Counting and Sampling **Fall 2017**

Lecture 10: More on Matchings, Conductance

Lecturer: Shayan Oveis Gharan *October 27*

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

This is supposed to be the last lecture on Markov chains. Firstly, let us prove several additional properties of the chain we studied last time:

Say, m_k is the number of matchings of size k in G. Last time we say how to sample matching M with probability proportional to $\lambda^{|M|}$. By the equivalence of counting and sampling, we can use that algorithm to estimate the partition function

$$Z = \sum_k m_k \lambda^k.$$

Now, let us use the sampling theorem to esimate m_k. We use the following lemma

Lemma 10.1. *The sequence m_k is log-concanve, i.e., for all k,*

$$m_{k-1} m_{k+1} \leq m_k^2.$$

This fact follows directly from the fact that the matching polynomial $\sum_k m_k x^k$ is real rooted. We will say more about this in future.

Claim 10.2. *If $\lambda = \frac{m_{k-1}}{m_k}$, then $m_{k'} \lambda^{k'}$ is maximized at $k' = k-1$ and $k' = k$.*

Proof. First notice that $m_k \lambda^k = m_{k-1} \lambda^{k-1}$. Secondly, since the log function is monotonic, it suffices to check that $\log(m_k \cdot \lambda^k) = \log m_k + k \cdot \log \lambda$ is maximized at $k' = k$. Since the sequence $\log m_{k'}$ is concave it suffices to show that $m_k \lambda^k \geq m_{k+1} \lambda^{k+1}$ and that $m_{k-1} \lambda^{k-1} \geq m_{k-2} \lambda^{k-2}$. To see the rst of these, note that by log-concavity

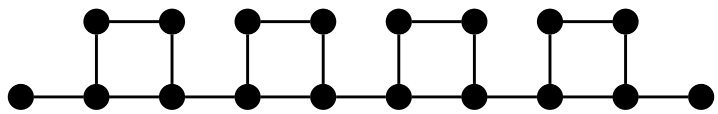
$$\frac{m_{k+1} \lambda^{k+1}}{m_k \lambda^k} = \lambda \frac{m_{k+1}}{m_k} = \frac{m_{k-1} m_{k+1}}{m_k^2} \leq 1.$$

The argument for the second inequality is entirely similar. □

Now, to estimate m_k, it is enough to gradually increase λ until we see many $k-1$ and k matchings in the samples. Then, we can estimate m_k easily by equivalence of counting and sampling. Note that this algorithm works fro any graph G as long as $\frac{m_{k-1}}{m_k}$ is polynomially small.

10.1 Estimating the number of Perfect Matchings

In principal one can use the above algorithm to estimate the number of perfect matchings of a graph G with $2n$ vertices as long as $\frac{m_{n-1}}{m_n}$ is polynomially small. This holds for many families of graphs including random graphs and dense graphs. However, it turns out that there are (bipartite) graphs (with $2n$ vertices) that have exponentially more near perfect matchings (i.e., matchings with $n-1$ edges) than the number of perfect matchings. The above algorithm fails See the following picture



It remained an open problem for two decades to design an FPRAS for counting the number of perfect matchings of a bipartite graph; equivalently, to design an algorithm that gives a $1+\epsilon$ approximation of the permanent of a nonnegative matrix. The question was resolved in breakthrough work of Jerrum, Sinclair and Vigoda [JSV04]. Let us discuss the main ideas of their proofs:

The first point to note is that one can run the same Markov chain that we discussed last time (with $\lambda = 1$) only on the space of perfect and near perfect matching. This was already observed in the work of [JS89]. It turns out that the chain mixes rapidly. But as we discussed above the stationary distribution may be concentrated on near perfect matchings.

The first idea of [JSV04] is a re-normalization idea. Let P be the set of all perfect matchings of G. We say a matching M has a hole at x, y if the x, y are unsaturated in M. Note that each near perfect matching has exactly two holes. Let $N(u, v)$ be the set of all near perfect matchings that have holes in u, v. The idea is to set the weight of all matchings in $M \in N(u, v)$ to be $\tilde{\pi}(M) = \frac{|P|}{|N(u,v)|}$. This implies that the sum of the weights of all class of near perfect matchings $N(u, v)$ is exactly $|P|$. Since there are only n^2 classes of holes, if we can generate a sample from $\tilde{\pi}$ with probability $1/(1 + n^2)$ it is a perfect matching. Note that we can naturally use the Metropolis rule to define a Markov chain with stationary distribution $\tilde{\pi}$. It turns out that one can extend the proof of [JS89] to work in this weighted regime. This is the most technical part of the proof of [JSV04], because the mixing time bound cannot depend on λ or the ratio of near prefect to perfect matchings. In fact, this is the main part of the proof which does not extend to the setting of general of graphs.

