CS 395T	Large-Scale Data Mining	Fall 2003
	Lecture 03 & 04	
Lecturer: Inderjit Dhillon		Date: Sept. 15 & 17, 2003

Keywords: *HITS, Clever Project, Google, PageRank*

1 Review

According to Kleinberg's algorithm, hubs correspond to large numbers in the left principal singular vector of the adjacency matrix $\mathbf{A}$, while authorities correspond to large numbers in the right principal singular vector of $\mathbf{A}$.

$$\begin{aligned}
\mathbf{A} &= \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^T \\
\mathbf{A}\mathbf{A}^T &= \sigma_1^2 \mathbf{u}_1 \mathbf{u}_1^T + \sigma_2^2 \mathbf{u}_2 \mathbf{u}_2^T + \cdots + \sigma_n^2 \mathbf{u}_n \mathbf{u}_n^T \\
\frac{(\mathbf{A}\mathbf{A}^T)^k \mathbf{h}_0}{||(\mathbf{A}\mathbf{A}^T)^k \mathbf{h}_0||} &= \frac{(\sigma_1^2)^k \mathbf{u}_1(\mathbf{u}_1^T \mathbf{h}_0) + (\sigma_2^2)^k \mathbf{u}_2(\mathbf{u}_2^T \mathbf{h}_0) + \cdots + (\sigma_n^2)^k \mathbf{u}_n(\mathbf{u}_n^T h_0)}{\sqrt{\{(\sigma_1^2)^k \mathbf{u}_1^T \mathbf{h}_0)^2 + ((\sigma_2^2)^k \mathbf{u}_2^T \mathbf{h}_0)^2 + \cdots + ((\sigma_n^2)^k \mathbf{u}_n^T \mathbf{h}_0\}^2}} \\
&= \frac{(\sigma_1^2)^k \mathbf{u}_1^T \mathbf{h}_0 \{\mathbf{u}_1 + (\frac{\sigma_2^2}{\sigma_1^2})^k (\frac{\mathbf{u}_2^T \mathbf{h}_0}{\mathbf{u}_1^T \mathbf{h}_0}) \mathbf{u}_2 + \cdots\}}{\sqrt{\{(\sigma_1^2)^k (\mathbf{u}_1^T \mathbf{h}_0)\}^2 \{1 + (\frac{\sigma_2}{\sigma_1})^{2k} (\frac{\mathbf{u}_2^T \mathbf{h}_0}{\mathbf{u}_1^T \mathbf{h}_0})^2 + \cdots\}}} \\
&\to \frac{\sigma_1^{2k} (\mathbf{u}_1^T \mathbf{h}_0) \mathbf{u}_0}{\sigma_1^{2k} (\mathbf{u}_1^T \mathbf{h}_0)} = \mathbf{u}_0, \text{ as } k \to \infty \text{ (if } \sigma_1 > \sigma_2 \text{ and } \mathbf{u}_1^T \mathbf{h}_0 \neq 0).
\end{aligned}$$

Thus, Kleinberg's algorithm converges if $\sigma_1^2 > \sigma_2^2$ (note that by the non-negativity of the matrix $\mathbf{A}$ it can be shown that $\mathbf{u}_1^T \mathbf{h}_0 \neq 0$ if $\mathbf{h}_0$ is the vector of all 1's). Rate of convergence $= \frac{\sigma_2^2}{\sigma_1^2}$ (i.e. the ratio of second largest eigenvalue of $\mathbf{A}\mathbf{A}^T$ to its largest eigenvalue). The smaller the above ratio the faster is the convergence of the Kleinberg algorithm also known as the HITS algorithm (HITS stands for Hypertext Induced Topic Search).

2 Improvements to basic HITS algorithm

1. Put weights on edges to reflect importance of links, e.g., put higher weight if anchor text associated with the link is relevant to the query.

2. Normalize weights outgoing from a single source or coming into a single sink. This alleviates spamming of query results since the influence due to 1 page is reduced.

