Mining di Dati Web

CORSO DI BIG DATA – MODULO ANALISI PER I BIG DATA

a.a. 2023/2024

Prof. Roberto Pirrone

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

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

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

Struttura dati usata per indicizzare dati appartenenti ad una rappresentazione sparsa di tipo insiemistico
(ad es. gli itemset nel mining di pattern frequenti)

- All'elemento viene associato un id generato tramite una funzione hash
- si crea una lista di id ognuno dei quali punta alla lista dei set che contengono l'elemento

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

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

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

Viene esplicitamente definita una "probabilità di transizione" da un nodo verso gli altri a questo connessi ovvero una probabilità di inseguire un determinato link

Google's PageRank (1)

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

Viene esplicitamente definita una "probabilità di transizione" da un nodo verso gli altri a questo connessi ovvero una probabilità di inseguire un determinato link

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

Il processo di definizione di queste frequenze a lungo termine di visita di una pagina è ottenuto tramite una "Catena di Markov" in cui le frequenze rappresentano le cosiddette "probabilità di stato stabile" di ciascun nodo

Catene di Markov

- Si consideri un sistema che transita da uno stato ad un altro, all'interno di uno spazio degli stati $S = \{i, j, k, \ldots\}$, con una certa *probabilità di transizione* p_{ij}

$$p_{ij} \in [0,1], \quad \sum_j p_{ij} = 1 \ \forall i$$

Catene di Markov

- Si consideri un sistema che transita da uno stato ad un altro, all'interno di uno spazio degli stati $S = \{i, j, k, \dots\}$, con una certa *probabilità di transizione* p_{ij}

$$p_{ij} \in [0,1], \quad \sum_j p_{ij} = 1 \ \forall i$$

- La variabile X_n che all'istante di tempo discreto n, contiene il valore dello stato i-esimo è una variabile aleatoria la cui probabilità è:

$$p_i^{(n)} = P\left(X_n = i\right)$$

Catene di Markov

- Si consideri un sistema che transita da uno stato ad un altro, all'interno di uno spazio degli stati $S = \{i, j, k, \dots\}$, con una certa *probabilità di transizione* p_{ij}

$$p_{ij} \in [0,1], \quad \sum_j p_{ij} = 1 \ \forall i$$

- La variabile X_n che all'istante di tempo discreto n, contiene il valore dello stato i-esimo è una variabile aleatoria la cui probabilità è:

$$p_i^{(n)} = P\left(X_n = i\right)$$

- L'evoluzione nel tempo di $X_o, X_1, \dots, X_n, \dots$ è un *processo stocastico*

Catene di Markov

- In generale l'evoluzione del processo stocastico è regolata come segue:

$$
\begin{aligned}
& P\left(X_0 = i_0, \ldots, X_{n+1} = i_{n+1}\right) \\
= & P\left(X_{n+1} = i_{n+1} \mid X_0 = i_0, \ldots, X_n = i_n\right) \cdot P\left(X_0 = i_0, \ldots, X_n = i_n\right)
\end{aligned}
$$

Catene di Markov

- In generale l'evoluzione del processo stocastico è regolata come segue:

$$
\begin{aligned}
& P\left(X_0=i_0, \ldots, X_{n+1}=i_{n+1}\right) \\
= & P\left(X_{n+1}=i_{n+1} \mid X_0=i_0, \ldots, X_n=i_n\right) \cdot P\left(X_0=i_0, \ldots, X_n=i_n\right)
\end{aligned}
$$

- Un processo stocastico è detto *processo di Markov* (o *catena di Markov*) se:

$$
P\left(X_{n+1}=i_{n+1} \mid X_0=i_0, \ldots, X_n=i_n\right)=P\left(X_{n+1}=i_{n+1} \mid X_n=i_n\right)
$$

Catene di Markov

- In generale l'evoluzione del processo stocastico è regolata come segue:

$$
\begin{aligned}
& P\left(X_0 = i_0, \ldots, X_{n+1} = i_{n+1}\right) \\
= & P\left(X_{n+1} = i_{n+1} \mid X_0 = i_0, \ldots, X_n = i_n\right) \cdot P\left(X_0 = i_0, \ldots, X_n = i_n\right)
\end{aligned}
$$

- Un processo stocastico è detto *processo di Markov* (o *catena di Markov*) se:

$$
P\left(X_{n+1} = i_{n+1} \mid X_0 = i_0, \ldots, X_n = i_n\right) = P\left(X_{n+1} = i_{n+1} \mid X_n = i_n\right)
$$

- Infine la catena di Markov è *omogenea* se:

$$
p_{ij} = P\left(X_{n+1} = j \mid X_n = i\right)
$$

non dipendono da *n*, ma solo dagli stati *i* e *j*.

