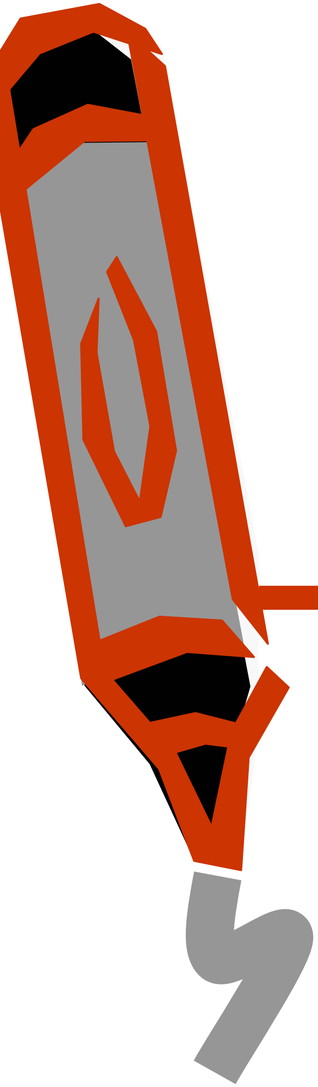
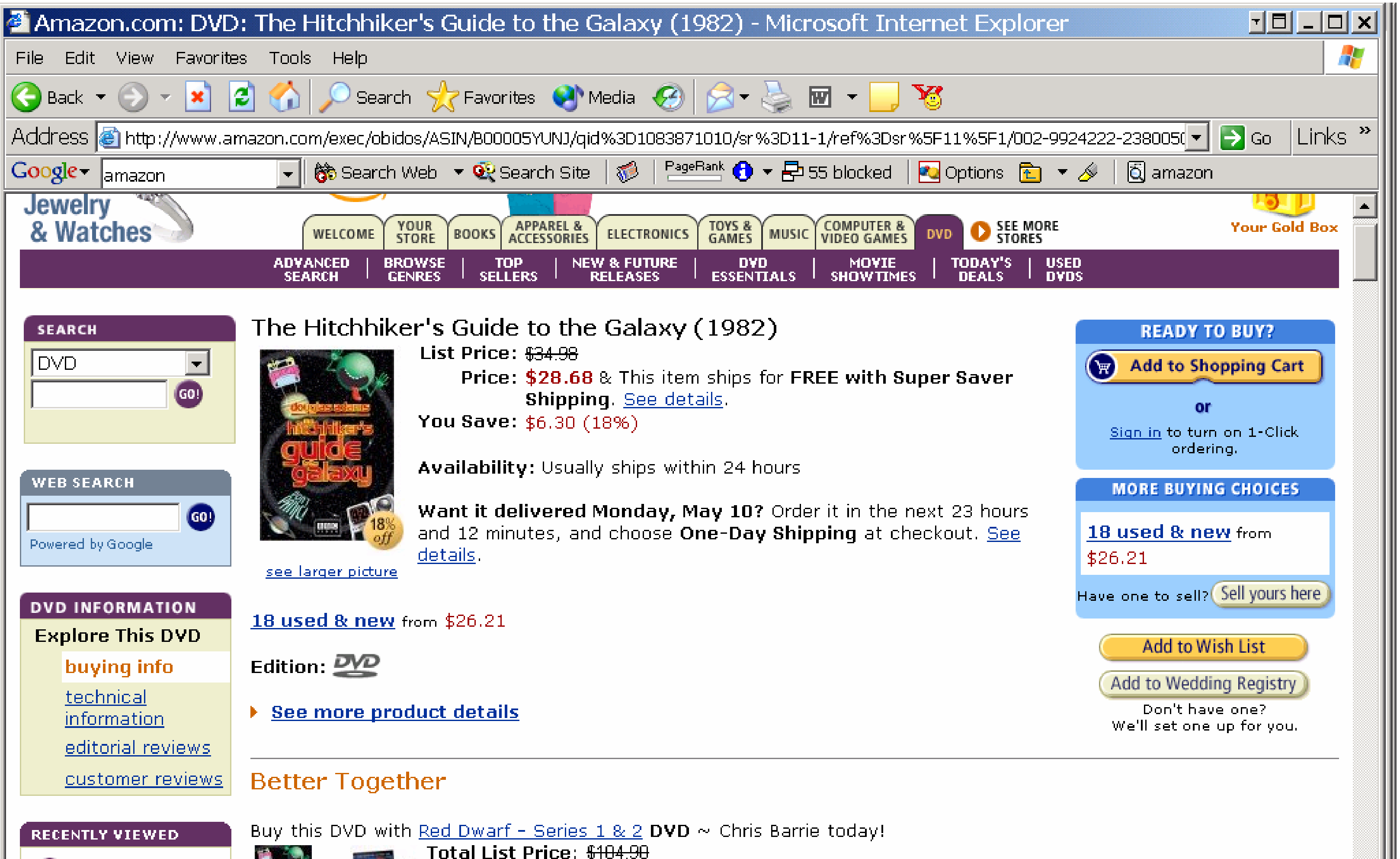


Collaborative Filtering

William W. Cohen
Center for Automated Learning and Discovery
Carnegie Mellon University

Everyday Examples of Collaborative Filtering...



The Hitchhiker's Guide to the Galaxy (1982)

List Price: ~~$34.98~~

Price: $28.68 & This item ships for **FREE with Super Saver Shipping**. See details.

You Save: $6.30 (18%)

Availability: Usually ships within 24 hours

Want it delivered Monday, May 10? Order it in the next 23 hours and 12 minutes, and choose **One-Day Shipping** at checkout. See details.

18 used & new from $26.21

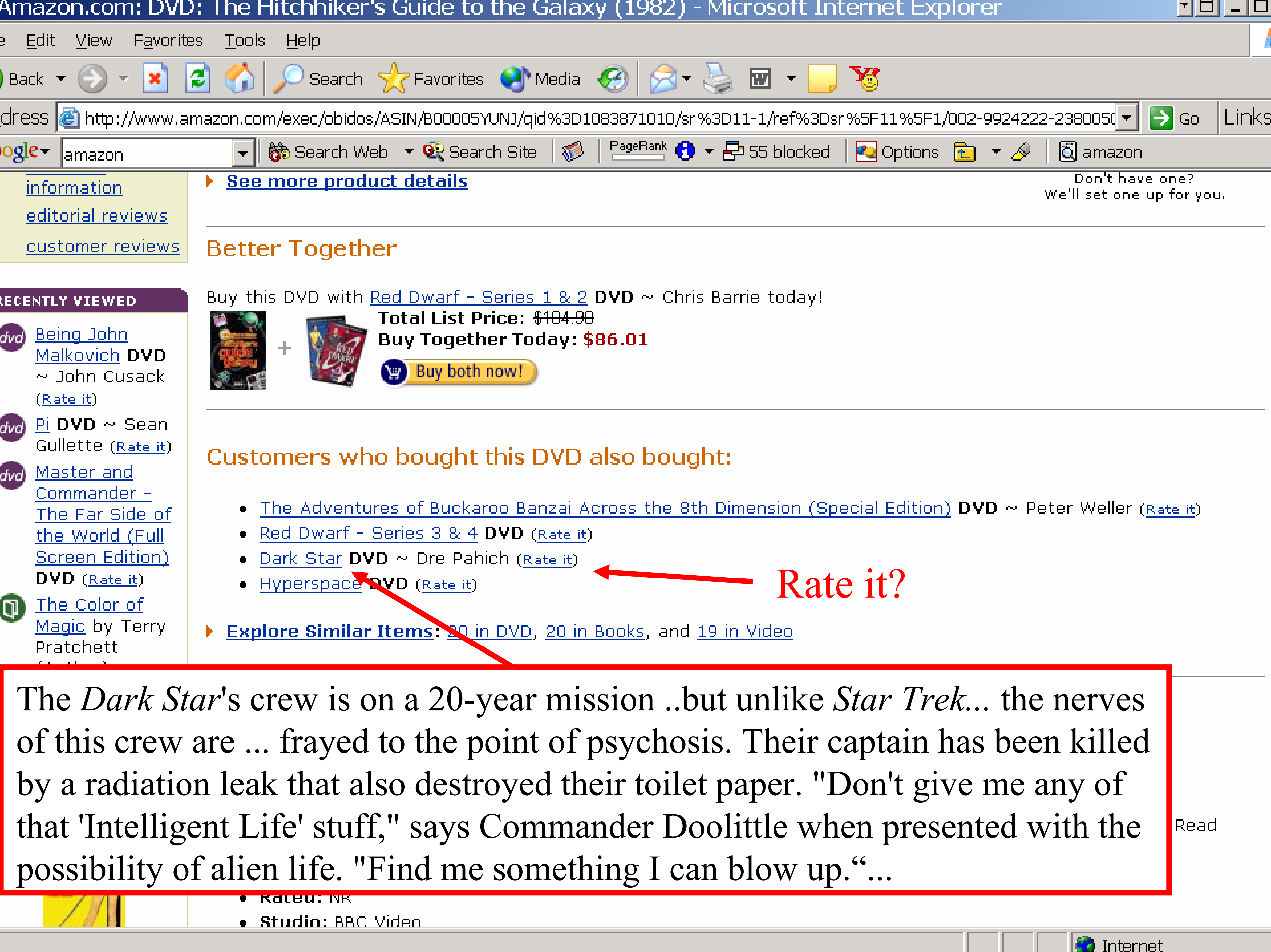
Edition: DVD

See more product details

Better Together

Buy this DVD with Red Dwarf - Series 1 & 2 **DVD** ~ Chris Barrie today!

Total List Price: ~~$104.98~~


Amazon.com: DVD: The Hitchhiker's Guide to the Galaxy (1982) - Microsoft Internet Explorer
Edit View Favorites Tools Help
Back Search Favorites Media
dress http://www.amazon.com/exec/obidos/ASIN/B00005YUNJ/qid%3D1083871010/sr%3D11-1/ref%3Dsr%5F11%5F1/002-9924222-238005
Go Links
amazon Search Web Search Site PageRank 55 blocked Options amazon
Internet

information
editorial reviews
customer reviews

RECENTLY VIEWED

Being John Malkovich DVD ~ John Cusack (Rate it)

Pi DVD ~ Sean Gullette (Rate it)

Master and Commander - The Far Side of the World (Full Screen Edition) DVD (Rate it)

The Color of Magic by Terry Pratchett

See more product details

Don't have one?
We'll set one up for you.

Better Together

Buy this DVD with Red Dwarf - Series 1 & 2 DVD ~ Chris Barrie today!

Total List Price: ~~$104.98~~
Buy Together Today: $86.01

Customers who bought this DVD also bought:

- The Adventures of Buckaroo Banzai Across the 8th Dimension (Special Edition) **DVD** ~ Peter Weller (Rate it)
- Red Dwarf - Series 3 & 4 **DVD** (Rate it)
- Dark Star **DVD** ~ Dre Pahich (Rate it)
- Hyperspace **DVD** (Rate it)

Explore Similar Items: 20 in DVD, 20 in Books, and 19 in Video

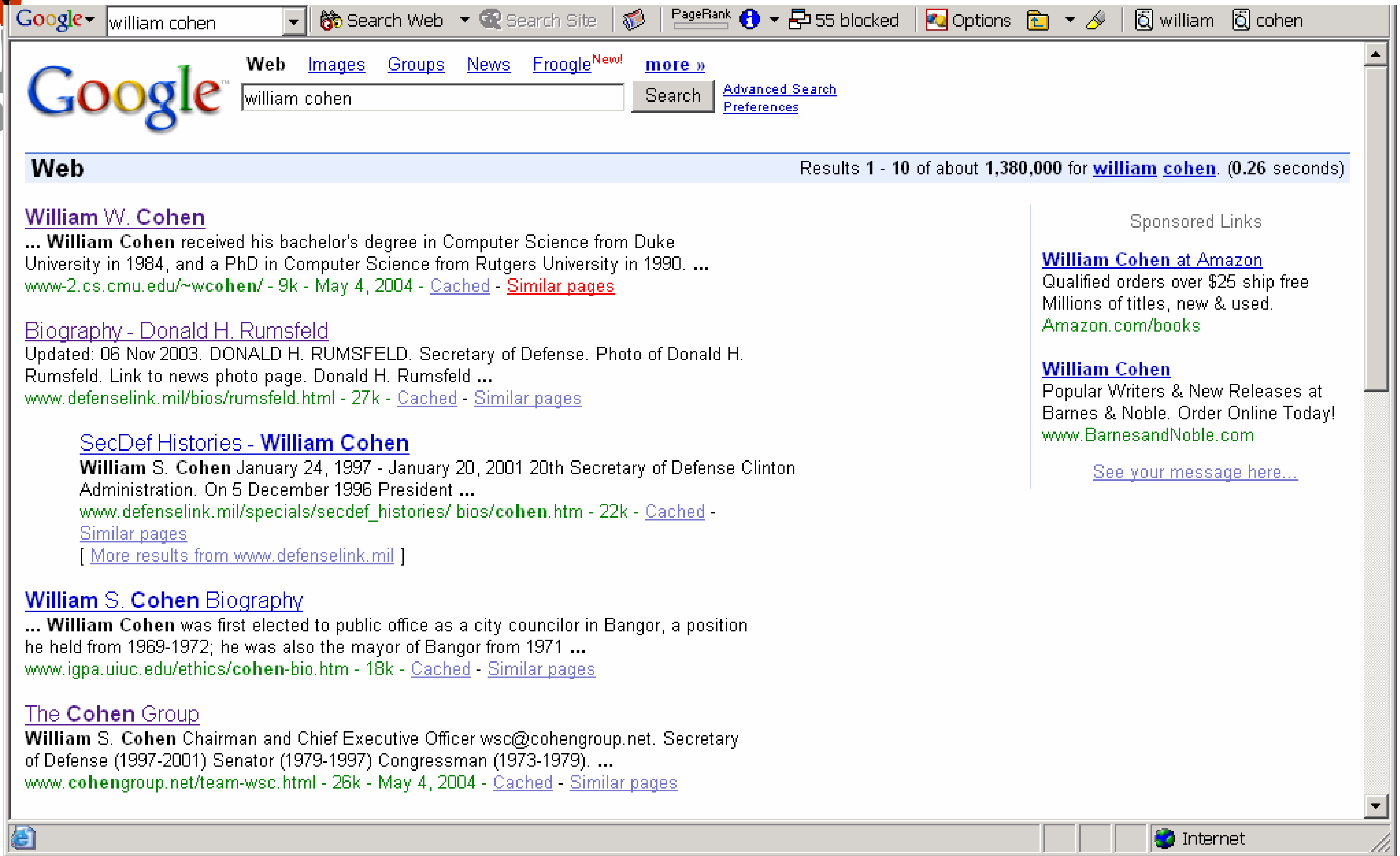
Read

- **Rated:** NR
- **Studio:** BBC Video

Rate it?

