MARKOV CHAINS AND PAGERANK

Francois Role - francois.role@parisdescartes.fr

September 2016

1 Markov Chains

Markov chains are used to model the evolution of a process, in situations where the future evolution of the process depends on where it is at present, but not on how it got there.

Let S be a finite or countably infinite set of **states** and let $\{X_n\}_{n\geq 0}$ be a sequence of random variables with values in S. The probability of being in state y at point $n+1$ is only conditioned by where we are at time n. The probability of transitioning to state y if we are in state x is denoted as:

$$P(X_{n+1} = y | X_n = x) = p_{i,j}$$

The fact that were we are at step $n+1$ only depends on where we were at step n, and not on the entire past, is the so-called **Markov property**.

The $p_{i,j}$ probabilities of jumping from state x to state y are collected in a **transition matrix** P.

At a a given point in time t, an individual can be in one of k states with a certain probability. This distribution over states is represented as a stochastic vector $\pi^{(t)}$. The initial distribution vector is $\pi^{(0)} = [P(X_0 = x | x \in S)]^T$.

Suppose that we perform an experiment consisting in looking at the state of a given person at a point in time. Let A_i be the event "the individual is in state i at time n". Let B_j be the event "the individual is in state j at time $n+1$".

The A_i form a complete system of events: $A_i \cap A_k$ if $i \neq k$ and $\bigcup_{A_i} = \Omega$. Therefore, we have:

$$B_j = (B_j \cap A_1) \bigcup (B_j \cap A_2) \bigcup ... \bigcup (B_j \cap A_n).$$

By the law of total probabilities, we have:

$$P(B_j) = P(B_j \cap A_1) + P(B_j \cap A_2) + ... + (B_j \cap A_n)$$

Also, since $P(B_j) = P(X_{n+1} = j)$, we have:

P($\mathrm{X}_{n+1} = j) = P(X_{n+1} = j, X_n = 1) + P(X_{n+1} = j, X_n = 2) + ... + P(X_{n+1} = j), X_n = 1)$

P($\mathrm{X}_{n+1} = j) = P(X_{n+1} = j | X_n = 1) P(X_n = 1) + ...$

$\pi_j^{(n+1)} = p_{j1}\pi_1^{(n)} + ...$

The j-th entry of the $\pi^{(n+1)}$ vector is obtained by the dot product of the $j-th$ row of P and the $\pi^{(n)}$ vector. Hence, we have:

$$\pi^{(n+1)} = P\pi^{(n)} for\ n = 0, 1, 2, ...$$

Given an initial vector $\pi^{(0)}$ and the equation $\pi^{(k+1)} = P\pi^{(k)}$ it follows that:

$$P^n \pi^{(0)} = \pi^{(n)}$$

where $P^n_{i,j}$ contains the conditional probability of reaching i in n steps if we start from j (if the columns in P represent the sources and the rows the targets).

We will see that, for some particular P, $P^n\pi^{(0)}$ tends towards a **stationary vector** as n tends towards infinity.

Note: in the following we will often note the stochastic distribution vector as x_i instead of $\pi^{(i)}$.

1.1 Find the stationary vector: brute force

Exercise

a) Suppose that we can be in one of three states (numbered from 1 to 3) and that at the beginning the probability of being in state 1 is 1. The transition matrix P is :

$$\begin{pmatrix} 0.7 & 0.1 \\ 0.3 & \\ 0.2 & 0.8 \\ 0.3 & \\ 0.1 & 0.1 \\ 0.4 & \end{pmatrix}$$

$p_{1,2} = 0.2$ is the probability of moving from state 1 to state 2.
Compute the distribution over states after 20 steps.
b) Suppose now that at the beginning the probability of being in one of the three states is given by the vector: $[0.2, 0.4, 0.4]\top$. Compute the distribution over states after n steps, and conclude.
c) What does the entry $P^3_{1,2}$ represent?

1.2 Why does the system converge to a stationary vector: a preliminary investigation

As suggested by the previous exercise, the system seems to converge to a stationary vector. The following exercise will confirm this intuition.

Exercise

Suppose we have the following transition matrix P:

$$\begin{pmatrix} 0.95 & 0.03 \\ 0.05 & 0.97 \end{pmatrix}$$

We want to study the behavior of the dynamic system defined by $x_{k+1} = Px_k$ $(k = 0, 1, 2, ...)$ with $x_0 = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$.

The goal is to find an explicit formula for x_k and compute the limit for it when $k \to \infty$.

Perform the following steps:

1. Compute the eigenvectors v_1 and v_2 of P.
2. Express x_0 in terms of the eigenvectors of P (find c_1 and c_2 such that $x_0 = c_1v_1 + c_2v_2$).
3. Express x_1 in terms of x_0, x_2 in terms of x_1, and generalize to find an expression for x_k. You should find $x_k = c_1{\lambda_1}^k v_1 + c^2{\lambda_2}^k v_2$.
4. Substituting the values for c_1 and c_2 found in step **2** determine $\lim\limits_{k\to\infty} x_k$. The key point is that $\lambda_1 = 1$ (Why?).

