6.S891 Lecture 1: Overview of Counting and Sampling

Kuikui Liu

September 7, 2023

The entirety of this course (and this field of study) is based on the following two fundamental problems. Suppose we have to *huge* collection Ω of objects (e.g. vectors in $\mathbb{R}^n$, n-bit strings in $\{0,1\}^n$), as well as a nonnegative weight function $w : \Omega \to \mathbb{R}_{\geq 0}$ (e.g. the $\{0,1\}$-indicator function of some meaningful subset of Ω).

Problem 1 (Counting)**.** *Compute the* partition function $Z \overset{\mathsf{def}}{=} \sum_{x \in \Omega} w(x)$.

Problem 2 (Sampling)**.** *Let μ be the probability distribution on Ω given by $\mu(x) \propto w(x)$, i.e. $\mu(x) = \frac{w(x)}{Z}$ for all $x \in \Omega$. Sample a random element X from Ω according to μ.*

Without getting bogged down with too much details, let's just say that in most settings, "have" means we can freely find/store/manipulate single elements in Ω, we can efficiently test membership in Ω if Ω lives within a larger ambient space, and we can efficiently query a blackbox which on input x, spits out the value of $w(x)$. One should think of Ω either as infinitely large, or as growing exponentially fast w.r.t. some natural underlying parameter, e.g. the dimension n in $\{\pm 1\}^n$, with n tending to infinity. Of course, we seek to develop efficient algorithms, and so such considerations preclude trivial algorithms based on brute force enumeration of Ω and querying $w(\cdot)$ everywhere.

The goal of this course is to show you mathematical tools for designing and analyzing efficient algorithms for solving these problems. We'll see a diverse mixture of approaches, including:

- statistical physics methods like Markov chain Monte Carlo and correlation decay
- algebraic ideas based on the geometry of multivariate polynomials (e.g. zero-freeness)
- algorithms based on optimization

We'll try to place further emphasis on *connections* between these approaches.

1 Why?

These problems occur in many settings of practical importance. Some more traditional applications include the following.

- **Statistical Inference:** How can we learn useful statistical models from data, and make accurate inferences about the world once we have them? *Bayesian statistics* provides a powerful framework for understanding these questions. In the Bayesian approach, we have a model of how observed data X (e.g. symptoms) is generated from unobserved variables Θ of interest (e.g. diseases), i.e. conditional probabilities $\Pr[X \mid \Theta]$. The goal is then to infer Θ given the observed data X (e.g. medical diagnosis from observed symptoms). It turns out, in a certain mathematically precise sense, the optimal thing to do is to draw samples from the *posterior distribution* $\Pr[\Theta \mid X]$, which via Bayes' Rule, can be expressed as

$$\Pr[\Theta \mid X] = \frac{\Pr[X \mid \Theta] \cdot \Pr[\Theta]}{\Pr[X]}$$

where $\mathsf{P}[\Theta]$ is a probability distribution encoding "prior knowledge" of the hidden parameters Θ, and $\mathsf{P}[X] = \int \Pr[X \mid \Theta] \cdot \Pr[\Theta] \, d\Theta$ is the unconditional probability of observing X. In this setting, typically the probabilities $\Pr[X \mid \Theta]$ and $\Pr[\Theta]$ are easy to compute since they are specified by the statistical model, whereas the normalizing constant $\Pr[X]$ (essentially, the partition function Z from Problem 1) is unknown and hard to compute [WJ08; KF09].

- **Statistical Mechanics and Phase Transitions:** How can we understand macroscopic properties of materials given only their microscopic composition? We often model such systems mathematically using *graphical models*, which have a wide variety of applications beyond statistical physics, including machine learning and social network analysis [WJ08; KF09]. Many objects of interest to pure mathematicians, including independent sets, matchings, colorings, cuts, etc. can be captured by graphical models.

  We have a graph $G = (V, E)$ (e.g. $\mathbb{Z}^d$), and we imagine each vertex v is occupied by a particle of some sort, labeled say by elements of $[q] = \{1, \ldots, q\}$. To first order, the particles only directly interact with their nearest neighbors, and so we have potentials $\psi_e : [q] \times [q] \to \mathbb{R}$ for each edge $e = \{u, v\} \in E$ describing the contribution of that edge to the energy of a global system configuration $\sigma : V \to [q]$. This energy is given by the *Hamiltonian*

  $$H(\sigma) \overset{\mathsf{def}}{=} -\sum_{e=\{u,v\}\in E} \psi_e(\sigma_u, \sigma_v),$$

  which gives rise to the *Gibbs distribution* (or *Boltzmann distribution*) μ over all possible configurations $[q]^V$ given by

  $$\mu_\beta(\sigma) \propto w(\sigma) = \exp\left(-\beta H(\sigma)\right).$$

  Here, the parameter $\beta \in \mathbb{R}_{\geq 0}$ is sometimes called the *(inverse) temperature*, and it controls how strong we want these interactions to be. This is an *extremely rich* class of distributions. Sampling from μ_β and/or computing the partition function $Z = \sum_{\sigma\in[q]^V} \exp\left(-\beta H(\sigma)\right)$ are essential to understanding the Gibbs distribution. One of the profound insights we'll see later in the course is the connection between phase transitions in the "qualitative properties" of μ_β as we vary β [Bov06; FV17], and the *computational complexity* of counting/sampling from μ_β.

