The Mathematics of Web Search

Random walk on a graph
Search Engine
Page Rank Algorithm
Perron-Frobenius Theorem
Power Method

Transition Matrices

Definition: In a directed graph, for every vertex i there is a number of edges that enter that vertex (i is a head) and a number of edges that exit that vertex (i is a tail). Thus we define the indegree of vertex i as the number of edges for which vertex i is a head. Similarly, the outdegree of vertex i as the number of edges for which i is a tail.

Mathematica has built-in functions for the indegree and the outdegree of the vertices of a graph:
VertexInDegree[graph] -- returns a list with the indegrees of the vertices of the graph specified in the argument
VertexOutDegree[graph] -- returns a list with the indegrees of the vertices of the graph

Definition: The transition matrix A of a directed graph is defined as follows. If there is an edge from i to j and the outdegree of vertex i is \(d_i\), then on column i and row j we put \(\frac{1}{d_i}\). Otherwise we mark the entry on column i and row j with 0.

Random Walk on a Graph: We use the transition matrix to model the behavior of a random surfer on a graph. The surfer chooses a node at random, then walks on the outgoing edges to other nodes for as long as he/she wishes. At each step the probability that the surfer moves from node i to node j is zero if there is no link from i to j and \(\frac{1}{d_i}\) otherwise. Recall that \(d_i\) is the outdegree of vertex i.

Question: What is the probability that a random surfer that starts at one of the nodes of the graph visits say, node j?
Search Engine

- gather information from Web pages.
  Crawlers crawl the Web following hyperlinks and they index the documents they find:
  \[ \text{document} \rightarrow \text{list of words, links, etc} \]

- process and store information in a database
  Indexer - computes the forward index
  \[ \text{document} \rightarrow \text{(number of occurrences of each word)} \]
  Sorter - computes the inverted index
  \[ \text{keyword} \rightarrow \{ \text{doc1, doc2, ..., docn} \} \]

- query the database in order to answer user's queries
  Query engine
  - uses the inverted index to compile a list of documents relevant to the keywords and phrases of the query.
  - lists the documents found in decreasing order, according to their relevance ("best" documents first)

\[ \text{Relevance}_{\text{Query, Page}}(\pi) = I(\pi, Q) \times P(\pi). \]

where \( I(\pi, Q) \) is a text-based ranking (on-page score of \( \pi \) for query \( Q \)) and \( P(\pi) \) is the quality factor of the page, or the page rank, independent of the textual content of the page.

\( P(\pi) \) is a number between 0 and 1, so multiplication by \( P(\pi) \) acts as a damping on the document score of \( \pi \).

Page Quality

- based on the reference(citation) ranking
- independent of the textual content of the page
- each page is assigned a quality value, based on the amount and quality of the citing pages.

The underlying assumption is that more important websites are likely to receive more links from other websites and therefore, the importance of any website can be judged by looking at the number and quality of the pages that link to it.

A Web citation is simply a link (page i "cites" page j is there is a link from page i to page j). In the next picture, Page 1 cites Pages 2, 3 and 4, Page 2 cites Pages 2 and 3, Page 3 cites Page 1, and finally Page 4 cites Pages 1 and 3.
Web Graph

Nodes = Web sites
Edges = Hyperlinks between web sites

We can "translate" our baby Internet model with 4 pages into a directed graph with 4 nodes, one for each web site. When web site i references j, we add a directed edge between node i and node j in the graph. For the purpose of computing their page rank, we ignore any navigational links such as back, next buttons, as we only care about the connections between different web sites.

```math
G = Graph[
    {1 -> 2, 1 -> 3, 1 -> 4, 3 -> 1, 2 -> 3, 2 -> 4, 4 -> 1, 4 -> 3},
    GraphStyle -> "SmallNetwork", VertexSize -> 0.15]
```

```
Lout = VertexOutDegree[G];
A = AdjacencyMatrix[G];
TA = Transpose[A];
Lout = VertexOutDegree[G];
f[i_, j_] := If[Lout[j] != 0, TA[[i]][[j]]/Lout[j], 0]
T = Table[f[i, j], {i, Length[TA]}, {j, Length[TA]}];
GraphicsRow[{G, MatrixForm[T]}]
```
Page Rank Algorithm

Suppose that initially the importance is uniformly distributed among the 4 nodes, each getting $\frac{1}{4}$. Denote by $v$ the initial rank vector, having all entries equal to $\frac{1}{4}$. Each incoming link increases the importance of a web page, so at step 1, we update the rank of each page by adding to the current value the importance of the incoming links. This is the same as multiplying the transition matrix $T$ with $v$.

