COMPSCI 630: Randomized Algorithms April 11, 2018

Lecture 21: Approximate Counting II – Matchings, MCMC

Lecturer: Rong Ge *Scribe: Abe Frandsen*

1 Overview

In this lecture, we explore the use of Markov chains for sampling in counting problems. We apply this technique to the problem of counting perfect matchings in a bipartite graph.

2 Introduction to MCMC

Recall the general Monte-Carlo FPRAS for counting problems:

- sample uniformly at random from some set S
- check what fraction $\hat{p}$ of samples have the property of interest
- output $|S| \cdot \hat{p}$.

As we discussed in the previous lecture, this approach has the following potential problems:

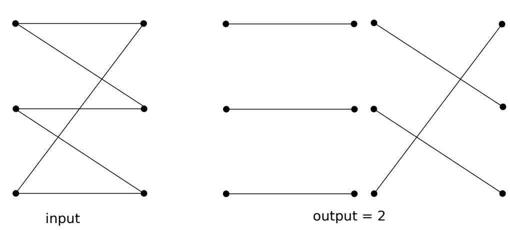
1. if the true fraction p is too small, we can't get a sufficiently accurate estimate
2. if S is a complicated set:
   (a) we may not know $|S|$
   (b) it is not easy to sample from S

We can use Markov chains to address these issues.

Recall that if a Markov chain can be described by an undirected, regular graph, then the stationary distribution is uniform. This leads to the following **idea (MCMC)**: construct a *fast-mixing* Markov chain on S whose stationary distribution is uniform, and run the Markov chain until mixing in order to sample from S. By fast-mixing, we mean something like $\log |S|$ time complexity. This idea is called Markov Chain Monte-Carlo, or MCMC.

3 Application: Counting Perfect Matchings

Given a bipartite graph with n vertices on each side and with each vertex having degree at least $n/2$, we seek to compute the number of perfect matchings. See the figure below for an example.



Note that in a complete bipartite graph with n vertices on each side, each perfect matching corresponds to a permutation on n elements, and so there are $n!$ such possible perfect matchings. Thus, it won't work to just sample uniformly from this set of all possible matchings, since the fraction of actual matchings in the graph might be too small.

Consider this **idea**: reduce the problem to simpler problems. In particular, suppose that we know the number of matchings with $n-1$ edges (we call this quantity $\#(n-1)$matchings). Let S_n be the set of $(n-1)$-matchings AND n-matchings. If we can sample from S_n, then we can estimate $\frac{\#n\text{matchings}}{\#(n-1)\text{matchings}}$, and this will be fairly accurate as long as there is not a huge difference between $\#n$matchings and $\#(n-1)$matchings. Finally, simply output

$$\#(n-1)\text{matchings}\left(\frac{\#n\text{matchings}}{\#(n-1)\text{matchings}}\right),$$

where the second term is our estimate of that fraction. Of course, this only works if we have a good estimate of $\#(n-1)$matchings. But we can estimate this number recursively by the same process.

In general, we let S_k be the set of $(k-1)$- and k-matchings, and then we sample from S_k to estimate the fraction $\frac{\#k\text{matchings}}{\#(k-1)\text{matchings}}$. For the base case, we simply observe that $\#1\text{matchings} = \#\text{edges}$. After successively estimating each fraction, we simply output

$$\#\text{edges}\cdot\frac{\#2\text{matchings}}{\#1\text{matchings}}\cdot\frac{\#3\text{matchings}}{\#2\text{matchings}}\cdots\frac{\#n\text{matchings}}{\#(n-1)\text{matchings}}.$$

Each individual ratio can be estimated independently using MCMC sampling on the sets S_k.

4 Analyzing the Matching-Counting Algorithm

In order to establish the correctness of the algorithm described above, we need to address the following points:

1. show the ratios $\frac{\#k\text{matchings}}{\#(k-1)\text{matchings}}$ are not too small or large
2. construct a Markov chain on S_k with uniform stationary distribution
3. prove that the chain is fast mixing

Let's look at each of these points in turn.

4.1 Bounding the Ratios

Claim 1.

