Web Mining

- Discovering interesting and useful information from Web content and usage

Web Mining Challenges

- In traditional data mining:“Structured” data; Scale: 10 million already quite big
- In web mining
  - Semi-structured: Html with hierarchies, Web information are linked
  - Scale: 25+ billion web pages
- Web: Contains information of almost anything, High redundancy, Noisy

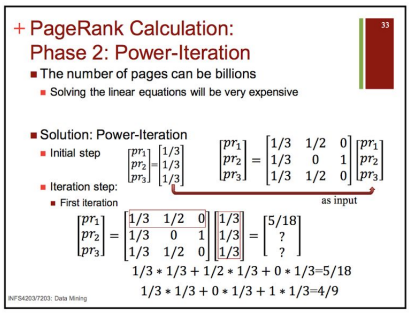
Content Mining

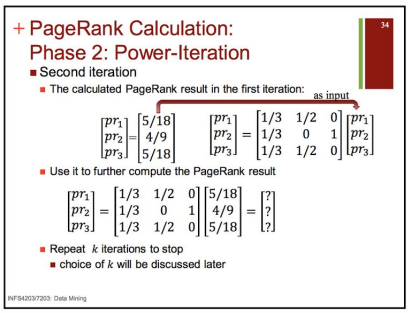
- **Content mining** extends the functionality of basic search engines.
- **Collaborative filtering** identifies preferences based on ratings of similar users (e.g. which pages did they visit).
- Steps:
  - Step0: Handling Missing values
  - Step1: Find similar (cosine similarity) users
  - Step2: Estimate user’s rating (Select Top-N (k) candidates)
  - Step 3: Return the recommendation (return Top-N ratings to users)
    - Estimate rate = Sum (top-N ratings) / N
- **Challenges**:
  - scalability (many users and items),
  - robustness (there will be noise),
  - sparsity (user-item rating matrix is very sparse),
  - and cold start (how to make recommendations to new users)

Structure Mining

- Mines the structure (links, graph) of the Web.
  - One technique is **PageRank** by Google.
- Motivation of Web structure mining:
  - Was only based on relevance of query
  - Hyperlinks(importance,similar web)
  - Applications(communities discover)
- Algorithms:
  - PageRank (Ranking web pages used by Google)
    - Importance: calculated numerically from the number of pages that point to it (backlinks)
    - Weighting is used to provide even more importance to those backlinks that come from other important pages.
    - Phase 1: Matrix Formulation (portion of page point to it)

- Be aware Matrix columns (Pages 0 -> n) and row(Pages 0 -> m)

- Phase 2: Power-Iteration (Repeat K iterations)
  - **Check** if NOT Spider Trap first. (No probability needed)
    - 

    - 

  - **Check** if is Spider Trap first.
    - Spider Trap: if there are no links from within the group to outside the group
    - Spider Trap Solution:
    - Old: Randomly pick a link in the page and visit that linking page
    -
      - Probability Matrix M
        - $M(i,j)$:probability to visit page P_i if we are currently at page P_j

$$\begin{bmatrix} pr_1 \\ pr_2 \\ \dots \\ pr_N \end{bmatrix} = \alpha M \begin{bmatrix} pr_1 \\ pr_2 \\ \dots \\ pr_N \end{bmatrix} + (1-\alpha) \begin{bmatrix} 1/N \\ 1/N \\ \dots \\ 1/N \end{bmatrix}$$

- How to choose K iterations?
  - Maximum error is 0.85^t

Choosing the number of iterations: Analysis

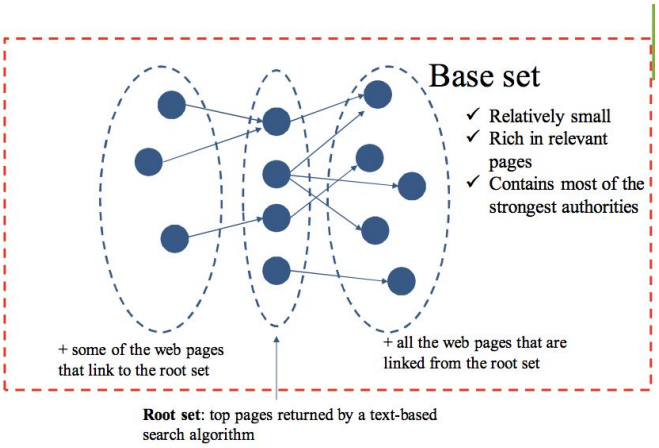
- Let $err(t) = \sum_{i=1}^{N} |pr_i(t) - pr_i|$ denotes the error in the t −th iteration
  - Property 1: $err(t) \leq \alpha \cdot err(t-1)$
  - Property 2: $err(0) \leq 1$
  - Question: If $\alpha = 0.85$, and we need to guarantee the error is no more than 0.0001, how to choose the number t of iterations?
    - $err(t) \leq err(t-1) \cdot \alpha \leq \cdots \leq err(0) \cdot \alpha^t \leq \alpha^t$
    - $err(t) \leq 0.85^t \leq 0.0001$
    - $t \geq 57$

$pr_i(t)$: the calculated PageRank of page P_i in the t-th iteration;
pr_i: the exact PageRank of page P_i.