Some more recent applications, which are no less important but unfortunately won't be discussed in lectures due to time constraints, are the following.

- **Fairness:** How can we detect gerrymandering in redistricting plans? This question is fundamental to the U.S. democracy, since this type of manipulation can unfairly influence the voting power of various groups, as well as bias the outcomes of elections. The methodology implemented in a number of recent court cases is to compare the redistricting plan in question with an ensemble of "typical" plans, randomly sampled from an appropriate probability distribution over all plans (see e.g. [DDS21]). Performing this sampling efficiently is incredibly difficult, since the number of possible plans is astronomically large, and one must account for a diverse set of constraints on allowable districts [DW22].

- **Privacy:** How can we conduct large-scale societal studies without compromising the privacy of each individual's data? This problem is primarily solved by making algorithms *differentially private* [DN04; Blu+05; Dwo+06], as was done in recent deployments of the U.S. Census [BUT23] and Apple products [Inc]. However, a difficult tradeoff must be made between the *accuracy* of the algorithm, and how well it maintains privacy. Remarkably, there is a generic method called the *exponential mechanism* which attains and characterizes the optimal theoretical tradeoff [MT06]. It works by taking the algorithm's output and adding noise drawn from a probability distribution specially tailored to the problem at hand. Efficiently sampling from this distribution would yield optimal algorithms which are guaranteed to be differentially private.

There are also tons of connections with optimization, combinatorics, optimal transport, geometry, etc.

2 Counting Complexity

Similar to how there is a formal complexity theory for decision problems, there is also a complexity theory for counting problems. In the land of counting problems, the analog of NP is $\#\mathsf{P}$. We won't need the formal definition of $\#\mathsf{P}$ but roughly speaking, any NP decision problem gives rise to a $\#\mathsf{P}$

counting problem in the natural way: Instead of producing a single solution to the NP-problem, the task is to count the number of such solutions.

Once we have such the complexity class #P, we can then ask what the hardest problems are in #P. Paralleling the theory for decision problems, we have #P-hard problems, i.e. those for which every counting problem in #P can be reduced to. The #P-hard problems within #P are then the #P-complete problems. So what problems are #P-complete? As you might guess, #SAT, the problem of counting solutions to an arbitrary Boolean formula is #P-complete. The proof actually follows from a direct adaptation of the classical proof that SAT is NP-complete (the Cook–Levin Theorem); the reduction preserves the number of solutions, i.e. it is *parsimonious*.

At first glance, one might expect that NP-complete decision problems always correspond to #P-complete problems, and vice versa. As far as I'm aware, we don't know that every NP-complete problem *necessarily* gives rise to a #P-complete problem. A striking theorem of Valiant says that the converse is false, namely there are actually #P-complete problems which correspond to decision problems in P.

Theorem 2.1 (Valiant; [Val79]). *The problem of exactly counting perfect matchings in an arbitrary bipartite graph is #P-complete.*

A mention in passing a seminal result of Jerrum–Sinclair–Vigoda, who gave an efficient algorithm to approximate the number of perfect matchings in any bipartite graph to arbitrary accuracy (formally, an FPRAS in the sense of Definition 1 below). Since the number of perfect matchings in a bipartite graph, with bipartition $L \sqcup R$ and $|L| = |R| = n$, is precisely the *permanent* of the $n \times n$ adjacency matrix $A \in \mathbb{R}^{L \times R}$, it immediately follows that computing the permanent if #P-hard.

Corollary 2.2. *The problem of computing the permanent*

$$\operatorname{per}(A) \overset{\mathsf{def}}{=} \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i,\sigma(i)}$$

of an arbitrary square matrix $A \in \mathbb{R}^{n \times n}$ is #P-hard.[1]

One perhaps surprising juxtaposition is that the determinant[2]

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sign}(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)}$$

is well-known to be efficiently computable. So even though these two quantities are cosmetically similar, they have very different computational complexities.[3]

Since the state spaces we're interested in are intractably large, we don't typically expect to be able to exactly count efficiently. However, there are some really interesting objects where we can count. We mention one neat example.

Theorem 2.3 (Kirchhoff's Matrix Tree Theorem). *Let $G = (V, E)$ be any graph, and let $L \in \mathbb{R}^{V \times V}$ denote its* Laplacian*:*

$$L(u,v) = \begin{cases} \deg(u), & \text{if } u = v \\ -1, & \text{if } u \neq v, \{u,v\} \in E \\ 0, & \text{otherwise} \end{cases}.$$

Then for any vertex $v \in V$, if L_v denotes the $(n-1) \times (n-1)$ submatrix of L where the row and column corresponding to v is removed, then $\det(L_v)$ equals the number of spanning trees of G. In particular, there is an efficient algorithm for exactly counting spanning trees in any graph.