At step 0, the importance is uniformly distributed, $v = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$.
At step 1, the new importance vector is $v_1 =Tv$.
At step 2, the updated importance vector is $v_2 = Tv_1 = T(Tv) = T^2 v$.

At step $k$, the updated importance vector is $v_k = T^k v$.

The following code in Mathematica tests this convergence process:

```mathematica
v = {0.25, 0.25, 0.25, 0.25}
For[k = 1, k <= 20, k++, Print[MatrixPower[T, k].v]]
```

The sequence converges to a vector $w \approx (0.3870, 0.1290, 0.2903, 0.1935)$. Note that this is only an approximation!

```mathematica
w = {0.3870, 0.1290, 0.2903, 0.1935};
```

The sum of the entries of $w$ is approximately 0.9998.
Page Rank Algorithm

* The limit vector \( w \) must satisfy the equation \( Tw = w \nabla \)

If \( v_k \rightarrow w \) as \( k \rightarrow \infty \) then \( Tv_k \rightarrow Tw \) as \( k \rightarrow \infty \) (because the map \( x \rightarrow Tx \) is continuous with respect to \( x \)).

However, \( Tv_k = v_{k+1} \) and \( v_{k+1} \rightarrow w \) as \( k \rightarrow \infty \)!

Therefore \( Tw = w \), so \( w \) belongs to the eigenspace of \( T \) corresponding to the eigenvalue 1.

```math
\text{eigen} = \text{Eigensystem}[T]
```

We do not have to compute all of the eigenvalues and eigenvectors of \( T \), because only the eigenspace corresponding to the eigenvalue 1 is relevant for our convergence problem.

```math
\text{evector} = \text{eigen}[[2]][[1]];
\text{Print}[
  "The eigenvectors corresponding to the eigenvalue 1 are scalar multiples of the vector ", \text{evector}]
```

The eigenvectors corresponding to the eigenvalue 1 are scalar multiples of the vector \( \begin{bmatrix} 2 & 3 & 2 & 1 \end{bmatrix} \)

We normalize so that the sum of the elements is equal to 1.

```math
w2 = 1 / (\text{evector}[[1]] + \text{evector}[[2]] + \text{evector}[[3]] + \text{evector}[[4]]) * \text{evector}
```

\[ w2 = \begin{bmatrix} 0.387097, 0.129032, 0.290323, 0.193548 \end{bmatrix} \]

This gives the exact value of the limit vector \( \begin{bmatrix} \frac{12}{31}, \frac{4}{31}, \frac{9}{31}, \frac{6}{31} \end{bmatrix} \).

Meaning of the Page Rank vector

The limit vector \( w \) is called the Page Rank vector of our graph. It provides a ranking system for the nodes of the graph. Each node is assigned an importance factor, based on the amount and relative importance of the nodes that link to it.

In our example, Node 1 (Page 1) has page rank 0.38 so it is the most important page. Page 3 has a score of 0.29, Page 2 has importance factor 0.12, so it is the least important, and Page 4 has importance factor 0.19.

Random Surfer Model: The page rank of Page 1 represents the probability that a random surfer on the Internet that opens a browser to any page and starts following hyperlinks, visits Page 1.
Perron-Frobenius Theorem

If \( M \) is a positive, column stochastic \( nxn \) matrix, then the following statements are true:

1. The number 1 is an eigenvalue of \( M \), of multiplicity one (that is, if \( u \) and \( v \) are two eigenvectors corresponding to the eigenvalue 1, then \( u \) is a scalar multiple of \( v \)).
2. The eigenvalue 1 is the largest eigenvalue of \( M \); all the other eigenvalues of \( M \) are strictly less than 1 in absolute value.
3. Any eigenvector corresponding to 1 has either positive or negative entries.
4. There exists a unique probabilistic eigenvector \( w \) corresponding to the eigenvalue 1.