The *Dark Star*'s crew is on a 20-year mission ..but unlike *Star Trek*... the nerves of this crew are ... frayed to the point of psychosis. Their captain has been killed by a radiation leak that also destroyed their toilet paper. "Don't give me any of that 'Intelligent Life' stuff," says Commander Doolittle when presented with the possibility of alien life. "Find me something I can blow up.“...

Everyday Examples of Collaborative Filtering...



william cohen

Web Results **1** - **10** of about **1,380,000** for **william** **cohen**. (**0.26** seconds)

William W. **Cohen**
... **William Cohen** received his bachelor's degree in Computer Science from Duke University in 1984, and a PhD in Computer Science from Rutgers University in 1990. ...
www-2.cs.cmu.edu/~**wcohen**/ - 9k - May 4, 2004 - Cached - Similar pages

Biography - Donald H. Rumsfeld
Updated: 06 Nov 2003. DONALD H. RUMSFELD. Secretary of Defense. Photo of Donald H. Rumsfeld. Link to news photo page. Donald H. Rumsfeld ...
www.defenselink.mil/bios/rumsfeld.html - 27k - Cached - Similar pages

SecDef Histories - **William Cohen**
William S. **Cohen** January 24, 1997 - January 20, 2001 20th Secretary of Defense Clinton Administration. On 5 December 1996 President ...
www.defenselink.mil/specials/secdef_histories/ bios/**cohen**.htm - 22k - Cached - Similar pages
[More results from www.defenselink.mil]

William S. **Cohen** Biography
... **William Cohen** was first elected to public office as a city councilor in Bangor, a position he held from 1969-1972; he was also the mayor of Bangor from 1971 ...
www.igpa.uiuc.edu/ethics/**cohen**-bio.htm - 18k - Cached - Similar pages

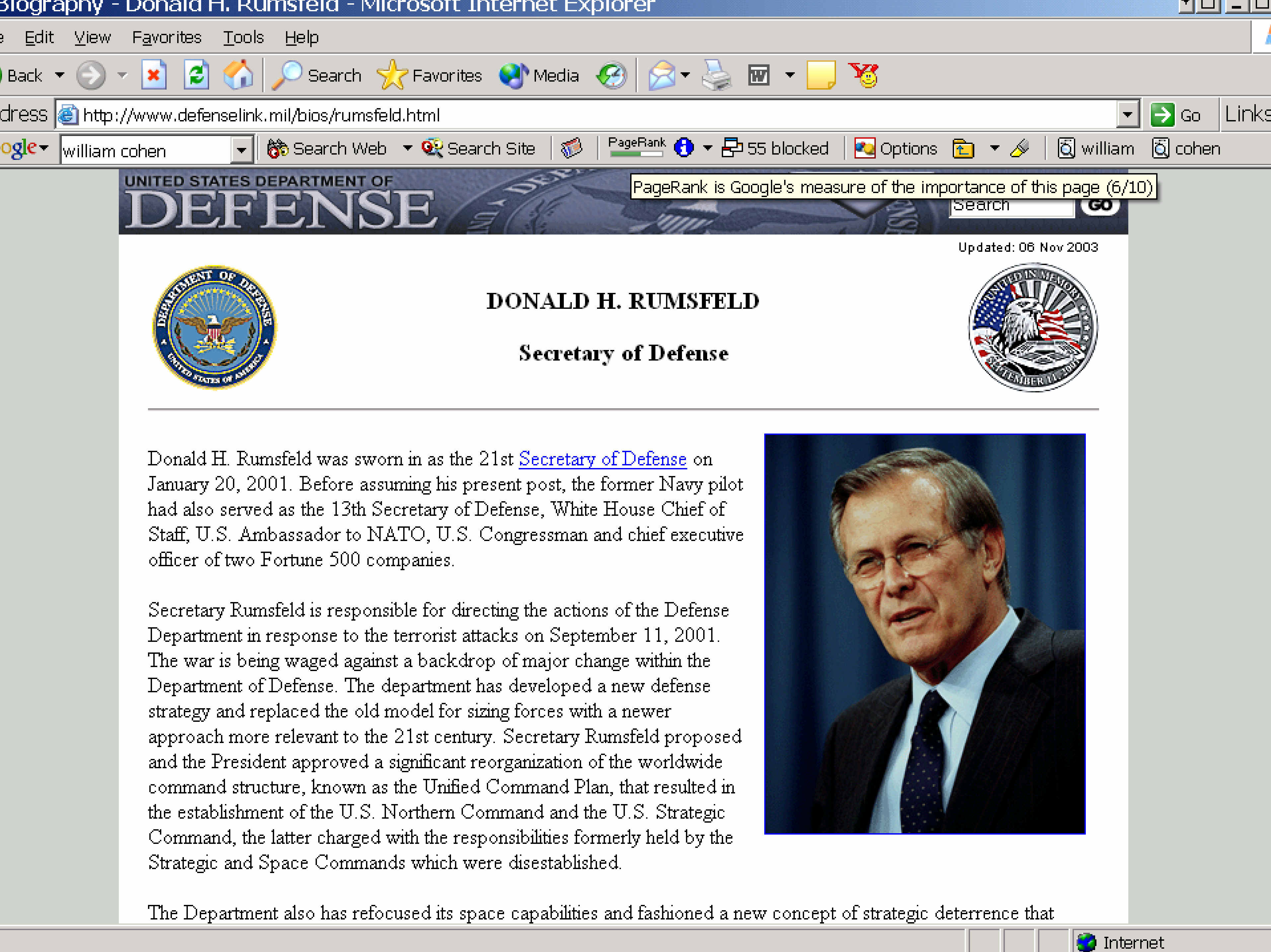
The **Cohen** Group
William S. **Cohen** Chairman and Chief Executive Officer wsc@cohengroup.net. Secretary of Defense (1997-2001) Senator (1979-1997) Congressman (1973-1979). ...
www.**cohen**group.net/team-wsc.html - 26k - May 4, 2004 - Cached - Similar pages

Sponsored Links

William Cohen at Amazon
Qualified orders over $25 ship free
Millions of titles, new & used.
Amazon.com/books

William Cohen
Popular Writers & New Releases at Barnes & Noble. Order Online Today!
www.BarnesandNoble.com

See your message here...


Biography - Donald H. Rumsfeld - Microsoft Internet Explorer
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
dress
http://www.defenselink.mil/bios/rumsfeld.html
Go
Links
ogle
william cohen
Search Web
Search Site
PageRank
55 blocked
Options
william
cohen
PageRank is Google's measure of the importance of this page (6/10)
UNITED STATES DEPARTMENT OF
DEFENSE
Search
GO
Internet

Updated: 06 Nov 2003

DONALD H. RUMSFELD

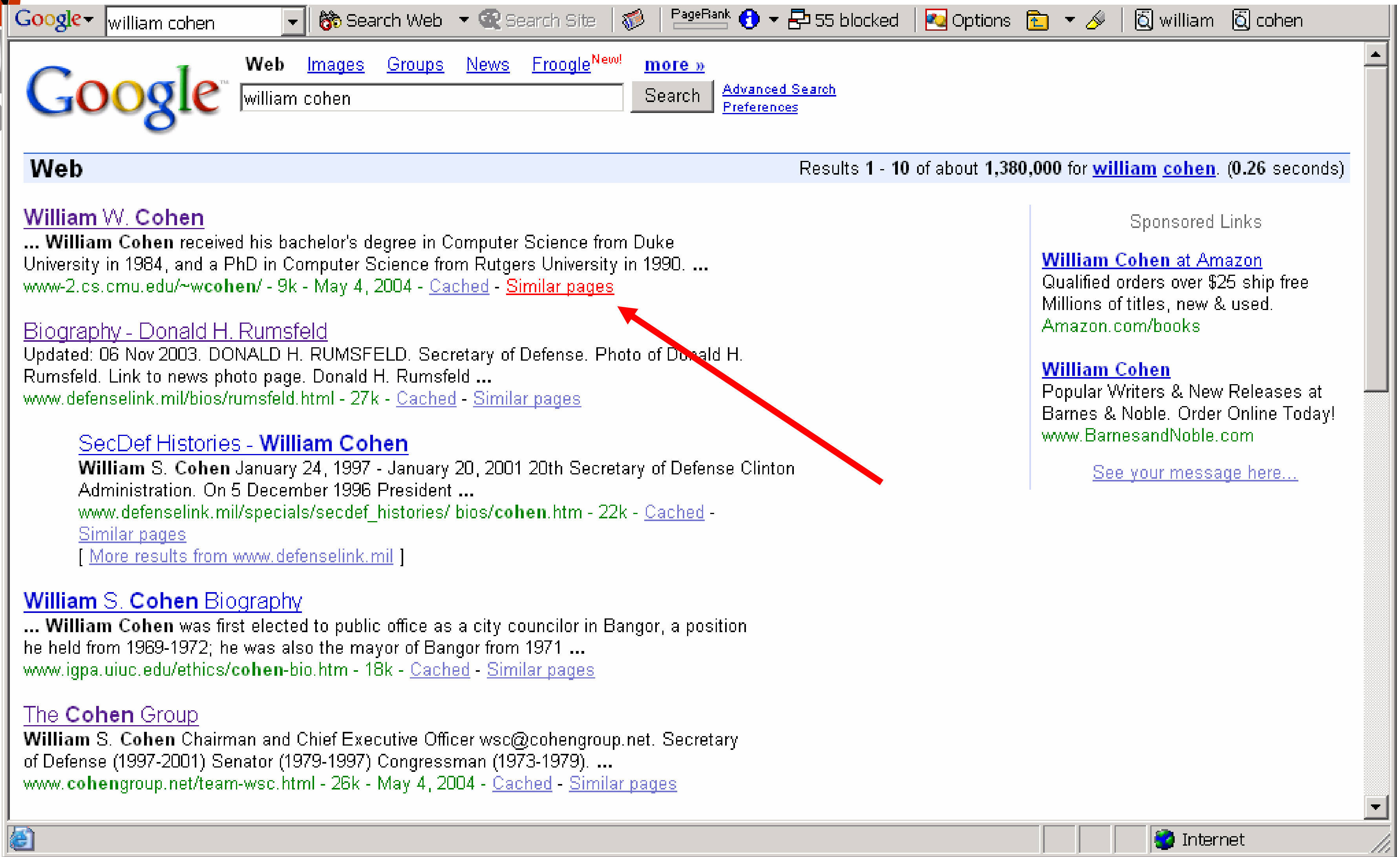
Secretary of Defense

Donald H. Rumsfeld was sworn in as the 21st Secretary of Defense on January 20, 2001. Before assuming his present post, the former Navy pilot had also served as the 13th Secretary of Defense, White House Chief of Staff, U.S. Ambassador to NATO, U.S. Congressman and chief executive officer of two Fortune 500 companies.

Secretary Rumsfeld is responsible for directing the actions of the Defense Department in response to the terrorist attacks on September 11, 2001. The war is being waged against a backdrop of major change within the Department of Defense. The department has developed a new defense strategy and replaced the old model for sizing forces with a newer approach more relevant to the 21st century. Secretary Rumsfeld proposed and the President approved a significant reorganization of the worldwide command structure, known as the Unified Command Plan, that resulted in the establishment of the U.S. Northern Command and the U.S. Strategic Command, the latter charged with the responsibilities formerly held by the Strategic and Space Commands which were disestablished.

The Department also has refocused its space capabilities and fashioned a new concept of strategic deterrence that

Everyday Examples of Collaborative Filtering...



william cohen

Web Results **1** - **10** of about **1,380,000** for **william** **cohen**. (**0.26** seconds)

William W. Cohen
... **William Cohen** received his bachelor's degree in Computer Science from Duke University in 1984, and a PhD in Computer Science from Rutgers University in 1990. ...
www-2.cs.cmu.edu/~w**cohen**/ - 9k - May 4, 2004 - Cached - Similar pages

Biography - Donald H. Rumsfeld
Updated: 06 Nov 2003. DONALD H. RUMSFELD. Secretary of Defense. Photo of Donald H. Rumsfeld. Link to news photo page. Donald H. Rumsfeld ...
www.defenselink.mil/bios/rumsfeld.html - 27k - Cached - Similar pages

SecDef Histories - **William Cohen**
William S. **Cohen** January 24, 1997 - January 20, 2001 20th Secretary of Defense Clinton Administration. On 5 December 1996 President ...
www.defenselink.mil/specials/secdef_histories/ bios/**cohen**.htm - 22k - Cached - Similar pages
[More results from www.defenselink.mil]

William S. **Cohen** Biography
... **William Cohen** was first elected to public office as a city councilor in Bangor, a position he held from 1969-1972; he was also the mayor of Bangor from 1971 ...
www.igpa.uiuc.edu/ethics/**cohen**-bio.htm - 18k - Cached - Similar pages

