Mining Web Data

Lijun Zhang
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

Outline

- **Introduction**
- Web Crawling and Resource Discovery
- Search Engine Indexing and Query Processing
- Ranking Algorithms
- Recommender Systems
- Summary

Introduction

- Web is an unique phenomenon
  - The scale, the distributed and uncoordinated nature of its creation, the openness of the underlying platform, and the diversity of applications
- Two Primary Types of Data
  - Web content information
    - ✓ Document data, Linkage data (Graph)
  - Web usage data
    - ✓ Web transactions, ratings, and user feedback, Web logs

Applications on the Web

- Content-Centric Applications
  - Data mining applications
    - Cluster or classify web documents
  - Web crawling and resource discovery
  - Web search
    - Linkage and content
  - Web linkage mining
- Usage-Centric Applications
  - Recommender systems
  - Web log analysis
    - Anomalous patterns, and Web site design

Outline

- Introduction
- **Web Crawling and Resource Discovery**
- Search Engine Indexing and Query Processing
- Ranking Algorithms
- Recommender Systems
- Summary

Web Crawling

- Web Crawlers or Spiders or Robots
- Motivations
  - Resources on the Web are dispensed widely across globally distributed sites
  - Sometimes, it is necessary to download all the relevant pages at a central location
- Universal Crawlers
  - Crawl all pages on the Web (Google, Bing)
- Preferential Crawlers
  - Crawl pages related to a particular subject or belong to a particular site

Crawler Algorithms

- A real crawler algorithm is complex
  - A selection Algorithm, Parsing, Distributed, multi-threads
- A Basic Crawler Algorithm

Algorithm *BasicCrawler*(Seed URLs: S, Selection Algorithm: $\mathcal{A}$)

begin

$FrontierList = S$;

repeat

Use algorithm $\mathcal{A}$ to select URL $X \in FrontierSet$;

$FrontierList = FrontierList - \{X\}$;

Fetch URL X and add to repository;

Add all relevant URLs in fetched document X to end of $FrontierList$;

until termination criterion;

end

Selection Algorithms

- Breadth-first
- Depth-first

- Frequency-Based
  - Most universal crawlers are incremental crawlers that are intended to refresh previous crawls
- PageRank-Based
  - Choose Web pages with high PageRank

Combatting Spider Traps

- The crawling algorithm maintains a list of previously visited URLs for comparison purposes
  - So, it always visits distinct Web pages
- However, many sites create dynamic URLs
  - http://www.examplesite.com/page1
  - http://www.examplesite.com/page1/page2
  - Limit the maximum size of the URL
  - Limit the number of URLs from a site

Outline

- Introduction
- Web Crawling and Resource Discovery
- **Search Engine Indexing and Query Processing**
- Ranking Algorithms
- Recommender Systems
- Summary

The Process of Search

- Offline Stage
  - The search engine preprocesses the crawled documents to extract the tokens and constructs an index
  - A quality-based ranking score is also computed for each page

- Online Query Processing
  - The relevant documents are accessed and then ranked using both their relevance to the query and their quality

Offline Stage

- The Preprocessing Steps
  - The relevant tokens are extracted and stemmed
  - Stop words are removed
- Construct the Inverted Index
  - Maps each word identifier to a list of document identifiers containing it
    - Document ID, Frequency, Position
- Construct the Vocabulary Index
  - Access the storage location of the inverted word

Ranking (1)

- Content-Based Score
  - A word is given different weights, depending upon whether it occurs in the title, body, URL token, or the anchor text
  - The number of occurrences of a keyword in a document will be used in the score
  - The prominence of a term in font size and color may be leveraged for scoring
  - When multiple keywords are specified, their relative positions in the documents are used as well

Ranking (2)