3 Approximate Counting and Sampling

Now let's discuss the approximate versions of counting and sampling, as well as their connections. We start with approximate counting since it is more straightforward.

[1]Here and throughout, S_n denotes the group of permutations on n letters.
[2]Here, $\operatorname{sign}(\sigma)$ is 1 if σ is made up of an even number of transpositions, and -1 otherwise.
[3]There is a whole field which studies the relationship between them from the complexity-theoretic perspective called *Geometric Complexity Theory*.

Definition 1 (FPRAS (Informal); see e.g. [JVV86])**.** *Given Ω and $w : \Omega \to \mathbb{R}_{\geq 0}$, a* fully polynomial-time randomized approximation scheme (FPRAS) *for estimating the partition function $Z = \sum_{x \in \Omega} w(x)$ is a (randomized) algorithm that, given an accuracy parameter $0 < \epsilon < 1$ and a failure probability tolerance $0 < \delta < 1$, outputs a number $\hat{Z}$ such that*

$$\Pr\left[(1-\epsilon)Z \leq \hat{Z} \leq (1+\epsilon)Z\right] \geq 1 - \delta$$

in time $\mathsf{poly}\left(n, \frac{1}{\epsilon}, \log \frac{1}{\delta}\right)$*, where n is the size of the input describing Ω and w.*

Note that it suffices to take δ to be some constant (e.g. $3/4$), since one can always boost the success probability to $1-\delta$ by running $O(\log(1/\delta))$ independent copies of the algorithm and using the "median-of-means estimator".

Now let's consider the approximate version of the sampling problem. A natural way to do this is say that our algorithm samples from a distribution which is "close" to the distribution $\mu(x) \propto w(x)$. A common way to quantify "closeness" is to use the *total variation distance* $\|\mu - \nu\|_{\mathsf{TV}} = \frac{1}{2}\sum_{x \in \Omega} |\mu(x) - \nu(x)|$ (sometimes also written $d_{\mathsf{TV}}(\mu, \nu)$). If X is a random variable taking values in Ω, we'll write $\mathsf{Law}(X)$ for the underlying probability distribution over Ω that X follows.

Definition 2 (FPAS (Informal); see e.g. [JVV86])**.** *Given Ω and $w : \Omega \to \mathbb{R}_{\geq 0}$, a* fully polynomial-time approximate sampler (FPAS)[4] *for the distribution $\mu(x) \propto w(x)$ on Ω is a randomized algorithm which, given an error tolerance $0 < \delta < 1$, outputs a random element $X \in \Omega$ such that $\|\mathsf{Law}(X) - \mu\|_{\mathsf{TV}} \leq \delta$ in time* $\mathsf{poly}\left(n, \log \frac{1}{\delta}\right)$*, where n is the size of the input describing Ω and w.*

First, why total variation distance? It turns out it implies closeness of all (bounded) statistics of the distribution simultaneously.

Lemma 3.1. *Let μ, ν be two probability measures on Ω. Then*

$$\|\mu - \nu\|_{\mathsf{TV}} = \sup_f |\mathbb{E}_\mu[f] - \mathbb{E}_\nu[f]|,$$

where the supremum is over all 1-bounded functions $f : \Omega \to [0,1]$.

In particular, $\|\mu - \nu\|_{\mathsf{TV}} \leq \epsilon$ says that uniformly over all possible events $A \subseteq \Omega$, the probability $\mu(A)$ is within ϵ of $\nu(A)$. Hence, if we have an approximate sampler, then we can approximate the expectation $\mathbb{E}_\mu[f]$ to within additive error δ for any bounded test function f. We'll discuss this in greater depth shortly.

Proof Sketch of Lemma 3.1. Since Ω is finite, we can view $f : \Omega \to [0,1]$ and μ, ν as big vectors in $[0,1]^\Omega$. Then $|\mathbb{E}_\mu[f] - \mathbb{E}_\nu[f]| = |\langle \mu - \nu, f\rangle|$, which is convex in f. This convexity immediately implies the right-hand side equals $\sup_{A \subseteq \Omega} |\mu(A) - \nu(A)|$ via the natural correspondence between $A \subseteq \Omega$ and its $\{0,1\}$-indicator function. Since $|\mu(A) - \nu(A)| = |\mu(\Omega \setminus A) - \nu(\Omega \setminus A)|$,

$$|\mu(A) - \nu(A)| = \frac{1}{2}\left|\sum_{x \in A}(\mu(x) - \nu(x))\right| + \frac{1}{2}\left|\sum_{x \in \Omega \setminus A}(\mu(x) - \nu(x))\right| \leq \|\mu - \nu\|_{\mathsf{TV}} \qquad \forall A \subseteq \Omega$$

by the Triangle Inequality. On the other hand, this inequality is saturated by the set $A = \{x \in \Omega : \mu(x) > \nu(x)\}$, and so we must have equality. □

3.1 The Monte Carlo Method

Suppose we wanted to estimate the expectation $\mathbb{E}_\mu[f]$ of some function $f : \Omega \to \mathbb{R}$. Assuming we have an exact sampler, one approach is to simply drawn a bunch of i.i.d. samples $X_1, \ldots, X_T$ and use the empirical mean $\frac{1}{T}\sum_{i=1}^T f(X_i)$. This is the essence of the *Monte Carlo method*, and it works thanks to the *concentration of measure* phenomenon.

