The PageRank/HITS algorithms

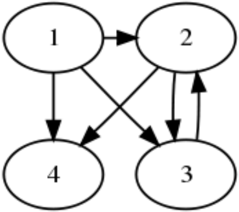
Joni Pajarinen, Joni.Pajarinen@tkk.fi

March 19, 2008

Outline

- 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

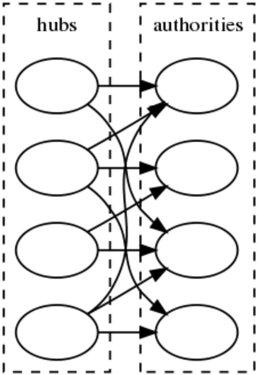
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$



- 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

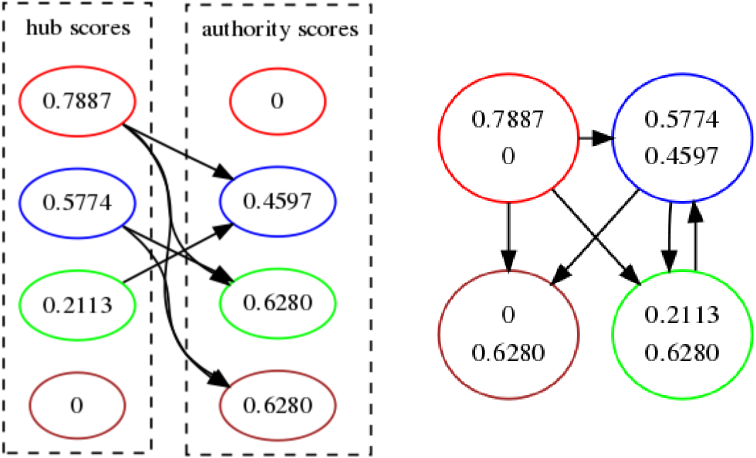


- The scores for authority nodes $\mathbf{x}$ can be determined from the hub scores $\mathbf{x} = A^T\mathbf{y}$
- And similarly the hub scores from the authority scores $\mathbf{y} = A\mathbf{x}$
- Substituting into the equations we get

$$\mathbf{x} = A^T A\mathbf{x}$$

$$\mathbf{y} = AA^T\mathbf{y}$$

$\|\|_2$ normalized hub and authority scores of example web graph



- Singular Value Decomposition (SVD)
- For a real valued $m \times n$ matrix A the SVD $A = USV^T$ 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 σ 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 $A = USV^T$, 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 $A = USV^T$
- $A^TA = VS^TU^TUSV^T = V\left(S^TS\right)V^T = V\ \Sigma\ V^T$
  $AA^T = USV^TVS^TU^T = U\left(SS^T\right)U^T = U\ \Sigma\ U^T$
  Σ 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
  - $\mathbf{x}_j = \sum_{a_{ij}=1} \mathbf{y}_i$
  - $\mathbf{y}_i = \sum_{a_{ij}=1} \mathbf{x}_j$
- The hub and authority scores are normalized using $\|\|_2$

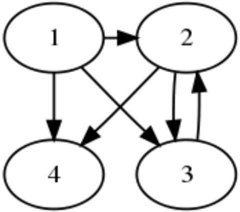
Input: Adjacency matrix A of size $n \times m$ and number of iterations
Output: Authority and hub score vectors $\mathbf{x}$ and $\mathbf{y}$ respectively
$\mathbf{x} = (1, 1, \ldots, 1) \in \mathcal{R}^m$; $\mathbf{y} = (1, 1, \ldots, 1) \in \mathcal{R}^n$;
while *Iterations still left* **do**
    for i=1,2,...,m **do**
        $x_j = \sum_{a_{ij}=1} y_i$;
    end
    for j=1,2,...,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