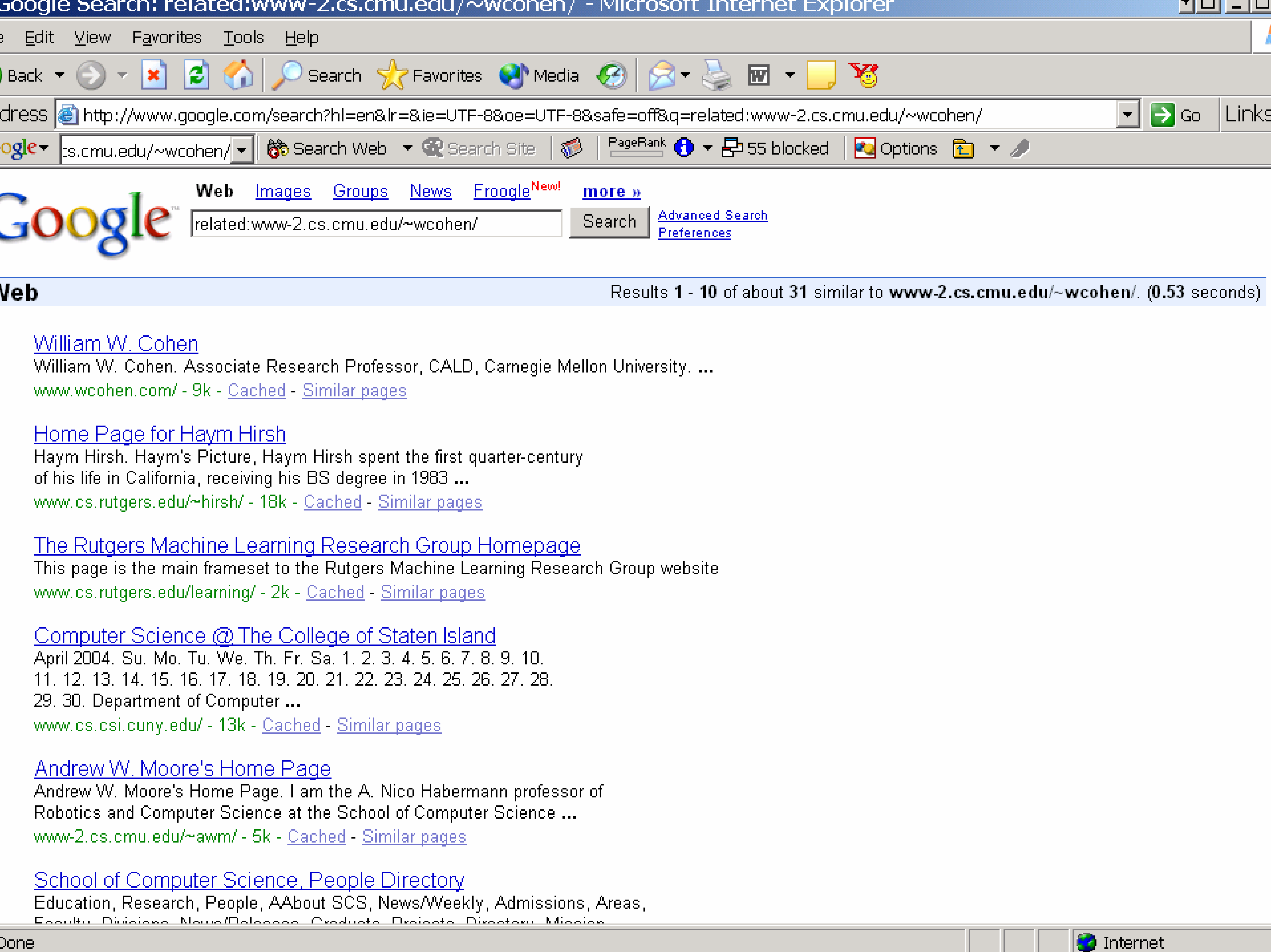
The **Cohen** Group
William S. **Cohen** Chairman and Chief Executive Officer wsc@cohengroup.net. Secretary of Defense (1997-2001) Senator (1979-1997) Congressman (1973-1979). ...
www.**cohen**group.net/team-wsc.html - 26k - May 4, 2004 - Cached - Similar pages

Sponsored Links

William Cohen at Amazon
Qualified orders over $25 ship free
Millions of titles, new & used.
Amazon.com/books

William Cohen
Popular Writers & New Releases at Barnes & Noble. Order Online Today!
www.BarnesandNoble.com

See your message here...



Google Search: related:www-2.cs.cmu.edu/~wcohen/ - Microsoft Internet Explorer
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
dress
http://www.google.com/search?hl=en&lr=&ie=UTF-8&oe=UTF-8&safe=off&q=related:www-2.cs.cmu.edu/~wcohen/
Go
Links
ogle
:s.cmu.edu/~wcohen/
Search Web
Search Site
PageRank
55 blocked
Options
Web
Images
Groups
News
Froogle New!
more »
related:www-2.cs.cmu.edu/~wcohen/
Search
Advanced Search
Preferences
Done
Internet

Veb Results **1** - **10** of about **31** similar to **www-2.cs.cmu.edu/~wcohen/**. (**0.53** seconds)

William W. Cohen
William W. Cohen. Associate Research Professor, CALD, Carnegie Mellon University. **...**
www.wcohen.com/ - 9k - Cached - Similar pages

Home Page for Haym Hirsh
Haym Hirsh. Haym's Picture, Haym Hirsh spent the first quarter-century of his life in California, receiving his BS degree in 1983 **...**
www.cs.rutgers.edu/~hirsh/ - 18k - Cached - Similar pages

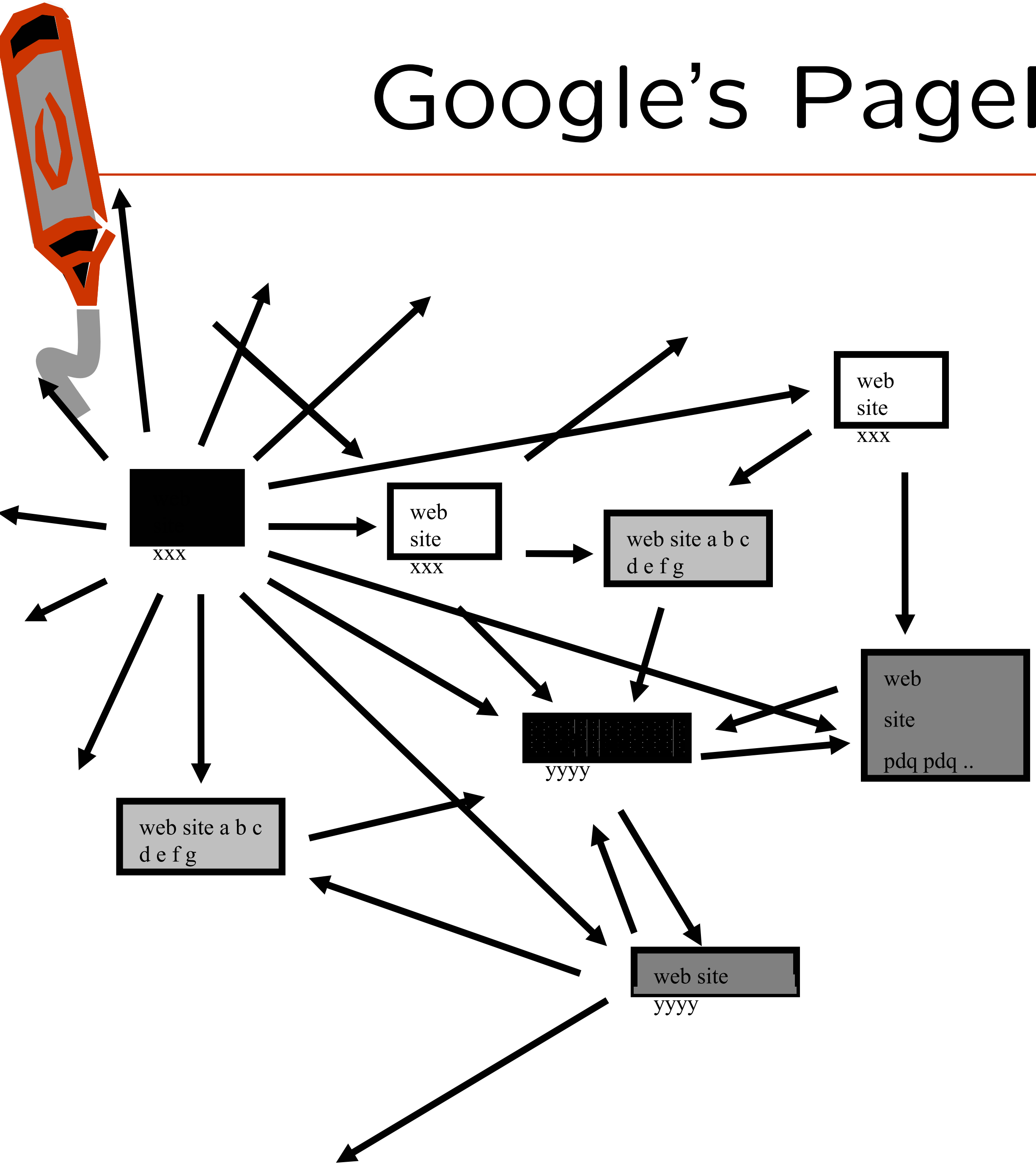
The Rutgers Machine Learning Research Group Homepage
This page is the main frameset to the Rutgers Machine Learning Research Group website
www.cs.rutgers.edu/learning/ - 2k - Cached - Similar pages

Computer Science @ The College of Staten Island
April 2004. Su. Mo. Tu. We. Th. Fr. Sa. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. Department of Computer **...**
www.cs.csi.cuny.edu/ - 13k - Cached - Similar pages

Andrew W. Moore's Home Page
Andrew W. Moore's Home Page. I am the A. Nico Habermann professor of Robotics and Computer Science at the School of Computer Science **...**
www-2.cs.cmu.edu/~awm/ - 5k - Cached - Similar pages

School of Computer Science, People Directory
Education, Research, People, AAbout SCS, News/Weekly, Admissions, Areas, Faculty, Divisions, News/Releases, Graduate, Projects, Directory, Mission

Google's PageRank



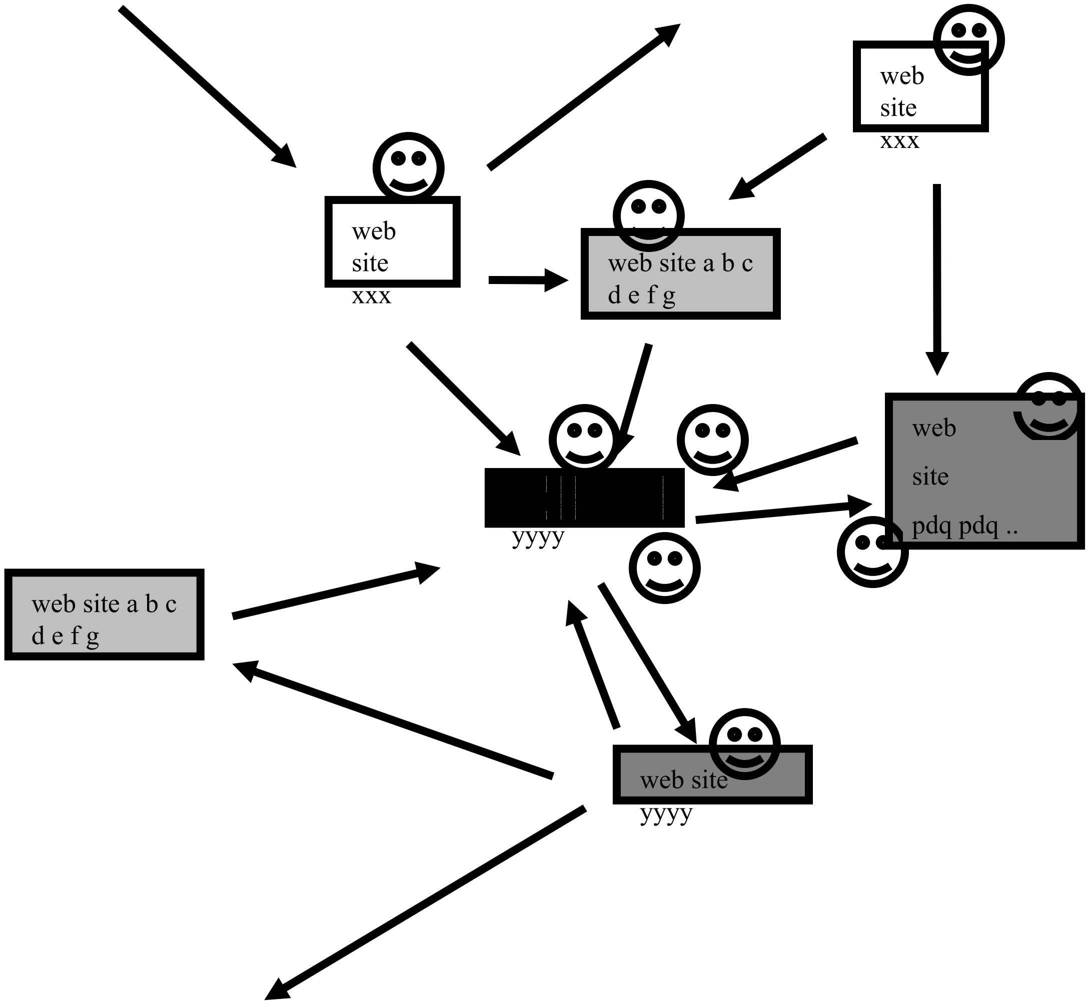
Inlinks are "good" (recommendations)

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

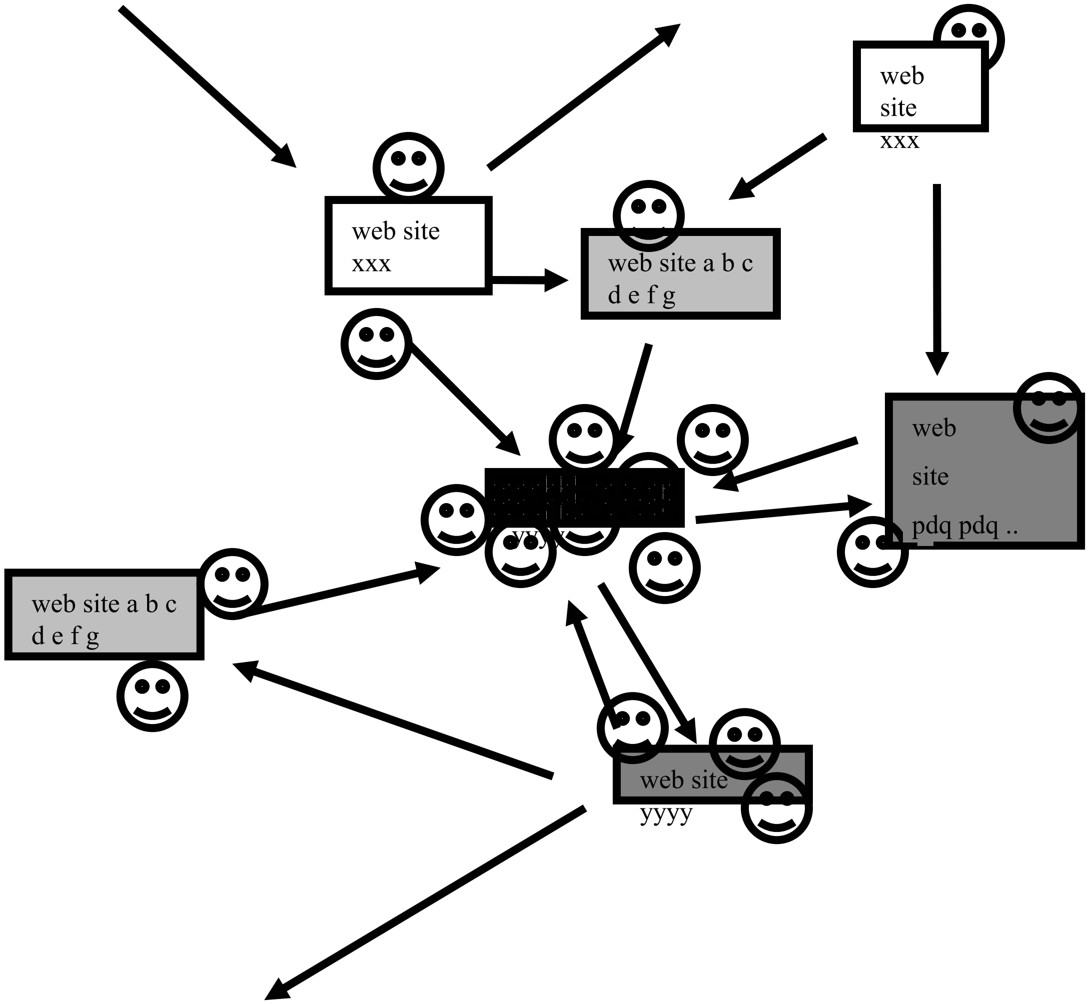


Imagine a "pagehopper" that always either ☺

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

Google's PageRank

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



web
site
xxx

web site
xxx

web site a b c
d e f g

yyyy

web
site
pdq pdq ..

web site a b c
d e f g

web site
yyyy

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"

Everyday Examples of Collaborative Filtering...

