

Lecture 9 (short summary)

Date: Tuesday, 9th February, 2016

Convergence -- standard random walk

In any connected graph, the standard random walk converges (to the stationary distribution) in time roughly

$$O\left(\frac{\log n}{\delta}\right), \text{ where } \delta = 1 - \max_{i \neq n} |\lambda_i|.$$

This follows from the analysis we did for power iteration. The *gap* δ thus controls the convergence time. If the graph is bipartite, $\delta = 0$, so the above does not give any guarantee about convergence to the stationary distribution. In fact, it is easy to see that the walk always *alternates*, thus it does not converge to the stationary distribution. A clean way to get around it is to define a slightly different walk.

Lazy random walks

Again, we have a particle that moves between vertices in G . In every step, it *stays put* with prob α , and it moves to a random neighbor with the remaining probability. We can define the “distribution at time t ” as we did for standard random walks.

Now, we can write

$$p^{(t)} = Zp^{(t-1)},$$

where Z is the transition matrix of the lazy random walk. We saw that

$$Z = \alpha I + (1 - \alpha)AD^{-1}.$$

The eigenvalues of AD^{-1} lie in $[-1, 1]$, and thus those of Z lie in $[-1 + 2\alpha, 1]$. We think of α as a constant in $(0, 1)$, so the smallest eigenvalue is bounded away from -1. The largest eigenvalue is still 1. For any connected graph, the gap δ defined as before is now simply $(1 - \alpha)(\lambda_n - \lambda_{n-1})$. For concreteness, let us think of α as 1/2.

Lazy random walks and sparse cuts

From the above, the convergence time of the lazy walk depends on $\lambda_n - \lambda_{n-1}$, which, when translated to the eigenvalues of the normalized Laplacian, is simply the second smallest eigenvalue -- $\lambda_2(L_G)$.

We saw earlier that Cheeger's inequality connects $\lambda_2(L_G)$ to the *expansion* of the graph -- i.e., $\Phi(G)$, the smallest expansion over all cuts in G .

Cheeger's inequality states that $\Phi(G)^2/4 \leq \lambda_2(L_G) \leq 2\Phi(G)$.

The discussion above says that convergence time for the lazy random walk is roughly $\log n / \lambda_2(L_G)$. Thus if the graph has small expansion -- $\Phi(G)$ is small -- walks take long to converge, by the second part of Cheeger, and if the graph has high expansion, walks converge quickly to the stationary distribution. (We call this phenomenon ***mixing***).

In many situations, we will be interested in showing that **walks mix fast**. The typical way to prove this is by showing that $\Phi(G)$ is large. In fact, there is a name for such graphs:

Expander graph: A graph in which $\Phi(G) \geq \Omega(1)$ is called an expander.

Typically, the challenging step is to prove that a graph of interest is an expander. We see methods to do this in the next class.

Directed graphs

Before completing our discussion on random walks, we discuss briefly walks in directed graphs. Walking in a graph can be done in the same way as before -- a particle starts at some vertex (picked possibly from some start distribution $p^{(0)}$), and takes a random edge *out* of its current location. One issue crops up right way -- if a vertex has out-degree 0, the particle is *stuck*. We can assume there are no such vertices (or add a self loop).

Once we do this, it is possible to define the walk matrix as before:

$$M = A_{out} D_{out}^{-1}.$$

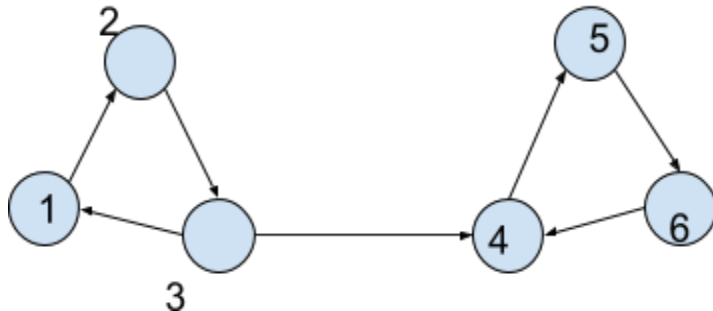
We would like to say that there exists a stationary distribution, and that every walk converges to it.

The first problem we have is similar to that in undirected graphs -- the bipartiteness. Here, this can manifest itself in a more sophisticated way. Suppose we have a directed triangle. Then at step i , we are in state $i \pmod{3}$. We can fix this problem as before -- by making the walk lazy. Thus we consider $Z = \alpha I + (1 - \alpha) A_{out} D_{out}^{-1}$.

Can we now say that (a) there is a unique stationary distribution, and (b) all walks converge to it? It turns out that this is true, and is a consequence of the Perron-Frobenius theorem. This theorem -- a fundamental one in linear algebra -- states that any $n \times n$ matrix with strictly positive entries has a strictly positive largest eigenvalue, and that the corresponding

eigenvector has positive entries. With some caveats, this can be extended to the case when the matrix has some zero entries. See [\[wiki\]](#) or [\[this\]](#).

We still have issues that we didn't in the undirected case. Consider the graph



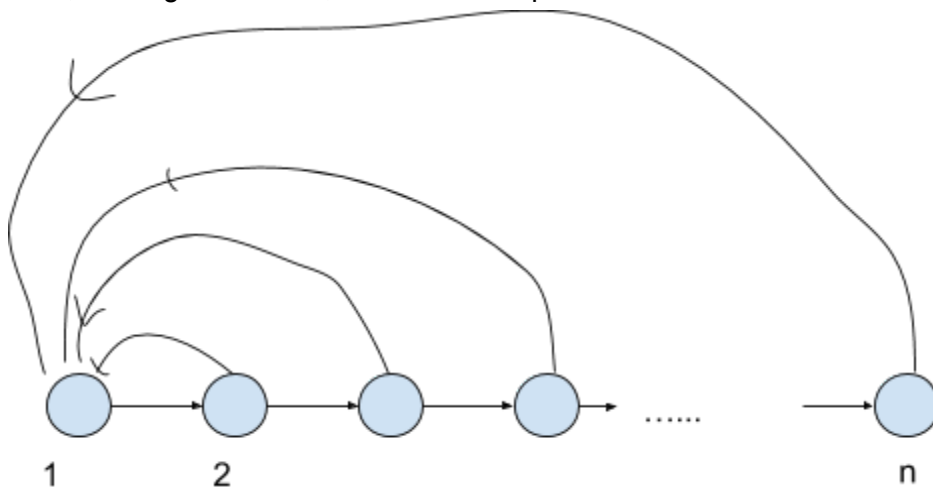
Every vertex has non-zero out-degree. But now, note that if we start *on the left*, we get to the right with some probability, and then we just cycle around (lazily) -- we never come back left! This shows that the stationary distribution, if it exists, must have all its *support* on vertices 4, 5, 6.

This is why it is necessary to start with a **strongly connected** graph G . [[for those unfamiliar, this basically means that for any vertices u, v , there is a path from u to v and vice-versa]].

Convergence time

Recall that in undirected graphs, we could show a lower bound on $\lambda_2(L_G)$ (homework), which implied that a lazy random walk visits every vertex w.h.p. after $\text{poly}(n)$ steps (a simple bound we showed gave n^4). This is not true in directed graphs. Consider the graph

Here, starting at vertex 1, we take 2^n steps to reach vertex n .



Thus, computing the stationary distribution of directed walks by simulation could be very inefficient..

So how do we handle directed graphs? In fact, many graphs of interest are *naturally* directed. The *page-rank* random walk is one that has good convergence properties, and also says something useful about the graph.

We study this briefly in the next class.