- Limitations of Content-Based Score
  - It does not account for the reputation, or the quality, of the page
    - ✓ A user may publish incorrect material
  - Web Spam
    - ✓ Content-spamming: The Web host owner fills up repeated keywords in the hosted Web page
    - ✓ Cloaking: The Web site serves different content to crawlers than it does to users
  - Search Engine Optimization (SEO)
    - ✓ The Web set owners attempt to optimize search results by using their knowledge

Ranking (3)

- Reputation-Based Score
  - Page citation mechanisms: When a page is of high quality, many other Web pages point to it
  - User feedback or behavioral analysis mechanisms: When a user chooses a Web page, this is clear evidence of the relevance of that page to the user
- The Final Ranking Score

$$RankScore = f(IRScore, RepScore).$$

  - Spams always exist

Outline

- Introduction
- Web Crawling and Resource Discovery
- Search Engine Indexing and Query Processing
- **Ranking Algorithms**
- Recommender Systems
- Summary

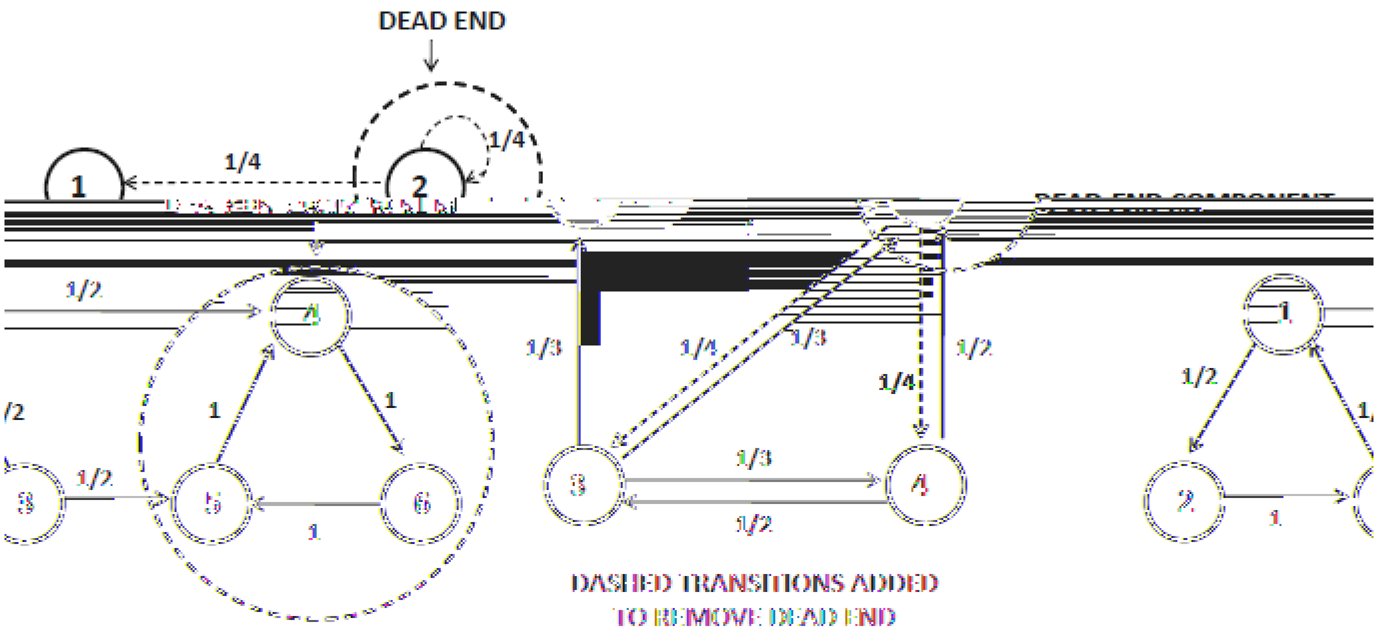
Google's PageRank (1)

- Random Walk Model
  - A random surfer who visits random pages on the Web by selecting random links on a page

1. The long-term relative frequency of visits to any particular page is clearly influenced by the number of in-linking pages to it
2. The long-term frequency of visits to any page will be higher if it is linked to by other frequently visited pages

