

Collaborative Filtering

- Introduction
- Search or Content based Method
- User-Based Collaborative Filtering
- Item-to-Item Collaborative Filtering
- Using Google's PageRank
- Memory-Based Algorithms (Breese et al, UAI98)

Collaborative Filtering

	Star Wars	Hoop Dreams	Contact	Titanic
Joe	5	2	5	4
John	2	5		3
Al	2	2	4	2
Nathan	5	1	5	?

Recommender systems: Systems that evaluate quality based on the preferences of others with a similar point of view

Collaborative Filtering: *The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given a set of historical preference judgments for a community of users.*

- Predict the opinion the user will have on the different items
- Recommend the 'best' items based on the user's previous likings and the opinions of like-minded users whose ratings are similar

Collaborative Filtering in our life

The screenshot shows a Mozilla Firefox browser window with the title 'data mining - Google Search - Mozilla Firefox'. The address bar contains the URL 'http://www.google.com/search?hl=en&client=firefox-a&rls=org.mozilla%3Aen-US%3Aofficial&hs=PUx&q=data mining' and a search bar with the text 'data mining'. The search results page displays the Google logo, a search bar, and a 'Search' button. Below the search bar, the results are categorized by 'Web' and 'Books'. The main results list includes:

- Data mining** - Wikipedia, the free encyclopedia
Data mining is the process of sorting through large amounts of data and picking out relevant information. It is usually used by business intelligence ...
en.wikipedia.org/wiki/Data_mining - 58k - [Cached](#) - [Similar pages](#)
- Data Mining: What is Data Mining?**
Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into ...
www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm - 13k - [Cached](#) - [Similar pages](#)
- Data Mining: Text Mining, Visualization and Social Media**
Commentary on text mining, data mining, social media and data visualization.

On the right side, there is a 'Sponsored Links' section with the following links:

- Data Analytical Services**
Web-Based Data Mining, Formatting, Merging, and Custom Query Services
www.dataanalyticalservices.com
- Analyze Unstructured Data**
with Business Objects. eMail, Docs, Blogs, & More. Free Whitepaper!
www.BusinessObjects.com
- Open Source Data Mining**

The status bar at the bottom of the browser window shows 'Done'.

Collaborative Filtering in our life

File Edit View History Bookmarks Tools Help

http://www.amazon.com/Principles-Adaptive-Computation-Machine-Learning/dp/026208290X/ref=pd_bbs_sr_1?ie=UTF8&

★★★★☆ (17 customer reviews)

List Price: \$65.00
Price: **\$52.00** & eligible for free shipping with **Amazon Prime**
You Save: \$13.00 (20%)
Availability: In Stock. Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Wednesday, May 7? Order it in the next 13 hours and 56 minutes, and choose **One-Day Shipping** at checkout. [See details](#)

28 used & new available from \$32.00

[See larger image](#)
[Share your own customer images](#)
Publisher: learn how customers can search inside this book.

Are You an Author or Publisher?
[Find out how to publish your own Kindle Books](#)

More Buying Choices
28 used & new from \$32.00
Have one to sell? [Sell yours here](#)

[Add to Wish List](#)
[Add to Shopping List](#)
[Add to Wedding Registry](#)
[Add to Baby Registry](#)
[Tell a friend](#)

Better Together
Buy this book with [The Elements of Statistical Learning](#) by T. Hastie today!
Buy Together Today: \$123.96
[Add both to Cart](#)

Customers Who Bought This Item Also Bought

Page 1 of 10

Data Mining: Practical Machine Learning To... by Ian H. Witten
★★★★☆ (21) \$39.66

Data Mining, Second Edition, by Jiawei Han
★★★★☆ (26) \$51.96

Pattern Recognition and Machine Learning (...) by Christopher M. Bishop
★★★★☆ (34) \$59.96

Introduction to Machine Learning (Adaptive... by Ethem Alpaydin
★★★★☆ (6) \$41.60

Pattern Classification (2nd Edition) by Richard O. Duda
★★★★☆ (26) \$120.00

Search or Content based Method

- Given the user's purchased and rated items, constructs a search query to find other popular items
- For example, same author, artist, director, or similar keywords/subjects
- Impractical to base a query on all the items

User-Based Collaborative Filtering

Some issues with User based collaborative filtering

- Complexity grows linearly with the number of customers and items
- The sparsity of recommendations on the data set
 - Even active customers may have purchased well under 1% of the products