Catene di Markov

- In una catena di Markov omogenea con probabilità di transizione p_{ij} e con distribuzione iniziale $p_i^{(0)} = P\left(X_0 = i\right)$:

$$P\left(X_0 = i_0, \ldots, X_n = i_n\right) = p_i^{(0)} \cdot p_{i_0 i_1} \cdots p_{i_{n-1} i_n}, \ldots$$

Catene di Markov

- In una catena di Markov omogenea con probabilità di transizione p_{ij} e con distribuzione iniziale $p_i^{(0)} = P\left(X_0 = i\right)$:

$$P\left(X_0 = i_0, \ldots, X_n = i_n\right) = p_i^{(0)} \cdot p_{i_0 i_1} \cdots p_{i_{n-1} i_n}, \ldots$$

  - Si dimostra ricorsivamente poiché, ad es:
    P(*X*, *Y*, *Z*, *K*) = *P*(*X* | *Y*, *Z*, *K*)*P*(Y, Z, *K*) =
    = *P*(*X* | *Y*, *Z*, *K*)*P*(Y | Z, *K*) *P*(Z, *K*) = /* usiamo la catena Markoviana */
    = *P*(*X* | *Y*) *P*(Y | Z) *P*(Z | K) *P*(K)

Catene di Markov

- In una catena di Markov omogenea con probabilità di transizione p_{ij} e con distribuzione iniziale $p_i^{(0)} = P\left(X_0 = i\right)$:

$$p_i^{(n)} = \sum_{i_0, i_1, \ldots, i_{n-1}} p_i^{(0)} \cdot p_{i_0 i_1} \cdots p_{i_{n-1} i}$$

  - Si dimostra ricordando che tutti gli eventi $X_k = j$ sono mutuamente esclusivi all'istante k per cui le probabilità di raggiungere lo stato i_n al tempo n è la somma di tutte le probabilità che il sistema si sia evoluto lungo una qualunque delle possibili combinazioni di stati da i_0 a i_n.

Catene di Markov

- Una catena di Markov può essere rappresentata tramite un *grafo* $G = (S, P)$ in cui gli stati rappresentano i nodi e la *matrice di transizione* $P = [p_{ij}]$ contiene le probabilità di transizione che rappresentano gli archi.

Catene di Markov

- Una catena di Markov può essere rappresentata tramite un *grafo* $G = (S, P)$ in cui gli stati rappresentano i nodi e la *matrice di transizione* $P = [p_{ij}]$ contiene le probabilità di transizione che rappresentano gli archi.

- Valgono le seguenti proprietà:

$$(P \cdot P)_{ij} = \sum_{i_1} p_{i,i_1} p_{i_1 j} \quad (P^n)_{ij} = \sum_{i_1,\ldots,i_{n-1}} p_{ii_1} \cdots p_{i_{n-1} j}$$

Catene di Markov

- Una catena di Markov può essere rappresentata tramite un *grafo* $G = (S, P)$ in cui gli stati rappresentano i nodi e la *matrice di transizione* $P = [p_{ij}]$ contiene le probabilità di transizione che rappresentano gli archi.

