Lecture 16: October 24

Lecturer: Alistair Sinclair *Scribes: Gábor Pete and Sam Riesenfeld*

We have seen some examples of using the multicommodity flow method to prove rapid mixing for a Markov chain, but it was usually enough to direct the flow between any pair of vertices through a single path to get a good lower bound on the eigenvalue gap. Now we will see a more sophisticated argument, based on [MS99].

16.1 Random walks on truncated hypercubes

Our sample space is the truncated hypercube $\Omega = \{x \in \{0,1\}^n : \sum_{i=1}^n a_i x_i \leq b\}$, where $a_i \geq 0$, $b \geq 0$. We will consider the following nearest neighbour random walk:

1. At a vertex $x \in \Omega$, do nothing with probability $1/2$;
2. pick an index $i \in \{1, \ldots, n\}$ randomly with probability $\frac{1}{2n}$, and flip the coordinate x_i if we remain inside Ω with this move; else do nothing.

This is a symmetric, irreducible and aperiodic, lazy chain, with uniform stationary distribution. Though rapid mixing $\Theta(n \log n)$ for the complete hypercube has been well-known for a long time, along with all the eigenvalues, etc., and it is hard to imagine how truncation by a hyperplane could create bottlenecks, it was a long-open problem to prove even arbitrary polynomial bound on the mixing time.

The continuous analogues, Brownian motion or continuous-step random walks in a convex body, are also well-known to be rapidly mixing, see e.g. [LS93], but we run into problems when discretizing the sample space. Note also that our sample space is very natural from the computer scientist's point of view: Ω is the set of feasible solutions to the "0-1 knapsack problem" with n objects with weights a_i and knapsack capacity b.

Theorem 16.1 *For any $\{a_i\}_{i=1}^n$ and b, $\tau_{\text{mix}} = O(n^{4.5+\epsilon})$.*

We will prove a weaker polynomial bound here, by constructing a multicommodity flow with low cost $\rho(f)$ and small length $l(f)$. We will look at the vertices $x \in \Omega$ as subsets of $\{1, \ldots, n\}$, i.e. $x \equiv \{i : x_i = 1\}$, and we will denote the symmetric difference of two sets by $x \oplus y$. Note that the shortest paths between x and y are just the permutations of the set $x \oplus y$.

Since we have lost the symmetries of the complete hypercube, we cannot hope for a nice deterministic rule to assign a single or a small set of flow-carrying paths to each pair of vertices x and y. So we are going to spread our flow somewhat randomly among the shortest paths. However, an ordinary uniform shortest path would usually leave the truncated hypercube, so we have to condition on staying inside Ω. But this heavy conditioning will tend to send too much flow away from the hyperplane, towards the origin, so the cost of the flow would go high by overloading the edges close to the origin. Therefore, we would like to restrict ourselves to shortest paths that stay near the straight line connecting the geometric points x and y, i.e. to paths which are somewhat "balanced", but we still would like to use the set of shortest paths in an "almost uniform" way, to avoid overloading any edges. We can formulate these conditions as follows.

Let $\{w_i\}_{i=1}^m$ be a set of arbitrary real weights, and set $w = \sum_{i=1}^m w_i = w$, $B = \max_i |w_i|$. In our application we will have $m = |x \oplus y|$, and $w_i = a_i$ for $i \in y \setminus x$, and $w_i = -a_i$ for $i \in x \setminus y$.

Definition 16.2 *A permutation $\pi \in S_m$ is called* **λ-balanced** *for some $\lambda > 0$, if for all $k \in \{1, \ldots, m\}$,*

$$\min\{0, w\} - \lambda B \leq \left| \sum_{i=1}^{k} w_{\pi(i)} \right| \leq \max\{0, w\} + \lambda B.$$

Definition 16.3 *A random variable π taking values in S_m is called* **γ-uniform** *for some $\gamma \geq 1$, if for all $k \in \{1, \ldots, m\}$ and $U \subseteq \{1, \ldots, m\}$ with $|U| = k$,*

$$\Pr\big[\pi(\{1, \ldots, k\}) = U\big] \leq \gamma \binom{m}{k}^{-1}.$$

Note that for a uniform random permutation π, each of the above probabilities is exactly $\binom{m}{k}^{-1}$.

Now the really interesting claim is that there exist random permutations satisfying both conditions.

Theorem 16.4 *For any set of weights $\{w_i\}_{i=1}^m$, there exists a random permutation $\pi \in S_m$ that is 7-balanced with probability 1, and $O(m^2)$-balanced.*

We will give the proof next time, while now we show how to construct a good flow using these random permutations.

For simplicity, let's assume $1 \leq a_i \leq B$, and let's use the notation $a(x) = \sum_{i=1}^n a_i x_i$. Consider any pair of vertices x and y. Since these vertices can lie very close to the boundary $\{z : a(z) = b\}$, first pick $x_1 \subseteq x$ with $a(x) - 7B \leq a(x_1) \leq b - 7B$. Because of our conditions on the weights a_i, this can be done by deleting at most $7B$ elements of x. Here we also assume that the boundary hyperplane is not very close to the origin — note that otherwise the whole problem would be trivial anyway. Pick $y_1 \subseteq y$ similarly. Now we are going to design the flow from x to y (with demand $D(x, y) = |\Omega|^{-2}$) in three stages:

1. Send all flow along a single path $x \to x_1$, i.e. delete the elements of $x \setminus x_1$ in some fixed order.
2. Route the flow from x_1 to y_1 by distributing it among the shortest paths according to a 7-balanced $O(m^2)$-uniform random permutation of the m elements of $x_1 \oplus y_1$. Since each permutation we use is 7-balanced, none of these paths from x_1 to y_1 will leave Ω.
3. Send all flow along a single path from y_1 to y.

