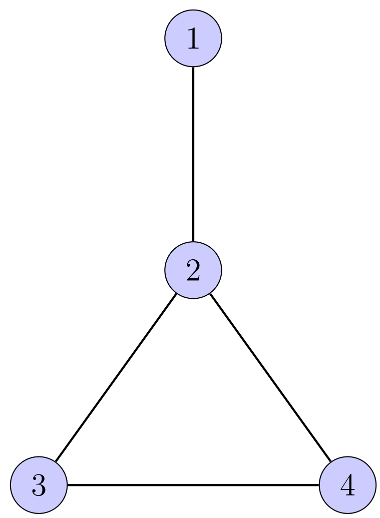
Lecture 22

Previously we saw that we could encode a directed graph (or digraph) in an incidence matrix relating edges to vertices.

In this lecture we'll look at an alternate method to encode a graph called the *adjacency matrix*. We'll start with *undirected graphs* (that is, graphs where all edges are bidirectional) and then look at directed graphs again.

Consider the following undirected graph, G,



Def: The adjacency matrix A of an undirected graph G, is a square $n \times n$ matrix where n is the number of vertices in the graph. The entries of A are given by

$$A_{ij} = \begin{cases} 1 & \text{if } i \leftrightarrow j \\ 0 & \text{otherwise} \end{cases}$$

This says that the (i, j)-entry of the adjacency matrix is 1 is there is an edge connecting vertices i and j and 0 if there is not.

Example 1: For the graph G pictured above, the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Fact 1: The adjacency matrix A of an undirected graph is always symmetric.

Def: The **degree** of a vertex is the number of vertices that it is connected to (or alternatively, the number of edges that it is a part of). We denote the degree of vertex i by d_i.

Example 2: For the graph G pictured above the vertex degrees are

$$d_1 = 1 \qquad d_2 = 3 \qquad d_3 = 2 \qquad d_4 = 2$$

Fact 2: The degree of vertex i is equal to the sum of row i.

Def: A walk is a list of vertices that can be traversed along edges in the graph. Note that there is no restriction on the number of times a vertex is touched during a walk.

Example 3. Denote the i^{th} vertex by v_i. Then the walk the walk that goes from vertex 1 to vertex 2 to vertex 4 is denoted by $v_1 - v_2 - v_4$. The walk that goes from vertex 1 to vertex 2 to vertex 3 and then back to vertex 2 is denoted by $v_1 - v_2 - v_3 - v_2$.

Fact 3: The (i, j)-entry of the matrix power A^k tell you how many walks of length k exist in the graph that start at vertex i and end at vertex j.

Example 4: We'll look at powers of the adjacency matrix corresponding to the the graph G pictured above. For the first power of A we have

$$A^1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

This is of course just the adjacency matrix of G itself. According to Fact 3, since $A_{12} = 1$ there is exactly one walk that starts at v_1 and ends at v_2. This is of course true because there is an edge between v_1 and v_2. Similarly, $A_{13} = 0$, indicating that there are no length-1 walks between v_1 and v_3.

Squaring the matrix A, we have

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

According to Fact 3, since $A_{14} = 1$ there is exactly one length-2 walk that starts at v_1 and ends at v_4. From the graph we see that this is the walk $v_1 - v_2 - v_4$. Notice that it is impossible to traverse a different sequences of vertices to get from v_1 to v_4. Similarly, since $A_{22} = 3$ there are exactly three length-2 walks that start at v_2 and end at v_2. These are $v_2 - v_1 - v_2$, $v_2 - v_3 - v_2$, and $v_2 - v_4 - v_2$.

Finally, cubing A we have

$$A^3 = \begin{bmatrix} 0 & 3 & 1 & 1 \\ 3 & 2 & 4 & 4 \\ 1 & 4 & 2 & 3 \\ 1 & 4 & 3 & 2 \end{bmatrix}$$

Since $A_{42} = 4$ there are exactly four length-3 paths that start at v_4 and end at v_2. Looking at the graph we see that these walks are $v_4 - v_2 - v_1 - v_2$, $v_4 - v_2 - v_3 - v_2$, $v_4 - v_2 - v_4 - v_2$, and $v_4 - v_3 - v_4 - v_2$.