- Bestseller lists
- Top 40 music lists
- The “recent returns” shelf at the library
- Unmarked but well-used paths thru the woods
- The printer room at work
- Many weblogs
- “Read any good books lately?”
- ....
- Common insight: personal tastes are *correlated*:
  - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y
  - especially (perhaps) if Bob knows Alice

Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
- Algorithms for CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings
- Conclusions/Summary

BellCore's MovieRecommender

- *Recommending And Evaluating Choices In A Virtual Community Of Use.* Will Hill, Larry Stead, Mark Rosenstein and George Furnas, Bellcore; CHI 1995

By **virtual community** we mean "a group of people who share characteristics and interact in essence or effect only". In other words, people in a Virtual Community influence each other *as though* they interacted but they *do not interact*. Thus we ask: "Is it possible to arrange for people to share some of the personalized informational benefits of community involvement without the associated communications costs?"

MovieRecommender Goals

Recommendations should:

- simultaneously ease and encourage rather than replace social processes....should make it easy to participate while leaving in hooks for people to pursue more personal relationships if they wish.
- be for sets of people not just individuals...multi-person recommending is often important, for example, when two or more people want to choose a video to watch together.
- be from people not a black box machine or so-called ”agent”.
- tell how much confidence to place in them, in other words they should include indications of how accurate they are.

BellCore’s MovieRecommender

- Participants sent email to videos@bellcore.com
- System replied with a list of 500 movies to rate on a 1-10 scale (250 random, 250 popular)
  - Only subset need to be rated
- New participant *P* sends in rated movies via email
- System compares ratings for *P* to ratings of (a random sample of) previous users
- Most *similar users* are used to predict scores for unrated movies (more later)
- System returns recommendations in an email message.

Suggested Videos for: John A. Jamus.

Your must-see list with predicted ratings:

- 7.0 "Alien (1979)"
- 6.5 "Blade Runner"
- 6.2 "Close Encounters Of The Third Kind (1977)"

Your video categories with average ratings:

- 6.7 "Action/Adventure"
- 6.5 "Science Fiction/Fantasy"
- 6.3 "Children/Family"
- 6.0 "Mystery/Suspense"
- 5.9 "Comedy"
- 5.8 "Drama"

The viewing patterns of 243 viewers were consulted. Patterns of 7 viewers were found to be most similar. Correlation with target viewer:

- 0.59 viewer-130 (unlisted@merl.com)
- 0.55 bullert,jane r (bullert@cc.bellcore.com)
- 0.51 jan_arst (jan_arst@khdld.decnet.philips.nl)
- 0.46 Ken Cross (moose@denali.EE.CORNELL.EDU)
- 0.42 rskt (rskt@cc.bellcore.com)
- 0.41 kkgg (kkgg@Athena.MIT.EDU)
- 0.41 bnn (bnn@cc.bellcore.com)

By category, their joint ratings recommend:

- Action/Adventure:
  - "Excalibur" 8.0, 4 viewers
  - "Apocalypse Now" 7.2, 4 viewers
  - "Platoon" 8.3, 3 viewers
- Science Fiction/Fantasy:
  - "Total Recall" 7.2, 5 viewers
- Children/Family:
  - "Wizard Of Oz, The" 8.5, 4 viewers
  - "Mary Poppins" 7.7, 3 viewers

Mystery/Suspense:

- "Silence Of The Lambs, The" 9.3, 3 viewers

Comedy:

- "National Lampoon's Animal House" 7.5, 4 viewers
- "Driving Miss Daisy" 7.5, 4 viewers
- "Hannah and Her Sisters" 8.0, 3 viewers

Drama:

- "It's A Wonderful Life" 8.0, 5 viewers
- "Dead Poets Society" 7.0, 5 viewers
- "Rain Man" 7.5, 4 viewers

Correlation of predicted ratings with your actual ratings is: 0.64 This number measures ability to evaluate movies accurately for you. 0.15 means low ability. 0.85 means very good ability. 0.50 means fair ability.

BellCore's MovieRecommender

- Evaluation:
  - Withhold 10% of the ratings of each user to use as a test set
  - Measure correlation between *predicted* ratings and *actual* ratings for test-set movie/user pairs

Figure 3 *Two Scatterplots of Actual Ratings by Predicted Ratings. Plot on left shows movie critics as predictor (r=0.22). Plot on right shows virtual community as predictor (r=0.62) (all values are jittered for the purpose of visual presentation, 3269 predictions each for 291 users)*

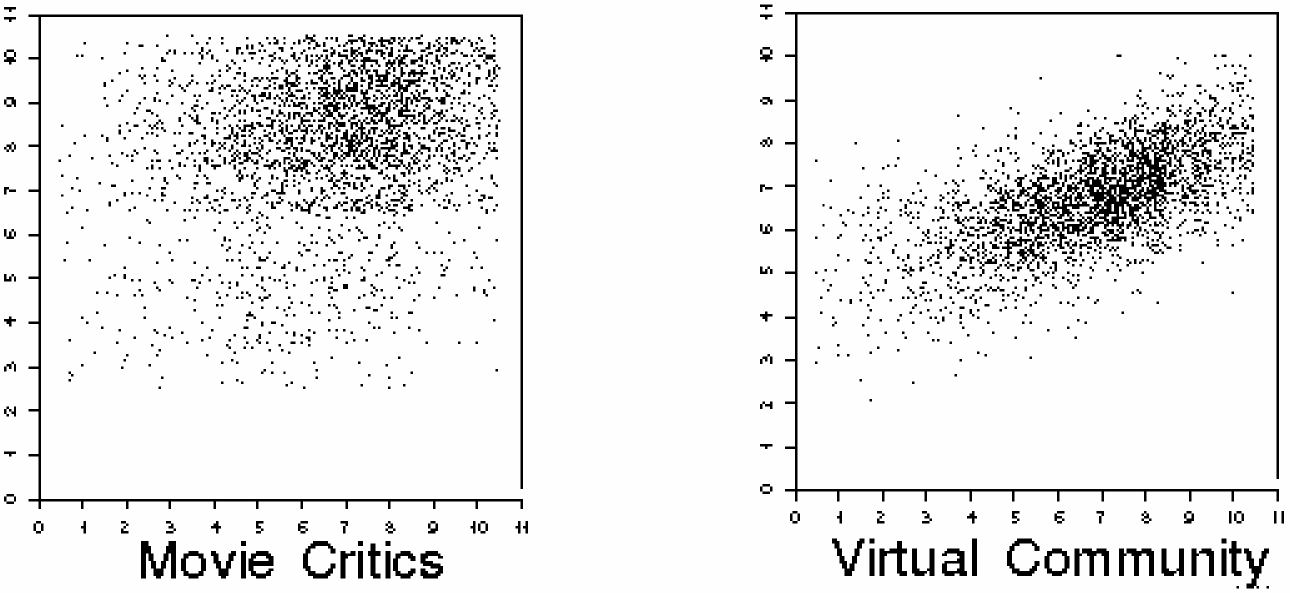
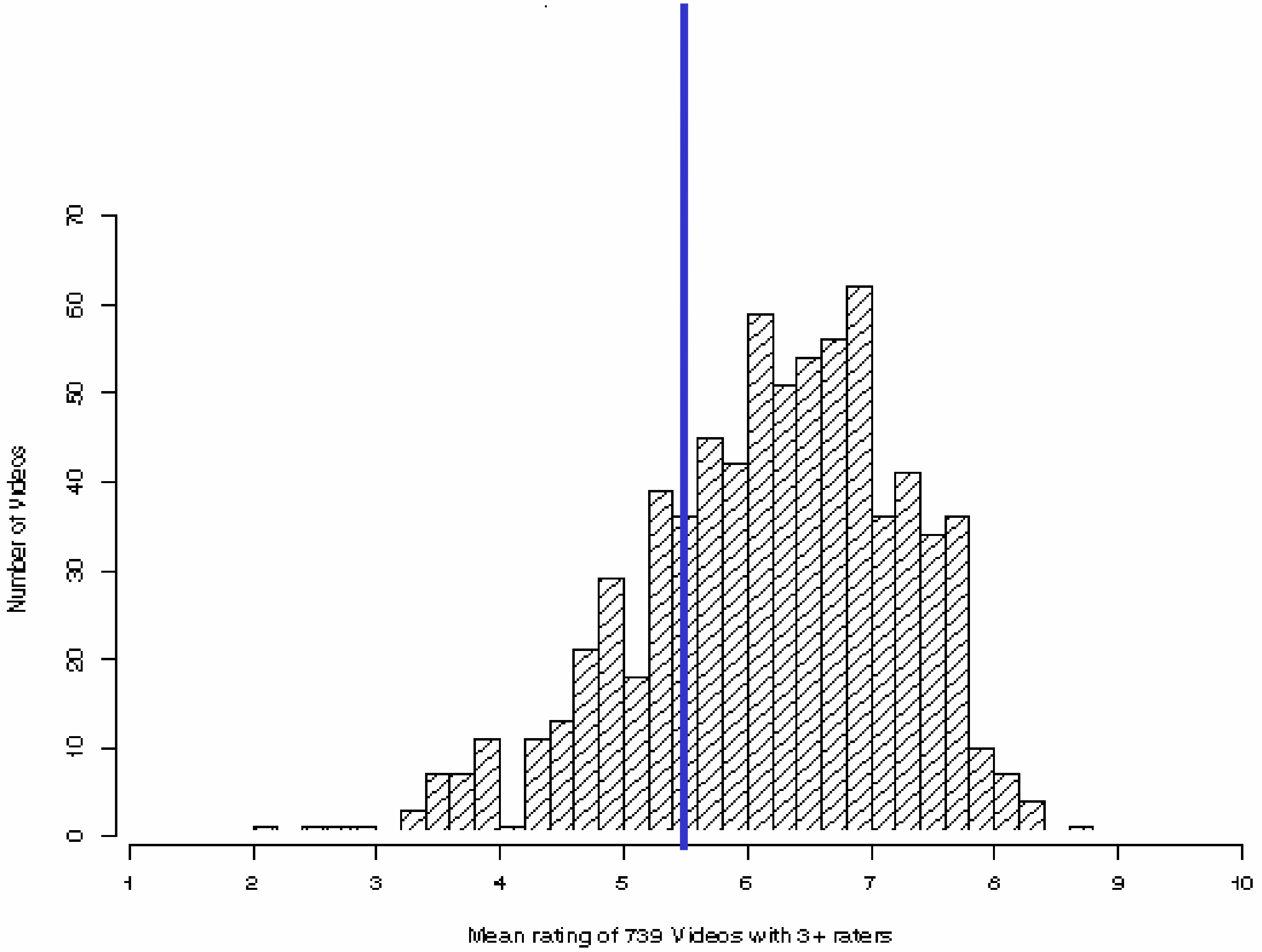


Figure 2 *Distribution of Video Mean Ratings*

Another key observation: *rated movies* tend to have *positive* ratings:

i.e., people rate what they watch, and watch what they like



Question: Can observation replace explicit rating?

BellCore's MovieRecommender

- Participants sent email to videos@bellcore.com
- System replied with a list of 500 movies to rate
  New participant *P* sends in rated movies via email
- System compares ratings for *P* to ratings of (a random sample of) previous users
- Most similar users are used to predict scores for unrated movies
  - *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* Breese, Heckerman, Kadie, UAI98
- System returns recommendations in an email message.