3. Eliminate edges between same domain.

(Is there still a way to spam a system based on the above algorithm?)

3 Further Application of HITS algorithm

The Clever Project at IBM Almaden used HITS for many applications.

History of such methods is in Citation Analysis, which examines citation patterns in scientific/technical journals.

Co-citation: Two papers share a co-citation if they are referred to by the same paper. $(\mathbf{A}^T\mathbf{A})_{ij}$ counts the number of co-citations shared by i and j.

Co-reference: Two papers share a co-reference if they refer to the same paper. $(\mathbf{A}\mathbf{A}^T)_{ij}$ counts the number of co-references shared by i and j.

Various applications of HITS:

1. Better query results by search engines (none of the popular search engines use HITS — IBM holds patent).

2. Automatically discovering emerging communities that are not yet indexed by Yahoo!, Excite and other major portals [KRRT99].

3. TaxMan (Taxonomy Manager for Taxonomy Generation): Use queries and feedback (include a web site, mark web site as relevant) to get good authoritative pages.

4. www.quiver.com – Authorities are web sites, Hubs are bookmark folders. Edge weights are weighted using user behavior and visit patterns.

5. Finding related pages on the web [DH99].

4 Google's PageRank

Again, a link from $p \to q$ gives importance to q.

$$\text{PageRank}(q) = \sum_{(p:p\to q)} \frac{\text{PageRank}(p)}{|F_p|} \tag{1}$$

where F_p is the set of pages p points to, c is a scaling constant chosen such that $\Sigma_q \text{PageRank}(q) = 1$. Thus, PageRank may be thought of as a probability distribution over all pages on the web. Consider the row-scaled adjacency Matrix $\mathbf{A}$ with

$$A_{ij} = \begin{cases} \frac{1}{|F_i|} & , \quad i \to j \\ 0 & , \quad i \not\to j. \end{cases} \tag{2}$$

Let $\mathbf{r}$ be the vector of PageRanks, $\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$.

The formula(1) may be written in the form

$$\begin{aligned} \mathbf{r}^{(k+1)} &= \mathbf{A}^T \mathbf{r}^{(k)}, \\ ||\mathbf{r}^{(k)}||_1 &= 1. \end{aligned}$$

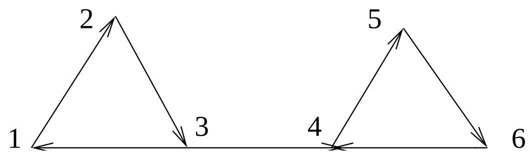

Figure 1: An example of rank sink.

Note that $\|\mathbf{r}^{(k+1)}\|_1 = 1$ as we show below:

$$\begin{aligned} r_j^{(k+1)} &= \sum_i (\mathbf{A}^T)_{ji} r_i^{(k)} \\ &= \sum_i A_{ij} r_i^{(k)}. \end{aligned}$$

Since all entries in $\mathbf{A}$ and $\mathbf{r}^{(k)}$ are non-negative,

$$\begin{aligned} \|\mathbf{r}^{(k+1)}\|_1 &= \sum_j \sum_i A_{ij} r_i^{(k)} \\ &= \sum_i \left(\sum_j A_{ij} \right) r_i^{(k)}. \end{aligned}$$

By (2), $\sum_j A_{ij} = 1$, and so substituting in the above equation gives us

$$\|\mathbf{r}^{(k+1)}\|_1 = \sum_i r_i^{(k)} = \|\mathbf{r}^{(k)}\|_1. \tag{3}$$

The iterate $\mathbf{r}^{(k)}$ will converge to the dominant (principal) eigenvector of $\mathbf{A}^T$ as $k \to \infty$ (under mild conditions). Note $\mathbf{A}$ is not symmetric and thus, the non-principal eigenvalues and eigenvectors may be complex.

Perron Theorem: A non-negative matrix has a non-negative principal eigenvalue and eigenvector.

