Graphs and Networks Lecture 18

PageRank and Random Walks on Directed Graphs

Daniel A. Spielman October 31, 2013

18.1 Overview

In this lecture, we will study random walks on directed graphs. This is the basis of Google's PageRank algorithm [**?**]. I remark that the idea for this algorithm was previously developed by Bonachic [**?**] and that a related algorithm was developed at the same time by Kleinberg [**?**].

We begin this lecture with the general theory of random walks on directed graphs. We conclude by examine some of the ways in which random walks on directed graphs can behave very differently from random walks on undirected graphs, and by explaining some of the advantages of PageRank.

The objective of the PageRank algorithm is to assign a measure of importance or authority to every web page. There is a good story to justify its measure. However, should we believe it? Well, those who remember how well Google performed when it first came on-line will consider this a strong justification. We can view this as a big experiment justifying the performance of PageRank. But, I presume that social scientists also found some way of justifying it before then.

In the next lecture we will see a very different approach to justifying PageRank though axioms. We will see that PageRank can be uniquely characterized as the ranking satisfying some pretty reasonable axioms.

18.2 PageRank

The idea of PageRank is that we want to assign some measure of importance to every web page. We will view links from one page to another as endorsements. So, once we have decided that one page is important, we will believe that each of the pages to which it points are important as well. To limit the influence of any one page, we divide its votes for importance over its out-links.

To state this mathematically, we view the web as a directed graph $G = (V, E)$, where (u, v) is an edge of E if page u has a link to page v. The PageRank vector $\boldsymbol{p}$ should satisfy the requirement that the rank of a page is the sum of the ranks of the pages that point to it, divided by their degrees. That is,

$$\boldsymbol{p}(v) = \sum_{u:(u,v)\in E} \boldsymbol{p}(u)/d_{out}(u).$$

To write this using matrices, we will let A be the adjacency matrix of the directed graph. We define this as follows:

$$A(v, u) = 1 \quad \text{if } (u, v) \in E.$$

Warning: My definition of A is the opposite of the one you would expect . When G has an edge from u to v, I have put a 1 in column u and row v. This is because I am going to make $\boldsymbol{p}$ be a column vector and multiply from the right.

Now, let $d_{out}(u)$ be the number of edges leaving a vertex u and let D_{out} be the diagonal matrix with the out-degrees of nodes on the diagonal. We then set

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

Then, $\boldsymbol{p}$ satisfies

$$\boldsymbol{p} = W_{out}\boldsymbol{p}.$$

So, $\boldsymbol{p}$ is an eigenvector of W_{out} of eigenvalue 1. We have seen this equation before. It tells us that $\boldsymbol{p}$ is a stable distribution for the random walk on G in which we only follow out-links.

But, you may notice a problem: D_{out}^{-1} is not defined if some vertex has zero out-degree. There are two ways to deal with this. The first is to make sure that there are no such vertices in your graph. The second is to ignore them initially, compute the PageRank for every other vertex, and then compute the ranks of those pages.

Now, the actual PageRank proposal is slightly different from this in a very useful way. There is some probability α so that at every step the walk has an α probability of jumping to a uniformly chosen random webpage. They tell us that α is set to some moderately small constant like 0.15. This is equivalent to adding a low-weight edge between every pair of vertices. For now, let's not add those edges and instead view it as changing the equation to

$$\boldsymbol{p} = \alpha\frac{1}{n}\mathbf{1} + (1-\alpha)W_{out}\boldsymbol{p}.$$

As in our lecture on Personal PageRank, we can see that this gives a convenient formula for $\boldsymbol{p}$:

$$\begin{aligned}
\boldsymbol{p} - (1-\alpha)W_{out}\boldsymbol{p} &= \alpha\frac{1}{n}\mathbf{1} \\
(I - (1-\alpha)W_{out})\boldsymbol{p} &= \alpha\frac{1}{n}\mathbf{1} \\
\boldsymbol{p} &= \alpha\frac{1}{n}(I - (1-\alpha)W_{out})^{-1}\mathbf{1}.
\end{aligned}$$

During this lecture we will show that the matrix in that last equation is in fact invertible, and so the vector $\boldsymbol{p}$ is well-defined.