Def: A graph is called *connected* if there exists a walk from v_i to v_j for all pairs i and j.

Fact 4: A graph G is connected if there exists some integer $k > 0$ such that $B_k = I + A + A^2 + \cdots + A^k$ has all positive entries.

Essentially this works because the entries of the powers of A give the number of walks of length k that connect each of the vertices. If B_k has all positive entries, then there exists a walk of at most length k between any two vertices.

Example 5: For the adjacency matrix for the example graph, we have

$$B_1 = I + A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

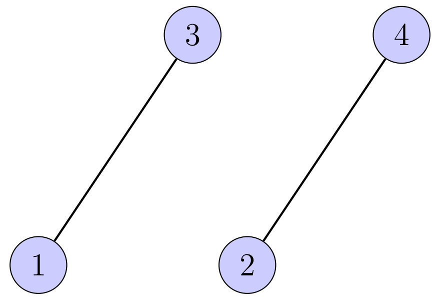
Since there are zero entries in B_1 it is not the case that you can get from every vertex to every other in walks of length 1 (for instance, it's impossible to get from v_1 to v_3 in one step).

We then have

$$B_2 = I + A + A^2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 4 & 2 & 2 \\ 1 & 2 & 3 & 2 \\ 1 & 2 & 2 & 3 \end{bmatrix}$$

Since B_2 has all positive entries we know that G is connected, and you can get from any vertex to any other in a walk of length 2 or less. This is clearly verified by looking at the graph.

Example 6: Consider the following disconnected graph



The associated adjacency matrix is $A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$

Taking powers of A we have

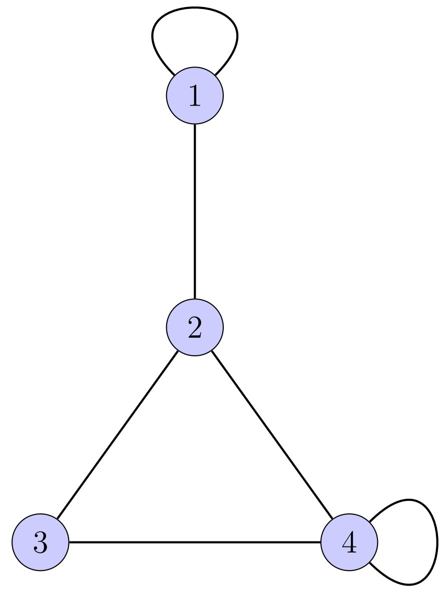
$$A^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A^3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad A^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Clearly this pattern repeats. Thus for any integer k the matrix B_k has the form

$$B_k = I + A + A^2 + \cdots + A^k = \begin{bmatrix} * & 0 & * & 0 \\ 0 & * & 0 & * \\ * & 0 & * & 0 \\ 0 & * & 0 & * \end{bmatrix}$$

Since B_k will not have all positive entries for any value of k the graph is not connected.

Example 7: Vertices can also have self-loops. Consider the modification of the graph in Example 1 where we add self-loops to vertices 1 and 4.

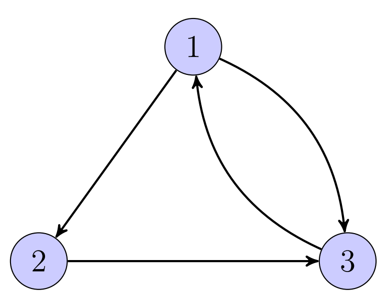


The adjacency matrix for the graph with self-loops is $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

Directed Graphs and The Google Matrix

Consider a directed graph where the vertices represent pages on the internet. Directed edges in the graph represent hyperlinks from one page to another. That is, if there a link from page 1 to page 2 then there is a directed edge in the graph from v_1 to v_2.