Item-to-Item Collaborative Filtering

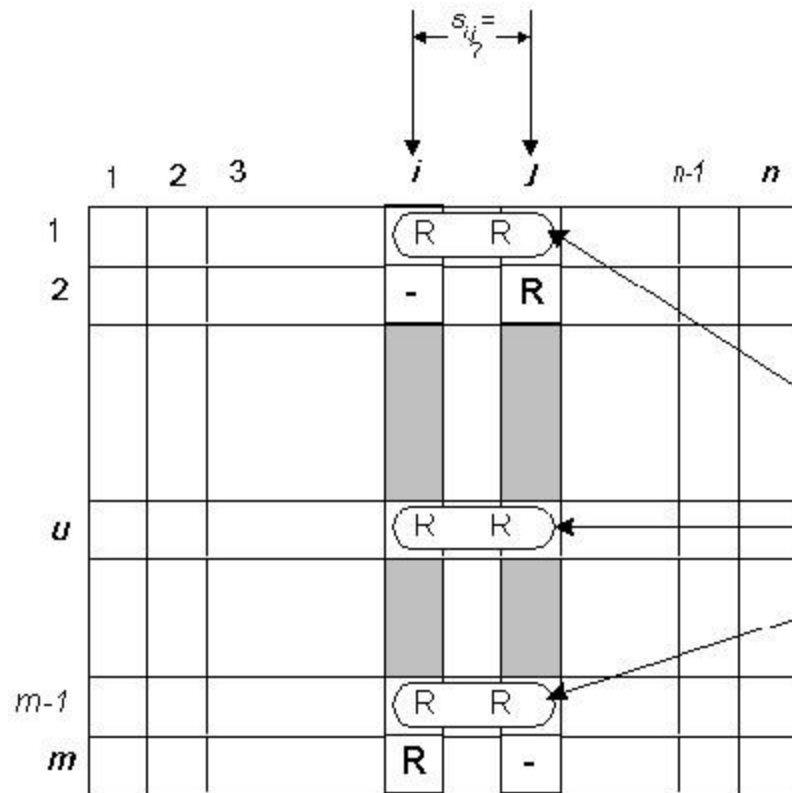
- Rather than matching the user to similar customers, build a similar-items table by finding that customers tend to purchase together
- Amazon.com used this method
- Scales independently of the catalog size or the total number of customers
- Acceptable performance by creating the expensive similar-item table offline

Item-to-Item CF Algorithm

```
For each item in product catalog,  $I_1$ 
  For each customer  $C$  who purchased  $I_1$ 
    For each item  $I_2$  purchased by
      customer  $C$ 
        Record that a customer purchased  $I_1$ 
          and  $I_2$ 
  For each item  $I_2$ 
    Compute the similarity between  $I_1$  and  $I_2$ 
```