In particular, we will show that all eigenvalues of W_{out} are at most 1 in absolute value. This means that, as in the lecture on Personal PageRank, we can express $\boldsymbol{p}$ as the sum of the infinite series:

$$\boldsymbol{p} = \frac{\alpha}{n}\sum_{t\geq 0}(1-\alpha)^t \boldsymbol{W}_{out}^t\mathbf{1}.$$

For α not too small, the terms $(1-\alpha)^t$ shrink very quickly. Moreover, the sum of the entries in each vector $\boldsymbol{W}_{out}^t\mathbf{1}$ is always the same. So, the terms for large t will have very little contribution to the sum.

18.3 Random Walks on Directed Graphs : Components

We will now study in general directed graphs. At the end of the lecture we will see how the uniform jump probability used in PageRank makes the walks much nicer.

First, let's think about when a random walk is reasonable. For example, what if there is a vertex with no outgoing edges? What would it mean to take a random step from that vertex? You could interpret this to mean that the random walk dies whenever it hits such a vertex. In this case, such vertices will suck the probability out of a walk. It is more convenient to either get rid of such vertices, or to add a self-loop from such a vertex back to itself. In the latter case, probability will accumulate at such a vertex.

If we want to have a unique stable distribution, then we should just get rid of such vertices. If we had two different vertices whose only out-edges were self-loops, then a walk that starts at one of these vertices will stay there. So, we would have at least two different stable distributions.

If we were to eliminate vertices with no out-edges, then we might wind up creating more vertices with no out-edges, and so on. So, we would have to keep eliminating until no vertices with no out-edges remained. Would the stable distribution now be unique?

Not necessarily. A graph with two different strongly-connected components could have more than one stable distribution. Recall that a set of vertices S is strongly-connected if there is a path from each vertex of S to every other vertex of S, using only vertices from S. The set S is a strongly-connected component if it is a maximal strongly-connected set. That means that one cannot add any set of vertices to S and still have a strongly-connected set. Let $S_1, \ldots, S_k$ be the set of strongly connected components of G, and let H be the directed graph on vertices $\{1, \ldots, k\}$ that contains an edge between vertices i and j if there is an edge from a vertex of S_i to an edge of S_j. Recall that the graph H cannot have any cycles (otherwise we could merge the sets in the cycle together to form a larger strongly-connected component). If H has two vertices with no out-edges, then G will have at least two stable distributions. To see this, consider starting the walk at one of the vertices in one of those strongly-connected components. As the component has no out-edges, the walk will never leave.

What about the components corresponding to vertices in H with out-edges? It turns out that these must have zero probability in the limit of a random walk. The reason is clear: probability mass will continue to escape from the out-edge. Moreover, as the component is strongly-connected, all probability mass in the component will eventually pass by a vertex containing an out-edge.

So, the only way we can have a unique stable distribution in which every vertex has non-zero probability is if G is itself one big strongly-connected component. In this case, we say that G is strongly-connected.

18.4 Eigenvalue 1

Lemma 18.4.1. *If G has no vertices of out-degree 0, then 1 is an eigenvalue of $\boldsymbol{W}_{out}$.*

Proof. If G has no vertices of out-degree 0, then every column of $\boldsymbol{A}_{out}$ has at least one non-zero entry. In fact, the uth column of $\boldsymbol{A}_{out}$ has $d_{out}(u)$ non-zero entries, so the uth column of $\boldsymbol{A}_{out}\boldsymbol{D}_{out}^{-1}$ has sum 1. This implies that

$$\mathbf{1}^T \boldsymbol{W}_{out} = \mathbf{1}^T,$$

and so $\boldsymbol{W}_{out}$ has an eigenvector of eigenvalue 1. □

This is one place where we must be careful. We now know that $\mathbf{1}^T$ is a left-eigenvector of $\boldsymbol{W}_{out}$. As the set of left-eigenvalues and right-eigenvalues of a matrix are the same, we know that $\boldsymbol{W}_{out}$ also has a right-eigenvector of eigenvalue 1. This is the vector $\boldsymbol{p}$ that we are looking for. However, we do not know any nice, simple formula for $\boldsymbol{p}$!