Algorithms for Collaborative Filtering 1: 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

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

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

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

Algorithms for Collaborative Filtering 1: Memory-Based Algorithms (Breese et al, UAI98)

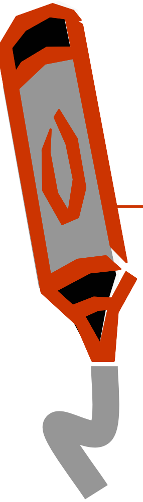
- K-nearest neighbor
- Pearson correlation coefficient (Resnick '94, Grouplens):

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

- Cosine distance (from IR)

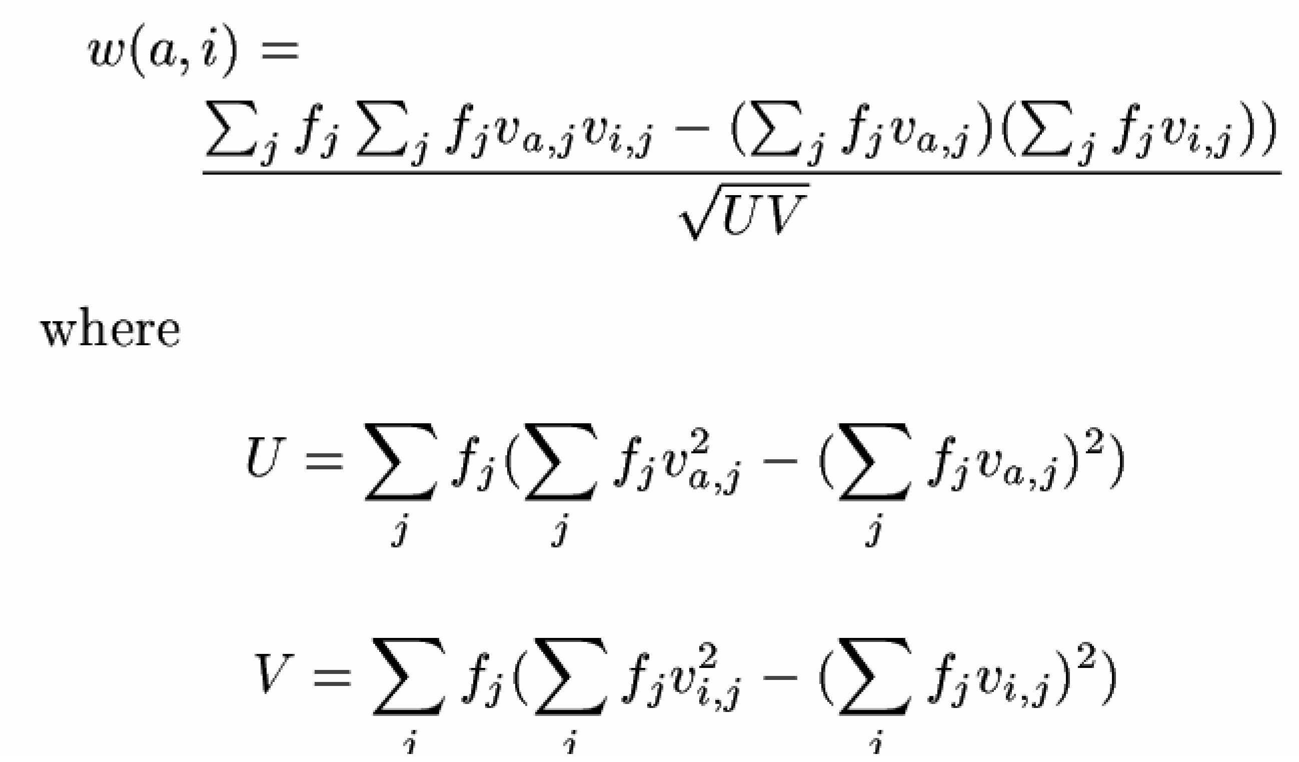
$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}}$$

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



Algorithms for Collaborative Filtering 1: 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

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

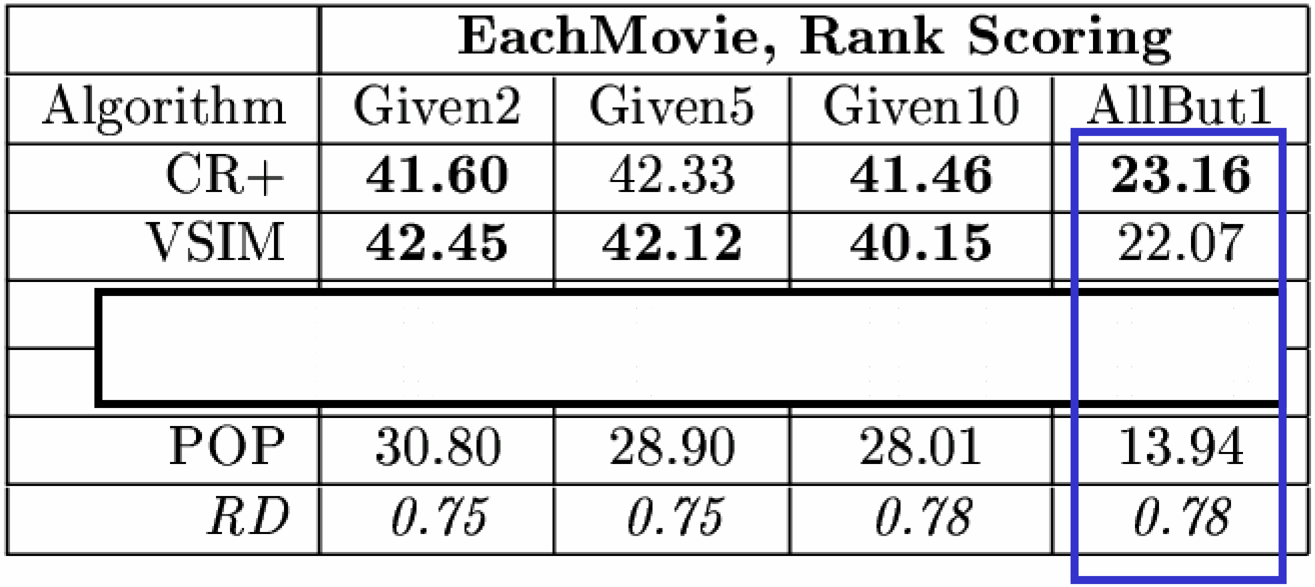
$$V = \sum_j f_j (\sum_j f_j v_{i,j}^2 - (\sum_j f_j v_{i,j})^2)$$

Algorithms for Collaborative Filtering 1: 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

Algorithms for Collaborative Filtering 1: Memory-Based Algorithms (Breese et al, UAI98)

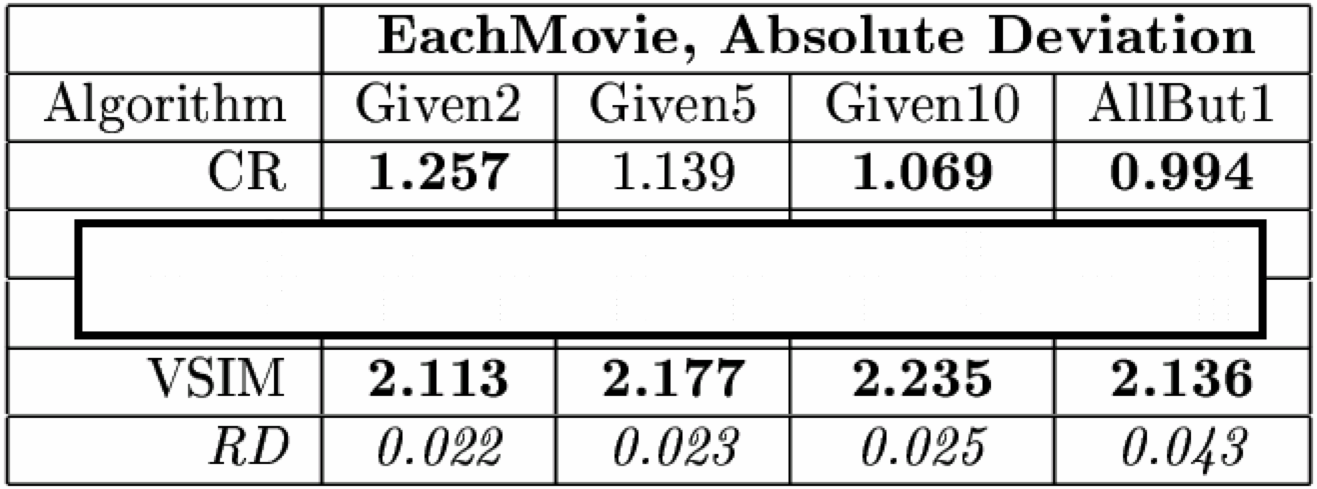
soccer score



	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
POP	30.80	28.90	28.01	13.94
RD	*0.75*	*0.75*	*0.78*	*0.78*

Why are these numbers **worse**?

golf score



	EachMovie, Absolute Deviation			
Algorithm	Given2	Given5	Given10	AllBut1
CR	**1.257**	1.139	**1.069**	**0.994**
VSIM	**2.113**	**2.177**	**2.235**	**2.136**
RD	*0.022*	*0.023*	*0.025*	*0.043*

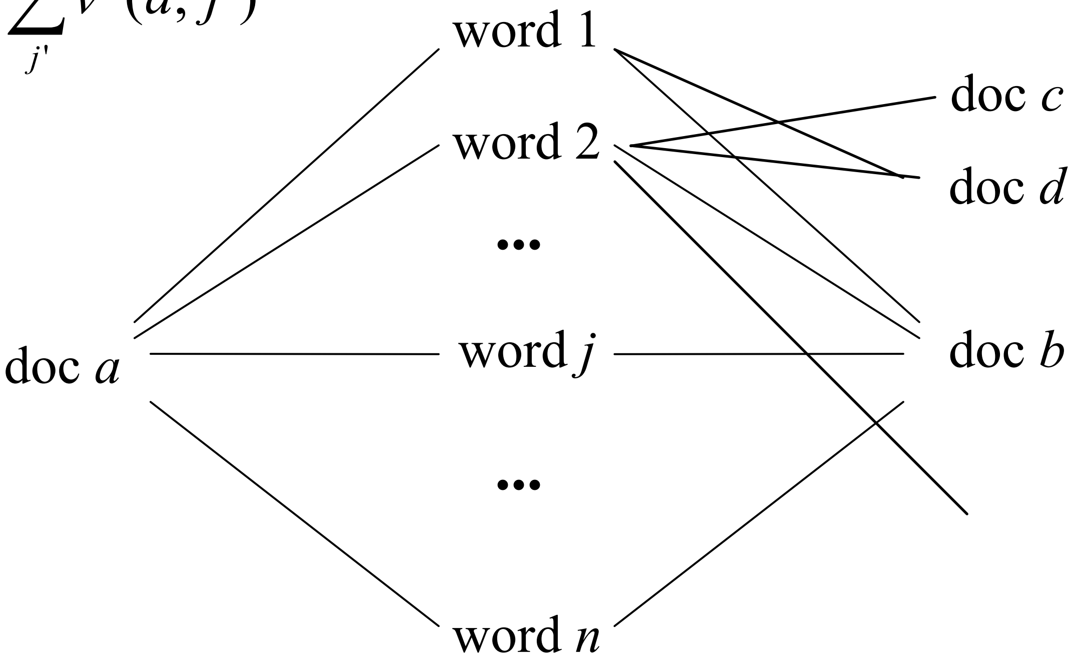
Visualizing Cosine Distance

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

$$= A' \cdot B'$$

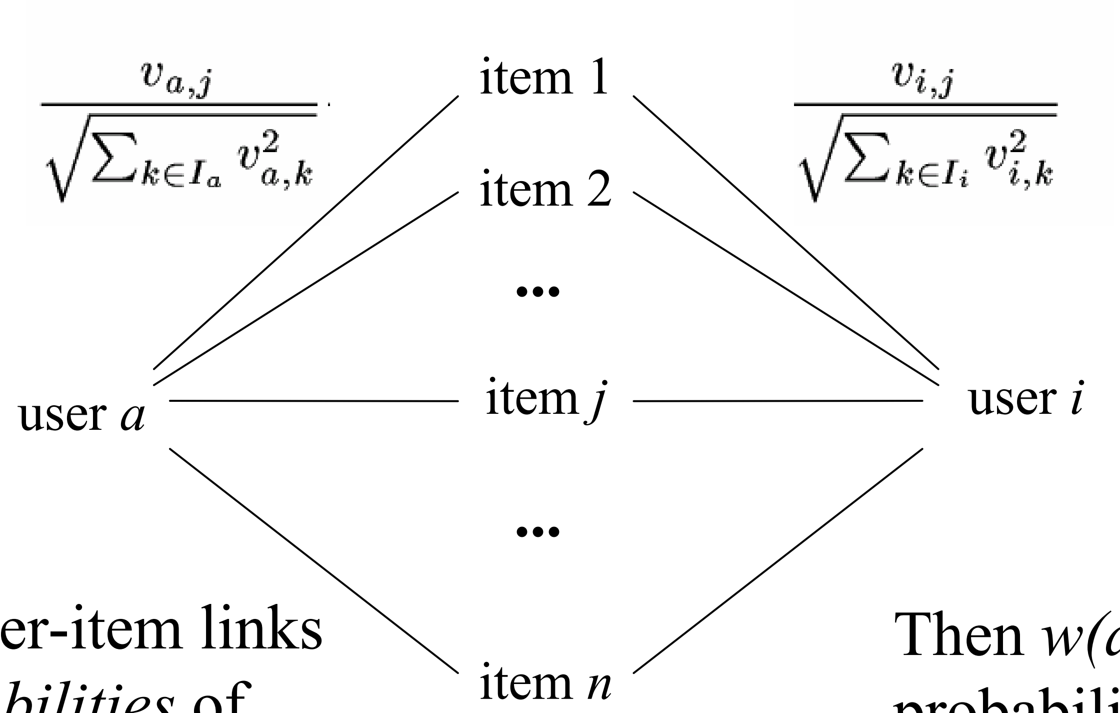
Let $\vec{A} = < \ldots, v(a,j), \ldots >$

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



Visualizing Cosine Distance

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



$\frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}}$

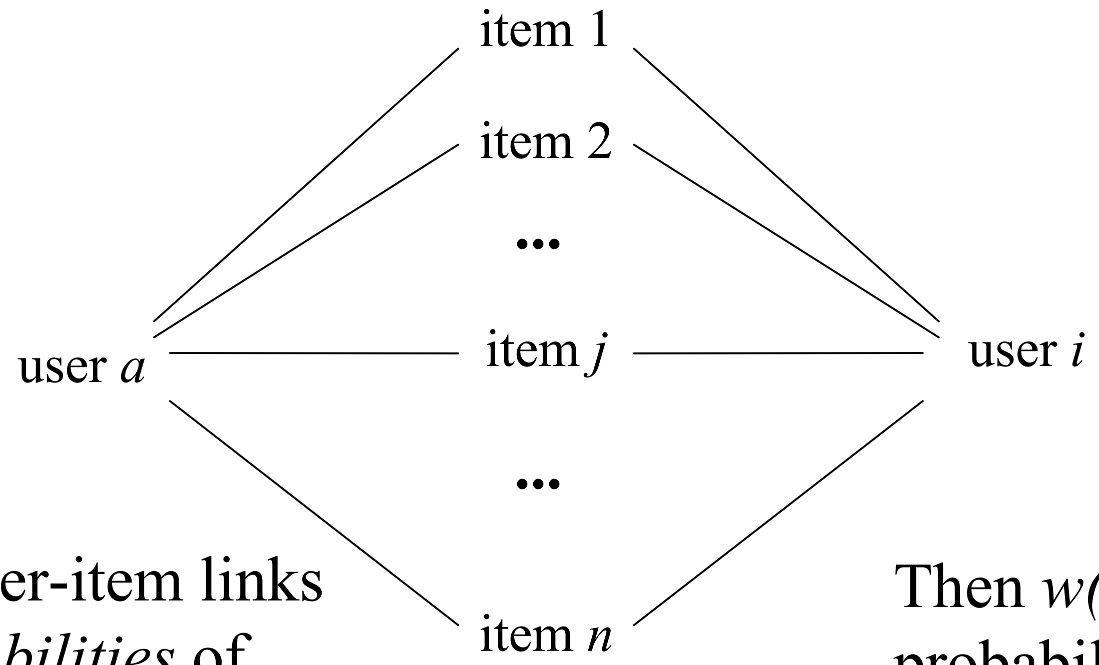
$\frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}}$