Consider the following directed graph



The adjacency matrix for a directed graph has a 1 in the (i,j)-position if there is a directed edge that starts at v_i and ends at v_j:

$$A_{ij} = \begin{cases} 1 & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$$

Example 8: The adjacency matrix for the graph above is $A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

For directed graphs we can talk about the *incoming degree* or *outgoing degree* of a vertex. The incoming degree of a vertex is the number of edges that terminate at the vertex. Similarly the outgoing degree of a vertex is the number of edges that start at the vertex. For our purposes, we'll only care about the outgoing degree.

The outgoing degrees of the three vertices in the example graph are given by

$$d_1 = 2 \qquad d_2 = 1 \qquad d_3 = 1$$

Note that the outgoing degree of a vertex is equal to the sum of the associated row in the adjacency matrix.

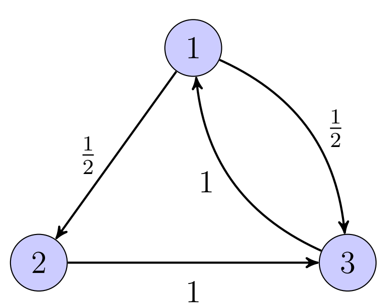
Def: The transition matrix of a directed graph T is found by transposing the adjacency matrix and dividing each column by the outgoing degree of the associated vertex. Mathematically, we have

$$T_{ij} = \frac{A_{ji}}{d_j}$$

For the example graph, we have

$$T = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix}$$

The transition matrix gives us an interesting way of interpreting the graph. Consider dropping a *random walker* on a vertex of the graph. The walker then moves along a directed edge to another vertex with a probability evenly distributed between each outgoing edges. For the example graph the probabilities look as follows



Let's see what happens when we drop a random walker at vertex 1 and then let it go.

Step 0: at v_1 with prob $p_1 = 1$

Step 1: at v_2 with prob $p_2 = 1/2$, at v_3 with prob $p_3 = 1/2$

Step 2: at v_1 with prob $p_1 = 1/2$, at v_3 with prob $p_3 = 1/2$

Step 3: at v_1 with prob $p_1 = 1/2$, at v_2 with prob $p_2 = 1/4$, at v_3 with prob $p_3 = 1/4$

The transition matrix T gives us a way to determine the probabilities of the walker using linear algebra. Since we begin with the walker at vertex 1 with probability 1 we denote the state of the system by the vector

$$\mathbf{x}^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

To move the walker one step we multiply the state vector by T. We have

$$\mathbf{x}^{(1)} = T\mathbf{x}^{(0)} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \end{bmatrix}$$

Note that the state vector after step one corresponds to the probabilities we deduced above for the random walker. To get the state of the system after two steps we multiply again by the transition matrix.

$$\mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix}$$

which again agrees with the previously computed probabilities. Taking one more step, we have

$$\mathbf{x}^{(3)} = T\mathbf{x}^{(2)} = \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/4 \\ 1/4 \end{bmatrix}$$

Now suppose that we let the random walker go for a large number of steps until the state vector no longer changes from step to step. The resulting vector is as follows

$$\mathbf{x} = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.4 \end{bmatrix}$$

This says that after a large number of time there is a 40% chance that the walker is at vertex 1, a 20% chance that it's at vertex 2, and a 40% chance that it's at vertex 3.

It's easy to check that the given vector is an eigenvector of the transition matrix T with associated eigenvalue $\lambda = 1$.

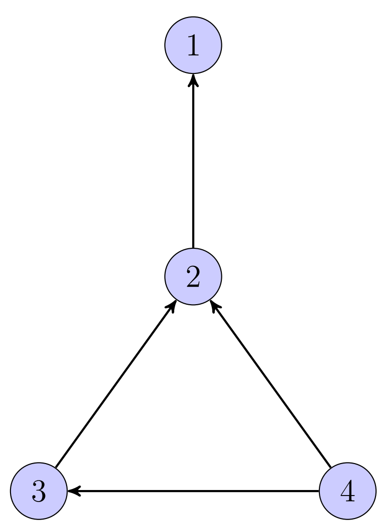
$$\begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.2 \\ 0.4 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.4 \end{bmatrix}$$

Note that any nonzero scalar multiple of this vector would also be an eigenvector with associated eigenvalue 1. When we interpret the vector as a probability distribution we always scale the vector so that it has all nonnegative entries that sum to 1.

