CS 395T	Large-Scale Data Mining	Fall 2003

Lecture 01

Lecturer: Inderjit Dhillon — Date: Sept 3, 2003

Keywords: *Connectivity analysis on the web, Hubs and Authorities, Singular Vectors*

References

1. "Authoritative sources in a hyperlinked environment" by Jon Kleinberg

2. "Improved Algorithms for Topic Distillation in a Hyperlinked Environment" by Krishna Bharat and Monika Henzinger

Web-Search:

- Concerned about *broad-topic* queries, e.g. "web-browsers".
- Fundamental difficulty: the *Abundance problem* (The number of pages that could be returned as relevant is far too large).
- Need a mechanism for ranking all these pages.

Hypothesis: Page p, by including a link to page q, *confers authority* on q

- How do you use link information in answering a broad topic query?
- Desire pages that are *relevant and authoritative.*
- Will look at link information in a "relevant" subgraph of the web.

Relevant subgraph of a query:

1. Take $\mathbf{S}$ to be the top k (say 200) pages indexed by Alta Vista (or any other search engine).
2. Grow $\mathbf{S}$ to $\mathbf{T}$,

$$\mathbf{T} = \mathbf{S} \cup \{p | \exists \text{ link from } p \text{ to a page in } \mathbf{S}\} \cup \{q | \exists \text{ link from a page in } \mathbf{S} \text{ to } q\} \qquad (1)$$

3. Consider the subgraph induced by $\mathbf{T}$. This is the "relevant" subgraph.

One idea is to give a high rank to pages with large number of in-links. However, this turns out to be unsatisfactory. It is not only the large in-degree of a page p that is important, but also the quality of pages that point to p. If more important pages point to p, that means p is more authoritative.

KEY IDEA: **Good "hub" pages have links to good "authority" pages**.

Let $G = (V, E)$ denote the "relevant" subgraph induced by $\mathbf{T}$. Let $V = \{p_1, p_2, \ldots, p_n\}$ be n pages in V. Associate nonnegative *authority weight* a_p and a non-negative *hub weight* h_p with each page $p \in V$. Normalize the weights such that:

$$\sum_{p \in V} a_p^2 = 1 \quad \text{and} \quad \sum_{p \in V} h_p^2 = 1. \tag{2}$$

Kleinberg's (Basic) Algorithm

1. Initialize: $h_p = 1$, $a_p = 1$ $\forall$ pages $p \in V$.
2. Iterate for $i = 1, 2, 3, \ldots$
   - Update authority weights using hub weights,
   $$a_p \leftarrow \sum_{q:(q,p) \in E} h_q, \quad \forall p \in V. \tag{3}$$
   - Update hub weights using authority weights,
   $$h_p \leftarrow \sum_{q:(p,q) \in E} a_q, \quad \forall p \in V. \tag{4}$$
   - Normalize a_p and h_p such that (2) holds.

Do the authority and hub weights converge as the iteration count increases?

Consider the adjacency matrix $\mathbf{A}$ of graph G. $\mathbf{A}$ is $n \times n$ and $A_{ij} = 1$ if $(p_i, p_j) \in E$ is an edge in G, and 0 otherwise. In terms of $\mathbf{A}$, (3) and (4) become:

$$\mathbf{a} \leftarrow \mathbf{A}^T \mathbf{h} \quad \text{and} \quad \mathbf{h} \leftarrow \mathbf{A}\mathbf{a}, \tag{5}$$

where $\mathbf{h}$ = vector of hub weights and $\mathbf{a}$ = vector of authority weights, both of sizes $n \times 1$. The iterative updates lead to the following:

$$\begin{array}{l} \mathbf{a}^{(1)} = \mathbf{A}^T \mathbf{h}^{(0)} \\ \mathbf{h}^{(1)} = \mathbf{A}\mathbf{a}^{(1)} \end{array} \Rightarrow \mathbf{h}^{(1)} = \mathbf{A}\mathbf{A}^T \mathbf{h}^{(0)} \Rightarrow \mathbf{h}^{(k)} = (\mathbf{A}\mathbf{A}^T)^k \mathbf{h}^{(0)}, \tag{6}$$

where $\mathbf{h}^{(k)}$ denotes the hub vector at k^{th} iteration. With normalization,

- $\mathbf{h}^{(k)}$ is a unit vector in the direction of $(\mathbf{A}\mathbf{A}^T)^k \mathbf{h}^{(0)}$, and
- $\mathbf{a}^{(k)}$ is a unit vector in the direction of $(\mathbf{A}^T\mathbf{A})^{k-1} \mathbf{A}^T \mathbf{h}^{(0)}$.

Under mild conditions (a proof will be given in the next lecture):

- $\mathbf{h}^{(k)}$ converges to the leading left singular vector of $\mathbf{A}$, and
- $\mathbf{a}^{(k)}$ converges to the leading right singular vector of $\mathbf{A}$.

Another problem is: **similar-page queries**. Given a page p, what pages are similar to p?
If p is a good authority on some topic, take T to be pages (200) that point to p. Then find authorities, and output them as similar pages.

Problems/Drawbacks with Hubs & Authorities approach:

1. Links may be within the same domain name.
2. There may be many sites within a domain name that contain many links to a particular page.
3. Links may be automatically generated and don't always confer authority on a page.
4. May get non-relevant pages due to "topic drift" or "topic dilution".

Solutions:

1. Remove links within the same domain name.
2. Normalize each link to have weight $1/k$, where k is the total number of links from the same domain name.

3 & 4. Do content analysis by looking at the text of the page to ensure that the page is relevant. Remove irrelevant pages and penalize pages that are less relevant.