Approximate Counting and Mixing Time of Markov chains **Fall 2024**

Lecture 1: Introduction: Equivalence of Counting and Sampling

Lecturer: Shayan Oveis Gharan *Sept 27*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

Assume we have a giant space Ω (in most of this class we assume Ω is finite) and suppose we have a nonnegative weight function $w : \Omega \to \mathbb{R}_+$, and we would like to sample an element $x \in \Omega$ with probability $\pi(x) = w(x)/Z$ where $Z = \sum_{x\in\Omega} w(x)$ is called the *partition function* is unknown to us.

Definition 1.1 (#P). *The class #P (called Sharp-P) is the class of all functions f for which there exists an NP language L with a non-deterministic Turing machine M such that $f(x)$ equals the number of accepting paths in M on input x.*

In other words, each problem in #P can always be seen as the problem of counting the number of witnesses for a given instance of an NP problem. Under the above definition it follows that #CIRCUIT-SAT is #P-complete. Also, that if #R is #P-complete, then L_R is NP-complete.

In this course we will discuss several class of approaches for these problems. The general framework that we study is as follows: We construct a Markov chain with stationary distribution the same as the probability distribution $\pi(.)$; then we compute the partition function by generating polynomially many samples.

The main goal in this course is to study tools to bound mixing time of Markov chains. In the first half we study more classical techniques such as coupling and the canonical path method and in the second half we study more recent developments such as the spectral and entropic independence.

1.1 Applications of Counting/Sampling

Let us discuss some applications of counting/sampling problems.

Combinatorics One specific topic of interest is to analyze Markov chains to generate random combinatorial structures such as matchings, spanning trees or independent sets. This chains have applications all over computer science, e.g., to obtain improved approximation algorithms for TSP.

Other interesting applications are in generating random graphs with fixed degree sequence, or random expander graphs.

Ising Model and Markov Random Fields Spin systems originated in statistical mechanics in the study of phase transition in magnets [Isi25]. Since then they have become important objects to study in probability theory and in computer science under the names of Markov random fields or Graphical models. A graphical model typically defined as follows: We are given a graph $G = (V, E)$. For a vector $x = (x_1, \dots, x_n)$ where each $x_i \in \{0, 1, \dots, q-1\}$ we define

$$w(x) = \frac{1}{Z} \prod_{i\sim j} \psi(x_i, x_j).$$

Here, $\psi : [q] \times [q] \to \mathbb{R}_+$ is a weight function, and Z is the partition function. The main important property is that any vertex is independent of all vertices of G conditioned on its neighbors. We typically, assume vertices have low degree.

In general one can consider different class of ψ functions. In the hard core model we have $x_i \in \{0, 1\}$ and $\psi(x_i, x_j) = x_i \text{NAND} x_j$. In other words, say $x_i = 1$ if the location corresponding to i is occupied. The ψ function puts a hard constraint that any two neighboring sites cannot be occupied simultaneously. In other words, any x with positive weight is an independent set of G. See [MRSZ02] for applications of the hard core model in telecommunication networks.

Next application of spin systems is in studying the Ising model which can be thought of as a model of magnets. Regard the magnet as made up of molecules which are constrained to lie on the sites of a regular lattice. Suppose there are N such sites and molecules, labelled $i = 1, \ldots, N$.

Now regard each molecule as a microscopic magnet, which either points along some preferred axis, or points in exactly the opposite direction. Thus each molecule i has two possible configurations, which can be labelled by a 'spin' variable σ_i with values $+1$ (parallel to axis) or -1 (anti-parallel). The spin is said to be 'up' when $\sigma_i = 1$, 'down' when $\sigma_i = -1$. Often these values are written more briefly as $+$ and $-$. The energy of a configuration σ is given by the so called *Hamiltonian function*

$$H(\sigma) = -\sum_{i \sim j} \sigma_i \sigma_j.$$

Note that the energy obviously increases with the number of pair of neighbors whose spins disagree. The corresponding Gibbs distribution is defined as follows:

$$\mu(\sigma) = \frac{1}{Z(\beta)} e^{-\beta H(\sigma)}.$$

where $Z(\beta)$ is the partition function. The parameter β which physically correspond to inverse of the termperature determines the influence of the energy. If $\beta \approx 0$ then all $\mu(.)$ is almost uniform. As $\beta \to \infty$ the bias of μ goes towards low energy states. Note that the above can also be seen as a spin system where $\psi(\sigma_i, \sigma_j) = e^{\beta \sigma_i \sigma_j}$. See [Bax82] for an in depth applications and the theory behind the Ising model.