$\lim\limits_{k\to\infty} x_k = v$ where v is the eigenvector associated with eigenvalue 1.

This is because, of all the eigenvalues, only eigenvalue 1 will not disappear when raised to an infinite power!

1.3 Find the stationary vector by solving $Px = x$

The previous exercised showed that the stationary vector is (a mutiple of the) eigenvector associated with eigenvalue 1. This in line with the fact that the stationary vector x is such that $Px = x$. So all we have to do for finding the stationary vector is to compute the eigenvector corresponding to eigenvalue 1.

NOTE: since $Px = x \Leftrightarrow (P - I)x = 0$, this is equivalent to forming the $(P - I)$ matrix and solve the equation $(P - I)x = 0$ for x.

Exercise

c) Given the matrix :

0.7 0.1 0.3
0.2 0.8 0.3
0.1 0.1 0.4

Find the stationary vector by computing the eigenvector of 1. Check that you get the same solution as in the first exercise.

1.4 Properties of markov chains and their associated matrices

A Markov chain is said to be **ergodic** (**irreducible** transition matrix) if it is possible to eventually get from every state to every state. Each state can move to each state for some step k in the future.

A Markov chain is **regular** (**primitive** transition matrix) if some power of its transition matrix has only positive entries. A strictly positive matrix is trivially primitive: the underlying graph is strongly connected. If A is a irreducible positive matrix then the spectral radius ρ of A is a simple eigenvalue of A. To be primitive, a matrix needs to be both irreducible and aperiodic. In summary, we have:

regular (primitive) = ergodic (irreducible) + acyclic (aperiodic)

If the transition matrix P is primitive we are guaranteed that the distribution for X_n will converge to a unique stationary distribution whatever starting distribution we use for X_0.

$Markov\ chain$	$P\ matrix$
$regular$	$primitive$
$ergodic$	$irreducible$
$acyclic$	$aperiodic$

Exercise

a. What can be said about $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Compute $P^{20}x$ for $x_0 = [0.2, 0.8]$.

b. Given matrix $P = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$, compute $P^{20}x$ for $x_0 = [0.2, 0.4, 0.4]$.

c. Let P denote the transition matrix for a regular Markov chain with finite state space (primitive matrix). Then $\lim_{n\to\infty} P^n$ is a matrix, all columns of which are the same strictly positive probability vector. Verify it with the primitive matrix P below.

2 Google Pagerank Matrix

Let H be a column normalized hyperlink matrix with $h_{ij} = \frac{1}{Degree_i j}$ if there is a link from j to i and 0 otherwise.

The matrix H is only substochastic. Some columns are likely to contain only 0, meaning that some nodes nay not have any outlinks. Transform it into a stochastic matrix S:

$$S = H + \frac{1}{n} e a^T$$

where e is a vector of all ones and a is a "dangling vector" whose component a_i is 1 if the i-th column of S contains only zeros and 0 otherwise.

We now a have a kind of Markov transition matrix. However, this is not enough to guarantee convergence to a unique stationary vector. So, we also perform a primitivity adjustment (the idea of the "random web surfer"):

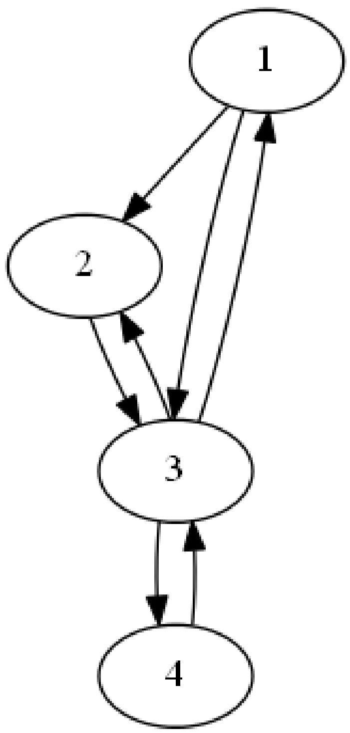
$$G = \alpha S + (1 - \alpha)\frac{1}{n} e e^T$$

The surfer can now reach any page at random. The matrix $\frac{1}{n}ee^T$ is called the “teleportation matrix”. Using the definition of S, G can also be expressed in terms of the original matrix as:

$$G = \alpha(H + \frac{1}{n}ea^T) + (1-\alpha)\frac{1}{n}ee^T$$
$$G = \alpha H + \frac{1}{n}e(\alpha a^T + (1-\alpha)e^T$$

Exercise

1. Compute the page rank scores for the following graph:



You should find that node 3 has the largest score, as shown below:
[0.15970516 0.23157248 **0.4490172** 0.15970516]