The PageRank/HITS algorithms

JONI PAJARINEN, Joni.Pajarinen@tkk.fi

March 19, 2008
Outline

Link Analysis
  WWW
  Other Applications
  Web page references

HITS
  Hypertext Induced Topics Search (HITS)
  Eigenvectors and SVD
  Iterative method

PageRank
  PageRank
  PageRank problems
  PageRank natural solution
  Computing the PageRank

Summary
The World Wide Web (WWW) consists of pages that reference (link to) each other.

The adjacency matrix $A$ of a set of pages (nodes) defines the linking structure.

Matrix element $a_{ij}$ is 1 if node $i$ references node $j$ and 0 otherwise.

$$A = \begin{pmatrix}
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}$$
Several other applications share same linking characteristics with the WWW

Article citations form a web of references

Journal importance could and has been analysed using link analysis

Social networks
What can we say about web page references?

- Interesting pages are referenced by several other pages
- Interesting pages are referenced by interesting pages
- A page, which references several interesting pages, might be itself interesting
Hypertext Induced Topics Search (HITS) developed by Jon Kleinberg

HITS is applied on a subgraph after a search is done on the complete graph

Uses hubs and authorities to define a recursive relationship between web pages

An authority is a page that many hubs link to

A hub is a page that links to many authorities
The scores for authority nodes $x$ can be determined from the hub scores $x = A^T y$.

And similarly the hub scores from the authority scores $y = Ax$.

Substituting into the equations we get

$$x = A^T Ax$$

$$y = AA^T y$$
$\|\|_2$ normalized hub and authority scores of example web graph
Singular Value Decomposition (SVD)

For a real valued $m \times n$ matrix $A$ the SVD $A = U S V^T$ consists of $U$, a $m \times n$ orthogonal matrix, $S$, a $m \times n$ matrix of singular values on the diagonal and $V$ an orthogonal matrix of size $n \times n$

A singular value $\sigma$ is such that $Av = \sigma u$ and $A^T u = \sigma v$, where $u$ is called the left-singular and $v$ the right-singular vector

For $A = U S V^T$, $U$ consists of left-singular vectors, $V$ of right-singular vectors and $S$ of the singular values
Finding eigenvectors for $AA^T$ and $A^TA$ solves the hub and authority score linear equations.

For the matrix $A$ we can use singular value decomposition (SVD) on $A = USV^T$.

- $A^TA = V S^T U^T U S V^T = V (S^T S) V^T = V \Sigma V^T$
- $A A^T = U S V^T V S^T U^T = U (S S^T) U^T = U \Sigma U^T$

$\Sigma$ is a diagonal matrix with the eigenvalues.

The first vectors of left and right matrices $U$ and $V$ are the first eigenvectors for $AA^T$ and $A^TA$ respectively, i.e. the hub and authority scores.
An iterative method suggested by Kleinberg for solving the linear equations

We use the following two operations to update the weights

\[ x_j = \sum_{a_{ij}=1} y_i \]
\[ y_i = \sum_{a_{ij}=1} x_j \]

The hub and authority scores are normalized using \( \| \|_2 \)
Input: Adjacency matrix $A$ of size $n \times m$ and number of iterations
Output: Authority and hub score vectors $\mathbf{x}$ and $\mathbf{y}$ respectively

$\mathbf{x} = (1, 1, \ldots, 1) \in \mathbb{R}^m$; $\mathbf{y} = (1, 1, \ldots, 1) \in \mathbb{R}^n$;

while Iterations still left do
  for $i=1,2,\ldots,m$ do
    $x_j = \sum_{a_{ij}=1} y_i$;
  end
  for $j=1,2,\ldots,n$ do
    $y_i = \sum_{a_{ij}=1} x_j$;
  end
  Normalize($\mathbf{x}$); Normalize($\mathbf{y}$);
end

Algorithm 1: Iterative algorithm for computing the authority and hub score vectors
PageRank developed by Larry Page and Sergey Brin at Stanford University

Based on the idea of a 'random surfer'

Pages as Markov Chain states

Probability for moving from a page to another page modelled as a state transition probability
The Markov Chain state transition probability matrix $P$

$$
P = \begin{pmatrix}
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & \frac{1}{2} & \frac{1}{2} \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
$$

The pagerank $r^T = r^T P$
- **Dead-end states** → matrix $P$ not stochastic
- **Transient states** → Markov Chain not irreducible
- **Periodic states** → no stable $r$
- $\mathbf{v}$ is the personalization stochastic vector
- The uniform vector $\mathbf{v} = \frac{\mathbf{e}}{|\mathbf{e}|}$, where $\mathbf{e} = (1, \ldots, 1)$, is used often
- Adding the possibility to jump from dead-end nodes to any node: $P_{stochastic} = P + D$, where $D = d\mathbf{v}^T$ and $d_i = 1$, when $i$ is a dead-end node
- Adding the possibility to teleport to any node: $P_{final} = \alpha P_{stochastic} + (1 - \alpha)\mathbf{e}\mathbf{v}^T$, where $\alpha$ is the dampening factor
  - $P_{final}$ is irreducible and all its states are aperiodic
\( \mathbf{r}^T = \mathbf{r}^T \mathbf{P}_{final} \) determines the unique stationary distribution \( \mathbf{r} \), because the Markov Chain is irreducible and its states are aperiodic.

Also \( \mathbf{r}^T = \mathbf{u}^T \lim_{k \to \infty} \mathbf{P}_{final}^k \), where \( \mathbf{u} \) is any stochastic vector.
PageRank example using the dampening factor $\alpha = 0.85$

$$P_{final} = \alpha (P + D) + (1 - \alpha) \frac{ee^T}{|e|}$$

$$P_{final} = \begin{pmatrix}
0.0375 & 0.3208 & 0.3208 & 0.3208 \\
0.0375 & 0.0375 & 0.4625 & 0.4625 \\
0.0375 & 0.8875 & 0.0375 & 0.0375 \\
0.25 & 0.25 & 0.25 & 0.25 \\
\end{pmatrix}$$
Storage and computational complexity problems

- $P$ is usually sparse, but $P_{final}$ is dense
- Computing the first left eigenvector of $P_{final}$ solves $r$ for the linear equation $r^T = r^T P_{final}$, but can be computationally demanding
Using the Power Iteration method we can calculate $r$ performing mostly sparse calculations

$$
\begin{align*}
  r_0^T &= \frac{e}{|e|} \\
  r_{i+1}^T &= r_i^T P_{final} \\
  &= r_i^T \left( \alpha P_{stochastic} + (1 - \alpha) \frac{e e^T}{|e|} \right) \\
  &= \alpha \left( r_i^T P + r_i^T D \right) + (1 - \alpha) r_i^T
\end{align*}
$$

Other methods for sparse computation of PageRank exist, e.g. solving $(I - \alpha P^T) y = v$ and then $r = \frac{y}{\|y\|_1}$ (proof in [1])
HITS is applied on a subgraph after a search is done on the complete graph.

HITS defines hubs and authorities recursively.

PageRank is used for ranking all the nodes of the complete graph and then applying a search.

PageRank is based on the 'random surfer' idea and the web is seen as a Markov Chain.

Power Iteration an efficient way to calculate with sparse matrices.