Lastly, there is a fundamental issue with the above approach; we do not know $|P|$ and $|N(u, v)|$'s, so it is impossible to implement the above Markov chain. The second idea is to use a *homotopy* method. Define an activity $\lambda_{x,y}$ for every pair of vertices x, y and weight each matching M with $\lambda(M) = \prod_{e \in M} \lambda_e$. Originally, let $\lambda_{x,y} = 1$ for all pairs (even the nonnedges). In such a case, we can exactly compute $\lambda(N(u, v))$ and $\lambda(P)$. Now, we gradually decrease the activity of nonnedges. Each time we use the Markov chain to re-compute a new estimate of $\lambda(N(u, v))$ and $\lambda(P)$ for all pairs u, v. Using that estimate we recompute the weight of each matching. We continue this until the activity of each nonnedge is as small as $1/n!$. In this case we can essentially ignore the nonnedges, and just sample a perfect matching.

Let us conclude this part with a few open problems:

i) An FPRAS for perfect matchings in general graphs

ii) A deterministic $1 + \epsilon$ approximation algorithm for bipartite graphs

iii) A different randomize $1 + \epsilon$ approximation algorithm for bipartite graphs. To this date the above algorithm of [JSV04] is the only algorithm that we are aware of to sample perfect matchings in bipartite graphs. Unfortunately, the algorithm is very slow for practical applications.

10.2 Conductance

Let us discuss another approach to bound the mixing time. Recall that for a reversible Markov chain we can construct a graph $G = (V, E)$ where the weight of every edge (x, y) is $w(x, y) = \pi(x)K(x, y)$. Given a (weighted) graph G, the *conductance* of a set S is defined as

$$\phi(S) = \frac{w(S,\overline{S})}{\sum_{x \in S} d_w(x)} = \frac{\sum_{x\in S, y \notin S} w(x,y)}{\sum_x d_w(x)}.$$

Now, suppose G is the graph constructed from a reversible Markov chain. It follows from the above definition that

$$\phi(S) = \frac{\sum_{x\in S, y\notin S} Q(x,y)}{\pi(S)}.$$

Let

$$\phi(G) = \min_{S:\pi(S)\le 1/2} \phi(S).$$

There is a fundamental inequality in spectral graph theory called the Cheeger's inequality which asserts the following:

Theorem 10.3 (Cheeger's Inequality)**.** *For any graph G,*

$$\frac{1-\lambda_2}{2} \le \phi(G) \le \sqrt{2(1-\lambda_2)}$$

where λ_2 is the second largest eigenvalue of K. Equivalently, $1-\lambda_2$ is the Poincaré constant of the chain

It follows from the above inequality that we can bound the mixing time as a function $1/\phi(G)$.

Corollary 10.4. *For any reversible Markov chain K, with associated graph G,*

$$\tau_x(\epsilon) \le O\left(\frac{\log \frac{1}{\epsilon\cdot\pi(x)}}{\phi(G)^2}\right)$$

where as usual $\tau_x(\epsilon)$ is the time to get to total variation distance ϵ for the chain started at state x.

Note that the above corollary also shows that there is at most a square root loss in using the conductance (as opposed to the Poincaré constant) to bound the mixing time. Therefore, if the inverse Poincaré constant is a polynomial in n, then so is $1/\phi(G)$.

10.3 Duality of Multicommodity Flow

Next, let us discuss the connection between the conductance and the path technology that we have been considering so far. It turns out that the sparsest cut problem is the dual to the multicommodity flow problem that we have been using to bound the mixing time.

First, let us define a modified conductance S,

$$\phi'(S) = \frac{\sum_{x\in S, y\notin S} Q(x,y)}{\sum_{x\in S, y\notin S} \pi(x)\pi(y)} = \frac{Q(S,\overline{S})}{\pi(S)\pi(\overline{S})}.$$

Observe that if $\pi(S) \le 1/2$, then

$$\phi'(S) \le \phi(S) \le 2\phi'(S).$$

Let

$$\phi'(G) = \min_S \phi'(S).$$

Observe that $\phi'(G) \leq \phi(G) \leq 2\phi'(G)$.

Now, suppose we have routed a multicommodity flow f, where the flow between each pair x, y is exactly $\pi(x)\pi(y)$. Observe that the maximum congestion is at least $1/\phi'(G)$. The reason is as follows: Say $S = \operatorname{argmin}_T \phi'(T)$. Observe that the total amount of flow out of the set S is at least $\pi(S)\pi(\overline{S})'$. But the sum of the congestion of all edges in the cut $(S, \overline{S})$ is exactly $Q(S, \overline{S})$. Therefore, there must be an edge in this cut with congestion $1/\phi'(S) = 1/\phi'(G)$.