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



item 1

item 2

…

user a

item j

user i

…

item n

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

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

Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
- Algorithms for CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings

LIBRA Book Recommender

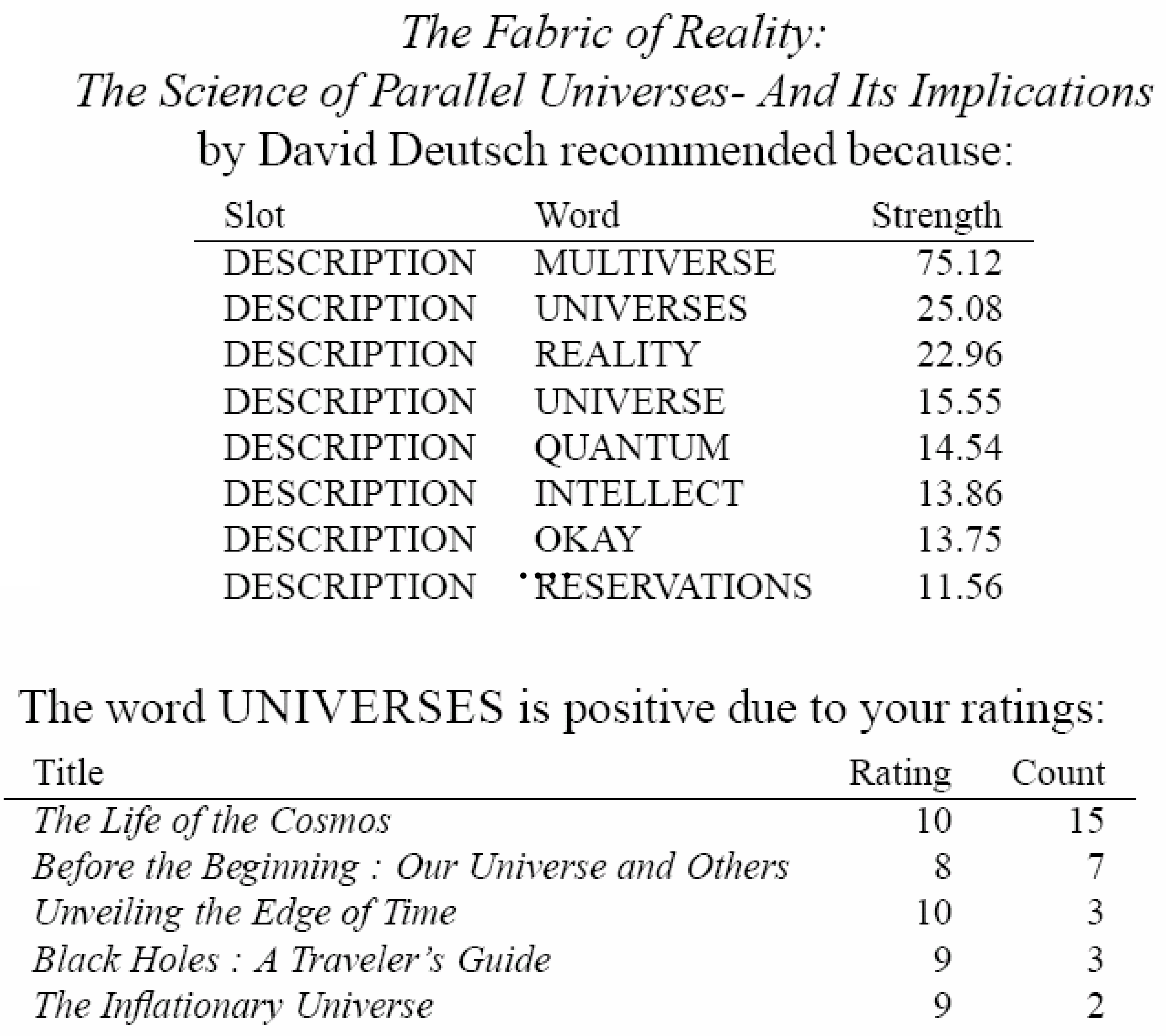
Content-Based Book Recommending Using Learning for Text Categorization. Raymond J. Mooney, Loriene Roy, Univ Texas/Austin; DL-2000

> [CF] assumes that a given user’s tastes are generally the same as another user ... Items that have not been rated by a sufficient number of users cannot be effectively recommended. Unfortunately, statistics on library use indicate that most books are utilized by very few patrons. ... [CF] approaches ... recommend popular titles, perpetuating homogeneity.... this approach raises concerns about privacy and access to proprietary customer data.

LIBRA Book Recommender

- Database of textual descriptions + meta-information about books (from Amazon.com's website)
  - title, authors, synopses, published reviews, customer comments, related authors, related titles, and subject terms.
- Users provides 1-10 rating for training books
- System learns a model of the user
  - Naive Bayes classifier predicts Prob(user rating>5|book)
- System explains ratings in terms of "informative features" and explains features in terms of examples

LIBRA Book Recommender



The Fabric of Reality:
The Science of Parallel Universes- And Its Implications
by David Deutsch recommended because:

Slot	Word	Strength
DESCRIPTION	MULTIVERSE	75.12
DESCRIPTION	UNIVERSES	25.08
DESCRIPTION	REALITY	22.96
DESCRIPTION	UNIVERSE	15.55
DESCRIPTION	QUANTUM	14.54
DESCRIPTION	INTELLECT	13.86
DESCRIPTION	OKAY	13.75
	...	
DESCRIPTION	RESERVATIONS	11.56

The word UNIVERSES is positive due to your ratings:

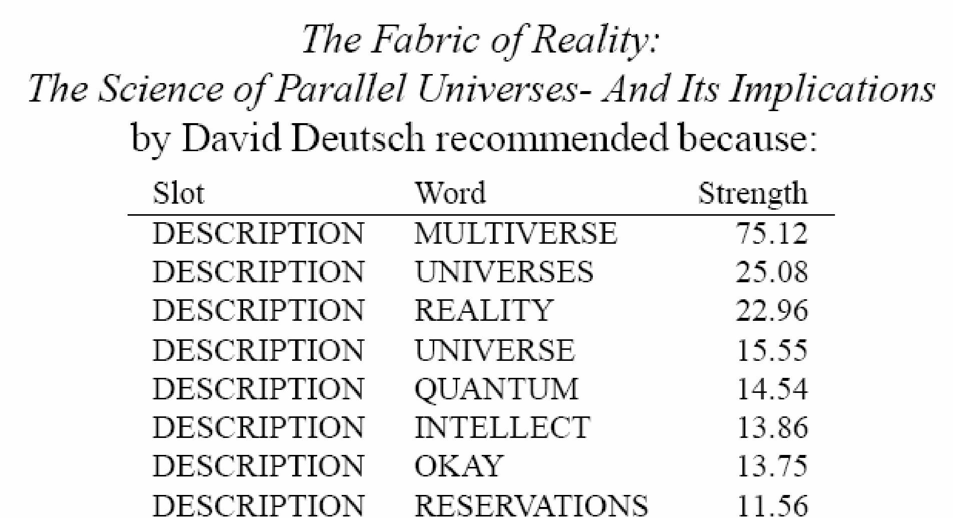
Title	Rating	Count
The Life of the Cosmos	10	15
Before the Beginning : Our Universe and Others	8	7
Unveiling the Edge of Time	10	3
Black Holes : A Traveler's Guide	9	3
The Inflationary Universe	9	2

LIBRA Book Recommender

Key differences from MovieRecommender:

- *vs* collaborative filtering, recommendation is based on properties of the *item being recommended,* not tastes of other users

- *vs* memory-based techniques, **LIBRA** builds an *explicit model* of the user's tastes (expressed as weights for different words)



The Fabric of Reality:
The Science of Parallel Universes- And Its Implications
by David Deutsch recommended because:

Slot	Word	Strength
DESCRIPTION	MULTIVERSE	75.12
DESCRIPTION	UNIVERSES	25.08
DESCRIPTION	REALITY	22.96
DESCRIPTION	UNIVERSE	15.55
DESCRIPTION	QUANTUM	14.54
DESCRIPTION	INTELLECT	13.86
DESCRIPTION	OKAY	13.75
DESCRIPTION	RESERVATIONS	11.56

....

LIBRA Book Recommender

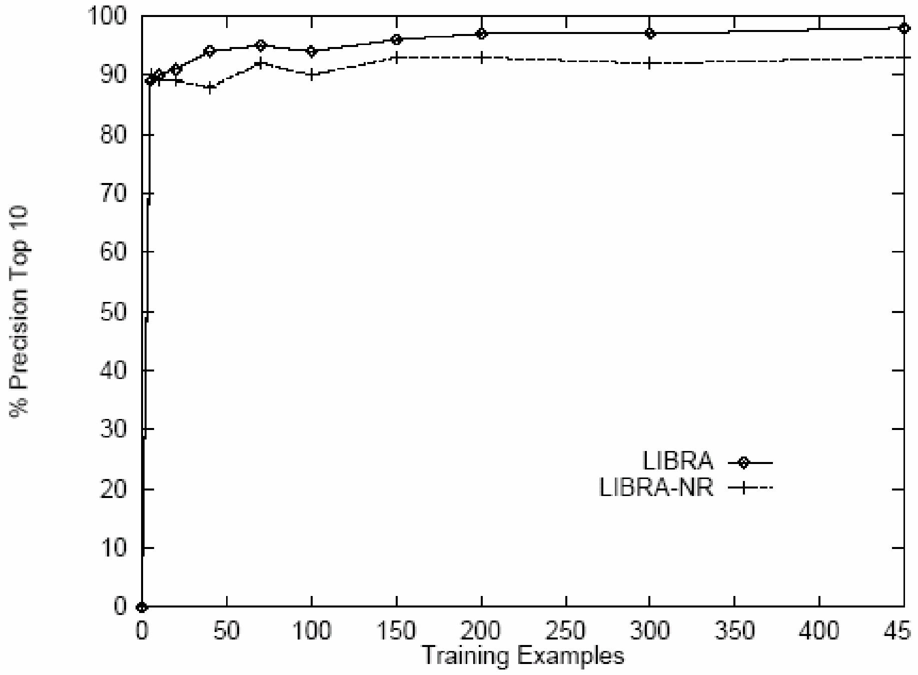


Figure 2: MYST **Precision at Top 10**

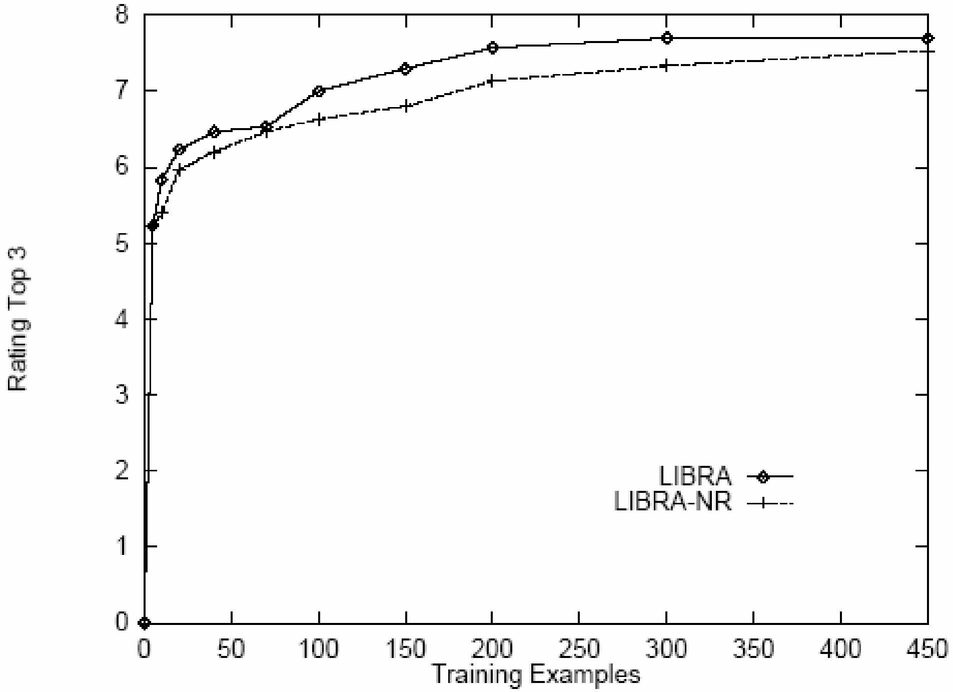
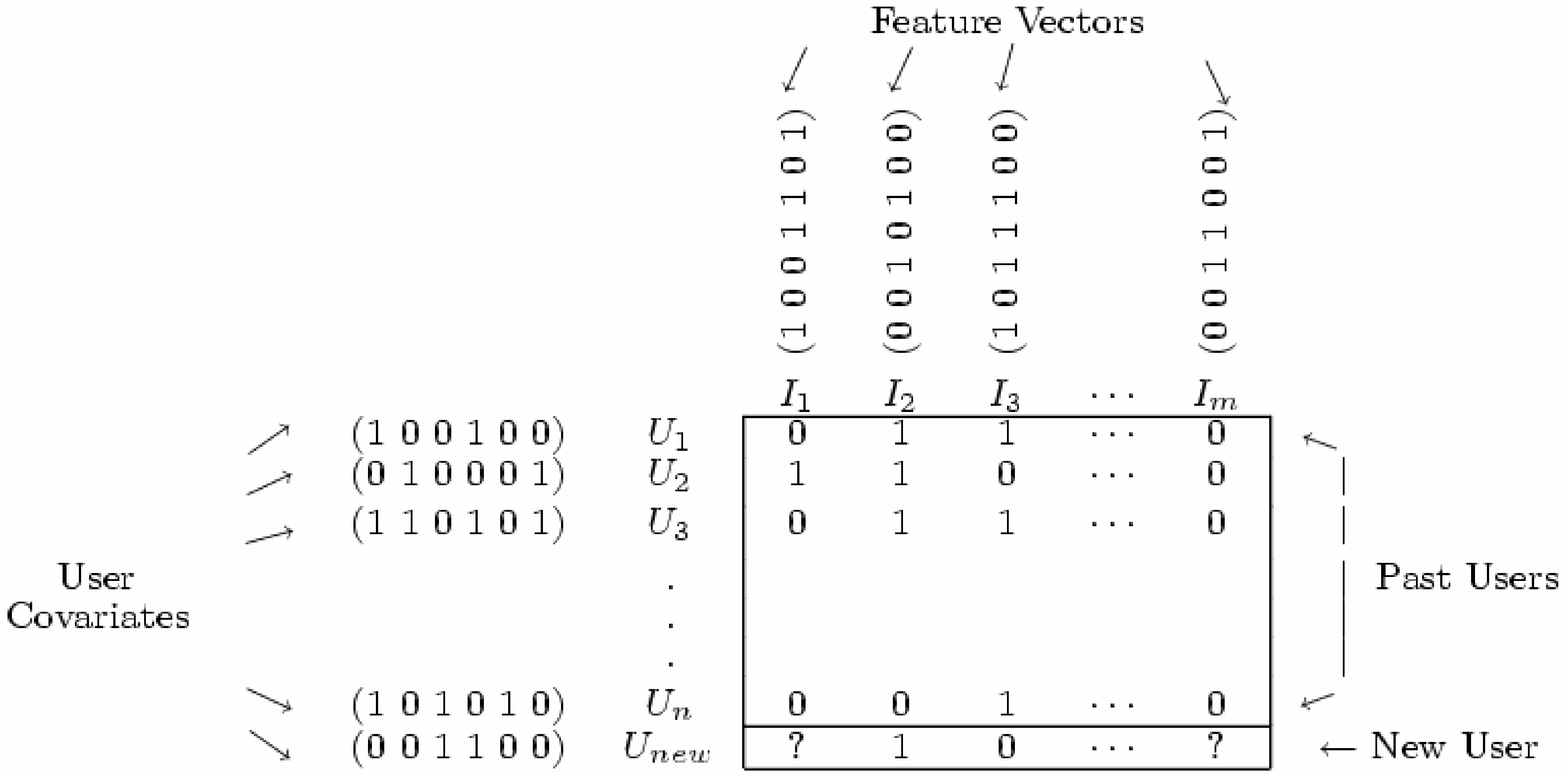


