Google’s Page Rank Algorithm

Searching the Web – Background

- When looking for information on the Web a standard approach is to type (or speak) relevant *query terms* to a *search engine*.
- For example: Bing, Yahoo, Ask (20 years ago, Lycos, Altavista).
- One (the most?) popular search engine is *Google*.
- The quality of such search engines is gauged by:

Speed of response

Number of results

Ordering of results

Relevance of results

Why is Google so dominant?

- Typically users have found the results reported are:

returned quickly

comprehensive

relevant

well-ordered

- While other search engines perform reasonably wrt to the *first three criteria* user experience suggests Google is *most accurate regarding the last* of these.
- eg using Altavista it was often needed to go through 70-80 *reported pages* before identifying *relevant links*.

Search Engines – Basic Ideas

- Consider a selection of pages as forming a *directed graph*.
- eg the selection might be all pages containing a given phrase.
- There is a *directed edge* from one node (*page*) to another if that page *links to it*.
- The "*search problem*" is then about *ordering* all of the nodes (pages) so that the "*most important*" appear *first* in this order.

Gauging Importance – Naïve Method

- If "*a lot of pages*" have a direct link to some page, P, then it may appear that this makes page P "*important*".
- So a naïve ranking method would be: in the selection of pages $\{p_1, p_2, \ldots, p_n\}$ *count* the *number of pages* that *link* to p_k (from within this set) and assign this total as "*the score for* p_k".
- The problem with this approach is that it *ignores the source of links*: a link from www.google.com is assigned *exactly the same weight* as one from www.csc.liv.ac.uk/~ped/ .
- A more "accurate" scheme would assign scores based on the *importance of sources*.

Improved scoring method

- Define the score for page p_k in terms of the scores of $\{q_1, q_2, \dots, q_t\}$ where each q_i has a direct link to p_k.
- Suppose we use r_k to denote "*the final score assigned to* p_k".
- This suggests,

$$r_k = \sum_{p_i : \langle p_i, p_k \rangle \text{ is a link}} r_i$$

- but this doesn't "*distribute*" the score of each page. Instead:

$$r_k = \sum_{p_i : \langle p_i, p_k \rangle \text{ is a link}} \frac{r_i}{\left|\{p_j : \langle p_i, p_j \rangle \text{ is a link}\}\right|}$$

Modelling by Matrices

- Although the expression on the previous slide looks very involved we can describe things simply by using *matrices*.
- $\{p_1, p_2, \ldots, p_n\}$: set of pages (*graph nodes*).
- $\{\langle p_i, p_j \rangle : \ p_i \ links\ to\ p_j\}$: set of links (*graph edges*)
- r_k : the *score* for page p_k that we want to compute.
- t_i : the *number* of links *out* of page p_i.

$$r_k = \sum_{\langle p_i, p_k \rangle} \frac{r_i}{t_i}$$

Summing up

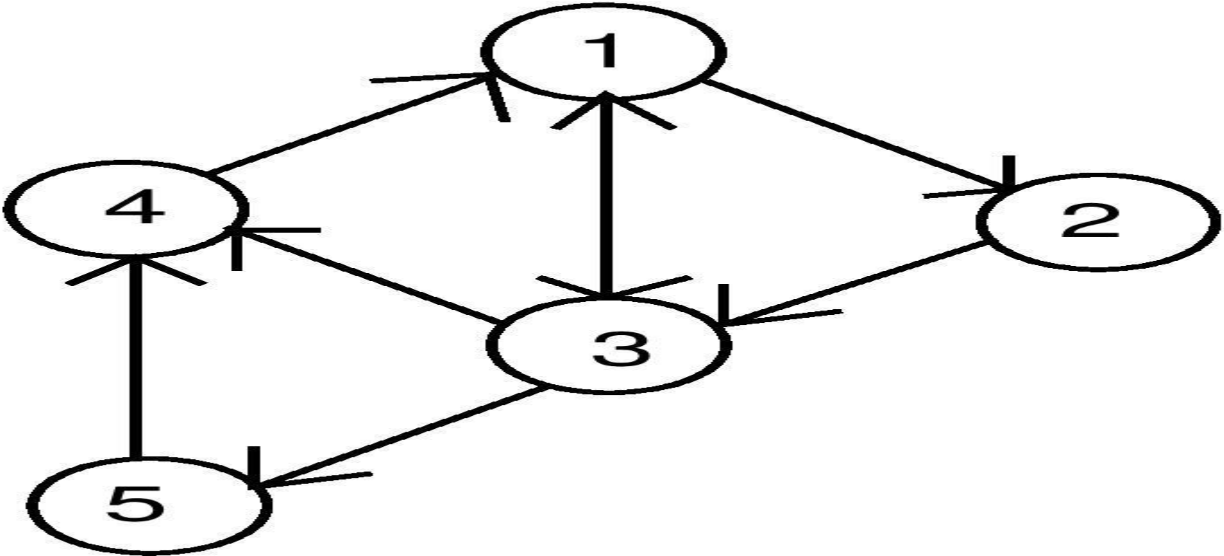
- The "score vector" $\underline{r} = \langle r_1, r_2, \dots, r_n \rangle$ must satisfy:

$$W \cdot \underline{r}^T = \underline{r}^T$$

where W is the $n \times n$ matrix with

$$w_{ij} = \begin{cases} 0 & if\ \langle p_j, p_i \rangle\ is \notin link \\ \dfrac{1}{t_j} & if\ \langle p_j, p_i \rangle\ is \in link \end{cases}$$

An Example



The “weight matrix” for this

$$W = \begin{pmatrix} 0 & 0 & 1/3 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1 \\ 0 & 0 & 1/3 & 0 & 0 \end{pmatrix}$$

- Notice that *all of the columns* add up to 1.

The connection with Spectra

- We are looking for a *score vector* that satisfies $W \cdot \underline{r}^T = \underline{r}^T$
- In other words "*the score vector is an eigenvector of W for an eigenvalue of 1*".
- It can be shown that 1 is *always* an eigenvalue of "*column stochastic*" matrices: those whose columns sum to 1.
- For "*suitable*" graphs this eigenvalue is dominant.
- This means that the "*score vector*" is effectively *unique*.

The example 5 page web

- For our example we find a score vector:

$$\underline{r} = \left\langle 1, \quad \frac{1}{2}, \quad 1, \quad \frac{2}{3}, \quad \frac{1}{3} \right\rangle$$

- Pages 1 and 3 are the "*highest ranked*".
- Page 5 is the "*lowest*".

Some Complications

- A web page that has no outgoing links is called a "*dangling page*": these require *some adjustments* to be made.
- The technical details (some of which are *commercially sensitive*) are outlined on pages 429–434 of the textbook.
- Another issue is *manipulation of outcomes* by altering the *link structures* (see eg page 422).
- Even the *basic* form of Google's method is fairly *robust in minimizing* the effects of *naïve manipulation*. (page 427)

Summary

- Google's use of spectral techniques derives from results dating back to the start of the 20^{th} century.
- Although requiring some adjustments for practical use (dangling pages for example) it offers a strong example of the importance of spectral analysis in Computer Science.
- Similar approaches have evolved for other "*ranking type*" problems many of which are a topic of current research.
- We give another very different example of spectral methods in the next lecture: *Image Compression via SVD.*