

Lecture 16 — March 15

Lecturer: Richard Karp

Scribe: Alexandre Bouchard-Côté

Approximate counting and sampling¹

The $\#P$ complexity class

The $\#P$ (“sharp P”) class will be important for the analysis of the algorithms considered in the next few lectures. We will start by reviewing its definition, its hardness and some canonical problems in this class.

Definition 1. *A problem is in $\#P$ if there is a non-deterministic, polynomial-time Turing machine that, for each instance I of the problem, has a number of accepting computations that is exactly equal to the number of distinct solutions for instance I .*

Counting the number of SAT assignments of a CNF formula is an example of $\#P$ -complete problem. Counting the number of perfect matching in a graph is also $\#P$ -complete. This is surprising since the corresponding decidability problem is in P . Another such example: counting the number of linear extensions of a partial ordering.

The class $\#P$ is believed to contain very hard problems. Obviously, a $\#P$ problem must be at least as hard as the corresponding NP problem. If it is easy to count answers, then it must be easy to tell whether there are any answers. Therefore, the $\#P$ problem corresponding to any NP -complete problem must be NP -hard.

We actually have, as a consequence of Toda’s theorem, the following:

Theorem 1. *If $\#P = P$, then the polynomial hierarchy collapses, i.e. $PH = P$.*

No deterministic algorithm is known that can even find the approximate answer of $\#P$ -complete problems to within some reasonable error bound. However, there are probabilistic algorithms that return good approximations to some $\#P$ -complete problems with high probability. We are therefore interested in *approximate counting*.

¹Reference: *The Markov Chain Monte Carlo Method: An Approach To Approximate Counting And Integration* (1996), Mark Jerrum, Alistair Sinclair

We will first approach the problem of *approximate sampling*, and then show a general method to do approximate counting using an approximate sampling algorithm as a black box.

Approximate sampling

From now on, S will be a finite set and π, π' , probability distributions on S . π will be called the *target distribution*, and π' will be the result of our sampling algorithm.

We now define a notion of distance between distributions:

Definition 2. For distributions π_1, π_2 over a common finite set S , the total variation distance is given by:

$$\begin{aligned} D(\pi_1, \pi_2) &:= \frac{1}{2} \sum_{s \in S} |\pi_1(s) - \pi_2(s)| \\ &= \max_{A \subseteq S} \left\{ |\pi_1(A) - \pi_2(A)| \right\} \end{aligned}$$

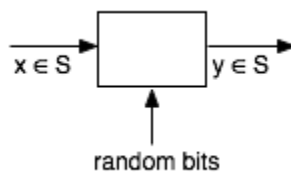
We are aiming for:

Definition 3. A fully polynomial approximate sampling scheme. The input takes the following form:

- X , which will implicitly specify S , π ,
- a tolerance $\delta > 0$.

The requirement is to sample according to π' in time polynomial in both $|X|$ and $\log(\frac{1}{\delta})$, where $D(\pi', \pi) \leq \delta$.

Markov chain Monte Carlo (MCMC) methods are a class of algorithms that will achieve the above goal. The basic idea is to construct a Markov chain that has the desired distribution as its stationary distribution. The state of the chain after a large number of steps is then used as a sample from the desired distribution.



We will discuss MCMC algorithms based on a *random walk* through the set S . The general idea is illustrated in the above diagram. Let us look at some examples before developing the general framework.

Example 1. Let S be the set of 0–1 vectors $\vec{x} = (x_1, \dots, x_n)$ satisfying

$$\vec{a} \cdot \vec{x} \leq b,$$

where $\vec{a} = (a_1, \dots, a_n)$ is a constant vector, and b , a fixed scalar.