Lemma 18.4.2. *If G is strongly connected, then the eigenvalue 1 has multiplicity 1. In particular, if*

$$\boldsymbol{v}^T \boldsymbol{W}_{out} = \boldsymbol{v}^T,$$

then we must have $\boldsymbol{v}^T = c\mathbf{1}$ for some constant c.

The proof of this is similar to the proof in the undirected case, so we will skip it. I might put it on a problem set.

Just as in the undirected case, the uniqueness of the eigenvector of eigenvalue 1 does not imply that a random walk will necessarily converge to this eigenvalue. However, there can be fancier obstructions in the directed case. Consider a directed cycle on k vertices. If we start the random walk at one of these vertices, it will just keep jumping around the cycle. So, it will go through the same configuration every k steps. Spectrally speaking, this is because such a graph has an eigenvalue that is a k-th root of 1:

$$e^{2\pi i/k}.$$

Of course, we could eliminate this issue by adding self-loops to every vertex (perhaps with small weight), or edges from each vertex to every other vertex, as in PageRank.

I should also mention the following theorem, which we can also prove as we did in the undirected case.

Theorem 18.4.3. *Every eigenvalue of $\boldsymbol{W}_{out}$ has absolute value at most 1.*

This could also show up in a problem set.

18.5 A Positive Stable Distribution

We will now prove that a strongly connected graph has a stable distribution by showing that the vector $\boldsymbol{p}$ for which

$$\boldsymbol{p} = \boldsymbol{W}_{out}\boldsymbol{p} \tag{18.1}$$

must be all positive or all negative. We begin by showing that a solution that is all non-negative must be all positive.

To do this, we will consider the matrix

$$\boldsymbol{W}^*_{out} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=0}^{n-1} \boldsymbol{W}^i_{out}.$$

We need to establish a few properties of $\boldsymbol{W}^*_{out}$.

Claim 18.5.1. *If $\boldsymbol{W}_{out}\boldsymbol{p} = \boldsymbol{p}$, then $\boldsymbol{W}^*_{out}\boldsymbol{p} = \boldsymbol{p}$. Similarly, $\mathbf{1}^T\boldsymbol{W}^*_{out} = \mathbf{1}^T$.*

Claim 18.5.2. *The matrix $\boldsymbol{W}^*_{out}$ has no negative or zero entries.*

Proof. As $\boldsymbol{W}_{out}$ is non-negative, it follows immediately that $\boldsymbol{W}^*_{out}$ is non-negative. To show that $\boldsymbol{W}^*_{out}$ has no zero entries, note that $\boldsymbol{W}^t_{out}(b,a)$ is equal to the probability that a random walk starting at a hits b in exactly t time steps. As the graph is strongly connected, for every pair of vertices a and b, there is some t less than n for which this probability is non-zero. As $\boldsymbol{W}^*_{out}(b,a)$ is the average of these probabilities for t between 0 and n, it is non-zero as well. □

Lemma 18.5.3. *If $\boldsymbol{p}$ satisfies* (18.1) *and $\boldsymbol{p}$ is non-negative, then $\boldsymbol{p}$ is strictly positive.*

Proof. Assuming that $\boldsymbol{p}$ is not the all-zero vector, it has some positive entry. Assume, without loss of generality, that $\boldsymbol{p}(1) > 0$. As both $\boldsymbol{p}$ and $\boldsymbol{W}^*_{out}$ are both non-negative,

$$\boldsymbol{p}(j) = \sum_i \boldsymbol{W}^*_{out}(j,i)\boldsymbol{p}(i) \geq \boldsymbol{W}^*_{out}(j,1)\boldsymbol{p}(1) > 0.$$

□

We now prove that (18.1) has a non-negative solution.

Theorem 18.5.4. *The equation $\boldsymbol{W}_{out}\boldsymbol{p} = \boldsymbol{p}$ has a non-negative solution.*

Proof. We will show that it has a solution in which all the signs are the same, which implies that it has a non-negative solution (flip all signs if necessary). Assume by way of contradiction that $\boldsymbol{p}$ is not sign-uniform. That is, that $\boldsymbol{p}$ has both positive and negative entries. We will use the fact that if $\boldsymbol{x}$ is some vector with both positive and negative entries, then

$$\left|\sum_u \boldsymbol{x}(u)\right| < \sum_u \left|\boldsymbol{x}(u)\right|.$$