- $O(N^2M)$ as worst case, $O(NM)$ in practical

Item-to-Item CF Algorithm Similarity Calculation

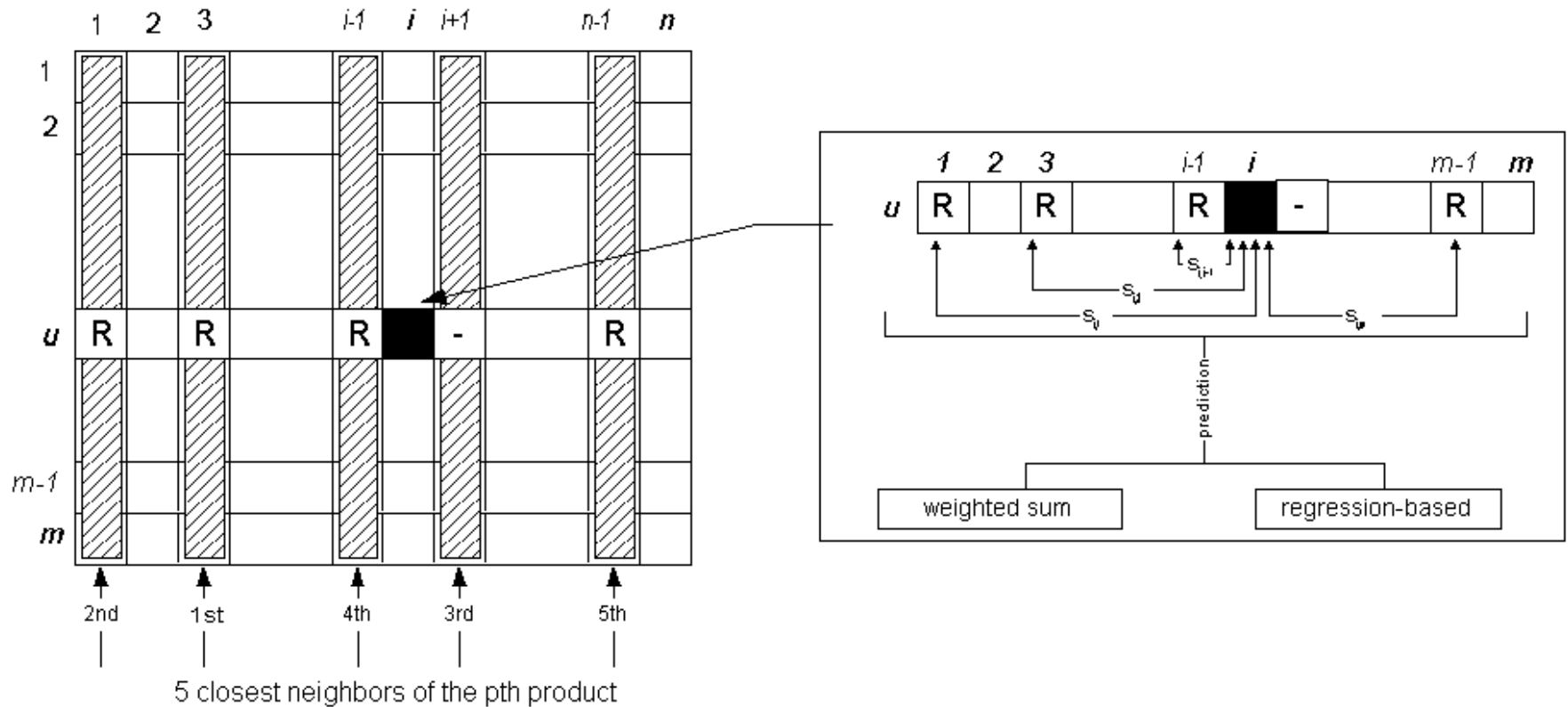


Item-to-Item CF Algorithm Similarity Calculation

- For similarity between two items i and j ,

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}.$$