Theorem 3.2 (Chernoff/Hoeffding Bound)**.** *Let $Y_1, \ldots, Y_T$ be independent random variables such that $0 \leq Y_i \leq 1$ w.p. 1 for all $i = 1, \ldots, T$. Let $Y \overset{\mathsf{def}}{=} \frac{1}{T}\sum_{i=1}^T Y_i$ denote the empirical mean. Then*

$$\Pr\left[|Y - \mathbb{E}[Y]| \geq \epsilon \cdot \mathbb{E}[Y]\right] \leq 2 \cdot \exp\left(-\frac{\epsilon^2 \cdot \mathbb{E}[Y] \cdot T}{3}\right).$$

[4]If you look in the literature, the term FPAUS is more common, but it is defined specifically for the uniform distribution.

In particular, setting $T = \Theta\left(\frac{1}{\epsilon^2 \cdot \mathbb{E}_\mu[f]} \log(1/\delta)\right)$ and taking $Y_i = f(X_i)$ for $X_1, \ldots, X_T \sim \mu$, we get a multiplicative $1 \pm \epsilon$ approximation to $\mathbb{E}_\mu[f]$ with probability at least $1 - \delta$. If we had access to only an approximate sampler such that $X_1, \ldots, X_T \sim \nu$ for $\|\mu - \nu\|_{\mathsf{TV}} \leq \eta$, then the same algorithm would incur an extra additive error η in the approximation.

Corollary 3.3 (Correctness of Monte Carlo Method)**.** *Let μ be a probability measure over Ω, and let $f : \Omega \to [0,1]$ be a 1-bounded function. Suppose have an* FPAS *for μ. Then we can compute a number $\hat{Z}$ such that $(1-\epsilon) \cdot \mathbb{E}_\mu[f] \leq \hat{Z} \leq (1+\epsilon) \cdot \mathbb{E}_\mu[f]$ with probability $\geq 1 - \delta$, using at most $O\left(\frac{1}{\epsilon^2 \cdot \mathbb{E}_\mu[f]} \log(1/\delta)\right)$ calls to the* FPAS *with input total variation error $\frac{\epsilon}{2} \cdot \mathbb{E}_\mu[f]$.*

Note that all of this only works well if $\mathbb{E}_\mu[f]$ isn't too small (e.g. $\mathbb{E}_\mu[f] \geq \Omega(1)$ or $\mathbb{E}_\mu[f] \gtrsim 1/\mathsf{poly}(n)$). We mention in passing that if one were instead interested in unbounded functions f but with finite variance, then one should instead use the *median-of-means method* (sometimes called the "median trick") to get adequate concentration; this is a much more robust estimator.

4 Equivalence Between Approximate Counting and Sampling

Now that we have the Monte Carlo method, it might seem that an approximate sampler is much more powerful than an approximate counter. After all, we can use it to estimate the expectation of almost any test function. Surprisingly, for a wide class of problems, it turns out these two notions are *equivalent*, at least up to polynomial losses in the running time.

Theorem 4.1 (Informal; [JVV86])**.** *For "self-reducible" Ω and $w : \Omega \to \mathbb{R}_{\geq 0}$, there exists an* FPRAS *for estimating $Z = \sum_{x \in \Omega} w(x)$ if and only if there exists an* FPAS *for approximately sampling from the distribution $\mu(x) \propto w(x)$.*

At a high level, "self-reducibility" captures the property that the given counting/sampling problem can be efficiently *decomposed* into counting/sampling subproblems of the "same type". Later on, we'll see For simplicity, we'll illustrate the idea of "self-reducibility", as well as the proof of Theorem 4.1, for a concrete class of objects which capture the core ideas: Let $G = (V, E)$ be a graph, and let Ω_G denote the collection of all matchings in G of all sizes; recall that a matching is a subset of edges $M \subseteq E$ such that no vertex in V is incident to more than one edge in M. Let $w \equiv 1$ so that the induced distribution μ_G is uniform over Ω_G.

4.1 From Sampling to Counting

Let's start by reducing counting to sampling, making use of the Monte Carlo method. This will also illustrate why "self-reducibility" is important. Fix an edge $e = \{u, v\} \in E$, and partition the set of matchings Ω_G into those which contain or don't contain e.

- Any matching in G containing e can be viewed as taking a matching in $G - u - v$ and adding e. Hence, the collection of matchings in G containing e is in one-to-one correspondence with the collection of matchings in $G - u - v$.
- Any matching in G not containing e is equivalently a matching in $G - e$.