From equation (18.1), we have that for all u,

$$\boldsymbol{p}(u) = \sum_v \boldsymbol{W}^*_{out}(u,v)\boldsymbol{p}(v),$$

and so

$$\left|\boldsymbol{p}(u)\right| = \left|\sum_v \boldsymbol{W}^*_{out}(u,v)\boldsymbol{p}(v)\right|.$$

As we have assumed that $\boldsymbol{p}$ is not sign-uniform, and $\boldsymbol{W}^*_{out}(u,v)$ is always positive, we have the inequality

$$\left|\sum_v \boldsymbol{W}^*_{out}(u,v)\boldsymbol{p}(v)\right| < \sum_v \boldsymbol{W}^*_{out}(u,v)\left|\boldsymbol{p}(v)\right|,$$

which implies

$$|\boldsymbol{p}(u)| < \sum_v \boldsymbol{W}^*_{out}(u,v)\left|\boldsymbol{p}(v)\right|.$$

If we now sum over all u, we get

$$\begin{aligned}\sum_u |\boldsymbol{p}(u)| &< \sum_u \sum_v \boldsymbol{W}^*_{out}(u,v)\left|\boldsymbol{p}(v)\right| \\ &= \sum_v \sum_u \boldsymbol{W}^*_{out}(u,v)\left|\boldsymbol{p}(v)\right| \\ &= \sum_v |\boldsymbol{p}(v)| \sum_u \boldsymbol{W}^*_{out}(u,v) \\ &= \sum_v \left|\boldsymbol{p}(v)\right|,\end{aligned}$$

as $\mathbf{1}^T \boldsymbol{W}^*_{out} = \mathbf{1}^T$ is equivalent to

$$\sum_u \boldsymbol{W}^*_{out}(u,v) = 1.$$

From the assumption that $\boldsymbol{p}$ is not sign-uniform, we have derived a contradiction. □

18.6 Warning about directed graphs

There are few important ways in which random walks on directed graphs differ from random walks on undirected graphs. The first is that the spectral theory is different. An asymmetric matrix like $\boldsymbol{W}_{out}$ might not be diagonalizable. That is, it might not have n eigenvalues. So, spectral-type analyses need to go through the Jordan normal form. Spectral-type analyses are also complicated by the fact that the eigenvectors of such a matrix are usually not orthogonal.

The next warning is that the probabilities of vertices under the stable distribution can vary by many orders of magnitude. For example, consider the graph on n vertices $\{1,\ldots,n\}$ with an edge from vertex i to vertex $i+1$ for every $1 \le i < n$, as well as an edge from every vertex $i \ge 2$ pointing back to vertex 1. If we start a random walk at vertex 1, it is very unlikely to reach vertex n. At every node i, the chance that it makes it to node $i+1$ is only $1/2$. Moreover, with probability $1/2$ is jumps all the way back to node 1. From this argument, we can see that the probability of being at node 1 is much much larger than the probability of being at node n.

This phenomenon can also be used to produce graphs in which the random walk is very slow to converge to the stable distribution. To see this, let G_1 and G_2 be two different copies of the graph that we just described. For each, call the vertex corresponding to vertex 1 the *first* node and call the vertex corresponding to vertex n to *last* vertex. Now, add an edge from the last vertex of G_1 to the first vertex of G_2 and an edge from the last vertex of G_2 to the first vertex of G_1. This is a

symmetrical construction. So, in the stable distribution the probabilities of corresponding nodes in graphs G_1 and G_2 must be the same. But, if we start a random walk at the first node of G_1, it will take time exponential in n before the probability mass between the two parts begins to equalize. To see this, note that the only mass that enters G_2 does so through the last node of G_1. But, the probability that the random walk even reaches the last node of G_1 in the first 2^{n-3} steps is at most $1/4$.

Research Question: Can you find a fast algorithm for approximating the stable distribution of the random walk on an arbitrary graph?

18.7 What PageRank Does

By adding in an α probability of jumping to a uniform random vertex, PageRank avoids many of the problems of random walks on directed graphs. First, it ensures that the walk graph is strongly connected. Second, it ensures that the probability of reaching every vertex is at least α/n, and so is not too small.

Third, as we explained at the start, it provides a fast algorithm for approximating the PageRank vector.