Item-to-Item CF Algorithm Prediction Computation



- Recommend items with high-ranking based on similarity

Item-to-Item CF Algorithm Prediction Computation

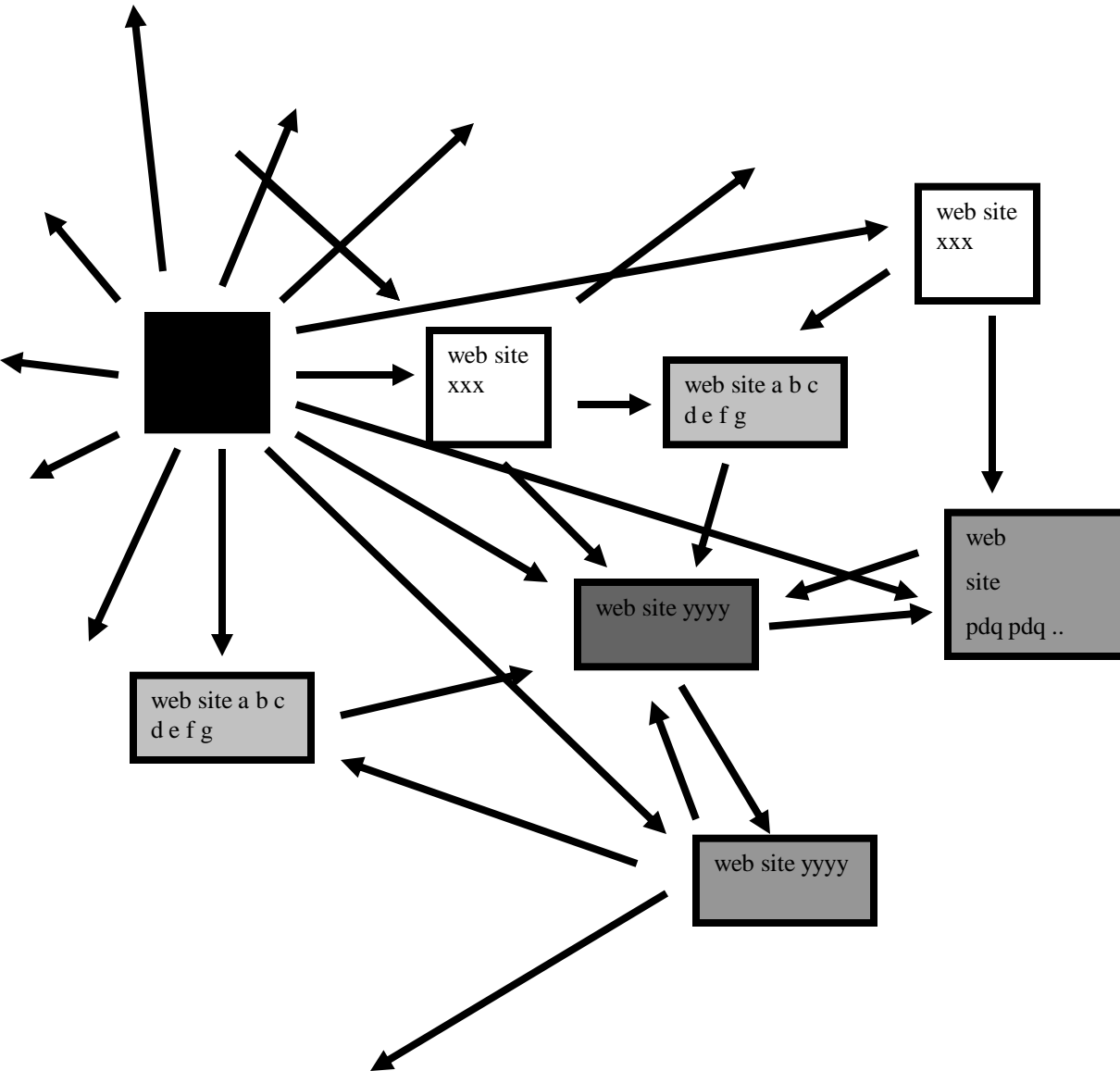
- Weighted Sum to capture how the active user rates the similar items