Google’s PageRank (2)

- Random Walk Model
  - Dead ends: pages with no outgoing links
  - Dead-end component



ead-end component (a) Dead-end node (b) De

Google's PageRank (3)

Random Walk Model

- Dead ends: pages with no outgoing links
  - ✓ Add links from the dead-end node (Web page) to all nodes (Web pages), including a self-loop to itself

- Dead-end component
  - ✓ A teleportation (restart) step: The random surfer may either jump to an arbitrary page with probability | , or it may follow one of the links on the page with probability 1 |

Steady-state Probabilities (1)

Steady-state Probabilities (2)

- The probability of a teleportation into i

|

—

ὲ

- The probability of a transition into i

(1 |) “(Ω) ὴ سش

ش ثص س

- Then, we have

$$\pi(i) = \alpha/n + (1-\alpha)\cdot \sum_{j\in In(i)} \pi(j)\cdot p_{ji}$$

Steady-state Probabilities (3)

- Let $\bar{\pi} = [\pi(1), \ldots, \pi(n)]^T$

$$\bar{e}/n = (1-\alpha)(1 - \pi\bar{\pi})P^T = \bar{\alpha}$$

  - With the constraint $\sum_{i=1}^{n} \pi(i) = 1$
- Optimization
  - $\bar{\pi}^{0} = \frac{\bar{e}}{n}$
  - $\bar{\pi}^{k+1} = \frac{\alpha\bar{e}}{n} + (1-\alpha)P^T\bar{\pi}^{k}$
  - $\bar{\pi}^{k+1} \leftarrow \frac{\bar{\pi}^{k+1}}{|\bar{\pi}^{k+1}|_1}$

Outline

- Introduction
- Web Crawling and Resource Discovery
- Search Engine Indexing and Query Processing
- Ranking Algorithms
- **Recommender Systems**
- Summary

Recommender Systems

- Data About User Buying Behaviors
  - User profiles, interests, browsing behavior, buying behavior, and ratings about various items

- The Goal
  - Leverage such data to make recommendations to customers about possible buying interests

Utility Matrix (1)

- For n users and d items, there is an $n \times d$ matrix D of utility values
  - The utility value for a user-item pair could correspond to either the buying behavior or the ratings of the user for the item

  - Typically, a small subset of the utility values are specified

Utility Matrix (2)

- For n users and d items, there is an $n \times d$ matrix D of utility values
  - Positive preferences only
    - ✓ A specification of a "like" option on a social networking site, the browsing of an item at an online site, the buying of a specified quantity of an item, or the raw quantities of the item bought by each user
  - Positive and negative preferences (ratings)
    - ✓ The user specifies the ratings that represent their like or dislike for the item

Utility Matrix (3)

- For n users and d items, there is an $n \times d$ matrix D of utility values

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	

[redacted]

Types of Recommendation

- **Content-Based Recommendations**
  - The users and items are both associated with feature-based descriptions
    - ✓ The text of the item description
    - ✓ The interests of user in a profile
- **Collaborative Filtering**
  - Leverage the user preferences in the form of ratings or buying behavior in a "collaborative" way
  - The utility matrix is used to determine either relevant users for specific items, or relevant items for specific users

Content-Based Recommendations (1)

- User is associated with some documents that describe his/her interests
  - Specified demographic profile
  - Specified interests at registration time
  - Descriptions of the items bought
- The items are also associated with textual descriptions

1. If no utility matrix is available
   - k-nearest neighbor approach: find the top-k items that are closest to the user
     - ✓ The cosine similarity with tf-idf can be used

Content-Based Recommendations (1)

- User is associated with some documents that describe his/her interests
  - Specified demographic profile
  - Specified interests at registration time
  - Descriptions of the items bought
- The items are also associated with textual descriptions