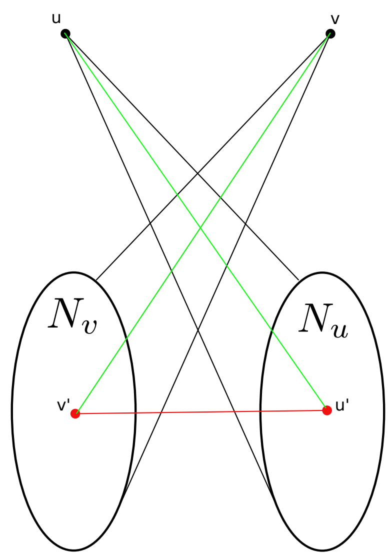
$$\frac{1}{n^4} \le \frac{\#kmatchings}{\#(k-1)matchings} \le n^2$$

Proof. We will construct mappings between the set of k-matchings and $(k-1)$-matchings. Define $f:\{k\text{-matchings}\}\to\{(k-1)\text{-matchings}\}$ as follows: given a k-matching, remove the first edge (relative to some pre-specified ordering on the edges). Note that this map may be neither one-to-one nor onto. However, for each $(k-1)$-matching, there are at most $\#\text{edges} \le n^2$ k-matchings that map to it, since each such k-matching corresponds to a single edge (the one that was removed). Hence, we have that

$$\frac{\#k\text{matchings}}{\#(k-1)\text{matchings}} \le n^2.$$

Now define the map $g : \{(k-1)\text{-matchings}\} \to \{k\text{-matchings}\}$ as follows: given a $(k-1)$-matching, find the first augmenting path of length at most 3, and apply this to obtain a k-matching. In this case, an augmenting path is a path between two unmatched vertices.

How do we know such an augmenting path of length at most 3 exists? Fix a $(k-1)$-matching as well as two unmatched vertices u and v. If there is an edge between u and v, this gives a direct augmenting path of length 1, and we can simply add the edge to the matching to obtain a k-matching. Similarly, if either u or v is connected to any other unmatched vertex, we can just add that edge to obtain a k-matching. So now assume that neither u nor v are connected to an unmatched vertex. Let N_u be the neighbors of u and let N_v be the neighbors of v. By assumption, both N_u and N_v have at least $n/2$ elements, and all of these elements are matched. Hence, at least one element of N_u must be matched to an element of N_v, since there are at most $\frac{n}{2}-1$ vertices outside of N_v that the elements of N_u could possibly be matched to. Let (u', v') be the match edge between N_u and N_v. We simply remove edge (u', v'), and add edges (u, u') and (v, v'), and this creates a k-matching using an augmenting path of length 3. See the figure below for an illustration of this.



These augmenting paths involve at most 4 vertices, so there are at most n^4 of them. Hence, for each k-matching, at most n^4 $(k-1)$-matchings can map to it. This implies that $\#(k-1)\text{matchings} \leq n^4 \#k\text{matchings}$, or in other words,

$$\frac{\#k\text{matchings}}{\#(k-1)\text{matchings}} \geq \frac{1}{n^4}.$$

□

4.2 Constructing the Markov Chain

We define a Markov chain on the set S_k via the following transition rules staring from some matching M:

- with probability $1/2$, do nothing
- randomly select an edge $e = (u, v)$
- if $|M| = k$:
  - if $e \in M$, remove e from M

- if $e \notin M$, do nothing

- if $|M| = k-1$:
  - if both u and v are unmatched, add e to M
  - else if u is matched to a vertex w and v is unmatched, swap (u,w) to (u,v)
  - else if v is matched to a vertex w and u is unmatched, swap (v,w) to (u,v)
  - else if both u and v are matched, do nothing

We need to show that $P(M_1 \to M_2) = P(M_2 \to M_1)$ for all matchings M_1 and M_2 in S_k, since this will imply a uniform stationary distribution. You can check that this is indeed the case by thinking about the different ways a transition from M_1 to M_2 can occur, and vice versa.

4.3 Establishing Fast Mixing

To show that our Markov chain has the fast mixing property, we consider the following **idea**: "embed" a complete graph as a "flow" to the Markov chain graph G (i.e. G is the graph corresponding to the Markov chain we constructed above, so each node of G corresponds to a matching, i.e. an element of S_k). Assume that each edge is used only bN times, where N is the number of nodes in G.

Claim 2.

$$\varphi(G) \geq \frac{1}{b}$$

Proof. Let S be some subset of the nodes of G. In a complete graph, there are $|S||\bar{S}|$ paths between S and $\bar{S}$, and each of these paths will flow through an edge in G. Since each edge in G is used at most bN times, we have

$$E(S,\bar{S}) \cdot b \cdot N \geq |S||\bar{S}|,$$

where $E(S,\bar{S})$ is the number of edges between S and $\bar{S}$ in G. In other words, we have

$$\frac{E(S,\bar{S})}{|S||\bar{S}|/N} \geq \frac{1}{b},$$

and this leads to the claimed result. □

It remais to construct this "embedding", but we leave that to the references.