Recall the following definitions:

A square matrix is called column stochastic if all its entries are greater than or equal to 0, and the sum of the entries in each column is 1.

A matrix is called positive if all of its entries are strictly greater than 0.

A vector is called probabilistic (or a probability distribution vector) if all its entries are greater than or equal to 0 and the sum of all entries is 1.

Power Method Convergence Theorem

Let \( M \) be a positive, column stochastic \( nxn \) matrix. Denote by \( w \) its unique probabilistic eigenvector corresponding to the eigenvalue 1.

Let \( v \) be the column vector with all entries equal to \( 1/n \). Then the sequence \( v, \ Mv, \ ... \ , \ M^kv \) converges to the vector \( w \) as \( k \) goes to \( \infty \).

Let \( z \) be any probabilistic vector of size \( n \). Then the sequence \( z, \ Mz, \ ... \ , \ M^kz \) converges to the vector \( w \) as \( k \) goes to \( \infty \).
Sketch of the proof

**Proposition 1:** If \( v \) is a probabilistic vector and \( M \) is a positive and column stochastic matrix, then \( Mv \) is a probabilistic vector with only positive entries.

**idea of proof:** If \( v \) is a probabilistic vector of dimension \( n \), and \( M \) is a column stochastic matrix, then \( Mv \) is also a probabilistic vector.

**Definition:** Let \( u \) and \( v \) be two vectors of size \( n \). We define the distance between \( u \) and \( v \) to be the non-negative real number \( d(u,v) = \frac{1}{2} \sum_{i=1}^{n} |u_i - v_i| \). Here \( u_i \) and \( v_i \) denote the \( i \)-th entries in the vectors \( u \) and respectively \( v \).

**Definition:** Consider now a sequence of vectors \( v_k, k>0 \), of size \( n \). We say that the sequence of vectors \( v_k \) converges to \( w \) as \( k \) tends to \( \infty \), if \( d(v_k, w) \to 0 \) as \( k \to \infty \).

**Proposition 2 (Contraction):** Let \( M \) be a positive and column stochastic \( nxn \) matrix. There exists a number \( r, 0<r<1 \), such that if \( v \) and \( u \) are two probabilistic vectors of size \( n \), then \( d(Mv, Mu)\leq(1-r) d(v,u) \). Therefore the distance between \( u \) and \( v \) decreases after left multiplication by \( M \).

**Proposition 3:** Let \( M \) be a positive and column stochastic \( nxn \) matrix and \( v \) any probabilistic vector of size \( n \). Denote by \( v_k=M^kv \). Then the sequence \( v_k \) converges exponentially fast to a unique limit vector \( w \).

**idea of proof:** Banach Fixed point Theorem!

Let \( j \) and \( k \) be any two positive integer. We can estimate the distance between any two terms of the sequence \( v_j \) and \( v_{jk} \) in the following way:

\[
d(v_{jk}, v_j) = d(M^{jk}v, M^jv) \leq (1-r)^j d(M^jv, v) \leq (1-r)^j d(v_k, v) \leq (1-r)^j \to 0 \quad \text{as} \quad j \to \infty,
\]

because \( 1-r \) is strictly less than \( 1 \). Therefore the sequence \( (v_k)_{k=0} \) is a Cauchy sequence in \( \mathbb{R}^n \), so there exists a vector \( w \) in \( \mathbb{R}^n \) such that \( v_k \to w \) as \( k \to \infty \). The limit \( w \) is unique! Suppose by contradiction, that there would be (at least) two limit vectors, \( w \) and \( z \). Then both \( w \) and \( z \) must satisfy \( Mw=w \) and \( Mz=z \). Then we must also have the inequalities \( 0 \leq d(w, z) = d(Mw, Mz) \leq (1-r)d(w, z) < d(w, z) \), because \( 1-r \) is strictly less than \( 1 \). We reached a contradiction, \( d(w, z)<d(w, z) \), which shows that our initial assumption was false and \( w \) and \( z \) must be equal.