$$P_{u,i} = \frac{\sum_{\text{all similar items, } N} (s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, } N} (|s_{i,N}|)}$$

- Regression to avoid misleading in the sense that two similarities may be distant yet may have very high similarities

$$\bar{R}_N^i = \alpha \bar{R}_i + \beta + \epsilon$$

Google's PageRank



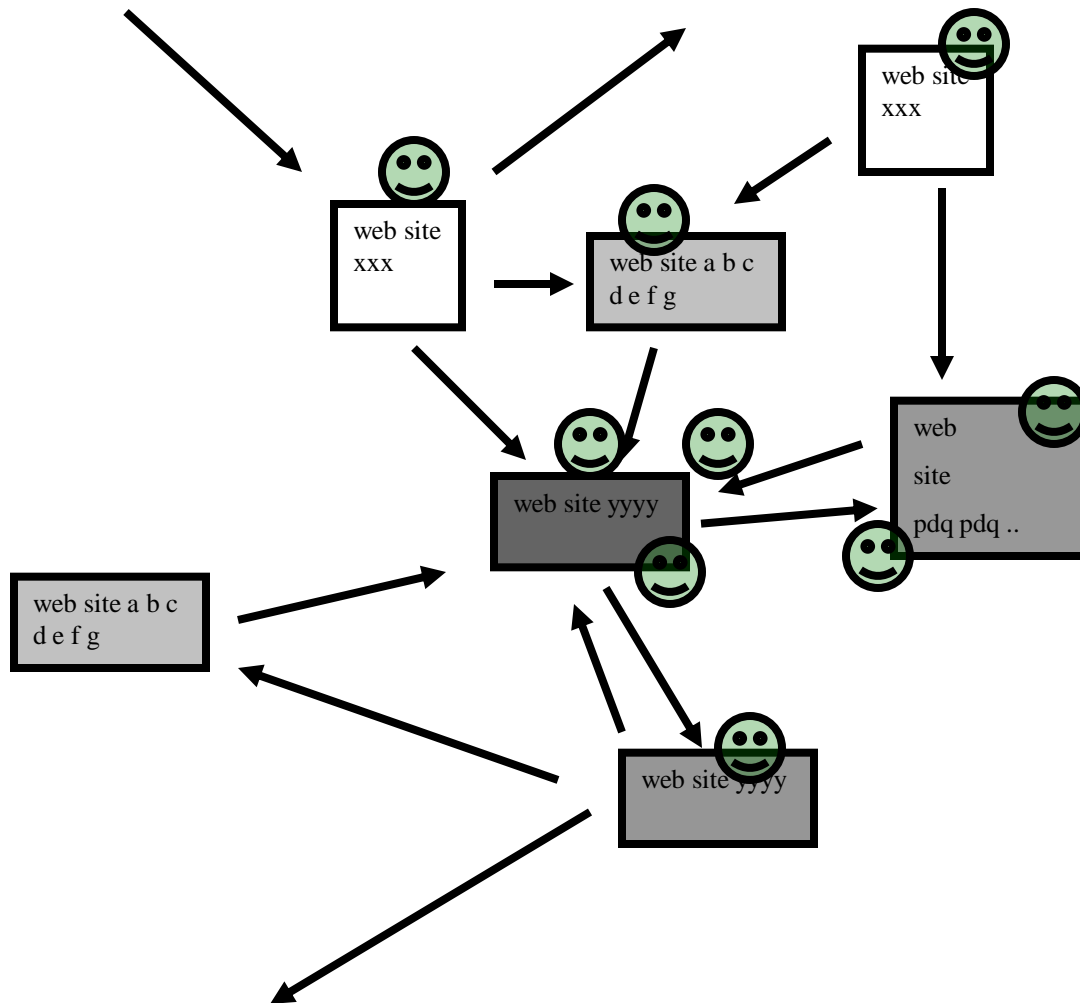
Inlinks are “good”
(recommendations)