➢ HITS (Hyperlink-induced Topic Search)
- **2 Scores** for each page:
  - Hub value: the value of its links to other pages
  - Authority value: the value of the content of the page

- Pages with highest authority values and hub values are the results of interest.

- **Sampling component**



- **weight propagation component**
- Use count frequency to construct Matrix **A^T**
- **A^T** has same column and row as PageRank

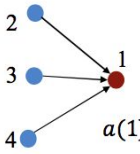
$$\begin{bmatrix} P1(p1) & P2(p2) & P3(p3) \\ P2(p1) & P2(p2) & P2(p3) \\ P3(p1) & P3(p2) & P3(p3) \end{bmatrix}$$

$P1....Pn : Column$

$p1....pn : Row$

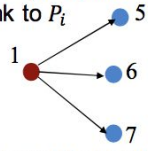
- I: Authority A^T
- O: Hub A

- Initially all hub values and authority values are 1.
- Iterative approach
  - I step:



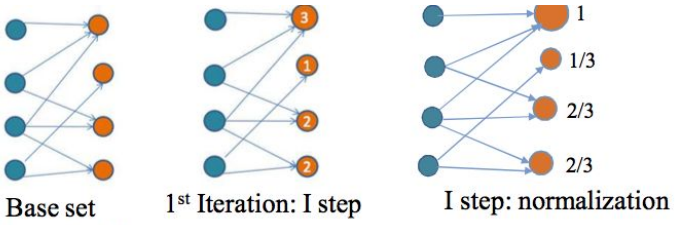
$a(1) = h(2) + h(3) + h(4)$

    - Authority value of page P_i is updated to the sum of the hub values of the pages link to P_i
  - O Step:



$h(1) = a(5) + a(6) + a(7)$

    - Hub value of page P_i is updated to the sum of the authority values of the pages P_i links to

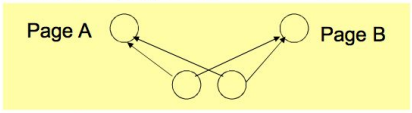


Base set 1st Iteration: I step I step: normalization

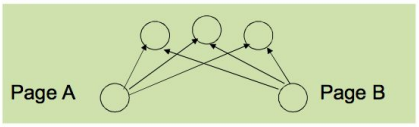
- Normalization
  - Scale a_i so that the maximum becomes 1 after I step
  - Scale h_i so that the maximum becomes 1 after O step
- Stopping condition
  - Repeat the I-step and O-step k times
  - When k is large enough, it converges (Proof of convergence is not required).

➢ Algorithms in web-community detection

- **Co-citation**: the similarity of A and B is measured by the number of pages cite both A and B.



- **Bibliographic coupling**: the similarity of A and B is measured by the number of pages cited by both A and B.



HITS Advantages Vs. Disadvantages

Advantages	Disadvantages
• Rank pages according to the query topic	• Does not have anti-spam capability: One may add out-links to his own page that points to many good authorities • Topic-drift: One may collect many pages that have nothing to do with the topic — by just pointing to them • Query-time evaluation: expensive

HITS Vs. PageRank

HITS	PageRank
• executed at query time	• Precomputed • Commonly used in search engine

• Not-commonly used in search engine • 2 scores for each page • Process on small subset of relevant doc	• 1 score for each page • Rank all web pages
Both are iterative algorithms based on the link structure of the Web	

Usage Mining

- Base upon how the Web is used, Predict user's actions
- OutComes:
  - Association rules – Find pages that are often viewed together
  - Clustering – Cluster users based on browsing patterns
  - Classification – Relate user attributes to patterns
- Data can be from clickstreams, user sessions, or a server session.
- To keep track of this data, use a log. But first, must cleanse and sessionize the data.
- **Pre-processing**
  - Web logs are raw data
    - Click stream: a sequential series of page view request
    - User session: a delimited set of user clicks (click stream) across one or more Web servers.
    - Server session (visit): a collection of user clicks to a single Web server during a user session.
  - Web log cleansing
    - Replace source IP address with unique but non-identifying ID
    - Replace exact URL of pages referenced with unique but non-identifying ID
    - Delete error records and records containing nonpage data (such as figures and code).
  - Sessionization
    - Identify consecutive page references from a IP address
      - occurring within a predefined time interval (e.g., 25 minutes).
      - Where the interclick time is less than a predefined threshold
- **Pattern Discovery**
  - Using statistical analysis, association rules, clustering,classification, sequential pattern, dependency modelling

- Example 1: use association rule algorithms to find pages frequently visited together:
  - Item set: visited pages
  - Help better website design
- Example 2: use statistical analysis to find user visit peaks
  - Use historical web logs to derive visiting histogram
  - Help better service
  - When need more machines

❖ **Issues:**

➢ identification of exact user not possible, single session isn't well defined

➢ exact sequence of pages references is unavailable due to caching of web pages. There is also privacy and legal issues.