CS 395T	Large-Scale Data Mining	Fall 2001

Lecture 03 & 04

Lecturer: Inderjit Dhillon Date: Sept 11&13, 2001

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

1 Review

According to Kleinberg, 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 $\mathbf{A}$.

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

Thus, Kleinberg's algorithm converges if $\sigma_1^2 > \sigma_2^2$ (note that 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's algorithm also known as the *HITS* algorithm (HITS stands for Hypertext Internet 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: Examines citation patterns in scientific/technical journals.

Co-citation: Common paper has references to 2 pages.

Bibliographic Coupling: Two papers have references to common papers.

Various applications of HITS:

1. Better query results by search engines (do not think that any 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]
6. Many others. Don't know if this has been applied to collaborative filtering. Possible project is to apply this to the Movies Database.

4 Google's Page Rank

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 adjacency Matrix $\mathbf{A}$ with

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

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

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

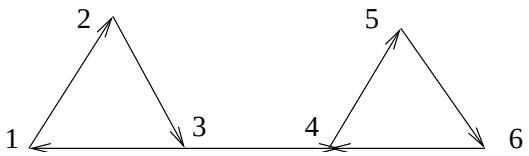

Figure 1: An example of rank sink.

Thus $\mathbf{r}^{(k)}$ 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: Non-negative matrix has non-negative principal eigenvalue and eigenvector.

Problems:

$$\mathbf{A} = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ \multicolumn{6}{c}{\cdots\cdots\cdots\cdots\cdots} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right] \tag{3}$$

All eigenvectors have 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

$$\text{PageRank}(q) = \sum_{(p:p\to q)} \frac{\text{PageRank}(p)}{|F_p|} + s_q \tag{4}$$

$$\text{where} \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \quad \text{is the} \quad \text{rank source vector} \tag{5}$$

$$\mathbf{r}^{(k+1)} = \mathbf{A}^T\mathbf{r}^{(k)} + \mathbf{s} \tag{6}$$

$$= (\mathbf{A}^T + \mathbf{s}\mathbf{e}^T)\mathbf{r}^{(k)} \quad \text{because} \quad ||\mathbf{r}^{(k)}||_1 = \mathbf{e}^T\mathbf{r}^{(k)} = 1. \tag{7}$$

Thus, PageRank is given by the dominant eigenvector of $\mathbf{A}^T + \mathbf{s}\mathbf{e}^T$.

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?" [BMaW98].

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 $\delta \geq \epsilon$

Thus,

$$||\mathbf{r}^{(i+1)}||_1 = ||\mathbf{A}^T\mathbf{r}^{(i)}||_1 + (||\mathbf{r}^{(i)}||_1 - ||\mathbf{A}^T\mathbf{r}^{(i)}||_1)||\mathbf{s}||_1 = 1 \quad \text{if} \quad ||\mathbf{s}||_1 = 1.$$

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.

[BMaW98] S. Brin, R. Motwani, and L. Page andT. 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.