Inlinks from a
“good” site are
better than inlinks
from a “bad” site

but inlinks from
sites with many
outlinks are not as
“good”...

“Good” and “bad”
are relative.

Google's PageRank

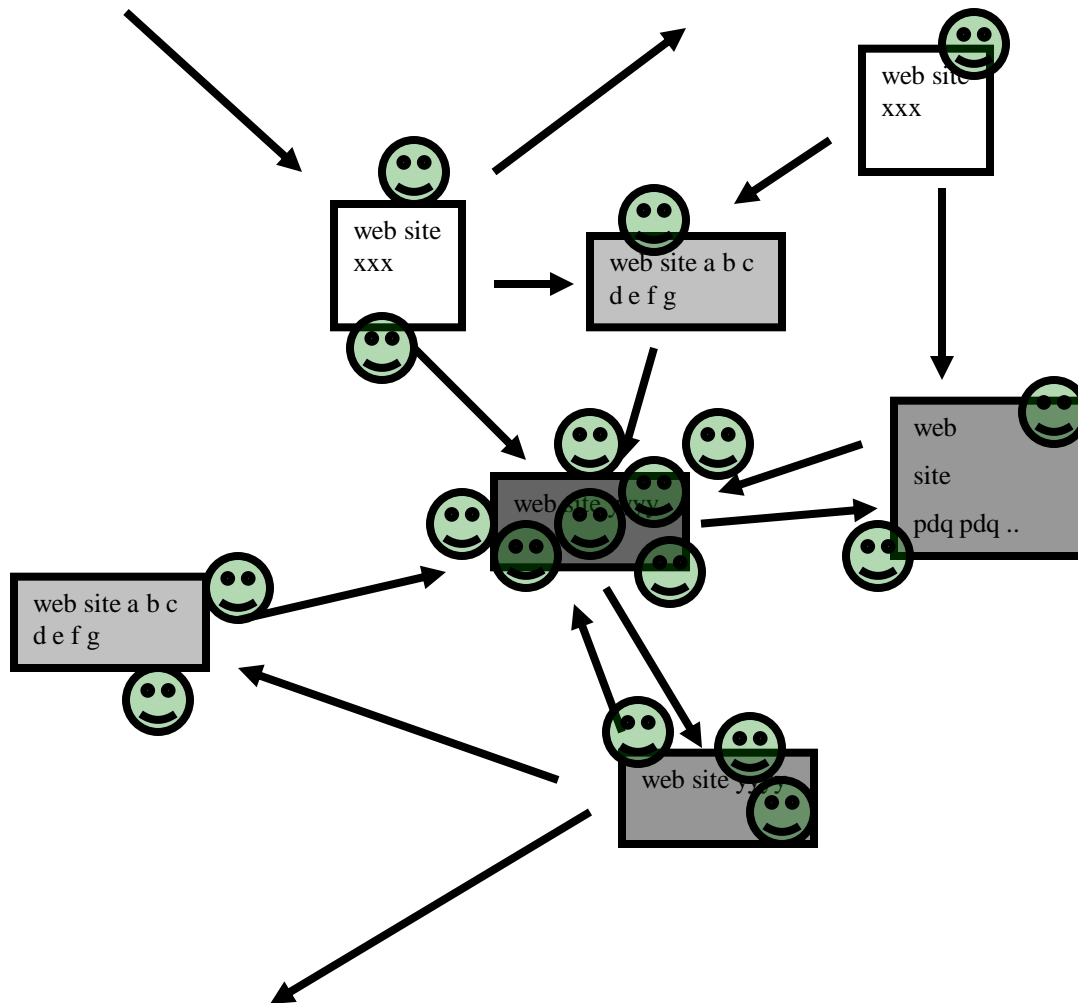


Imagine a “pagehopper” that always either

- follows a random link, or
- jumps to random page

Google's PageRank

(Brin & Page, <http://www-db.stanford.edu/~backrub/google.html>)



Imagine a “pagehopper” that always either

- follows a random link, or
- jumps to random page

PageRank ranks pages by the amount of time the pagehopper spends on a page:

- or, if there were many pagehoppers, PageRank is the expected “crowd size”

Memory-Based Algorithms (Breese et al, UAI98)

- $v_{i,j}$ = vote of user i on item j
- I_i = items for which user i has voted
- Mean vote for i is

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

- Predicted vote for “active user” a is weighted sum

$$p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^n \underbrace{w(a,i)}_{\text{weights of } n \text{ similar users}} (v_{i,j} - \bar{v}_i)$$

normalizer \nearrow

Memory-Based Algorithms (Breese et al, UAI98)

- K-nearest neighbor

$$w(a, i) = \begin{cases} 1 & \text{if } i \in \text{neighbors}(a) \\ 0 & \text{else} \end{cases}$$

- Pearson correlation coefficient (Resnick '94, Grouplens):

$$w(a, i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}}$$

- Cosine distance (from IR)

$$w(a, i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$$

Memory-Based Algorithms (Breese et al, UAI98)

- Cosine with “inverse user frequency” $f_i = \log(n/n_j)$, where n is number of users, n_j is number of users voting for item j

$$w(a, i) = \frac{\sum_j f_j \sum_j f_j v_{a,j} v_{i,j} - (\sum_j f_j v_{a,j})(\sum_j f_j v_{i,j})}{\sqrt{UV}}$$

where

$$U = \sum_j f_j (\sum_j f_j v_{a,j}^2 - (\sum_j f_j v_{a,j})^2)$$

$$V = \sum_i f_i (\sum_i f_i v_{i,j}^2 - (\sum_i f_i v_{i,j})^2)$$

Memory-Based Algorithms (Breese et al, UAI98)

- Evaluation:
 - split users into train/test sets
 - for each user a in the test set:
 - split a 's votes into observed (I) and to-predict (P)
 - measure average absolute **deviation** between predicted and actual votes in P
 - predict votes in P , and form a **ranked list**
 - assume (a) utility of k -th item in list is $\max(v_{a,j}-d, 0)$, where d is a “default vote” (b) probability of reaching rank k drops exponentially in k . Score a list by its expected utility R_a
 - average R_a over all test users

Memory-Based Algorithms (Breese et al, UAI98)

soccer score ↑

	EachMovie, Rank Scoring			
Algorithm	Given2	Given5	Given10	AllBut1
CR+	41.60	42.33	41.46	23.16
VSIM	42.45	42.12	40.15	22.07
BC	38.06	36.68	34.98	21.38
BN	28.64	30.50	33.16	23.49
POP	30.80	28.90	28.01	13.94
<i>RD</i>	<i>0.75</i>	<i>0.75</i>	<i>0.78</i>	<i>0.78</i>

Why are these numbers worse?