A potential solution is to use the following random walk:

```

1  repeat  Given a satisfying solution,  $\vec{x}$ 
2            $i$  is drawn uniformly at random from  $\{1, \dots, n\}$ 
3            $y_j \leftarrow \begin{cases} 1 - x_j & \text{if } j = i \\ x_j & \text{otherwise} \end{cases}$ 
4           if  $\vec{a} \cdot \vec{y} \leq b$ 
5             then return  $\vec{y}$ 
6           else return  $\vec{x}$ 

```

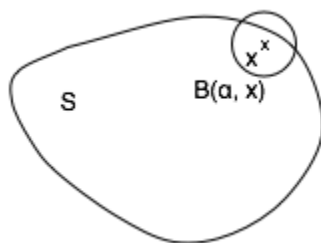
Example 2. Let S be the set of points in a d -dimensional convex body K . Consider now the following algorithm:

```

1  repeat  Given  $\vec{x} \in K$ 
2            $\vec{y}$  is drawn uniformly at random from a ball of radius  $\alpha$  about  $\vec{x}$ 
3           if  $\vec{y} \in K$ 
4             then return  $\vec{y}$ 
5           else return  $\vec{x}$ 

```

In other words, we pick a point at random in a ball of radius α about the current position, and we move there iff the new point falls in S :



Example 3. In this third example we consider the problem of sampling matchings in a fixed graph $G = (V, E)$. We propose the following random walk:

```

1  repeat  Given a matching  $M$ ,
2            $e$  is an edge drawn uniformly at random  $(u, v)$ 
3           if  $e \in M$ 
4             then  $M \leftarrow M \setminus \{e\}$ 
5           if  $e \notin M$  and  $u, v$  are both free
6             then  $M \leftarrow M \cup \{e\}$ 
7           if  $e \notin M$  and neither  $u$  nor  $v$  is free
8             then return  $M$ 
9           if  $e \notin M$ ,  $u$  is free and  $v$  is matched with  $w$ 
10            then  $M \leftarrow M \setminus \{\{v, w\}\} \cup \{e\}$ 
11           if  $e \notin M$ ,  $v$  is free and  $u$  is matched with  $w$ 
12            then  $M \leftarrow M \setminus \{\{u, w\}\} \cup \{e\}$ 

```

We are interested in assessing how rapidly, if at all, do these randomized algorithms achieve a distribution whose total variation distance from the uniform distribution is bounded by δ . In order to answer these questions, we will need some theory of Markov chains.

Markov chains

The general framework to study these algorithms is taken from the (finite state) Markov chain theory. In this subsection, $P_{x,y}$ will denote the transition probability $Pr[X_{t+1} = y | X_t = x]$, and $P_{x,y}^s$, the s -step transition probability $Pr[X_{t+s} = y | X_t = x]$. We start with some terminology.

Definition 4. A Markov chain is irreducible if any state is reachable from any other, i.e.:

$$\forall x, \forall y, \exists t \text{ s.t. } P_{x,y}^t > 0$$

It is aperiodic if it satisfies the following²

$$\forall g \geq 2, \exists x, r \text{ with } g \nmid r \text{ s.t. } P_{x,x}^r > 0$$

It is ergodic if it is both irreducible and aperiodic.

²alternatively, the negation of this property, periodicity, can be defined as follow: a process is periodic if there exists at least one state to which the process will continually return with a fixed time period (greater than one). Aperiodic means that there is no such state.

Ergodic chains will be of special interest to us because they are guaranteed to have a unique *stationary distribution*³.

Theorem 2. *An ergodic distribution has a unique stationary distribution π , i.e. a distribution satisfying:*

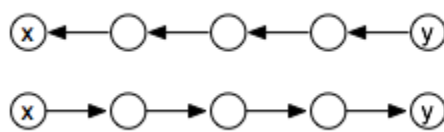
$$\forall x, \forall y, \lim_{t \rightarrow \infty} P^t(x, y) = \pi(y)$$

We will also need the notion of reversibility:

Definition 5. *A Markov chain is reversible if*

$$\forall x, \forall y, \pi(x)P_{x,y} = \pi(y)P_{y,x}$$

Pictorially, reversibility tells us that the following two paths:



have the same probability under the stationary distribution π . This effectively enables us to see transitions as undirected edges between states.

Example 4. Random walk on a graph

Let $G = (V, E)$, $V = \{1, 2, \dots, n\}$, $|E| = m$ and $d(i) :=$ degree of vertex i . We define the transition matrix as:

$$P_{i,j} := \begin{cases} \frac{1}{d(i)} & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that the stationary distribution in this case is:

$$\pi(i) = \frac{d(i)}{2m}$$

Moreover, the chain is reversible. Indeed, whenever $\{i, j\} \in E$ we have:

$$\pi(i)P_{i,j} = \frac{d(i)}{2m} \cdot \frac{1}{d(i)} = \frac{1}{2m} = \pi(j)P_{j,i}$$

³Note that since we are going to *build* Markov chains, it is going to be easy to make sure that these properties hold by applying simple transformations on the chains, for instance by introducing self-loops.