16.1.1 Analysis of flow

We assume that the total $x \to y$ flow is $\frac{1}{N^2}$, where $N = |\Omega|$. Then the capacity of edge $e = (z, z')$ is:

$$C(e) = N^2 \cdot (\pi(z) P(z, z')) = N^2 \frac{1}{N} \frac{1}{2n} = \frac{N}{2n}.$$

Our aim is to show that the total flow along e is at most αN, since then the cost of the flow will be bounded by

$$\frac{f(e)}{C(e)} \leq \frac{\alpha N}{\frac{N}{2n}} = 2n\alpha.$$

If α is small, i.e. polynomial in n, then this gives a good bound on the mixing time.

Instead of finding a bound on the total flow through any edge directly, we will get a bound on the total flow through any vertex, which then gives a bound on the flow through edges. It is sufficient to prove that the total flow through any vertex of G_Ω is at most αN.

Fix an arbitrary vertex z. The total flow through z from Stage 1 and Stage 3 paths is at most $2 \cdot 7B \cdot n^{7B} = O(n^{7B})$. (Notice that as $7B$ appears in the exponent, this flow and the analysis depend upon a fixed lower and upper bound on the weights; they are too crude for arbitrary weights.) For stage 2, we encode each (x_1, y_1) pair that sends flow through z using a pair (E, U), where $E \in \Omega$ and $U \subseteq \{1, \ldots, m\}$ for $m = |x_1 \oplus y_1|$, i.e. m is the size of the symmetric difference between x_1 and y_1.

Note that whatever result we get must be multipled by $O(n^{14B})$ to take into account all the possible pairs (x, y) that use the path $x \to x_1 \to y_1 \to y$.

For the encoding, let $E = x_1 \oplus y_1 \oplus z$, i.e. E is the part of $x_1 \oplus y_1$ that is missing from z. Given E, we can clearly recover $x_1 \oplus y_1 = z \oplus E$, and we can recover $x_1 \cap y_1 = z \cap E$. Thinking of this in terms of the knapsack problem, it is like the situation where we know which items belong to both knapsacks x_1 and y_1 and which items belong to one and not the other, but we still do not know which items belong to x_1 and which belong to y_1. This problem arises because we are not processing the items in a deterministic order, whereas previously in similar contexts we used a deterministic order. So we need another part U of the encoding to ensure complete information. U specifies those coordinates, i.e. items, that have already been processed going from x_1 to z. Now it is clear that the encoding (E, U) uniquely determines x_1, y_1.

Assuming π is from a family of permutations that are 7-balanced and $O(m^2)$-uniform (which we know exists), the flow through z is

$$\begin{aligned}
f(z) &= \sum_{(E,U)} [\text{amount of flow sent through } z \text{ by pair } (x_1, y_1) \leftrightarrow (E, U)] \\
&= \sum_{(E,U)} \mathbf{Pr}[\pi\{1, \ldots, |U|\} = U] \\
&= \sum_{E} \sum_{U} \mathbf{Pr}[\pi\{1, \ldots, |U|\} = U] \\
&= \sum_{E} \sum_{k} \sum_{U:\ |U|=k} \mathbf{Pr}[\pi\{1, \ldots, |U|\} = U] \\
&\leq \sum_{E} \sum_{k} \sum_{U:\ |U|=k} \gamma \binom{m}{k}^{-1} \\
&= \sum_{E} \sum_{k} \gamma = \sum_{E} n\gamma = n\gamma N = O(n^3)N,
\end{aligned}$$

since $\gamma = O(m^2)$. This shows that $\alpha = O(n^{3+14B})$.

Note that so far we have only claimed that E is an element of the state space, but this has not been justified. It is left as an exercise to check that $a(E) \leq b$, so $E \in G_\Omega$. Intuitively, the reason is that

$$a(E) + a(z) = a(x_1) + a(y_1).$$

Neither $a(x_1)$ nor $a(y_1)$ can be too big, and $a(z)$ cannot be too small since z is not allowed to drop too far below the hyperplane, so $a(E)$ does not get too large, and it stays below the hyperplane. E moves oppositely to z so if z stays within the bounds from below, E stays within the bounds from above. This is why we had to make sure that the path from x_1 to y_1 did not deviate too far *below* the straight line path.

This is an example of a situation that required us to spread out the flow instead of sending it all down one path. There are a number of as yet unanalyzed problems where this kind of combinatorics may help.

References

[MS99] B. Morris and A. Sinclair, "Random walks on truncated cubes and sampling 0-1 knapsack solutions (preliminary version)" *Proceedings of the 40th IEEE FOC (New York)*, 1999, pp. 230–240.

[LS93] L. Lovász and M. Simonovits, "Random walks in a convex body and an improved volume algorithm" *Random Structures Algorithms 4* (1993), no. 4, 359–412.