Problems:

$$\mathbf{A} = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right]$$

The dominant eigenvector has zeros in the first 3 entries, i.e., PageRank for pages 1, 2, 3 are 0. This is because the cycle $4 \to 5 \to 6$ acts as a rank sink. (see Fig. 1)

Solution: Introduce rank source

$$
\begin{aligned}
\text{PageRank}(q) &= \sum_{(p:p\rightarrow q)} c \left(\frac{\text{PageRank}(p)}{|F_p|}\right) + (1-c)s_q, \quad 0 < c < 1 \\
\text{where} \quad \mathbf{s} &= \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \quad \text{is the rank source vector} \\
\mathbf{r}^{(k+1)} &= c\mathbf{A}^T\mathbf{r}^{(k)} + (1-c)\mathbf{s} \\
&= (c\mathbf{A}^T + (1-c)\mathbf{s}\mathbf{e}^T)\mathbf{r}^{(k)} \quad \text{because} \quad ||\mathbf{r}^{(k)}||_1 = \mathbf{e}^T\mathbf{r}^{(k)} = 1.
\end{aligned}
$$

Thus, PageRank is given by the dominant eigenvector of $c\mathbf{A}^T + (1-c)\mathbf{s}\mathbf{e}^T$. Usually, c is chosen to be 0.85.

Another way to think about PageRank is the *Random Surfer Model.* The "random" surfer can follow any outlink from a page with equal probability. Periodically the "random" surfer "gets bored" and jumps to a random page on the Web.

Dominant Eigenvector Computation

The following iteration is given in page 8 of the paper "What can you do with a Web in your Pocket?" [BMPW98].

1. Initialize $\mathbf{r}^{(0)}$ such that $||\mathbf{r}^{(0)}||_1 = 1$
2. Loop:
   (a) $\mathbf{r}^{(i+1)} = \mathbf{A}^T\mathbf{r}^{(i)}$
   (b) $\alpha = ||\mathbf{r}^{(i)}||_1 - ||\mathbf{r}^{(i+1)}||_1$. (Is $\alpha \geq 0$?)
   (c) $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i+1)} + \alpha\mathbf{s}$
   (d) $\delta = ||\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}||_1$
3. while $\delta \geq \epsilon$

There is a flaw in the above algorithm since α is always 0 (see (3)). So, $\mathbf{s}$ does not come into the picture.

5 Graph Structure in the Web

Understanding the various properties of the web graph (e.g., diameter, degree distributions, connected components) can help various applications:

1. Designing better crawl strategies on the web (Cho, Garcia-Molina).
2. Analyzing behavior of web algorithms that use link information, e.g., how quickly/slowly will the PageRank and HITS algorithm converge.
3. Prediction of web structures, such as bipartite cores and better algorithms to compute them.
4. Predict emergence of new, yet unexploited, phenomena in the web graph.

The following are some highlights of web graph properties:

1. "Six" degrees of separation between any 2 web pages (almost true in the strongly connected component part of web graph if you allow traversal of both out-links and in-links). Not true in general. Has led to small-world network models for the graph (Barabasi *et.al.*, Watts and Strogatz).

2. Probability that a web page has in-degree i is proportional to $1/i^x$ for positive $x > 1$. ($x = 2.1$ in latest experiments).

3. Average path length between two web pages is 19 (Barabasi). Raghavan, *et. al.* claim this is not true in their paper "Graph Structure in the Web" [BKM+00].

The above material is covered in great detail in Raghavan, *et. al.*.

References

[BKM+00] A. Broder, R. Kumar, F. Maghouland, P. Raghavan, S. Rajagopalan, and R. Stata. Graph structure in the web. In *Proceedings of the 6th International World Wide Web Conference*, 2000.

[BMPW98] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

[DH99] J. Dean and M. Henzinger. Finding related pages in world wide web. In *Proceedings of the 8th International World Wide Web Conference*, May 1999.

[KRRT99] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cybercommunities. In *Proceedings of the 8th International World Wide Web Conference*, pages 403–415, May 1999.