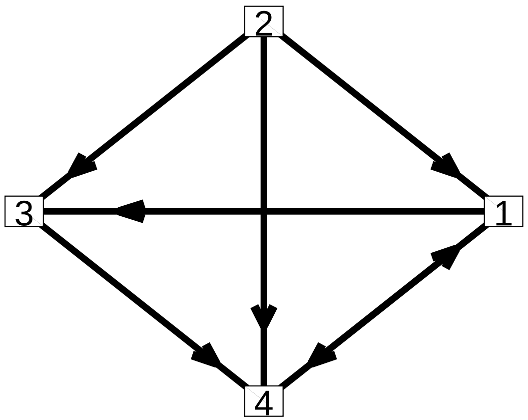
The google PageRank algorithm, session 2



Adjacency matrix.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

The **out-degrees** are the column sums and these are 2, 3, 1 and 1.

Out-degree and In-degree

Let `A` be the adjacency matrix in a Matlab program.

All the entries in column `j` are given by `A(:, j)`.

All the entries in row `i` are given by `A(i, :)`.

The **out-degree** of node j is the number of nodes that you can go to directly from node j. It is the sum of the entries in column j.

In Matlab all the out-degrees are generated by the statement

```
outdeg=sum(A)
```

The **in-degree** of node i is the number of nodes that go to directly to node i. It is the sum of the entries in row i.

In Matlab all the in-degrees are generated by the statement

```
indeg=sum(A')
```

The probability matrix C

Adjacency matrix.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

The out-degrees are the column sums and these are 2, 3, 1 and 1.
The probability matrix C is

$$C = \begin{pmatrix} 0 & 1/3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1/2 & 1/3 & 0 & 0 \\ 1/2 & 1/3 & 1 & 0 \end{pmatrix}$$

The entries in each column have been divided by the out-degree associated with the column. With this matrix this is okay as all the out-degrees are greater than 0.

A function file to get C given A

A version of a function m-file to get the probability matrix C from a valid adjacency matrix A can be as follows.

```
function C = cmat1_no_checks(A)
% function C = cmat1_no_checks(A) determines the
% probability matrix C from an adjacency matrix A

% get the dimensions, it needs to be a square matrix
n=size(A, 1);

% get the out degrees which are the column sums
outdeg=sum(A);

% set C column-by-column
C = zeros(n, n);
for from=1:n
  C(:, from) = A(:, from)/outdeg(from);
end
```

Probability vector at each step

Let $\underline{p}^{(k)}$ be a 4×1 vector with each entry giving the probability to be at that node at step k. If we start at node 2 then

$$\underline{p}^{(0)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Assuming that we can move to each of the other nodes with equal probability we get for the next step that

$$\underline{p}^{(1)} = \begin{pmatrix} 1/3 \\ 0 \\ 1/3 \\ 1/3 \end{pmatrix}.$$

How do we get $\underline{p}^{(2)}$, $\underline{p}^{(3)}$, ... in a systematic way with the computer generating the vectors?

Getting $\underline{p}^{(2)}$

To get to node 1 we must have to have previously been at node 2 or 4. At this stage this means we must have previously been at node 4. From the start the route was $2 \to 4 \to 1$ with probability

$$\left(\frac{1}{3}\right) 1 = \frac{1}{3}.$$

There is no route to node 2.

The only route to node 3 at this stage is $2 \to 1 \to 3$ and the probability of this is

$$\left(\frac{1}{3}\right)\left(\frac{1}{2}\right) = \frac{1}{6}.$$

There are 2 routes to node 4 and correspond to $2 \to 1 \to 4$ and $2 \to 3 \to 4$. Overall the probability is

$$\left(\frac{1}{3}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{3}\right) 1 = \frac{1}{2}.$$

Generalise

If we forget the specific numbers are just consider all possibilities then the probability to be at node i given what the probabilities were at the previous stage is

$$\Pr(i|1)(\underline{p}^{(1)})_1 + \Pr(i|2)(\underline{p}^{(1)})_2 + \Pr(i|3)(\underline{p}^{(1)})_3 + \Pr(i|4)(\underline{p}^{(1)})_4 = (i\text{th row of } C)\underline{p}^{(1)}.$$

We get all the probabilities by multiplying the vector $\underline{p}^{(1)}$ by C, i.e.

$$\underline{p}^{(2)} = C\underline{p}^{(1)} = C(C\underline{p}^{(0)}) = C^2\underline{p}^{(0)}.$$

In general

$$\underline{p}^{(k)} = C\underline{p}^{(k-1)} = \cdots = C^k\underline{p}^{(0)}, \quad k = 1, 2, \ldots.$$

Is there a limit as $k \to \infty$?

$$C = \begin{pmatrix} 0 & 1/3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1/2 & 1/3 & 0 & 0 \\ 1/2 & 1/3 & 1 & 0 \end{pmatrix}$$

Whatever we take for $\underline{p}^{(0)}$ the sequence of vectors converge to

$$\underline{p} = \begin{pmatrix} 2/5 \\ 0 \\ 1/5 \\ 2/5 \end{pmatrix}.$$

The program can verify this.

$\underline{p}$ is such that

$$C\underline{p} = \underline{p}$$

which will be a topic of the next sessions. $\underline{p}$ is an eigenvector of C with eigenvalue 1.

What if an out-degree is 0?

Suppose we replace

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

by

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

The out-degree of node 3 is 0 as all entries in column 3 are 0.

We cannot construct C in this case as we did before.

Group homework – enhancing the model

Instead of only following links the enhanced model introduces a parameter α, $0 \le \alpha \le 1$ such that the probability of following the links is $1 - \alpha$ and the probability of randomly going to another node is α. In this version when all out-degrees are greater than 0 the probability matrix changes to

$$C = (1-\alpha)C_{oldv} + \frac{\alpha}{n-1}\begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ \vdots & 1 & \ddots & 1 & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}.$$

If C_{oldv} has a column of zeros then in the enhanced model this column of C has entries of $1/(n-1)$ for every entry except the diagonal entry.

When there is a link between nodes there are two parts to consider. The probability is $(1-\alpha)$ times the previous value plus $\alpha/(n-1)$ as we also may move between the nodes by not following the links.

Group homework – task 1

This is the adjacency matrix for network 8 on page 14.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

Compute the matrix C in the enhanced version when $\alpha = 0.3$.

Group homework – task 2

This is the adjacency matrix for network 8 on page 14.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

Compute the matrix C in the enhanced version for a general value of α, i.e. apart from the diagonal entries the entries will now depend on α.

Group homework – task 3

Modify the Matlab function to a version which generates `C` in the enhanced model. The function should have two arguments which are `A` and `alpha`.