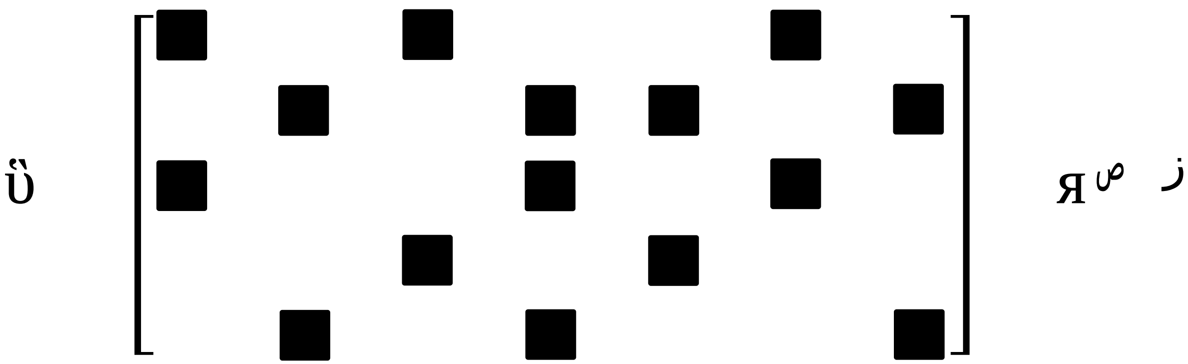
1. If no utility matrix is available
   -

Content-Based Recommendations (2)

2. If a utility matrix is available
   - Classification-Based Approach
     - ✓ Training documents representing the descriptions of the items for which that user has specified utilities
     - ✓ The labels represent the utility values.
     - ✓ The descriptions of the remaining items for that user can be viewed as the test documents
   - Regression-Based Approach
- Limitations
   - Depends on the quality of features

Collaborative Filtering

- Missing-value Estimation or Matrix Completion



  - The Matrix is extremely large
  - The Matrix is extremely sparse

Algorithms for Collaborative Filtering

- Neighborhood-Based Methods for Collaborative Filtering
  - **User-Based Similarity with Ratings**
  - Item-Based Similarity with Ratings
- Graph-Based Methods
- Clustering Methods
  - **Adapting ▓-Means Clustering**
  - Adapting Co-Clustering
- Latent Factor Models
  - Singular Value Decomposition
  - Matrix Factorization
  - **Matrix Completion**

User-Based Similarity with Ratings

- A Similarity Function between Users
  - $\overline{X} = x_1, \ldots, x_s$ and $\overline{Y} = y_1, \ldots, y_s$ be the common ratings between a pair of users
  - The Pearson correlation coefficient

$$\text{Pearson}(\overline{X}, \overline{Y}) = \frac{\sum_{i=1}^{s} (x_i - \hat{x}) \cdot (y_i - \hat{y})}{\sqrt{\sum_{i=1}^{s} (x_i - \hat{x})^2} \cdot \sqrt{\sum_{i=1}^{s} (y_i - \hat{y})^2}}$$

    - ✓ $\hat{x} = \sum_{i=1}^{s} x_i / s$ and $\hat{y} = \sum_{i=1}^{s} y_i / s$

1. Identify the peer group of the target user
   - Top-k users with the highest Pearson coefficient
2. Return the weighted average ratings of each of the items of this peer group
   - Normalization is needed

Clustering Methods (1)

- Motivations
  - Reduce the computational cost
  - Address the issue of data sparsity to some extent
- The Result of Clustering
  - Clusters of users
    - ✓ User-user similarity recommendations
  - Clusters of items
    - ✓ Item-item similarity recommendations

Clustering Methods (2)

- User-User Recommendation Approach
  1. Cluster all the users into $n_{\text{س}}$ groups of users using any clustering algorithm
  2. For any user i, compute the average

Adapting k-Means Clustering

1. In an iteration of k-means, centroids are computed by averaging each dimension over the number of specified values in the cluster members
   - Furthermore, the centroid itself may not be fully specified
2. The distance between a data point and a centroid is computed only over the specified dimensions in both
   - Furthermore, the distance is divided by the number of such dimensions in order to fairly compare different data points