It follows that

$$\Pr_{M \sim \mu_G}[e \notin M] = \frac{|\Omega_{G-e}|}{|\Omega_G|},$$

and so

$$|\Omega_G| = \frac{1}{\Pr_{M \sim \mu_G}[e \notin M]} \cdot |\Omega_{G-e}|.$$

Since we can sample, we can estimate $\Pr_{M \sim \mu_G}[e \notin M]$ using the Monte Carlo method. Furthermore, Ω_{G-e} is the collection of all matchings in a smaller graph where we have deleted e, so presumably we can estimate it by induction. In this sense, we have effectively reduced the problem of counting matchings in G to the same counting problem but in a smaller graph. Let's now do the full reduction.

Claim 4.2. *If there exists an* FPAS *for sampling a uniformly random matching in an arbitrary graph, then there exists an* FPRAS *for estimating the number of matchings in an arbitrary graph.*

Proof. Order the edges $e_1, \ldots, e_m$ of E arbitrarily, and define a sequence of graph $G_0, \ldots, G_m$ where $G_0 = G$ and $G_i = G_{i-1} - e_i$ for all $i = 1, \ldots, m$; for convenience, write $\Omega_i = \Omega_{G_i}$. Note that G_m is the empty graph with no edges, and so $|\Omega_m| = 1$. Hence,

$$|\Omega_G| = \prod_{i=1}^{m} \frac{|\Omega_{i-1}|}{|\Omega_i|} = \prod_{i=1}^{m} \frac{1}{p_i},$$

where $p_i = \frac{|\Omega_i|}{|\Omega_{i-1}|} = \Pr_{M\sim\mu_{i-1}}[e_i \notin M]$. If we can compute estimates $\hat{p}_1, \ldots, \hat{p}_m$ which are $(1 \pm O(\epsilon/m))$-multiplicative approximations to the true $p_1, \ldots, p_m$, then taking the product $\prod_{i=1}^{m} \frac{1}{\hat{p}_i}$ gives a $(1 \pm O(\epsilon))$-multiplicative approximation to the total number of matchings $|\Omega_G|$.

We compute the $\hat{p}_i$ via the Monte Carlo method (Corollary 3.3), taking the test function f_i on Ω_{i-1} to be the indicator $f_i(M) = \mathbb{I}[e_i \in M]$. For this to be efficient, we need that $\mathbb{E}_{\mu_{i-1}}[f_i] = p_i$ is not too small. For this, we claim that we always have $p_i \geq 1/2$. This is because if $e_i = \{u_i, v_i\}$, then every matching containing e_i can be mapped injectively to a matching not containing e_i just by removing e_i. In particular, $\left|\Omega_{G_{i-1}-u_i-v_i}\right| \leq \left|\Omega_{G_{i-1}-e_i}\right|$ and so

$$p_i = \frac{\left|\Omega_{G_{i-1}-e_i}\right|}{\left|\Omega_{G_{i-1}-e_i}\right| + \left|\Omega_{G_{i-1}-u_i-v_i}\right|} = \frac{1}{1 + \frac{\left|\Omega_{G_{i-1}-u_i-v_i}\right|}{\left|\Omega_{G_{i-1}-e_i}\right|}} \geq \frac{1}{2}.$$

Applying Corollary 3.3 with error probability δ/m and taking an appropriate Union Bound finishes the proof. □

Remark 1. Going beyond matchings, e.g. say to some distribution μ over $[q]^n = \{1, \ldots, q\}^n$ for some $q \geq 2$, one might worry that we don't a priori know which of the probabilities $\{\Pr_{\sigma\sim\mu}[\sigma(i) = \mathfrak{c}]\}_{\mathfrak{c}\in[q]}$ satisfies a nice lower bound. However, this isn't really an issue, since at least one of them must be at least $1/q$. Furthermore, by estimating all of these probabilities using the Monte Carlo method with total variation error, say $1/2q$, we can identify any $\mathfrak{c} \in [q]$ such that $\Pr_{\sigma\sim\mu}[\sigma(i) = \mathfrak{c}] \geq 1/2q$, and then estimate that specific probability to $(1 \pm \epsilon)$-multiplicative error.

4.2 From Counting to Sampling

The basic idea to go from counting to sampling is to do everything we did above "in reverse". In particular, if we have an approximate counter, then given an edge $e = \{u, v\} \in E$, we can accurately estimate $q_e = \Pr_{M\sim\mu_G}[e \in M] = \frac{|\Omega_{G-u-v}|}{|\Omega_G|}$ by estimating both numerator and denominator. We can then do the following:

- Include e in our sampled matching with probability q_e, and exclude e with probability $1 - q_e$.
- If we included e, then delete the endpoints u, v of e and recursively sample a uniformly random matching of $G - u - v$.
- If we excluded e, then delete only the edge e (but keep its endpoints), and recursively sample a uniformly random matching of $G - e$.