If we return to the analogy that the graph represents a network of webpages, then the eigenvector of interest indicates the long term behavior of a web surfer that is randomly clicking on hyperlinks. The pages that have high probabilities are deemed more important than the others, and would therefore be displayed higher up in the list by Google. The eigenvector is called the **PageRank** vector and the i^{th} entry in the vector is called the PageRank of page i.

Note that this was a simple case where the transition matrix T was a stochastic matrix (all nonnegative entries with column sums of 1). But this isn't always the case for every directed graph.

Consider the following digraph



The associated adjacency matrix is $A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

The outgoing degrees of the vertices are

$$d_1 = 0 \qquad d_2 = 1 \qquad d_3 = 1 \qquad d_4 = 2$$

Ignoring the zero outgoing degree of vertex 1, the transition matrix T is given by

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Suppose we start with a random walker at vertex 4. Then the first several state vectors are

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{x}^{(1)} = T\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix} \quad \mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \\ 0 \end{bmatrix}$$

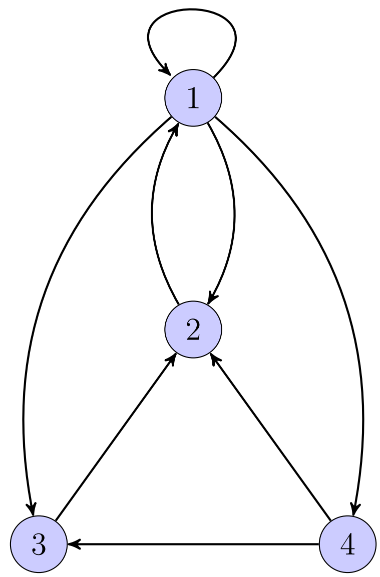
$$\mathbf{x}^{(3)} = T\mathbf{x}^{(2)} = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x}^{(4)} = T\mathbf{x}^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

We immediately see a problem at Step 3. The resulting state vector is no longer a probability distribution because the entries of the vector don't sum to 1.

The problem is that vertex 1 does not have any outgoing nodes, so the probability of the random walker doesn't have anywhere to go and sort of falls off the edge. We call vertex 1 a **dangling vertex** or **dangling node**.

Note also that the transition matrix T for the graph with the dangling vertex is not a stochastic matrix. The column associated with the dangling vertex is all zeros and thus does not sum to 1.

OK, so how do we fix this? Google's solution is to add a directed edge from the dangling vertex to every other vertex in the graph (including itself). The web analogy is that when a random surfer reaches a dangling node, it jumps randomly to any page in the web with equal probability. After fixing the dangling node, the directed graph looks as follows



The associated transition matrix for the new graph is as follows:

$$T = \begin{bmatrix} 1/4 & 1 & 0 & 0 \\ 1/4 & 0 & 1 & 1/2 \\ 1/4 & 0 & 0 & 1/2 \\ 1/4 & 0 & 0 & 0 \end{bmatrix}$$

Notice that the new transition matrix is in fact stochastic. It's PageRank vector is

$$\mathbf{x} = \begin{bmatrix} 0.42 \\ 0.32 \\ 0.15 \\ 0.10 \end{bmatrix}$$

OK, so the PageRank vector is the eigenvector associated with eigenvalue $\lambda = 1$.

How do we know that a stochastic matrix T always has an eigenvalue $\lambda = 1$?

How that there isn't an eigenvalue bigger than 1?

Are there multiple eigenvectors with associated eigenvalue $\lambda = 1$?

Fact 5: If T is stochastic then $\lambda = 1$ is an eigenvalue.

Before we can prove this we need to prove an important lemma about eigenvalues.

Lemma: If λ is an eigenvalue of A then it is also an eigenvalue of A^T.