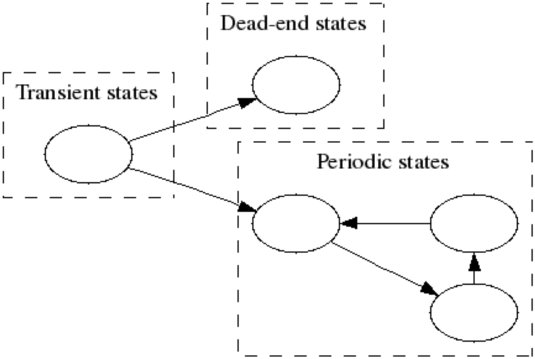
- The Markov Chain state transition probability matrix P

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



- The pagerank $\mathbf{r}^T = \mathbf{r}^T P$

- Dead-end states $\longrightarrow$ matrix P not stochastic
- Transient states $\longrightarrow$ Markov Chain not irreducible
- Periodic states $\longrightarrow$ no stable $\mathbf{r}$

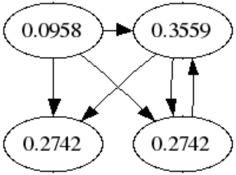


- $\mathbf{v}$ is the personalization stochastic vector
- The uniform vector $\mathbf{v} = \frac{\mathbf{e}}{|\mathbf{e}|}$, where $\mathbf{e} = (1, \ldots, 1)$, is used often
- Adding the possibility to jump from dead-end nodes to any node: $P_{stochastic} = P + D$, where $D = \mathbf{d}\mathbf{v}^T$ 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 α is the dampening factor
  - P_{final} is irreducible and all its states are aperiodic

- $\mathbf{r}^T = \mathbf{r}^T P_{final}$ determines the unique stationary distribution $\mathbf{r}$, because the Markov Chain is irreducible and its states are aperiodic
- Also $\mathbf{r}^T = \mathbf{u}^T \lim_{k\to\infty} P_{final}^k$, where $\mathbf{u}$ is any stochastic vector

- PageRank example using the dampening factor $\alpha = 0.85$
- $P_{final} = \alpha (P + D) + (1 - \alpha) \frac{\mathbf{e}\mathbf{e}^T}{|\mathbf{e}|}$

$$P_{final} = \begin{pmatrix} 0.0375 & 0.3208 & 0.3208 & 0.3208 \\ 0.0375 & 0.0375 & 0.4625 & 0.4625 \\ 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$



- Storage and computational complexity problems
- P is usually sparse, but P_{final} is dense
- Computing the first left eigenvector of P_{final} solves $\mathbf{r}$ for the linear equation $\mathbf{r}^T = \mathbf{r}^T P_{final}$, but can be computationally demanding

- Using the Power Iteration method we can calculate $\mathbf{r}$ performing mostly sparse calculations

$$
\begin{aligned}
\mathbf{r}_0^T &= \frac{\mathbf{e}}{|\mathbf{e}|} \\
\mathbf{r}_{i+1}^T &= \mathbf{r}_i^T P_{final} \\
&= \mathbf{r}_i^T \left(\alpha P_{stochastic} + (1-\alpha)\,\mathbf{e}\frac{\mathbf{e}^T}{|\mathbf{e}|} \right) \\
&= \alpha \left(\mathbf{r}_i^T P + \mathbf{r}_i^T D \right) + (1-\alpha)\,\mathbf{r}_i^T
\end{aligned}
$$

- Other methods for sparse computation of PageRank exist, e.g. solving $\left(I - \alpha\ P^T\right)\mathbf{y} = \mathbf{v}$ and then $\mathbf{r} = \frac{\mathbf{y}}{\|\mathbf{y}\|_1}$ (proof in [1])

Summary

- 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

References

- G. Del Corso, A. Gullí, and F. Romani, "Fast PageRank Computation via a Sparse Linear System," *Internet Mathematics*, 2005.
- J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.
- A. Langville and C. Meyer, "A Survey of Eigenvector Methods for Web Information Retrieval," *SIAM Review*, vol. 47, no. 1, pp. 135–161, 2005.
- S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 107–117, 1998.