Latent Factor Models

- The Key Idea
  - Summarize the correlations across rows and columns in the form of lower dimensional vectors, or latent factors
  - These latent factors become hidden variables that encode the correlations in the data matrix and can be used to make predictions
  - Estimation of the k-dimensional dominant latent factors is often possible even from incompletely specified data

Modeling

- The n users are represented by n factors: $\overline{U}_{\text{پ}}, \ldots, \overline{U}_{\text{ص}} \in \mathbb{R}^{\text{ش}}$
- The d items are represented by d factors: $\overline{I}_{\text{پ}}, \ldots, \overline{I}_{\text{ز}} \in \mathbb{R}^{\text{ش}}$
- The rating $r_{\text{سش}}$ for user i and item j

ì$_{\text{سش}}$ ⟨Ÿ$_{\text{س}}$, Ő$_{\text{ش}}$⟩ Ÿ$_{\text{س}}$ Ő$_{\text{ش}}$ Ő$_{\text{ش}}$ Ÿ$_{\text{س}}$

- The rating matrix $D = [r_{\text{سش}}]_{\text{ص} \quad \text{ز}}$

’O ”O$_{\text{ضصضزض}}$”O$_{\text{شزضس}}$

  - $F_{\text{ضصضزض}} \in \mathbb{R}^{\text{ص} \quad \text{ش}}$ and $F_{\text{شزضس}} \in \mathbb{R}^{\text{ز} \quad \text{ش}}$

Matrix Factorization (MF)

- The Goal

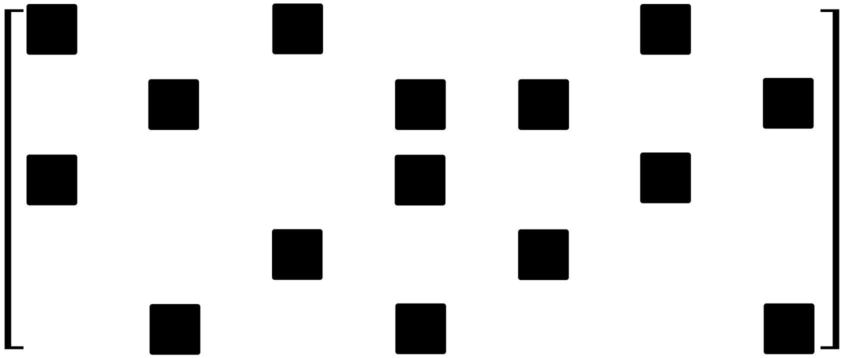
- The objective when D is fully observed

- The objective when D is partially observed

  - Ω is the set of observed indices
  - Constrains can be added: $U \geq 0$ and $V \geq$

Matrix Completion

- Assuming the Utility matrix is low-rank

$M =$  $\in \mathbb{R}^{m \times n}$

- The Optimization Problem

$$\min_{X \in \mathbb{R}^{m \times n}} \ \operatorname{rank}(X) \quad \text{s.t.} \quad X_{ij} = M_{ij}, \ (i,j) \in \Omega \quad \Rightarrow \quad \min_{X \in \mathbb{R}^{m \times n}} \ \|X\|_* \quad \text{s.t.} \quad X_{ij} = M_{ij}, \ (i,j) \in \Omega$$

  - Ω is the set of observed indices

Outline

- Introduction
- Web Crawling and Resource Discovery
- Search Engine Indexing and Query Processing
- Ranking Algorithms
- Recommender Systems
- **Summary**

Summary

- Web Crawling and Resource Discovery
  - Universal, Preferential, Spider Traps
- Search Engine Indexing and Query Processing
  - Content-based score, reputation-based scores
- Ranking Algorithms
  - PageRank and its variants, HITS
- Recommender Systems
  - Content-Based, Collaborative Filtering