1.2 Equivalence of Counting and Sampling

Let us discuss the equivalence of counting and sampling.

Definition 1.2 (FPRAS)**.** *Given a set Ω and a weight function $w : \Omega \to \mathbb{R}_+$, with partition function $Z = \sum_x w(x)$, a fully polynomial time randomized approximation scheme for Z is an algorithm that for a given error parameter $0 < \epsilon < 1$ and a confidence interval $0 < \delta < 1$ returns a number $\tilde{Z}$ such that*

$$\mathbb{P}\left[(1-\epsilon)Z \le \tilde{Z} \le (1+\epsilon)Z\right] \ge 1 - \delta.$$

The algorithm must run in time polynomial in the input size, $1/\epsilon$ and $\log(1/\delta)$.

Note that in the above definition it is enough to let $\delta = 1/4$ be an absolute constant. Because we can simply boost the probability geometrically by running multiple copies of the counting algorithm independently and returning the median of the returned outputs.

Before defining the notion of a uniform sampler we need to define a distance function between probability distributions. For two probability distributions $\mu, \nu : \Omega \to \mathbb{R}_+$ we write

$$\|\mu - \nu\|_{TV} = \frac{1}{2}\|\mu - \nu\|_1 = \frac{1}{2}\sum_x |\mu(x) - \nu(x)|$$

to denote the *total variation distance* of μ and ν.

Claim 1.3. *For any two probability distributions μ, ν,*

$$\|\mu - \nu\|_{TV} = \max_{A \subseteq \Omega} |\mu(A) - \nu(A)|$$

where $\mu(A) = \sum_{x \in A} \mu(x)$ is the probability of the even A under μ. In other words, if μ is close in total variation distance to ν it means that any event event has almost the same probability in μ, ν

Proof. First, we show that there exists A such that $|\mu(A) - \nu(A)| = \|\mu - \nu\|_{TV}$. Let $A = \{x : \mu(x) \geq \nu(x)\}$. Observe that since μ, ν are distributions

$$\sum_{x \in A} \mu(x) - \nu(x) = \sum_{x \notin A} \nu(x) - \mu(x) = \frac{1}{2}\|\mu - \nu\|_1 = \|\mu - \nu\|_{TV}.$$

On the other hand, for any set $B \neq A$,

$$\sum_{x \in B} \mu(x) - \nu(x) \leq \sum_{x \in A} \mu(x) - \nu(x).$$

as required. □

Note that we usually say μ, ν are close $\|\mu - \nu\|_{TV} \leq \epsilon$ for $\epsilon = 0.1$ or $\epsilon = 1/n$, however we usually work with distributions of exponential support size in n. So, the definition does not imply that $\mu(x) \approx \nu(x)$ for all x if we have such a big ϵ. The main advantage of the above definition is when we study "probable" probability events under μ, ν.

Definition 1.4 (FPAUS)**.** *Given a set Ω and a weight function $w : \Omega \to \mathbb{R}_+$, a fully polynomial time almost uniform sampler is an algorithm that for a given error parameter δ returns a point x sampled from a distribution μ such that*

$$\|\pi - \mu\|_{TV} \leq \delta.$$

The algorithm runs in time polynomial in the input and $\log(1/\delta)$*.*

It turns out that counting and sampling are closely related as summarized in the following table. The equivalence between FPRAS and FPAUS was first establish by Jerrum, Valiant and Vazirani for a class of problems known as self-reducible.

$$\begin{tikzcd}
\text{Exact Counter} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Exact Sampler} \arrow[d, Rightarrow] \\
\text{Approximate Counter} \arrow[r, Leftrightarrow] & \text{Approximate Sampler}
\end{tikzcd}$$

Definition 1.5 (Self-Reducible Problems)**.** *An NP search problem is self-reducible if the set of solutions can be partitioned into polynomially many sets each of which is in a one-to-one correspondence with the set of solutions of a smaller instance of the problem, and the polynomial size set of smaller instances are efficiently computable.*

They showed that for any self-reducible problem there is an FPRAS if and only if there is an FPAUS [**?**].