Figure 3: SF **Average Rating of Top 3**

LIBRA-NR = no related author/title features

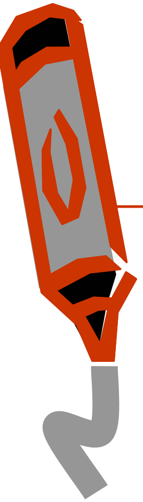
Collaborative + Content Filtering

(Basu et al, AAAI98; Condliff et al, AI-STATS99)



Feature Vectors

$$\begin{array}{ccc|ccccc|c}
 & & & (1\,0\,0\,1\,1\,0\,1) & (0\,0\,1\,0\,1\,0\,0) & (1\,0\,1\,1\,1\,0\,0) & & (0\,0\,1\,1\,0\,0\,1) & \\
 & & & I_1 & I_2 & I_3 & \cdots & I_m & \\
\hline
 & (1\,0\,0\,1\,0\,0) & U_1 & 0 & 1 & 1 & \cdots & 0 & \\
 & (0\,1\,0\,0\,0\,1) & U_2 & 1 & 1 & 0 & \cdots & 0 & \\
 & (1\,1\,0\,1\,0\,1) & U_3 & 0 & 1 & 1 & \cdots & 0 & \\
\text{User Covariates} & & \vdots & & & & & & \text{Past Users} \\
 & (1\,0\,1\,0\,1\,0) & U_n & 0 & 0 & 1 & \cdots & 0 & \\
\hline
 & (0\,0\,1\,1\,0\,0) & U_{new} & ? & 1 & 0 & \cdots & ? & \leftarrow \text{New User} \\
\hline
\end{array}$$

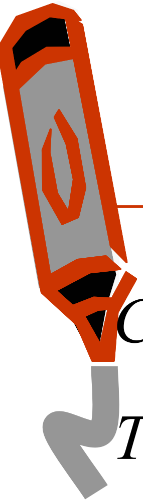


Collaborative + Content Filtering

(Basu et al, AAAI98; Condliff et al, AI-STATS99)

		Airplane	Matrix	Room with a View	...	Hidalgo
		comedy	action	romance	...	action
Joe	*27,M,70k*	9	7	2		7
Carol	*53,F,20k*	8		9		
…						
Kumar	*25,M,22k*	9	3			6
U_a	*48,M,81k*	4	7	?	?	?

Collaborative + Content Filtering As Classification (Basu, Hirsh, Cohen, AAAI98)



Classification task: map **(user,movie)** pair into **{likes,dislikes}**

Training data: known likes/dislikes

Test data: active users

Features: **any** properties of user/movie pair

		Airplane	**Matrix**	**Room with a View**	**…**	**Hidalgo**
		comedy	**action**	**romance**	**…**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
…						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	0	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

Examples: *genre(U,M), age(U,M), income(U,M),...*

- *genre(Carol,Matrix) = action*
- *income(Kumar,Hidalgo) = 22k/year*

Features: **any** properties of user/movie pair *(U,M)*

		Airplane	**Matrix**	**Room with a View**	**...**	**Hidalgo**
		comedy	**action**	**romance**	**...**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
...						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	0	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

Examples: *usersWhoLikedMovie(U,M):*

- *usersWhoLikedMovie(Carol,Hidalgo) = {Joe,...,Kumar}*
- *usersWhoLikedMovie(U_a, Matrix) = {Joe,...}*

Features: **any** properties of user/movie pair *(U,M)*

		Airplane	**Matrix**	**Room with a View**	**...**	**Hidalgo**
		comedy	**action**	**romance**	**...**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
...						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	0	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

Examples: *moviesLikedByUser(M,U):*

- *moviesLikedByUser(*,Joe) = {Airplane,Matrix,…,Hidalgo}*
- *actionMoviesLikedByUser(*,Joe)={Matrix,Hidalgo}*

Features: **any** properties of user/movie pair *(U,M)*

		Airplane	**Matrix**	**Room with a View**	**…**	**Hidalgo**
		comedy	**action**	**romance**	**…**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
…						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	0	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

genre={romance}, age=48, sex=male, income=81k,
usersWhoLikedMovie={Carol}, moviesLikedByUser={Matrix,Airplane}, ...

Features: **any** properties of user/movie pair *(U,M)*

		Airplane	**Matrix**	**Room with a View**	**...**	**Hidalgo**
		comedy	**action**	**romance**	**...**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
...						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	1	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

genre={romance}, age=48, sex=male, income=81k, usersWhoLikedMovie={Carol}, moviesLikedByUser={Matrix,Airplane}, ...

genre={action}, age=48, sex=male, income=81k, usersWhoLikedMovie = {Joe,Kumar}, moviesLikedByUser={Matrix,Airplane},...

		Airplane	**Matrix**	**Room with a View**	**...**	**Hidalgo**
		comedy	**action**	**romance**	**...**	**action**
Joe	*27,M,70k*	1	1	0		1
Carol	*53,F,20k*	1		1		0
…						
Kumar	*25,M,22k*	1	0	0		1
U_a	*48,M,81k*	1	1	?	?	?

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

genre={romance}, age=48, sex=male, income=81k, usersWhoLikedMovie={Carol}, moviesLikedByUser={Matrix,Airplane}, ...

genre={action}, age=48, sex=male, income=81k, usersWhoLikedMovie = {Joe,Kumar}, moviesLikedByUser={Matrix,Airplane},...

- Classification learning algorithm: rule learning (RIPPER)
  - If *NakedGun33/13* ∈*moviesLikedByUser* and *Joe* ∈ *usersWhoLikedMovie* and *genre=comedy* then predict *likes(U,M)*
  - If *age>12* and *age<17* and *HolyGrail* ∈ *moviesLikedByUser* ***and*** *director=MelBrooks* then predict *likes(U,M)*
  - If *Ishtar* ∈ *moviesLikedByUser* then predict *likes(U,M)*

Collaborative + Content Filtering As Classification (Basu et al, AAAI98)

Classification learning algorithm: rule learning (RIPPER)

- If *NakedGun33/13* ∈*moviesLikedByUser* and *Joe* ∈ *usersWhoLikedMovie* and *genre=comedy* then predict *likes(U,M)*

- If *age>12* and *age<17* and *HolyGrail* ∈ *moviesLikedByUser* ***and*** *director=MelBrooks* then predict *likes(U,M)*

- If *Ishtar* ∈*moviesLikedByUser* then predict *likes(U,M)*

- Important difference from memory-based approaches:
  - again, Ripper builds an explicit model—of how user's tastes relate items, and to the tastes of other users

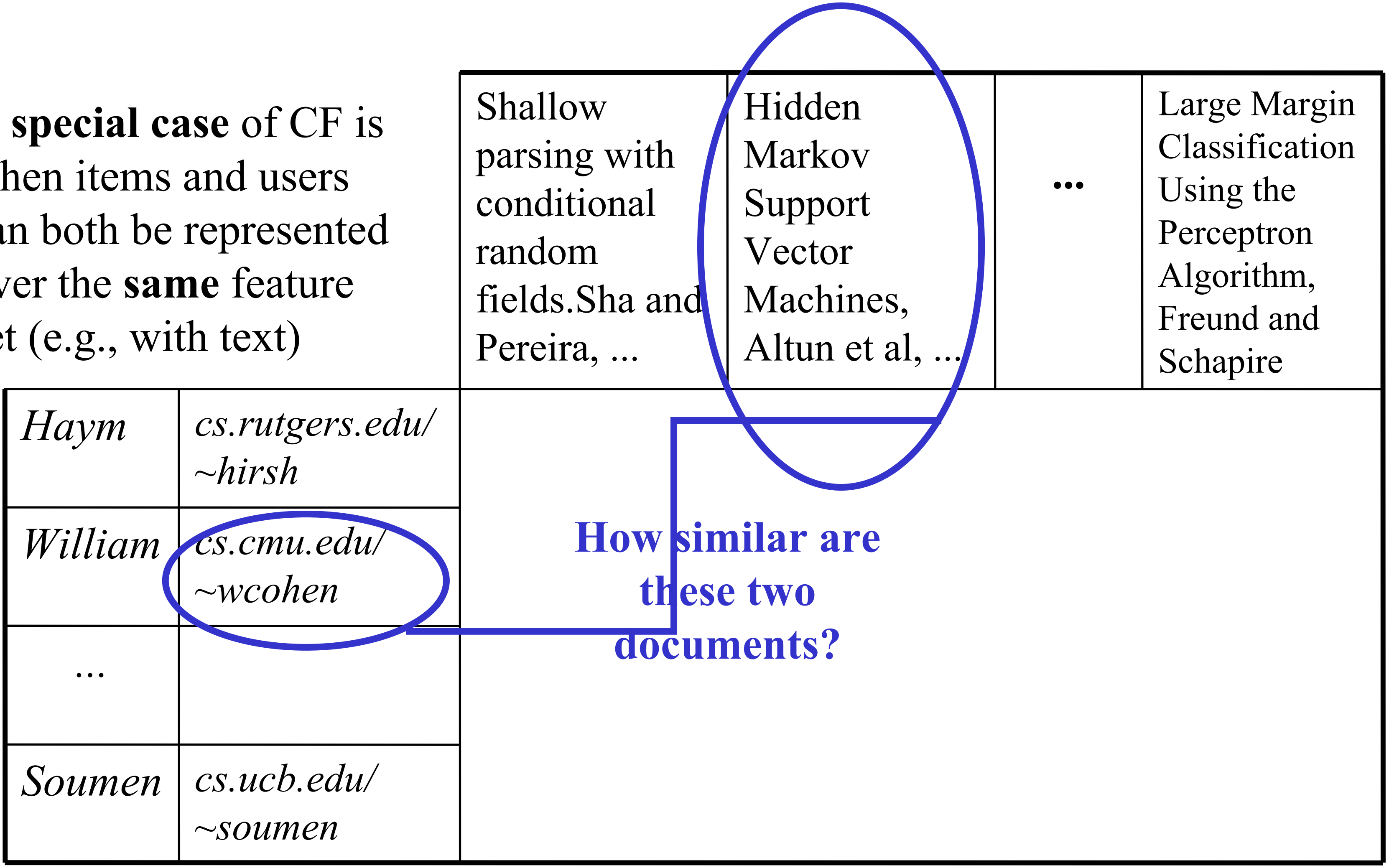
Basu et al 98 - results

- Evaluation:
  - Predict *liked(U,M)=“M in top quartile of U’s ranking*” from features, evaluate recall and precision
  - Features:
    - Collaborative*: UsersWhoLikedMovie, UsersWhoDislikedMovie, MoviesLikedByUser*
    - Content: *Actors, Directors, Genre, MPAA rating, ...*
    - Hybrid: *ComediesLikedByUser, DramasLikedByUser, UsersWhoLikedFewDramas, ...*
- Results: at same level of recall (about 33%)
  - *Ripper* with collaborative features only is worse than the original *MovieRecommender* (by about 5 pts precision – 73 vs 78)
  - *Ripper* with hybrid features is better than *MovieRecommender* (by about 5 pts precision)

Technical Paper Recommendation

(Basu, Hirsh, Cohen, Neville-Manning, JAIR 2001)

A **special case** of CF is when items and users can both be represented over the **same** feature set (e.g., with text)



		Shallow parsing with conditional random fields.Sha and Pereira, …	Hidden Markov Support Vector Machines, Altun et al, …	…	Large Margin Classification Using the Perceptron Algorithm, Freund and Schapire
Haym	*cs.rutgers.edu/ ~hirsh*				
William	*cs.cmu.edu/ ~wcohen*				
…					
Soumen	*cs.ucb.edu/ ~soumen*				

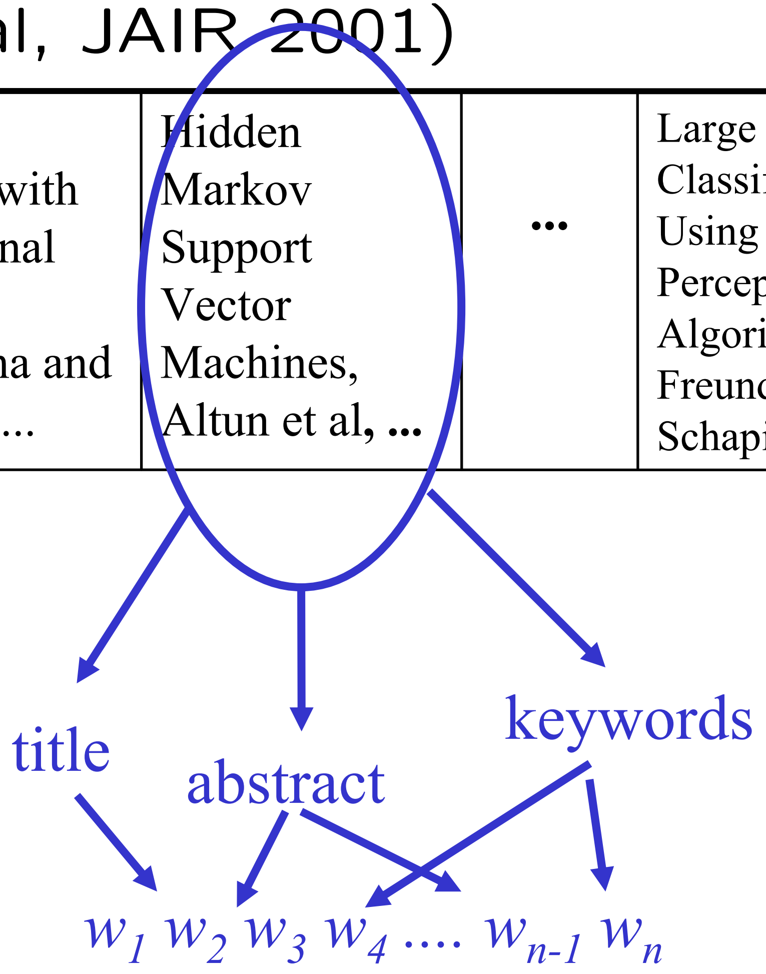
How similar are these two documents?