We will need more than just convergence in the subsequent analyses. A concept of rate of convergence, namely *mixing times*, will be useful.

Definition 6. *The mixing time is defined as:*

$$\tau_x(\epsilon) := \min \{t : \forall t' \geq t . D(P_x^{t'}, \pi) \leq \epsilon\},$$

where π is the stationary distribution and D denotes the total variation distance.

Hereafter, we assume that the chain is ergodic and reversible, and that, for all x , $P_{x,x} \geq \frac{1}{2}$.

The problem is, for a particular Markov chain (proposed as a solution to a sampling problem):

- to show that the stationary distribution is the desired target distribution (often the uniform distribution),
- to derive an upper bound on the mixing time.

We will assume the following facts:

- the eigenvalues of P are nonnegative,
- the largest eigenvalue is 1.

Let us call λ_2 the second-largest eigenvalue. It turns out that λ_2 plays an important role for bounding the mixing time:

Theorem 3.

$$\tau_x(\epsilon) \leq \frac{1}{1 - \lambda_2} \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)$$

It is, in many cases, not convenient to estimate λ_2 and hence the leading factor of the above expression. For this reason, other ways to bound the mixing time have been explored. The concept of *conductance*, a combinatorial property often easier to check, provides an interesting alternative:

Definition 7. *The conductance Φ is defined as*

$$\Phi := \min_{\{A \subseteq S : 0 < \pi(A) \leq \frac{1}{2}\}} \frac{Q(A, \bar{A})}{\pi(A)}$$

Conductance has the following property:

Theorem 4.

$$2\Psi^2 \leq 1 - \lambda_2,$$

and this gives us a bound on the mixing time:

Corollary 1.

$$\tau_x(\epsilon) \leq 2\Phi^{-2} \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)$$

Example 5. An h -expander is a regular graph $G = (V, E)$ of degree D such that, for every set $S \subseteq V$ such that $0 < |S| \leq \frac{|V|}{2}$, the number of edges between S and \bar{S} is at least $h|S|$.

Assume $V = \{1, 2, \dots, n\}$, and define a Markov chain as follows:

$$P_{i,j} = \begin{cases} \frac{1}{2} & \text{if } j = i \\ \frac{1}{2D} & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

This is a reversible ergodic chain with $\pi(i) = \frac{1}{n}$ for all i . Observe that:

$$Q(S, \bar{S}) \geq \frac{1}{n} h |S|,$$

$$\pi(S) = \frac{1}{n} |S|,$$

$$\Phi \geq h,$$

$$\tau_x(\epsilon) \leq \frac{2}{h^2} \log \left(\frac{n}{\epsilon} \right)$$

Thus, a random walk on an expander mixes in logarithmic time.

We now introduce the last tool that we will need for the analysis of mixing times.

Method of canonical paths

We can cast the analysis of mixing times in a flow problem: let V be the set of states, and define a graph $G = (V, E)$, where $E = \{\{x, y\} : P_{x,y} > 0\}$. For each ordered pair x, y of states, define a *canonical path* $\gamma_{x,y}$ from x to y . Let

$$\zeta := \max_{\{u,v\} \in E} \frac{\sum \pi(x)\pi(y)}{Q(u,v)},$$

where the sum is over all pairs x, y such that e lies in $\gamma_{x,y}$.

With this definition, the following holds:

Theorem 5.

$$\Phi \geq \frac{1}{2\zeta}$$

Again, we obtain as a corollary a bound on the mixing time:

Corollary 2.

$$\tau_x(\epsilon) \leq 8\zeta^2 \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right)$$

To summarize, we have the following three bounds on τ_x :

$$\begin{aligned} \tau_x(\epsilon) &\leq \frac{1}{1 - \lambda_2} \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) \\ &\leq 2\Phi^{-2} \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) \\ &\leq 8\zeta^2 \left(\log \left(\frac{1}{\pi(x)} \right) + \log \left(\frac{1}{\epsilon} \right) \right) \end{aligned}$$

In practice the 3 bounds above, second eigenvalue, conductance, and canonical paths bounds, are typically in increasing order of usability.