Here we prove this equivalence for the problem of counting matchings in a given graph. Given a graph $G = (V, E)$, let $\mathcal{M}$ be the of all matchings of G, in particular, the empty set is also a matching. We are interested in computing $|\mathcal{M}|$ or in generating (almost) uniform random samples from $\mathcal{M}(G)$.

1.2.1 FPAUS ⇒ FPRAS

First, we construct a FPRAS given an approximate sampler.

Lemma 1.6. *Given an FPAUS for sampling matchings in an arbitrary graph we given algorithm FPRAS for estimating $|\mathcal{M}(G)|$.*

First, let us name the edges of G as follows: $e_1, e_2, \ldots, e_m$. Consider the following sequence of graphs: $G_0 = G$ and $G_i = G_{i-1} - e_i$ for all $i \geq 1$. Note that G_m is the empty graph, so $|\mathcal{M}(G_m)| = 1$. We can rewrite $|\mathcal{M}(G)|$ as follows:

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

where we used that $|\mathcal{M}(G)| = |\mathcal{M}(G_0)|$ and that $|\mathcal{M}(G_m)| = 1$. So, the approximate the LHS it is enough to estimate each of the ratios in the RHS. Let

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

And, we have $|\mathcal{M}(G)| = \prod_i \frac{1}{p_i}$. In particular, if we approximate each p_i *multiplicatively* up to an $(1 + \epsilon/4m)$ error with probability $1 - \delta/m$, then by union bound we have $(1 + \epsilon/4m)$ approximation of all p_i's with probability $1 - \delta$. This already gives a $1 + \epsilon/2m$ approximation of $1/p_i$'s for all i. Taking the product, we obtain a $(1 + \epsilon/2m)^m < 1 + \epsilon$ approximation of $|\mathcal{M}(G)|$.

So, all we need to do estimate p_i, i.e., the fraction of matchings in G_{i-1} that do not contain e_i up to an $(1+\epsilon/4m)$ multiplicative error. First, observe that an additive approximation of p_i with $\epsilon/8m$ error is indeed a $1 + \epsilon/4m$ multiplicative approximation. This is because for all i, $1/2 \leq p_i \leq 1$. In particular, note that for any matching $M \in G_{i-1}$ that contains e_i, there is a matching $M \setminus \{e_i\}$ that does not contain e_i, so

$$|\mathcal{M}(G_{i-1}) - A| \leq |A| \Rightarrow \frac{|\mathcal{M}(G_i)|}{|\mathcal{M}(G_{i-1})|} = \frac{|A|}{|A| + |\overline{A}|} \geq 1/2$$

So, it remains to find an additive approximation of p_i within $\epsilon/8m$ error. Here we use the uniform sampler. Let π be the uniform distribution on matchings of G_{i-1}, i.e., $\pi(M) = 1/|\mathcal{M}(G_{i-1})|$. Let A be the set of all matchings of G_{i-1} that do not contain e_i,

$$A = \{M \in G_{i-1} : e_i \notin M\}.$$

We use an almost uniform sampler that returns a matching M from a distribution μ such that $\|\mu - \pi\|_{TV} \leq \epsilon/10m$. It follows by Claim 1.3 that

$$|\mathbb{P}_{M\sim\mu}[M \in A] - \mathbb{P}_{M\sim\pi}[M \in A]| \leq \epsilon/20m.$$

In other words,

$$p_i - \frac{\epsilon}{20m} \leq \mathbb{P}_{M\sim\mu}[M \in A] \leq p_i + \frac{\epsilon}{20m}.$$

So, to obtain a $\epsilon/10m$ additive approximation of p_i it is enough to find a $\epsilon/20m$ additive approximation of $\mathbb{P}_{M\sim\mu}[M \in A]$. By Chernoff bound it is enough to generate $O((\frac{m}{\epsilon^2})\log(m/\delta))$ matchings from μ and compute the fraction of matchings that are in A, i.e., those that do not contain e_i.