Proof: Recall that the determinant of a matrix and its transpose are the same. We then have

$$\det(A - \lambda I) = \det(A - \lambda I)^T = \det\left(A^T - \lambda I\right)$$

This means that A and A^T have the same characteristic polynomial. Since the eigenvalues of a matrix are the roots of its characteristic polynomial it must be the case that the eigenvalues of A and A^T are the same.

Caveat: Even though the eigenvalues of A and A^T are the same it is usually **NOT** the case that their associated eigenvectors are the same.

OK, now we're ready to prove Fact 5.

Proof: Let $\mathbf{1}$ be the vector of all 1's. Since the columns of T sum to 1 we have

$$T^T\mathbf{1} = \mathbf{1} \quad \Rightarrow \quad \lambda = 1 \text{ is an eigenvalue of } T \text{ and } T^T$$

Fact 6: If T is stochastic then it has no eigenvalue λ such that $|\lambda| > 1$.

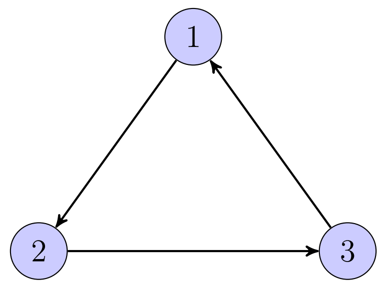
Proof: Recall that if T is stochastic $\mathbf{x}$ is such that it's entries are nonnegative and sum to 1 then the entries of $T\mathbf{x}$ also sum to 1. Suppose that $\mathbf{x}$ is an eigenvector with associated eigenvalue $|\lambda| > 1$. Then

$$T^n\mathbf{x} = \lambda^n\mathbf{x}$$

For large n it must be the case that some entry of $\lambda^n\mathbf{x}$ is larger than 1. But this can't be the case since all we've done is repeatedly multiply a vector with sum 1 by a transition matrix, so $\lambda^n\mathbf{x}$ must also have entries that sum to 1. This is a contradiction, so it must be the case that $|\lambda| \leq 1$ for all λ.

The last question we posed was whether T could have multiple eigenvectors associated with $\lambda = 1$. This is an important question if we are to interpret the dominant eigenvector as the PageRank vector. It would be weird if there were multiple PageRank vectors associated with a web.

It turns out that there are a couple of cases that can lead to non-unique dominant eigenvectors. The first is the case when the graph has multiple disconnected components. We've looked at this case before, so we won't beat it to death again. The new case is the following:



which has transition matrix $T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$.

If we drop a random walker at vertex 1 then we have $\mathbf{x}^{(0)} = \mathbf{e}_1$. Then

$$\mathbf{x}^{(1)} = T\mathbf{x}^{(0)} = \mathbf{e}_2 \qquad \mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = \mathbf{e}_3 \qquad \mathbf{x}^{(3)} = T\mathbf{x}^{(2)} = \mathbf{e}_1$$

It's clear that if we let the walker run for a long time, it will never settle down into a stationary state. It will instead skip between being located at each of the vertices with probability 1. When this happens we say that the graph is *periodic*.

It turns out that the reason this happens is because T has three distinct eigenvalues each with magnitude equal to 1. They are

$$\lambda_1 = 1 \qquad \lambda_{2,3} = 0.5 \pm \frac{\sqrt{3}}{2}i$$

OK, so what do we do to avoid these special cases? Google's solution is to **first fix dangling nodes** to obtain a stochastic matrix T and then add outgoing links from each page to every other page in the web. This can be justified intuitively by the notion that occasionally web surfers opt to type a new URL into the browser rather than clicking on a hyperlink. We need to make this modification to the transition matrix in such a way that we still obtained a column-stochastic matrix. We define the new matrix

$$G = (1-p)\,T + p\frac{1}{n}\mathbf{1}\mathbf{1}^T$$

where $\mathbf{1}$ is n-dimensional vector of all 1's. Notice that G is a column-stochastic matrix as well. The parameter p is a tuning parameter that sets the probability that the surfer makes a random jump. The tuning parameter p is usually chosen to be $p = 0.15$.