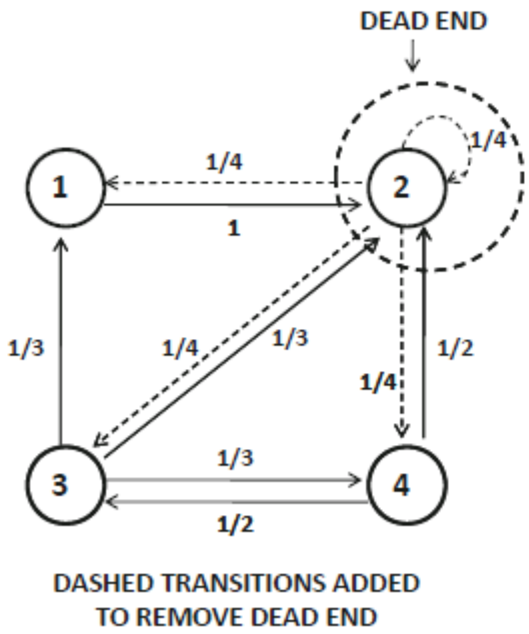
- Valgono le seguenti proprietà:

$$(P \cdot P)_{ij} = \sum_{i_1} p_{i,i_1} p_{i_1 j} \qquad (P^n)_{ij} = \sum_{i_1,\ldots,i_{n-1}} p_{ii_1} \cdots p_{i_{n-1}j}$$

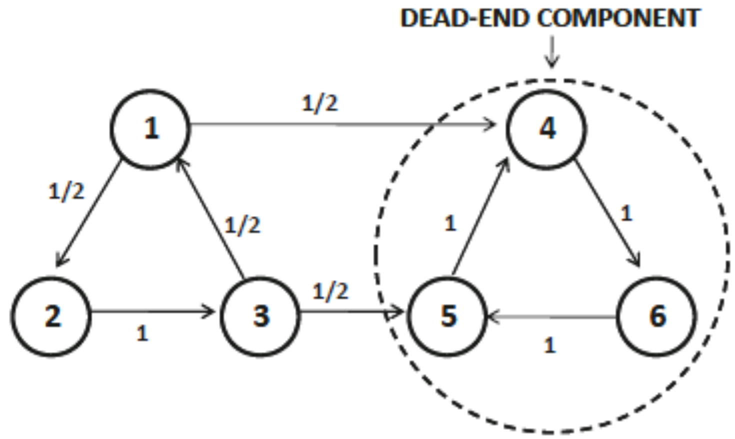
- In una catena di Markov omogenea: $p_j^{(n)} = \sum_i p_i^{(0)} \cdot (P^n)_{ij}$

Google's PageRank (2)

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



(a) Dead-end node



(b) Dead-end component

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 α, or it may follow one of the links on the page with probability $1-\alpha$

Steady-state Probabilities (1)

- $G = (N, A)$ be the directed Web graph
  - Nodes correspond to pages
  - Edges correspond to hyperlinks
    - ✓ Include added edges for dead-end nodes

Steady-state Probabilities (1)

- $G = (N, A)$ be the directed Web graph
  - Nodes correspond to pages
  - Edges correspond to hyperlinks
    - ✓ Include added edges for dead-end nodes
  - $\pi(i)$: the steady-state probability at i

Steady-state Probabilities (1)

- $G = (N, A)$ be the directed Web graph
  - Nodes correspond to pages
  - Edges correspond to hyperlinks
    - ✓ Include added edges for dead-end nodes
  - $\pi(i)$: the steady-state probability at i
  - $In(i)$: set of nodes incident on i

Steady-state Probabilities (1)

- $G = (N, A)$ be the directed Web graph
  - Nodes correspond to pages
  - Edges correspond to hyperlinks
    - ✓ Include added edges for dead-end nodes
  - $\pi(i)$: the steady-state probability at i
  - $In(i)$: set of nodes incident on i
  - $Out(i)$: the set of end points of the outgoing links of node i

Steady-state Probabilities (1)

- $G = (N, A)$ be the directed Web graph
  - Nodes correspond to pages
  - Edges correspond to hyperlinks
    - ✓ Include added edges for dead-end nodes
  - $\pi(i)$: the steady-state probability at i
  - $In(i)$: set of nodes incident on i
  - $Out(i)$: the set of end points of the outgoing links of node i
  - Transition matrix P of the Markov chain

$$p_{ij} = \frac{1}{|Out(i)|} \quad \text{if there is an edge form } i \text{ to } j$$

Steady-state Probabilities (2)

