Introduction

Today we will describe a technique for proving rapidly mixing. It is called coupling and we will exemplify it with an algorithm to uniformly generate spanning trees of a given graph.

1 Basics

Let $G = (V, E)$ be an undirected connected graph with n vertices and m edges. We define the *random walk* on G as the following Markov chain:

- start at some vertex v_0
- from a vertex v in one step choose uniformly (with probability $\frac{1}{d(v)}$) u, one of the neighbors of v, and move to u.

If the graph is not bipartite the Markov chain is irreducible, aperiodic and reversible. So there is a unique stationary distribution Π. In fact Π is given by: $\Pi(v) = \frac{d(v)}{2m}$. Indeed,

$$\Pi(v) = \sum_u \Pi(u)P(u,v) = \sum_{u \in N(v)} \frac{d(u)}{2m}\frac{1}{d(u)} = \frac{d(v)}{2m}$$

where by $N(v)$ we denote the set of neighbors of v.

Note that the reasoning above shows that if G is directed and for any vertex v of G, $indegree(v) = oudegree(v)$, then $\Pi(v) = \frac{d(v)}{2m}$ is the stationary distribution of the appropriately defined random walk on the directed graph G.

We will give some definitions:
h_{ij} = hitting time = E[time to reach j starting at i]
c_{ij} = commute time = E[time to return to i after visiting j starting at i]
Then we have that $c_{ij} = h_{ij} + h_{ji} = c_{ji}$.
C_i = E[time to hit all vertices starting at vertex i]
We define the cover time of a graph $C(G) = \max_i C_i$.
Examples:

1. $G = K_n$. Then for all vertices i, $C_i = C \sim n \log n$. This is the coupon collector problem.
2. G is a path of length n. Then one can compute (I had this as a homework) that $h_{1n} = \Theta(n^2)$ and also $C = \Theta(n^2)$.
3. G is lollipop graph: vertices $1, 2, \ldots n/2$ form a path and vertices $n/2, n/2+1, \ldots n$ form a clique. For this graph we have $C = \Theta(n^3)$.

Theorem : . For any graph G, $C(G) = O(n^3)$.

First we are going to prove the following lemma:

Lemma :

If $[uv] \in E(G)$, $c_{uv} \le 2m$.

Proof :

Let G_1 be the directed graph obtained from G by replacing each edge of G by two arcs going in both directions. Let G' be the directed line graph of the directed G_1. Formally, $V(G') = E(G_1)$ and $E(G') = \{([uv], [vw]),\ \text{where } [uv], [vw] \in E(G_1)\}$

Consider the random walk on the directed graph G'. The transition probabilities of this random walk are given by the matrix Q, where

$$Q_{[uv][vw]} = \frac{1}{d(v)} = P_{vw}$$

Fix an edge $[vw] \in V(G')$. Then:

$$\sum_{[uv]:([uv][vw]) \in E(G')} Q_{[uv][vw]} = \sum_{u \in N(v)} P_{vw} = \sum_{u \in N(v)} \frac{1}{d(v)} = 1$$

It follows that the matrix Q is doubly stochastic. This implies that the stationary distribution of the random walk on G' is uniform. So, for all $[uv] \in V(G')$,

$$\Pi'([uv]) = \frac{1}{|V(G')|} = \frac{1}{2m}.$$

For the proof of the lemma we will use the following fact:
Fact : $h_{ii} = \frac{1}{\Pi(i)}$.

For this fact we will only give intuition/sketch: for a random walk of length t, where t is very large, we expect to be at vertex i a $\Pi(i)$ fraction of the time. So we expect to be at vertex i $t\Pi(i)$ times. Then the expected time spend from one visit of i to the next visit is $t/t\Pi(i) = 1/\Pi(i)$.

We resume the proof of the lemma. So we have that $h'_{[uv][uv]} = 2m$ for all vertices $[uv] \in V(G')$.

Note that the 'original' random walk on G and the random walk on G' are closely related. Being at a vertex of G' corresponds to traversing the corresponding 'oriented' edge of G. So $h'_{[uv][uv]}$ gives the average number of steps from one transversal of the 'oriented' edge $[uv]$ to the next transversal of the 'oriented' edge $[uv]$. In this case then, in the 'original' random walk, a walk from v to u and back to v (the second transversal of the 'oriented' edge $[uv]$) takes place. It follows that $c_{uv} \leq h'_{[uv][uv]} = 2m$.

□

We continue now with the proof of Theorem **??**.

Proof :

Choose a spanning tree T of G and a DFS transversal of T:

$$S = (v_0, v_1, \ldots v_{n-1} = v_0)$$

Then:

$$C(G) \leq \sum_{j=0}^{2n-3} h_{v_j v_{j+1}} = \sum_{(uv) \in T} C_{uv} \leq \sum_{(uv) \in T} 2m = 2m(n-1) = O(n^3)$$

□

2 Uniformly Generating Spanning Trees

In this section G is an undirected graph. Note that finding one spanning tree is easy. Counting the number of spanning trees can also be done in polynomial time since all we have to do is compute the determinant given by the 'matrix-tree' theorem. (By a result due to Jerrum, Valiant and Vazirani which we will cover later in the course it will follow that generating spanning trees from the uniform distribution is also easy (polynomial time) solvable.)

We will be working with rooted spanning tree: choose a vertex as the root and orient all edges inward. To generate a spanning tree simply ignore the root.

Now we describe the Markov Chain $MC(ST)$:

1. Start with any rooted spanning tree ST_0.
2. Choose uniformly an edge (r, v) incident to the root r.
3. Add the directed edge (r, v) to ST (we have exactly one directed cycle).
4. Delete the outgoing edge from v.
5. Reroot so that v is the new root .
6. Repeat steps 2-5.

