geq 1 - \delta/m$, then this will give a $(1 \pm \epsilon)$ approximation of $|\mathcal{M}(G)|$ with probability $\geq 1 - \delta$. By Chernoff's inequality,

$$\Pr\left(\ |q_i - \mu_i| > \mu_i \epsilon/12m \ \right) < \delta/m,$$

for

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

We can conclude that with probability $\geq 1 - \delta$, for all i, we have

$$p_i(1 - \epsilon/3m) \leq q_i \leq p_i(1 + \epsilon/3m).$$

Since $(1 - \epsilon/3m)^m \geq (1 - \epsilon)$ and $(1 + \epsilon/3m)^m \leq (1 + \epsilon)$, then for

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

we have

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

□

Note in the above proof, the total number of samples needed is $O^*(m^3)$ ignoring the log factors and the dependence on ϵ. It turns out that by using Chebyshev's inequality, only $O^*(m^2)$ samples are needed in total. This reduces the running time for the FPRAS we are constructing. The interesting aspect of the proof using Chebyshev's is that we apply Chebyshev's for the random variable OUT. However it does not work out if we apply Chebyshev's for each q_i. This approach was first done by Dyer and Frieze [2], who used it to improve the running time of the FPRAS for estimating the volume of a convex body.

For the improvement using Chebyshev's inequality see Jerrum's monograph [4, Chapter 3].

6 FPAUS $\leq$ FPRAS

We first show the trivial reduction from exact sampling to exact counting.

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 show the more interesting reduction from approximate sampling to approximate counting.

Lemma 4. *Given an FPRAS for estimating the number of matchings, we can then construct an FPAUS for generating a random matching.*

The proof we present follows the approach in Sinclair [7].

Proof. Let us first assume that we have a deterministic algorithm to estimate the number of matchings of any graph G within a factor $(1 \pm \epsilon)$ in time polynomial in $|G|$ and $1/\epsilon$. Let δ denote the desired error probability for the FPAUS. Let $\eta = n^{-3}$.

We first show a scheme which, upon successful completion, generates matchings uniformly at random, but will only successfully complete with some probability $> 1/2$. By then running the procedure until the first time it successfully completes, then at most $\log(1/\delta)$ trials will suffice with probability $\geq 1 - \delta$.

The basic scheme is the same approach as in the proof of Lemma 3. Let G^i denote the graph considered in stage i of that algorithm (i.e., we've decided to include or not include $i - 1$ edges so far). Let e_i denote the current edge under consideration, and let G_1^i and G_2^i denote the corresponding subgraphs. In each stage, we use our deterministic approximate counter to estimate $|\mathcal{M}(G_2^i)|$, $|\mathcal{M}(()G_1^i)|$ and $|\mathcal{M}(G^i)|$ within factors $(1 \pm \eta)$ where η is defined with respect to the original input graph G. For each edge e_i considered, we may overestimate the probability for including or not including e_i in our recursive algorithm by at most a factor

$$\frac{1+\eta}{1-\eta} \leq (1+2\eta)^2 \leq \exp(4\eta).$$

For a matching N, let $p(N)$ denote the probability that the algorithm from the proof of Lemma 3 outputs N using the approximate estimates returned by the approximate counter. Our goal is to output N with probability $1/|\Omega|$ where $\Omega = \mathcal{M}(G)$. The recursive algorithm has at most $|E| \leq n^2$ rounds, hence

$$p(N) \leq \frac{\exp(4n^2\eta)}{|\Omega|}.$$

Suppose after running the recursive algorithm, and ending at matching N, we then outputed N with probability $p_{\text{accept}}(N) := (|\Omega| p(N))^{-1}$ and with probability $1 - p_{\text{accept}}(N)$ just called this run of the algorithm a failure and tried the entire recursive algorithm again. Then, note that the probability of outputting N is $p(N)p_{\text{accept}}(N) = 1/|\Omega|$ as desired. And the probability of the algorithm succeeding (i.e., outputting N) is $\geq \exp(-4n^2\eta) \geq (1-1/n)$ for n sufficiently large, which is clearly $> 1/2$.

The only catch is that we don't know $p_{\text{accept}}(N)$. The quantity $p(N)$ we do know. We also can compute a reasonable estimate of $|\Omega|$. We just run our deterministic approximate counter and get an estimate of $|\mathcal{M}(G)| = |\Omega|$ within a factor $(1 \pm \eta)$. Call our estimate S. Since $|\Omega| \leq S(1+\eta)$ it suffices to output N with probability

$$\frac{1}{p(N)|S|(1+\eta)}.$$

Then the probability of outputting N is $(|S|(1+\eta))^{-1}$, which is the same for all N, and one can check that the probability of the algorithm succeeding is still $> 1/2$.

Now it remains to consider a *randomized* approximate counter. Let 2δ denote the desired error probability for the FPAUS we are constructing. Let $\delta' = \delta/3n^2$. We will use the same approach as above, and just account for the error probabilities.

For each call of the FPRAS to estimate $|\mathcal{M}(G_2^i)|, |\mathcal{M}(()G_1^i)|$ and $|\mathcal{M}(G^i)|$ we will set the desired error probability to δ'. Since there are at most $3n^2$ calls to our FPRAS, with probability $\geq 1 - \delta$, all of the estimates are within a factor $(1 \pm \eta)$. Hence, with probability $\geq 1 - \delta$, after $\leq \log(1/\delta)$ trials of the above algorithm, with probability $\geq 1 - \delta$ we will generate a matching uniformly at random. Hence, with probability $\geq 1 - 2\delta$, we can generate a random matching in time polynomial in $|G|$ and $\log(1/\delta)$. Regardless of what is outputted with the remaining probability of $\leq 2\delta$, we are generating a matching from a distribution that is within variation distance $\leq 2\delta$ of the uniform distribution. □

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] M. E. Dyer and A. Frieze. Computing the volume of a convex body: a case where randomness provably helps. In *Proceedings of AMS Symposium on Probabilistic Combinatorics and Its Applications*, pages 123–170, 1991.

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

[4] M. R. Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Birkhauser Verlag, Basel, Switzerland, 2003.

[5] 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.

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

[7] A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach*. Birkhauser Verlag, Basel, Switzerland, 1993.