- The probability of a teleportation into i

$$\frac{\alpha}{n}$$

Steady-state Probabilities (2)

- The probability of a teleportation into i

$$\frac{\alpha}{n}$$

- The probability of a transition into i

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

Steady-state Probabilities (2)

- The probability of a teleportation into i

$$\frac{\alpha}{n}$$

- The probability of a transition into i

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

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

Steady-state Probabilities (3)

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

$$\bar{\pi} = \alpha\bar{e}/n + (1-\alpha)P^T\bar{\pi} \qquad \bar{e} = [1_1, \dots, 1_n]^T$$

  - With the constraint $\sum_{i=1}^{n} \pi(i) = 1$

Steady-state Probabilities (3)

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

$$\overline{\pi} = \alpha\overline{e}/n + (1-\alpha)P^T\overline{\pi} \qquad \overline{e} = [1_1, \dots, 1_n]^T$$

  - With the constraint $\sum_{i=1}^{n} \pi(i) = 1$
- Optimization
  - $\bar{\pi}^{(0)} = \frac{\bar{e}}{n}$

Steady-state Probabilities (3)

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

$$\overline{\pi} = \alpha\overline{e}/n + (1-\alpha)P^T\overline{\pi} \qquad \overline{e} = [1_1, \dots, 1_n]^T$$

  - With the constraint $\sum_{i=1}^{n} \pi(i) = 1$

- Optimization
  - $\bar{\pi}^{(0)} = \frac{\bar{e}}{n}$
  - $\bar{\pi}^{(t+1)} = \frac{\alpha\bar{e}}{n} + (1-\alpha)P^\top\bar{\pi}^{(t)}$
  - $\bar{\pi}^{(t+1)} \leftarrow \frac{\bar{\pi}^{(t+1)}}{|\bar{\pi}^{(t+1)}|_1}$

Fino a raggiungere la convergenza

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

La matrice D è sparsa!!!!

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	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ratings-based utility

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Positive-preference utility

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 (2)

2. If a utility matrix is available

- Classification-Based Approach *D contiene like*
  - ✓ 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 *D contiene valori di rating*

Limitations

- Depends on the quality of features

Collaborative Filtering

- Missing-value Estimation or Matrix Completion

$$M = \begin{bmatrix} \blacksquare & & \blacksquare & & & \blacksquare & \\ & \blacksquare & & \blacksquare & \blacksquare & & \blacksquare \\ \blacksquare & & & \blacksquare & & \blacksquare & \\ & & \blacksquare & & \blacksquare & & \\ & \blacksquare & & \blacksquare & & & \blacksquare \end{bmatrix} \in \mathbb{R}^{n \times d}$$

  - 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 k-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
  - $\bar{X} = (x_1, \ldots, x_s)$ and $\bar{Y} = (y_1, \ldots, y_s)$ 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

Item based Similarity with Ratings

- Un item è caratterizzato da un insieme di utenti che lo preferiscono o meno

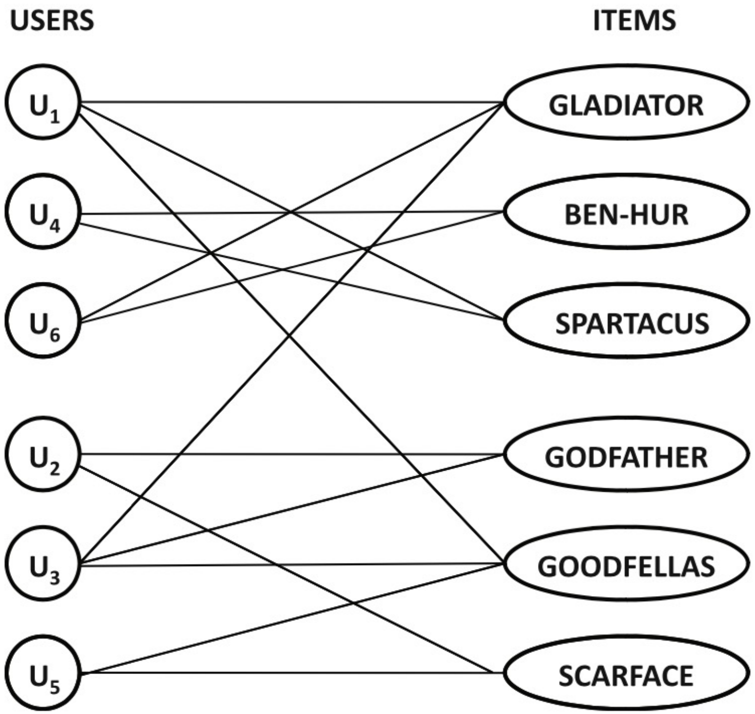
- Si confrontano le *colonne* della matrice M.