golf score ↓

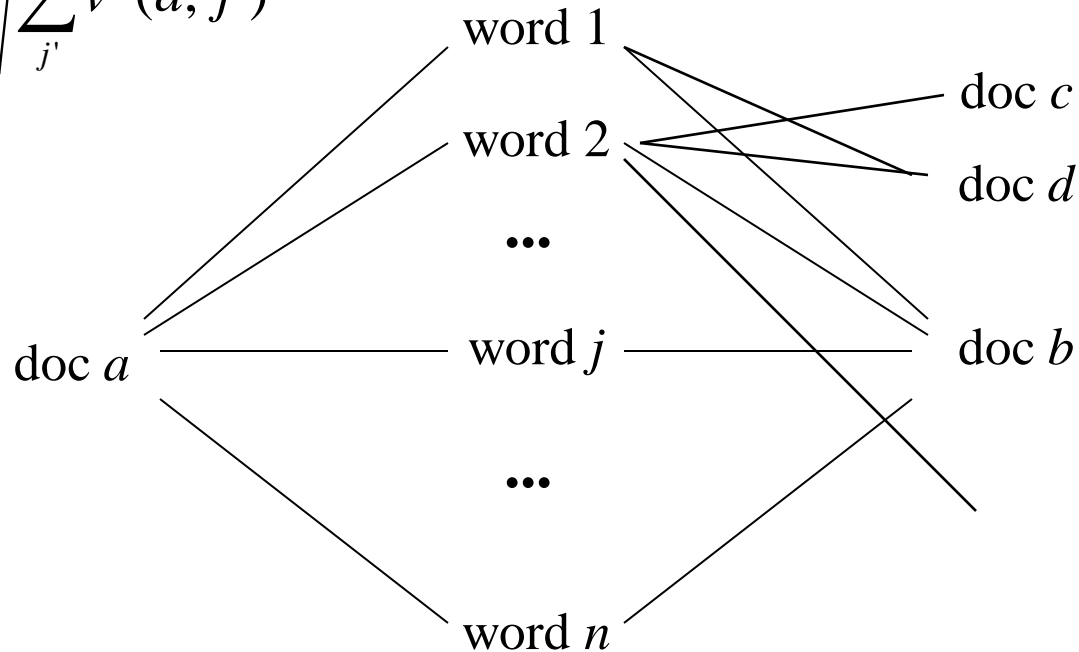
	EachMovie, Absolute Deviation			
Algorithm	Given2	Given5	Given10	AllBut1
CR	1.257	1.139	1.069	0.994
BC	1.127	1.144	1.138	1.103
BN	1.143	1.154	1.139	1.066
VSIM	2.113	2.177	2.235	2.136
<i>RD</i>	<i>0.022</i>	<i>0.023</i>	<i>0.025</i>	<i>0.043</i>

Visualizing Cosine Distance

similarity of doc a to doc $b = \text{sim}(a, b) = \sum_{\text{word } i} \frac{v(a, j)}{\sqrt{\sum_{j'} v^2(a, j')}} \cdot \frac{v(b, j)}{\sqrt{\sum_{j'} v^2(b, j')}} = A' \cdot B'$

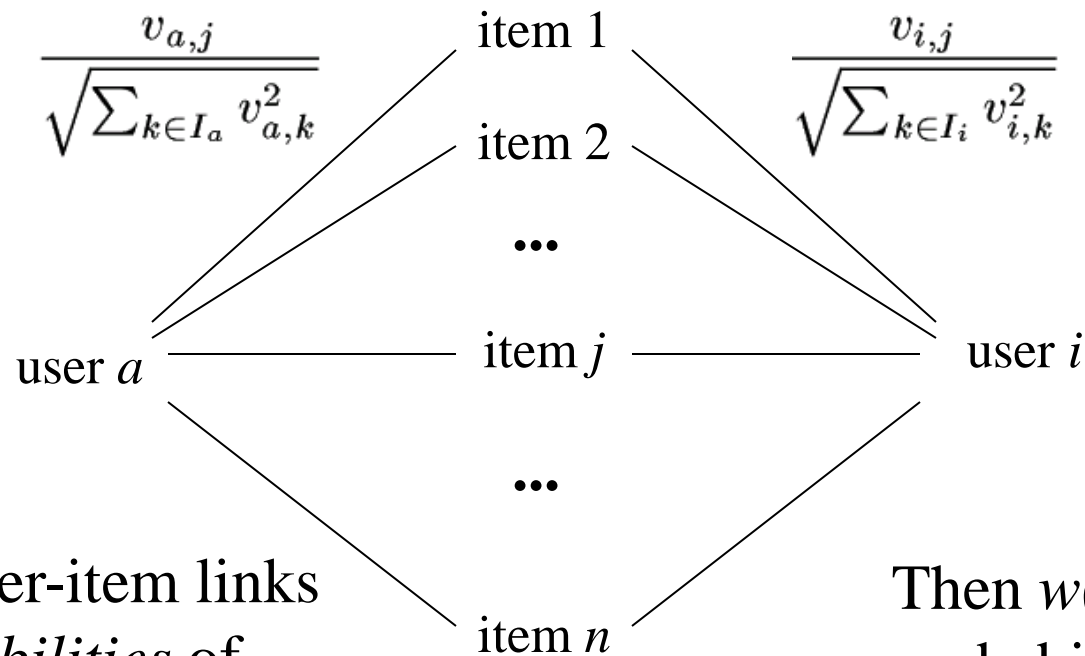
Let $\vec{A} = \langle \dots, v(a, j), \dots \rangle$