Theorem 1.7 (Chernoff-Hoeffding Bound)**.** *Let $X_1, \ldots, X_n$ be independent random variables such that for all i, $0 \leq X_i \leq 1$. We define the empirical mean of these variables by $\overline{X} = \frac{1}{n}(X_1 + \cdots + X_n)$. Then, for any $t > 0$*

$$\mathbb{P}\left[|\overline{X} - \mathbb{E}[X]| \geq \alpha \cdot \mathbb{E}[X]\right] \leq 2e^{-n\alpha^2\mathbb{E}[X]/3}.$$

Note that since $\mathbb{P}_{M\sim\mu}[M \in A] \geq 1/4$ if we have $O((\frac{m}{\epsilon^2})\log(m/\delta))$ samples with probability $1 - \delta/m$ we find an $\epsilon/20m$ additive approximation of $\mathbb{P}_{M\sim\mu}[M \in A]$ as desired. This completes the proof of Lemma 1.6.

1.2.2 FPRAS $\Rightarrow$ FPAUS

Next, we prove the reverse direction.

Lemma 1.8. *Given an FPRAS for estimating the number of matchings of an arbitrary graph G, there is an algorithm that generates almost uniform samples of $\mathcal{M}(G)$.*

Before proving the above lemma let us first prove a weaker statement. Suppose we have an exact counter. We show how to use it to obtain an exact sampler. Given a graph G, consider an arbitrary edge $e = (u, v)$, and consider the graphs $G_1 = G \setminus \{e\}$ and the induced graph $G_2 = G[V \setminus \{u, v\}]$. We claim that

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

This is because any matching M in G either has e in which case $M \setminus \{e\}$ is a matching of G_2 or it does not contain e in which case M is a matching of G_1. Now, using the exact counter we can compute the ratio

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

Now, we toss a coin; with probability p we remove e from G and recurse and with the remaining probability we include e in the output matching, erase its endpoints and recurse.

Now, suppose we have an approximate counter. First, of all it is not hard to see that we can construct an approximate sampler by following the above technique and use the number returned by counter as an approximation of p. This way we can approximate p multiplicatively up to a $1 + \eta$ error. Since the recursion terminates in $O(n^2)$ steps, it follows that the probability of choosing any matching M is at most $\frac{(1+\eta)^{n^2}}{|\mathcal{M}(G)|}$. Note that we may choose $\eta < 1/n^3$ and that gives a an approximate sampler with distribution that is within $1/n$ total variation distance of $\pi(.)$. But, note that we have spent time polynomial in $1/\eta$ to construct such a distribution. Next, we construct an approximate sampler that for any $\epsilon > 0$ in time polynomial in n and $\log(1/\epsilon)$ generates a sample from a distribution of total variation distance ϵ of π.

Rejection Sampling. For simplicity, suppose we have a deterministic approximate counter that gives a $1 + \eta$ multiplicative approximation of $|\mathcal{M}(G)|$ for any graph G. We use this to generate an approximate sampler. We design an algorithm that with probability $1 - 1/n$ returns an exact sample of π and with the remaining probability fails. So, to obtain a success probability of δ it is enough to run the algorithm $\log(1/\delta)$ times until it is successful. If the algorithm fails in all those iterations we just return an empty matching. The output distribution obviously would have a total variation distance of δ with π.