Say, $\rho(f) = \max_e \frac{f(e)}{Q(e)}$ is the maximum congestion of the multicommodity flow f. We just showed that for any flow f,

$$\rho(f) \geq \frac{1}{\phi'(G)}.$$

It turns out that the inverse of the above inequality is also true up to a factor $O(\log(|\Omega|))$. In other words, there exists a multicommodity flow such that $\rho(f) \leq \frac{\log(|\Omega|}{\phi'(G)}$. Let us say a little bit more about this proof.

It turns out that one can write a linear program to find the optimum multicommodity flow. For all x, y, let $\mathcal{P}_{x,y}$ be the set of all paths from x to y. Also, for a path P, let $f(P)$ be the flow that we route along the path P.

$$\begin{aligned}
\min \quad & \rho & \\
\text{s. t.} \quad & \sum_{P \in \mathcal{P}_{x,y}} f(P) \geq \pi(x)\pi(y) & \forall x, y \\
& \rho \cdot Q(e) \geq \sum_{P : e \in P} f(P) & \forall e, \\
& f(P) \geq 0 & \forall e.
\end{aligned} \tag{10.1}$$

Observe that the optimum ρ gives the maximum congestion of the optimum flow. Note that there are exponentially many variables in the above Linear program. But, we are using this program only for the sake of the analysis. Furthermore, one can write an equivalent description of this program with only polynomially in $|\Omega|$ many variables. The dual of the above program is as follows:

$$\begin{aligned}
\max \quad & \sum_{x,y} \lambda_{x,y} \pi(x)\pi(y) & \\
\text{s. t.} \quad & \sum_{e \in P} \alpha_e \geq \lambda_{x,y} & \forall P \in \mathcal{P}_{x,y} \\
& \sum_e \alpha_e Q(e) = 1 & \\
& \alpha_e \geq 0 & \forall e.
\end{aligned}$$

Observe that since we can scale α_e's with $\lambda_{x,y}$'s while still satisfying the first constraint of the above program, we can rewrite the above linear program as follows:

$$\begin{aligned}
\max \quad & \frac{\sum_{x,y} \lambda_{x,y} \pi(x)\pi(y)}{\sum_e \alpha_e \cdot Q(e)} & \\
\text{s. t.} \quad & \sum_{e \in P} \alpha_e \geq \lambda_{x,y} & \forall P \in \mathcal{P}_{x,y}, \\
& \alpha_e \geq 0 & \forall e.
\end{aligned}$$

Fix, an optimum α_e, and think of α_e's as weights on the edges. It follows that the optimum $\lambda_{x,y}$'s is the shortest path metric with respect to α_e's. So, we can think of the above program as follows: We construct a metric on the graph G by assigned a nonnegative weight to every edge, such that the ratio of the average distance between random pairs of vertices to the average length of the edges is maximized. So, the follows is equivalent to the above:

$$\begin{aligned} \min \quad & \frac{\mathbb{E}_{e\sim Q}\left[d(e)\right]}{\mathbb{E}_{x,y\sim\pi}\left[d(x,y)\right]} \\ \text{s. t.} \quad & d(x,y) \le d(x,z) + d(z,y) \quad \forall x,y,z \\ & d(x,y) \ge 0 \qquad\qquad\qquad\qquad \forall x,y. \end{aligned} \tag{10.2}$$

It turns out that the above program is a linear programming relaxation of the conductance problem. To see that observe that if we define the metric to be the cut metric with respect to the set optimum S of minimum conductance, i.e., we set

$$d(x,y) = \begin{cases} 1 & \text{if } x \in S, y \notin S \text{ or } x \notin S, y \in S \\ 0 & \text{otherwise} \end{cases}$$

then,

$$\mathbb{E}_{e\sim Q}\left[d(e)\right] = \sum_e Q(e)d(e) = Q(S,\overline{S})$$

and

$$\mathbb{E}_{x,y\sim\pi}\left[d(x,y)\right] = \sum_{x,y} \pi(x)\pi(y)d(x,y) = \pi(S)\pi(\overline{S}).$$

This shows that the optimum of (10.2) is always smaller than $\phi'(G)$. But by duality we know that the optimum of (10.2) is equal the maximum congestion of the optimum flow. Leighton and Rao [LR99] have shown that LP (10.2) can be rounded to a set S losing a factor at most $\log(|\Omega|)$. That is, there is a set S such that $\phi'(S)$ is at most $\log(|\Omega|)$ times the optimum of (10.2). This shows that the congestion of the optimum multicommodity flow f^* is at most

$$\rho(f^*) \le O\left(\frac{\log|\Omega|}{\phi'(G)}\right).$$

References

[JS89] M.R. Jerrum and A.J. Sinclair. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82:93–133, 1989. 10-2

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, July 2004. 10-2

[LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of The ACM*, 46:787–832, November 1999. 10-5