Remark 2. You might have noticed a little sleight of hand we did; we're now using $\Pr_{M\sim\mu_G}[e \in M]$ rather than $\Pr_{M\sim\mu_G}[e \notin M]$. The reason is that by the argument above, $\Pr_{M\sim\mu_G}[e \in M] \leq \Pr_{M\sim\mu_G}[e \notin M]$, and so a $(1\pm\epsilon)$-multiplicative approximation to $\Pr_{M\sim\mu_G}[e \in M]$ implies a $(1\pm\epsilon)$-multiplicative approximation to $\Pr_{M\sim\mu_G}[e \notin M]$ (but not vice versa). One might worry this won't generalize beyond matchings, but this isn't a significant issue: We could use our approximate counter to compute an estimate $\hat{q}$ for $\Pr_{M\sim\mu_G}[e \in M]$ and another estimate $\hat{p}$ for $\Pr_{M\sim\mu_G}[e \notin M]$. While $\hat{p}, \hat{q}$ might not sum to 1, we have $1 - \epsilon \leq \hat{p} + \hat{q} \leq 1 + \epsilon$ and so we can instead use the distribution which outputs 1 with probability $\frac{\hat{p}}{\hat{p}+\hat{q}}$ and 0 with probability $\frac{\hat{q}}{\hat{p}+\hat{q}}$. This works more generally, e.g. in the setting of Remark 1.

Claim 4.3. *If there exists an* FPRAS *for estimating the number of matchings in an arbitrary graph, then there exists an* FPAS *for sampling a uniformly random matching in an arbitrary graph.*

Proof (Assuming Deterministic Counter). For simplicity, so that we don't have to deal with the internal randomness of the FPRAS, let's first assume the approximate counter is *deterministic*. So, with probability 1, it will produce a $(1 \pm \epsilon)$-multiplicative approximation in time $\mathsf{poly}(n, 1/\epsilon)$; this is an FPTAS. We'll deal with the general FPRAS case later.

Fix an arbitrary ordering of the edges $e_1, \ldots, e_m$. Our algorithm proceeds as follows: Let $G_0 = G$, let $\widehat{M}$ denote the current matching the algorithm has built up, and for each $i = 1, \ldots, m$, do the following:

- If e_i is incident to an edge already in $\widehat{M}$, i.e. at least one of its endpoints is no longer present in G_{i-1}, skip e_i and move on.
- Otherwise, call our approximate counter to compute a $(1 \pm \eta)$-multiplicative approximation $\hat{q}_i$ to the probability $\Pr_{M \sim \mu_{G_{i-1}}}[e_i \in M]$, where η is some parameter to be determined later (intuitively, $\eta \leq O(\delta/m)$ where δ is the total variation distance we wish to achieve).
- With probability $\hat{q}_i$, add $e_i = \{u_i, v_i\}$ to $\widehat{M}$ and set $G_i = G_{i-1} - u_i - v_i$. Otherwise, don't modify $\widehat{M}$ and set $G_i = G_{i-1} - e_i$.

Then for every matching $M \subseteq E$, writing E_i for the event that e_i has the "correct status" w.r.t. M (i.e. we include e_i in the algorithm's output if $e_i \in M$ and exclude e_i otherwise), then

$$\begin{aligned}\nu(M) &= \prod_{i=1}^{m} \Pr_{\nu}[E_i \mid E_1, \ldots, E_{i-1}] \\ &\leq (1+\eta)^m \cdot \prod_{i=1}^{m} \Pr_{\mu_G}[E_i \mid E_1, \ldots, E_{i-1}] \\ &\leq \exp(\eta m) \cdot \mu_G(M).\end{aligned}$$

Similarly, $\nu(M) \geq \exp(-\eta m) \cdot \mu_G(M)$. It follows by direct calculation that $\|\mu_G - \nu\|_{\mathsf{TV}} \leq 1 - \exp(-\eta m)$, which is $\approx \eta m$ for $\eta \leq O(1/m)$. At this point, we are already quite happy setting $\eta \leq O(\delta/m)$ and getting our sampler. Except, there is a slight subtlety. If we do this, then the running time guarantee from our counting algorithm yields a sampler with running time dependence $\mathsf{poly}(1/\delta)$ instead of $\mathsf{poly}(\log(1/\delta))$. Our final trick is to add a *rejection sampling* step at the very end. To avoid confusion, let's now call the above procedure the "inner algorithm".

Set $\eta \leq O(1/m)$ (now independent of δ), and suppose the above algorithm outputs some matching M. We will accept this matching and output it with probability $\frac{\mu_G(M)}{\nu(M)}$; otherwise, we throw away M and rerun the whole algorithm. Note that we can compute $\nu(M)$ exactly just via the estimates $\hat{q}_i$ we received along the way, since the counting algorithm is deterministic. Furthermore, our approximate counter allows us to efficiently compute a $(1 \pm \eta)$-multiplicative approximation $\hat{Z}$ of $|\Omega_G|$, and so we instead use

$$\Pr[\mathsf{accept} \mid \text{propose } M] = \frac{C}{\nu(M) \cdot \hat{Z}}$$

as our acceptance probability, where $0 < C < 1$ is some universal constant chosen so that the right-hand side doesn't exceed 1. Since $\hat{Z}$ and C are independent of M,

$$\Pr[\text{overall algorithm outputs } M] = \underbrace{\Pr[\text{propose } M]}_{=\nu(M)} \cdot \Pr[\mathsf{accept} \mid \text{propose } M] = \frac{C}{\hat{Z}}.$$