This algorithm is due to Broder and Aldous.

Facts about MC(ST):

1. MC(ST) is not reversible. In general we cannot go backwards at all.
2. MC(ST) is irreducible. Intuition: start with the leaves.
3. In MC(ST), for any tree T rooted at v, $indegree(T) = outdegree(T) = d(v)$.

Here is an argument for the fact that $indegree(T) = d(v)$. Suppose the children of v in T are $v_1, \dots v_k$. If in the previous tree T' the edge out of v (since v is not a root in T') leads to a vertex in the subtree rooted at v_j, then it must be the case that v_j is the root of T' (to make sure T' is connected) and then we can reconstruct T' using T and the edge out of v in T': remove from T the directed edge (v_j, v) and add this edge out of v. These reconstructed trees T' are all distinct (each has a specific edge out of v), so we have exactly one for each edge incident to v.

Since the indegree equals the outdegree in MC(ST), we have the stationary distribution:

$$\Pi(T) = \frac{d(v)}{Z},$$

where v is the root of T and Z is some normalizing constant. In fact one can check that $Z = 2m \times$ (number of unrooted trees).

Here we assume that the original graph G is not bipartite, so that MC(ST) is aperiodic and therefore the stationary distribution is unique.

For an unrooted tree UT, we get

$$\sum_{T \text{ rooted version of } UT} \Pi(T) = \sum_{v\ vertex} \frac{d(v)}{Z} = \frac{2m}{Z}$$

It follows that for unrooted trees, the stationary distribution is uniform.

3 General idea of coupling

Suppose we have a Markov chain MC on a finite space S. We want to prove that MC is rapidly mixing. We construct another Markov chain MC^2 on $S \times S$, such that the projection of MC^2 on the first coordinate is MC and the projection of MC^2 on the second coordinate is MC. Also MC^2 (by the way we construct it) has the property that once the two coordinates agree, they will always remain in agreement.

We then have to show that, for any starting configuration in $S \times S$, in expected polynomial time the Markov chain MC^2 will be in a configuration of the type (s_t, s_t), that is, the two coordinates agree. Then it will follow that MC is rapidly mixing starting from any configuration s. Indeed, if we start MC^2 from (s, s'), s as in the sentence above and s' given by the stationary distribution of MC, after an expected polynomial number of steps MC^2 will be in a configuration where the two coordinates agree. The projection on the second coordinate of MC^2 will remain in the stationary distribution at all times and therefore when the two coordinates agree the projection on the first coordinate of MC^2, the original Markov chain, will be at the stationary distribution *exactly*.

As an example, consider the following algorithm for shuffling playing cards: choose uniformly a position (in $\{1, \ldots 52\}$) at random and bring the card at that position to the top (position 1) of the (first) deck. We define a coupled Markov chain on the same space: in the second deck, choose and bring to the top the same card (same face, like hearts 8) as in the first deck. Then one can check that the second Markov chain is a copy of the first one: the position to be brought at the top of the deck is chosen uniformly.

The configuration of the two Markov chain agrees at the moment all cards have been chosen - one can easily check that once a card is chosen it will stay at the same position in both stacks. The expected time for agreement is then given by the coupon collector problem and it is $\Theta(n \log n)$.

We return to the general case. Note again that $X_t = Y_t$ implies $X_{t+1} = Y_{t+1}$. where we denoted by (X_t, Y_t) is the configuration of the Markov chain MC^2 at time t. Note also that MC^2 has an 'absorbing set of states' (those where the two coordinates agree) - MC^2 is not irreducible. We define the random variable $t_c = \min\{t : X_t = Y_t\}$. We call $E[t_c]$ the *coupling time.*

We will show a more general results showing the dependence of the mixing time on the coupling time:

Theorem : The mixing time $\Delta_0(\epsilon) \leq \{t : P[X_t \neq Y_t] \leq \epsilon\}$.

Proof :

We will use X and Y omitting the subscript on t. Let $P(s) = P[X = s \;\&\; Y = s]$. Then

$$P[X \neq Y] = 1 - \sum_s P(s) = \frac{1}{2} \sum_s (P[X = s] - P(s)) + \frac{1}{2} \sum_s (P[Y = s] - P(s))$$

Using the fact that $P[X = s] \geq P(s)$ and $P[Y = s] \geq P(s)$, we get:

$$P[X \neq Y] = \frac{1}{2} \sum_s (|P[X = s] - P(s)| + |P(s) - P[Y = s]|)$$

Using $|a - c| + |c - b| \geq |a - b|$, we obtain:

$$P[X \neq Y] \geq \frac{1}{2} \sum_s |P[X = s] - P[Y = s]| = \|X - Y\|$$

From this we conclude $\Delta_0(\epsilon) \leq \{t : P[X_t \neq Y_t] \leq \epsilon\}$. □

4 Back to Generating Spanning Trees

Now we define the coupled Markov chain $MC(ST)'$ by the following rules:

1. Run the two Markov chains independently until the roots agree.

2. From this moment on choose the same edge (r, v) for both MC(ST) and MC(ST)'

The expected time until the two roots agree is bounded by $C(G)$: indeed, each of the two roots performs a random walk on G and by a result not shown in this class (but found in some books like ...) the expected time until the two roots meet is bounded by $C(G)$.

From the moment the two Markov chains have the same root they will always have the same root and one can check that after every vertex is chosen at least once as the root, the two rooted trees will agree. The common root performs a random walk on G and therefore the expected time until it visits all the vertices (each vertex becomes a root) is $C(G)$. Since $C(G) \in O(n^3)$, we conclude that the mixing time is $O(n^3 \log(1/\epsilon))$.