Not Kevin Bacon

Suppose we have a graph or network with n nodes and some connections between them. The matrix $\boldsymbol{A}$ whose entries are

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

is the **adjacency matrix** of the network. Note that these connections have direction, and i connecting to j does not automatically imply that j connects to i (i.e., this is a directed graph). Let $s_i = \sum_{j=1}^{n} A_{ij}$ be the number of nodes that node i connects to. If we were to randomly select a link leaving node i, each link would have probability $1/s_i$ of being selected.

Let $\boldsymbol{x}$ be a vector of positive values. We will require that $\sum_i x_i = 1$, so that $\boldsymbol{x}$ has the interpretation of a probability distribution over the nodes. If all connections are given equal weighting, the probability of following a connection from any node to node i is

$$z_i = \sum_{j \in P_i} \frac{x_j}{s_j}, \tag{2}$$

where P_i is the set of nodes that connect to node i. These are the rows with ones in column i of $\boldsymbol{A}$. By the definition of $\boldsymbol{A}$, this is the same as

$$z_i = \sum_{j=1}^{n} \frac{A_{ji} x_j}{s_j} = \sum_{j=1}^{n} B_{ij} x_j, \tag{3}$$

where we defined $B_{ij} = A_{ji}/s_j$. Put simply, $\boldsymbol{z} = \boldsymbol{B}\boldsymbol{x}$.

It's important to introduce some overall randomness into the jumps between nodes—this is the only way to escape a self-contained clique (disconnected subgraph). The probability of hopping to any node i entirely at random is just $1/n$. We blend link-following with random hopping as follows. Choose some $p \in [0, 1]$, and suppose that a hop between nodes follows one of the connections with probability p, or is a random hop with probability $1 - p$. Using $\mathbf{1}$ to denote the n-vector of all ones, then

$$\boldsymbol{y} = p\boldsymbol{B}\boldsymbol{x} + \frac{1-p}{n}\mathbf{1} \tag{4}$$

describes how to update probabilities after each hop. Finally, the fact that

$$1 = \sum_i x_i = \mathbf{1}^T \boldsymbol{x}, \tag{5}$$

allows us to express the map (4) as

$$\boldsymbol{y} = \left[p\boldsymbol{B} + \frac{1-p}{n}\boldsymbol{1}\boldsymbol{1}^T \right] \boldsymbol{x} = \boldsymbol{R}\boldsymbol{x}, \tag{6}$$

for a square matrix $\boldsymbol{R}$. If the probabilities are unchanged by hopping (i.e., $\boldsymbol{z} = \boldsymbol{x}$), then $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{R}$ with associated eigenvalue $\lambda = 1$. We won't prove this, but $\boldsymbol{R}$ is guaranteed to have $\lambda = 1$ as the leading eigenvalue, making power iteration possible. The resulting eigenvector $\boldsymbol{x}$ can be sorted to find out which nodes are most likely to be visited in the long run.

Note that $\boldsymbol{R}$ is *not* sparse and should never be formed. However, $\boldsymbol{R}\boldsymbol{x}$ as defined in (4) can be computed efficiently if $\boldsymbol{B}$ is sparse. That is all we need to do a power iteration with $\boldsymbol{R}$. Since we are working with probability, normalization is not required in the power iteration: $\boldsymbol{x}_{k+1} = \boldsymbol{R}\boldsymbol{x}_k$, provided $\boldsymbol{x}_1$ has positive entries and $\|\boldsymbol{x}\|_1 = 1$.

Goals

You will use an adjacency matrix for movie actors to perform the power iteration and find the leading eigenvector of $\boldsymbol{R}$, and using that vector to rank the actors in influence.

Preparation

Read section 8.2. Answer the following questions based on the above description.

1. Show using (3) that $\displaystyle\sum_{i=1}^{n} z_i = 1$. This proves that $\boldsymbol{z}$ is also a probability distribution.
2. Show using (4) that $\boldsymbol{y}$ is a probability distribution.

Procedure

Download the script template and the data file `actornetwork.mat`.

1. Load `actornetwork.mat` file from the assignment site. It has a vector `actor` of unique actor names for all credited roles in films released from 2004 through 2013. It also has a sparse adjacency matrix `A`, where a (symmetric) link between actors means that they appeared in at least one film together.

2. Use `nnz` to compute the density (number of nonzeros over total number of elements) of `A`. Use `whos` to find the memory usage of `A` in bytes. Also calculate the memory usage of an equivalent full (non-sparse) matrix.

3. Compute the vector $\mathbf{s}$ whose entries are s_i as defined above. Make a histogram of its entries using 32 bins.

4. Construct the matrix $\boldsymbol{B}$ appearing in (3) above. It helps to use the fact that $\boldsymbol{A}$ is symmetric. (It's reasonable to loop over one dimension of the matrix, but not over all of the elements.)

5. Set $p = 0.9$ and let $\boldsymbol{x}_1$ be a random vector of positive numbers. Normalize $\boldsymbol{x}_1$ to be a probability distribution. By repeatedly applying (4), do 100 power iterations to get $\boldsymbol{x}_{101}$. *Important: Do not attempt to define the matrix* $\boldsymbol{R}$, and do *not* use the book's power iteration function; instead use (**??**) to compute $\boldsymbol{Rx}$. Check that $\|\boldsymbol{x}_{101} - \boldsymbol{x}_{100}\|_1$ is less than 10^{-6}.

6. Sort the entries of $\boldsymbol{x}$ in descending order, using the second output to print out the names of the 10 "most collaborative" actors.

Discussion

1. The data file also includes an $m \times n$ sparse matrix $\boldsymbol{M}$. Its (i, j) entry is one if actor j appeared in film i, and zero otherwise. Give a simple interpretation of the matrix $\boldsymbol{M}^T\boldsymbol{M}$.

2. What is the interpretation of $\boldsymbol{M}\boldsymbol{M}^T$? What would a ranking using this matrix reveal?