is uniform over all matchings M, i.e. any accepted matching M is perfectly distributed according to μ_G. Furthermore,

$$\Pr[\mathsf{accept} \mid \text{propose } M] = C \cdot \frac{\mu_G(M)}{\nu(M)} \cdot \frac{1}{\mu_G(M) \cdot \hat{Z}} \geq C \cdot \exp(-\eta m) \cdot (1 - \eta),$$

which we can ensure is at least $\Omega(1)$. At the same time,

$$\Pr[\mathsf{accept} \mid \text{propose } M] \leq C \cdot \exp(\eta m) \cdot (1 + \eta) \leq 1,$$

since $\eta \le O(1/m)$ and C is an appropriate constant. Hence, in each call of the "inner algorithm", we accept with constant probability. So, with probability $1-\delta$, we will accept some output after $O(\log(1/\delta))$ attempts; if we reject all of the proposed matchings (which occurs only with probability $\le \delta$), we can output an arbitrary matching (e.g. $\emptyset$), and only incur an additional δ error in the total variation distance. Ultimately, there are only one source of error in the total variation distance (again, assuming our approximate counter is deterministic), namely there is a possibility of rejecting all proposed matchings. $\square$

Remark 3. If instead of stopping after $O(\log(1/\delta))$ attempts, we just keep running the algorithm until a matching is expected, then we would get a *perfect sampler.* Of course, this comes at the expense of having a nondeterministic running time (which is constant in expectation, and $\mathsf{poly}(m)$ with very high probability).

Proof (with Generic FPRAS*).* Now let's suppose we have a randomized approximate counter. Due to the internal randomness of the FPRAS, our estimates $\hat{q}_i$ in the inner algorithm above are random numbers, leading to a random output distribution ν over matchings. Nonetheless, if set the failure probability of the approximate counter to be $\le \tilde{O}(\delta/m)$, then by the Union Bound, with probability $\ge 1-\delta$, every estimate $\hat{q}_i$ over all $O(\log(1/\delta))$ attempts of the inner algorithm will be close to correct within the desired accuracy. Hence, if A is the event that all estimates are accurate and $\widehat{M}$ denotes the output of the overall algorithm, then by the Law of Total Probability

$$\left\|\mathsf{Law}\left(\widehat{M}\right) - \mu_G\right\|_{\mathsf{TV}} \le \underbrace{\left\|\mathsf{Law}\left(\widehat{M} \mid A\right) - \mu\right\|_{\mathsf{TV}}}_{\le\delta \text{ by above}} \cdot \underbrace{\Pr[A]}_{\le 1} + \underbrace{\left\|\mathsf{Law}\left(\widehat{M} \mid \overline{A}\right) - \mu\right\|_{\mathsf{TV}}}_{\le 1} \cdot \underbrace{\Pr\left[\overline{A}\right]}_{\le\delta} \le 2\delta.$$

At the same time, we've only incurred a logarithmic in m/δ overhead. $\square$

Remark 4. The randomness of the output distribution ν might be slightly confusing, e.g. since its randomness is seemingly intertwined with the random coin flips we used in the inner algorithm to decide whether or not to add e_i to $\widehat{M}$. Note however that these sources of randomness are independent.

Another way to think about it is that imagine in background, for the sake of analysis, the approximate counter was used to estimate every marginal probability for every possible execution path of the inner algorithm. Suppose these estimates are all recorded as labels in a depth-m binary tree, where each level-i node corresponds to having made a (valid) decision for $e_1, \ldots, e_i$; see Fig. 1. The root corresponds to initialization of the inner algorithm, the leaves correspond to possibilities for $\widehat{M}$, and the probabilities labeling the nodes tell you the probability of picking one of the children.

Then when the inner algorithm is run, we traverse this tree from the root to a leaf by look up the corresponding estimate $\hat{q}_i$ in a node, and selecting one of the children with probability $\hat{q}_i$. At the end, we pay the running time cost for the m (random) nodes that we query in this process.

5 The "All-or-Nothing" Theorem

We conclude with a very surprising and beautiful theorem due to Jerrum–Sinclair.

Theorem 5.1 (Informal; [SJ89]). *Let Ω and $w : \Omega \to \mathbb{R}_{\ge 0}$ be a "self-reducible" counting problem, and write n for the size of the input. Then for every constant $C > 0$, if there exists a polynomial-time randomized algorithm which outputs an estimate $\hat{Z}$ satisfying $n^{-C} \cdot Z \le \hat{Z} \le n^{C} \cdot Z$ with probability $3/4$, then there is an* FPRAS *for estimating $Z = \sum_{x\in\Omega} w(x)$.*

This result is striking for multiples reasons. First, it says that for self-reducible problems, we can always boost an algorithm producing a n^{10000}-multiplicative approximation to one that produces a $(1 \pm \epsilon)$-multiplicative approximation for any $\epsilon > 0$. Conversely, if there is no FPRAS, then we cannot even hope to get a n^{10000}-multiplicative approximation. This is in stark contrast to the world of combinatorial optimization, where many many different levels of approximation are possible.