Fix an ordering of the edges $e_1, \ldots, e_m$. Suppose we have decided already on $e_1, \ldots, e_{i-1}$, and based on that we have $G^i = (V, E)$ as the remaining graph. If $e_i = (u_i, v_i)$ is not an edge of G^i we let $p_i = 1$ and we recurse. Otherwise, consider two graph $G_1^i = G^i \setminus \{e_i\}$ and $G_2^i = G_i[V \setminus \{u_i, v_i\}]$ and we estimate

$$p_i = \frac{|\mathcal{M}(G_1^i)|}{|\mathcal{M}(G_1^i)| + |\mathcal{M}(G_2^i)|}.$$

Note that by estimating each of these quantities we obtain a $(1 + \eta)^2$ approximation of p_i. In particular, we will use $|\tilde{\mathcal{M}}(G_1^i)|$ to denote the approximation of $|\mathcal{M}(G_1^i)|$ returned by the approximate counter. So, we toss a coin and with probability p_i we delete e_i and with probability $1 - p_i$ we include e_i remove its endpoints and recurse. Let M be the matching that this algorithm produces. Obviously, the probability that M is produced is

$$\tilde{\pi}(M) := \prod_{i=1}^{m} \max \left\{ \mathbb{I}\left[e_i \notin G_i\right], \frac{\mathbb{I}\left[e_i \notin M\right] \cdot |\tilde{M}(G_1^i)| + \mathbb{I}\left[e_i \notin M\right] \cdot |\tilde{\mathcal{M}}(G_2^i)|}{|\tilde{\mathcal{M}}(G_1^i)| + |\tilde{\mathcal{M}}(G_2^i)|} \right\}.$$

Ideally we would like $\tilde{\pi}(M) = \pi(M) = 1/|\mathcal{M}(G)|$, but because of the $(1+\eta)^2$ errors we have

$$\tilde{\pi}(M) \geq \frac{(1-\eta)^{n^2}}{|\mathcal{M}(G)|} \geq \frac{(1-\eta)^{n^2+1}}{|\tilde{\mathcal{M}}(G)|} =: \alpha$$

The algorithm is as follows: Once we construct M we calculate $\tilde{\pi}(M)$, now with probability

$$p_{accept}(M) = \frac{\alpha}{\tilde{\pi}(M)}$$

we accept M and output it, and with the remaining probability we reject it. Observe that since $\alpha \leq \tilde{\pi}(M)$ for all M, the above is indeed a probability. The probability that we output a fixed matching M is exactly $p_{accept(M)} \cdot \tilde{\pi}(M) = \alpha$ which is uniform among all matchings as desired. For $\eta < 1/n^3$ the algorithm outputs a matching with probability at least

$$\sum_M p_{accept} \geq |\mathcal{M}(G)| \cdot \alpha \geq 1 - \Omega(1/n).$$

To complete the proof of Lemma 1.8 we need to extend the above construction to an randomized approximate counter. In that case we need to choose the success probability of the approximate counter to be at least $1 - \delta/10n^2$. Therefore, by union bound all queries that we make to the counter are correct with probability at least $1 - \delta/2$ as desired.

1.3 All or Nothing

Jerrum and Sinclair [JS89] prove the following surprising fact that a counting problem either has a FPRAS, or cannot be approximated in any reasonable sense in polynomial time. Specifically,

Theorem 1.9. *For a self-reducible problem, if there exists a polynomial time randomized algorithm for counting within a factor of* $(1 + poly(|x|)) \pm 1$, *then there exists a FPRAS.*

Note that this theorem says that, if we can approximately count colorings (say) in polynomial time within a factor of 1000, or even within a factor of n^{1000}, then we can get an FPRAS for colorings!

Corollary 1.10. *For a self-reducible counting problem, one of the following two holds:*

i) There exists a FPRAS;

ii) There does not exist a polynomial time approximation algorithm within any polynomial factor.

This dichotomy between approximable and non-approximable is very different from the situation with optimization problems, for which many different degrees of approximability exist (e.g., approximation schemes $1 + \epsilon$ for any ϵ); constant factor; logarithmic factor, polynomial factor etc.) We will prove this theorem later in the course.

References

[Bax82] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982. 1-2

[Isi25] E. Ising. Beitrag zur theorie des ferromagnetismus. *Z. Phys.*, 31:253–258, 1925. 1-1

[JS89] M.R. Jerrum and A.J. Sinclair. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82:93–133, 1989. 1-6

[MRSZ02] P. Mitra, K. Ramanan, A. Sengupta, and I. Ziedins. Markov random field models of multicasting in tree networks. *Advances in Applied Probability*, 34(1):1–27, 2002. 1-2