Let $\vec{A}' = \frac{\vec{A}}{\|\vec{A}\|} = \frac{\vec{A}}{\sqrt{\sum_{j'} v^2(a, j')}} = A'$



Visualizing Cosine Distance

distance from user a to user i = $w(a, i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$

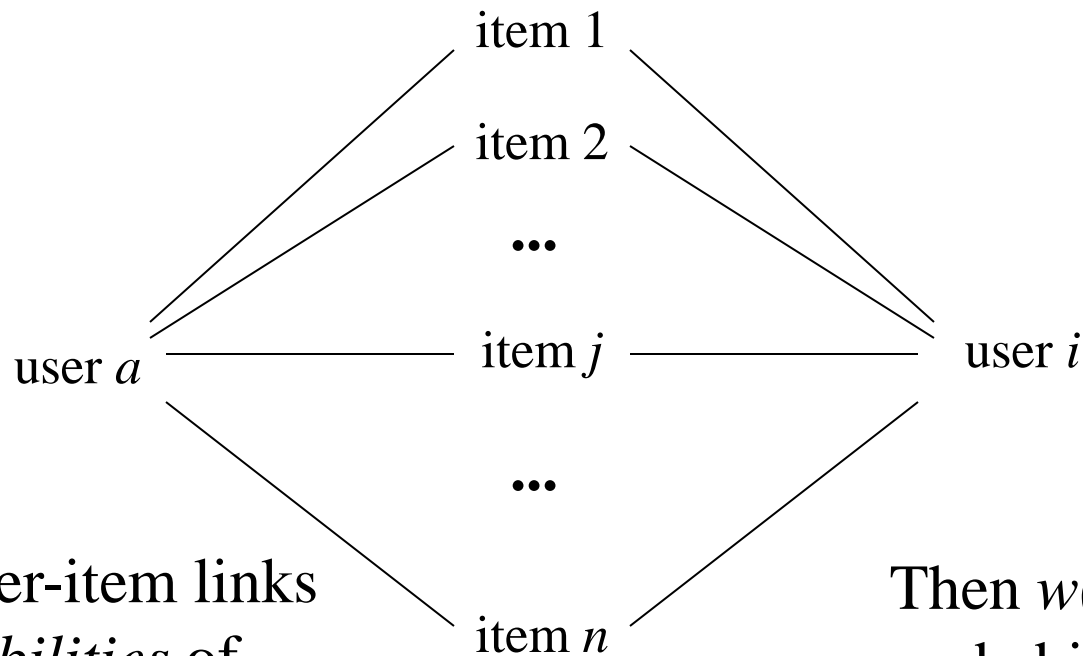


Suppose user-item links
were *probabilities* of
following a link

Then $w(a, i)$ is
probability of a and i
“meeting”

Visualizing Cosine Distance

Approximating Matrix Multiplication for Pattern Recognition Tasks, Cohen & Lewis, SODA 97—explores connection between cosine distance/inner product and random walks



Suppose user-item links were *probabilities* of following a link

Then $w(a,i)$ is probability of a and i “meeting”

References

- E-Commerce Recommendation Applications:
<http://citeseer.ist.psu.edu/cache/papers/cs/14532/http:zSzzSzwww.cs.umn.edu:zSzResearchSzGroupLensSzECRA.pdf/schafer01ecommerce.pdf>
- Amazon.com Recommendations: Item-to-Item Collaborative Filtering
<http://www.win.tue.nl/~laroyo/2L340/resources/Amazon-Recommendations.pdf>
- Item-based Collaborative Filtering Recommendation Algorithms
http://www.grouplens.org/papers/pdf/www10_sarwar.pdf
- John S. Breese, David Heckerman, Carl Myers Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. UAI 1998: 43-52
- Chumki Basu, Haym Hirsh, William W. Cohen: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. AAAI/IAAI 1998: 714-720
- Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, Steve Lawrence: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. UAI 2001: 437-444