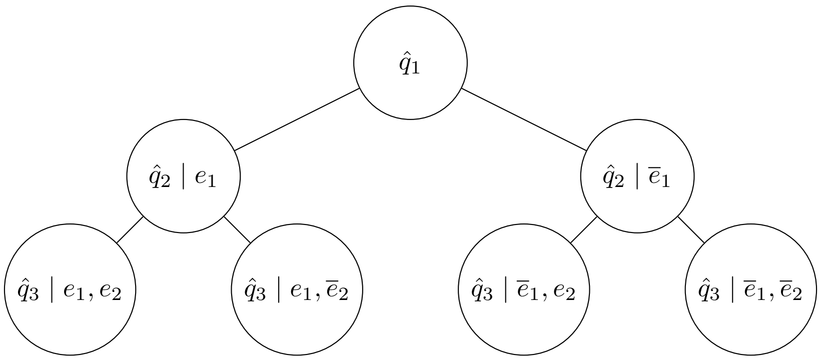
$\hat{q}_1$
$\hat{q}_2 \mid e_1$
$\hat{q}_2 \mid \overline{e}_1$
$\hat{q}_3 \mid e_1, e_2$
$\hat{q}_3 \mid e_1, \overline{e}_2$
$\hat{q}_3 \mid \overline{e}_1, e_2$
$\hat{q}_3 \mid \overline{e}_1, \overline{e}_2$

Figure 1: Small example of the "tree of execution paths" of the inner algorithm. Here, the $\hat{q}_i$ gives the estimate of the marginal of e_i conditioned on all previous decisions in the tree. We go to the left child if we include e_i (assuming this child exists, i.e. e_i doesn't intersect any previously included edge), and go right otherwise. For example in the second level $\hat{q}_2 \mid e_1$ is an estimate of $\Pr_{M\sim\mu_G}[e_2 \in M \mid e_1 \in M]$ and $\hat{q}_2 \mid \overline{e}_1$ is an estimate of $\Pr_{M\sim\mu_G}[e_2 \in M \mid e_1 \notin M]$.

References

[Blu+05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. "Practical Privacy: The SuLQ Framework". In: PODS '05. Baltimore, Maryland: Association for Computing Machinery, 2005, pp. 128–138. ISBN: 1595930620 (cit. on p. 2).

[Bov06] Anton Bovier. *Statistical Mechanics of Disordered Systems. A Mathematical Perspective.* Cambridge Series in Statistical and Probabilistic Mathematics 18. Cambridge University Press, 2006 (cit. on p. 2).

[BUT23] Population Reference Bureau, the U.S. Census Bureau's 2020 Census Data Products, and Dissemination Team. *Why the Census Bureau Chose Differential Privacy.* 2023. (Visited on 2023) (cit. on p. 2).

[DDS21] Daryl DeFord, Moon Duchin, and Justin Solomon. "Recombination: A Family of Markov Chains for Redistricting". In: *Harvard Data Science Review* 3.1 (2021) (cit. on p. 2).

[DN04] Cynthia Dwork and Kobbi Nissim. "Privacy-Preserving Datamining on Vertically Partitioned Databases". In: *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference.* Vol. 3152. Lecture Notes in Computer Science. Santa Barbara, California, USA: Springer, 2004, pp. 528–544 (cit. on p. 2).

[DW22] Moon Duchin and Olivia Walch. *Political Geometry. Rethinking Redistricting in the US with Math, Law, and Everything In Between.* 1st ed. Birkhäuser Cham, 2022 (cit. on p. 2).

[Dwo+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Proceedings of the Third Conference on Theory of Cryptography.* TCC'06. New York, NY: Springer-Verlag, 2006, pp. 265–284. ISBN: 3540327312 (cit. on p. 2).

[FV17] Sacha Friedli and Yvan Velenik. "Statistical Mechanics of Lattice Systems. A Concrete Mathematical Introduction". In: (2017) (cit. on p. 2).

[Inc] Apple Inc. *Differential Privacy.* (Visited on 2023) (cit. on p. 2).

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. "Random generation of combinatorial structures from a uniform distribution". In: *Theoretical Computer Science* 43 (1986), pp. 169–188 (cit. on pp. 4, 5).

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.* The MIT Press, 2009 (cit. on pp. 1, 2).

[MT06] Ravi Montenegro and Prasad Tetali. "Mathematical Aspects of Mixing Times in Markov Chains". In: *Foundations and Trends in Theoretical Computer Science* 1.3 (2006), pp. 237–354 (cit. on p. 2).

[SJ89] Alistair Sinclair and Mark Jerrum. "Approximate counting, uniform generation and rapidly mixing Markov chains". In: *Information and Computation* 82.1 (1989), pp. 93–133. ISSN: 0890-5401 (cit. on p. 8).

[Val79] L.G. Valiant. "The complexity of computing the permanent". In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(79)90044-6 (cit. on p. 3).

[WJ08] Martin J. Wainwright and Michael I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Vol. 1. Foundations and Trends in Machine Learning 1–2. Now Publishers Inc, 2008, pp. 1–305 (cit. on pp. 1, 2).