- Si usa la *distanza coseno normalizzata*:

$$\text{Cosine}(\bar{U}, \bar{V}) = \frac{\sum_{i=1}^{s} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{s} u_i^2} \cdot \sqrt{\sum_{i=1}^{s} v_i^2}}$$

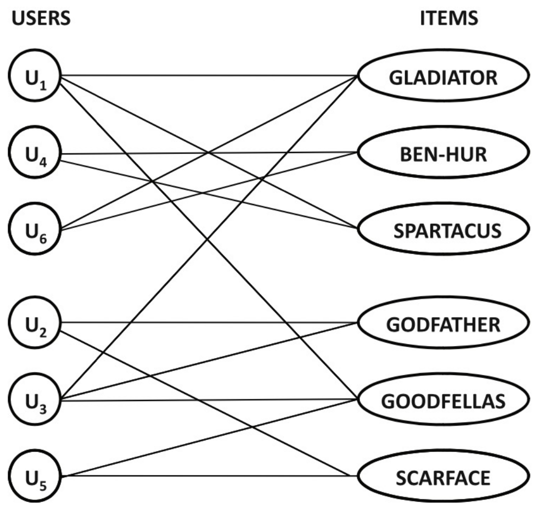
Graph Based Methods

- Si può pensare di costruire il grafo «bipartito» utenti-item, in cui c'è un arco se u_i ha espresso una valutazione su i_j



Graph Based Methods

- PageRank può essere utilizzato per individuare
  - I *k* item con migliore score con un random walk a partire dal nodo u_i
  - I *k* utenti con migliore score con un random walk a partire dal nodo i_j



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_g groups of users using any clustering algorithm
  2. For any user i, compute the average (normalized) rating of the specified items in its cluster
  3. Report these ratings for user i
- Item–Item Recommendation Approach
  1. Cluster all the items into n_g groups of items
  2. The rest is the same as "Item-Based Similarity with Ratings"

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: $\overline{U_1}, \dots, \overline{U_n} \in \mathbb{R}^k$
- The d items are represented by d factors: $\overline{I_1}, \dots, \overline{I_d} \in \mathbb{R}^k$
- The rating r_{ij} for user i and item j

$$r_{ij} \approx \langle \overline{U_i}, \overline{I_j} \rangle = \overline{U_i}^{\top} \overline{I_j} = \overline{I_j}^{\top} \overline{U_i}$$

- The rating matrix $D = [r_{ij}]_{n \times d}$

$$D \approx F_{user} F_{item}^{T}$$

  - $F_{user} \in \mathbb{R}^{n \times k}$ and $F_{item} \in \mathbb{R}^{d \times k}$

Matrix Factorization (MF)

- The Goal

$$D \approx UV^\top$$

- The objective when D is fully observed

$$J = \left\| D - UV^\top \right\|_F^2$$

- The objective when D is partially observed

$$J = \sum_{(i,j)\in\Omega} \left(D_{ij} - \overline{U_i}^\top \overline{V_j} \right)^2$$

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

Matrix Completion

- Assuming the Utility matrix is low-rank

$$M = \begin{bmatrix} \blacksquare & & \blacksquare & & & \blacksquare & \\ & \blacksquare & & \blacksquare & \blacksquare & & \blacksquare \\ \blacksquare & & & \blacksquare & & \blacksquare & \\ & & \blacksquare & & \blacksquare & & \\ & \blacksquare & & \blacksquare & & & \blacksquare \end{bmatrix} \in \mathbb{R}^{n \times d}$$

- The Optimization Problem

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

  - 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