Technical Paper Recommendation

(Basu et al, JAIR 2001)

A **special case** of CF is when items and users can both be represented over the **same** feature set (e.g., with text)

		Shallow parsing with conditional random fields.Sha and Pereira, …	Hidden Markov Support Vector Machines, Altun et al, **…**	**…**	Large Margin Classification Using the Perceptron Algorithm, Freund and Schapire
Haym	*cs.rutgers.edu/ ~hirsh*				
William	*cs.cmu.edu/ ~wcohen*				
…					
Soumen	*cs.ucb.edu/ ~soumen*				

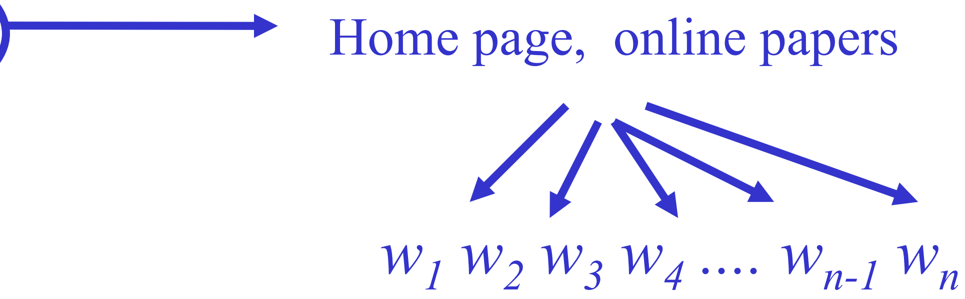


Technical Paper Recommendation

(Basu et al, JAIR 2001)

A **special case** of CF is when items and users can both be represented over the **same** feature set (e.g., with text)

		Shallow parsing with conditional random fields.Sha and Pereira, …	Hidden Markov Support Vector Machines, Altun et al, **…**	**…**	Large Margin Classification Using the Perceptron Algorithm, Freund and Schapire
Haym	*cs.rutgers.edu/ ~hirsh*				
William	*cs.cmu.edu/ ~wcohen*				
…					
Soumen	*cs.ucb.edu/ ~soumen*				



Home page, online papers

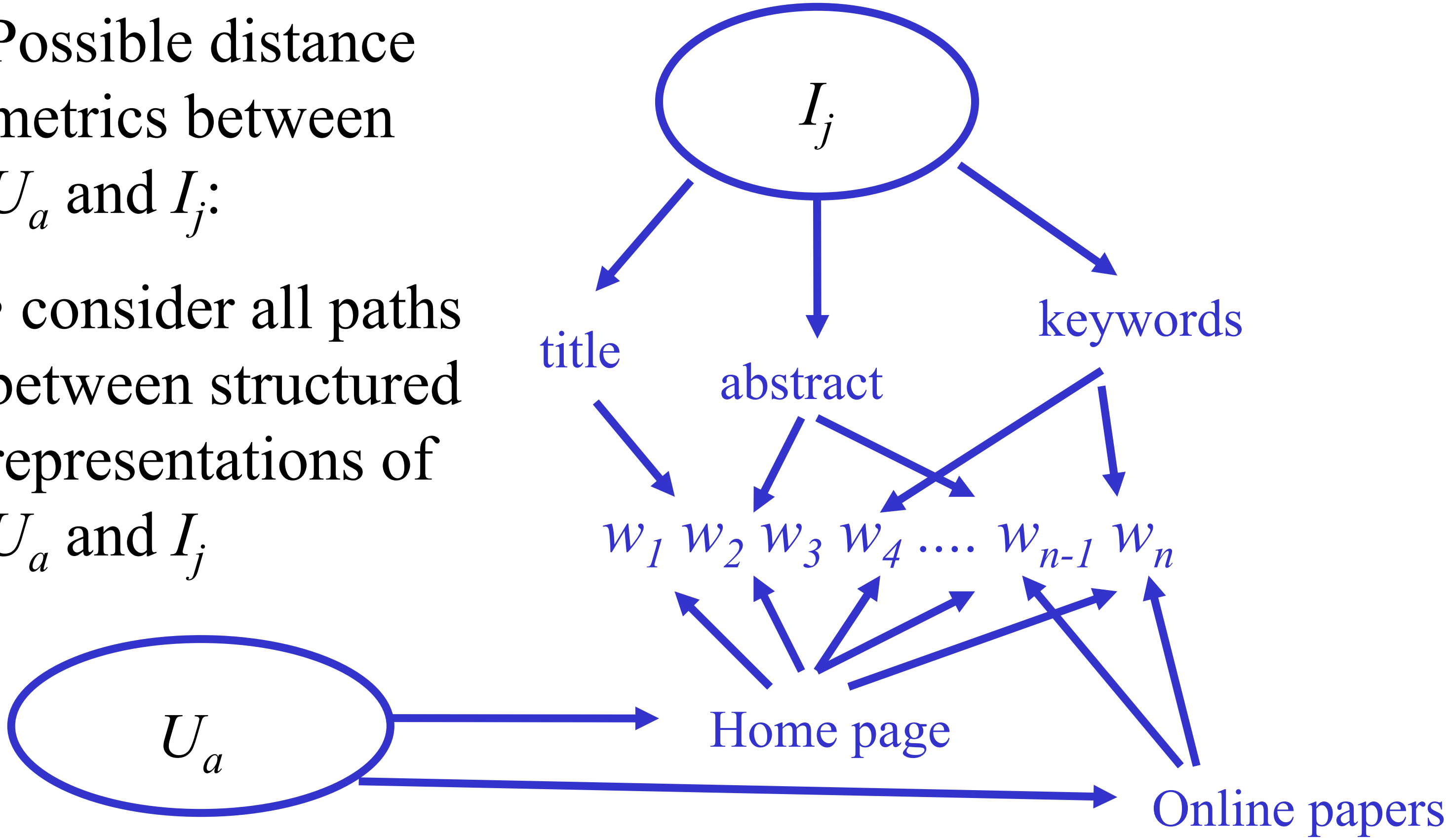
$w_1\ w_2\ w_3\ w_4\ \ldots.\ w_{n-1}\ w_n$

Technical Paper Recommendation

(Basu et al, JAIR 2001)

Possible distance metrics between U_a and I_j:

- consider all paths between structured representations of U_a and I_j



I_j

title

abstract

keywords

w_1 w_2 w_3 w_4 …. w_{n-1} w_n

U_a

Home page

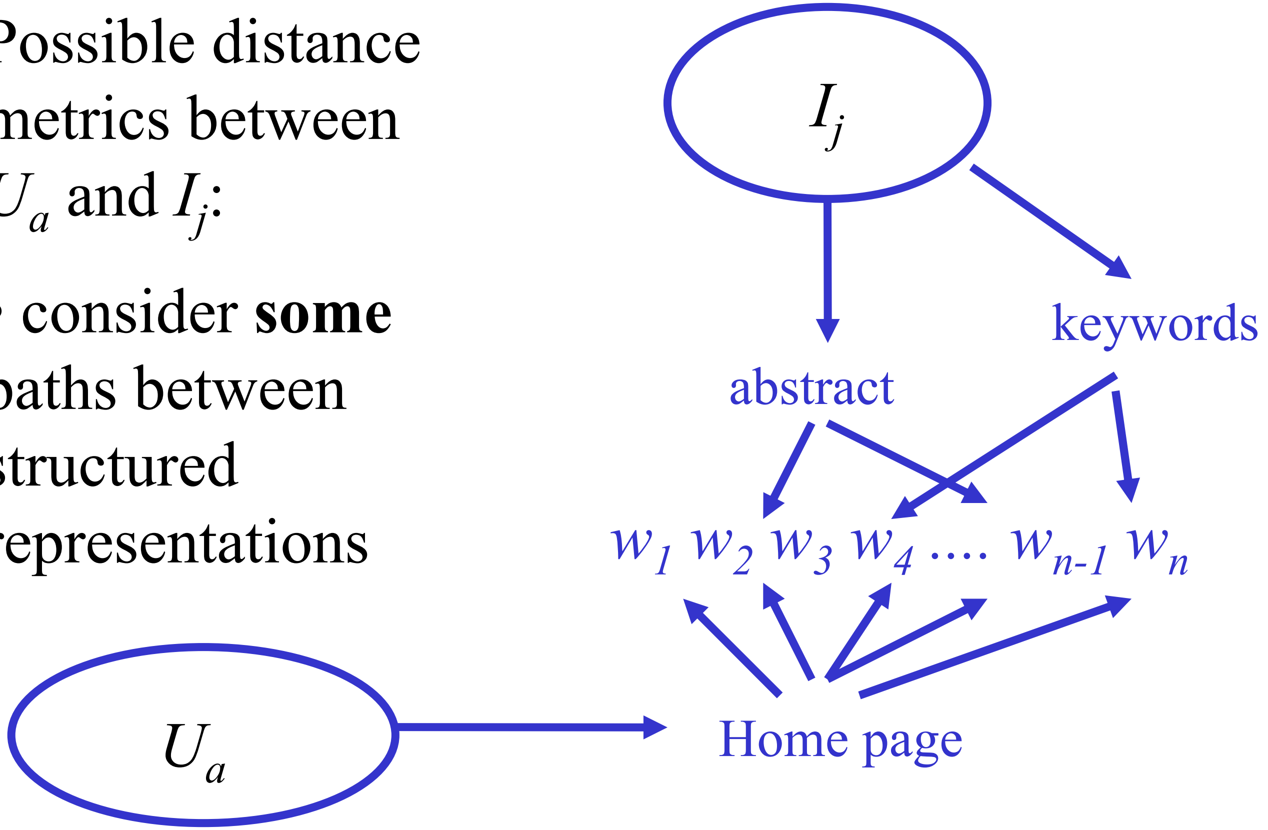
Online papers

Technical Paper Recommendation

(Basu et al, JAIR 2001)

Possible distance metrics between U_a and I_j:

- consider **some** paths between structured representations

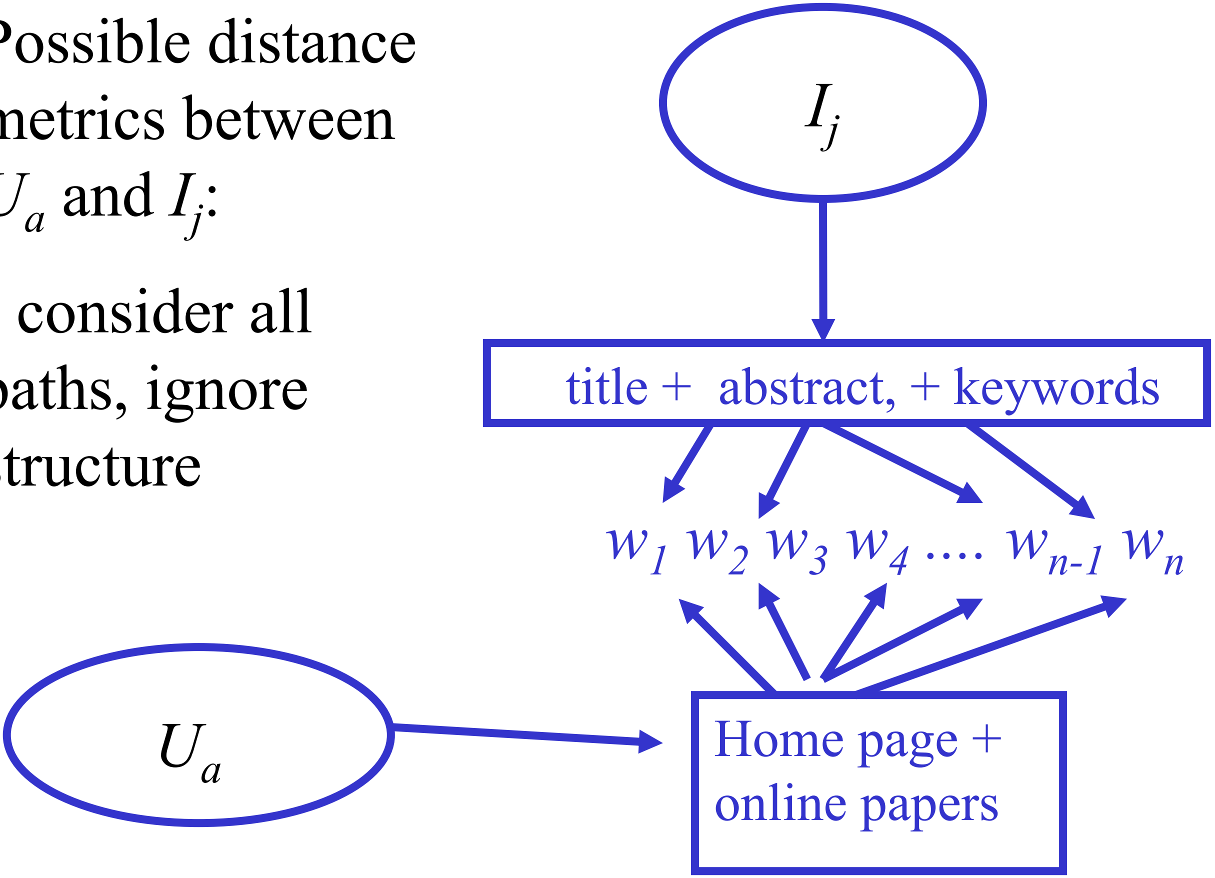


Technical Paper Recommendation

(Basu et al, JAIR 2001)

Possible distance metrics between U_a and I_j:

- consider all paths, ignore structure



I_j

title + abstract, + keywords

w_1 w_2 w_3 w_4 …. w_{n-1} w_n

U_a

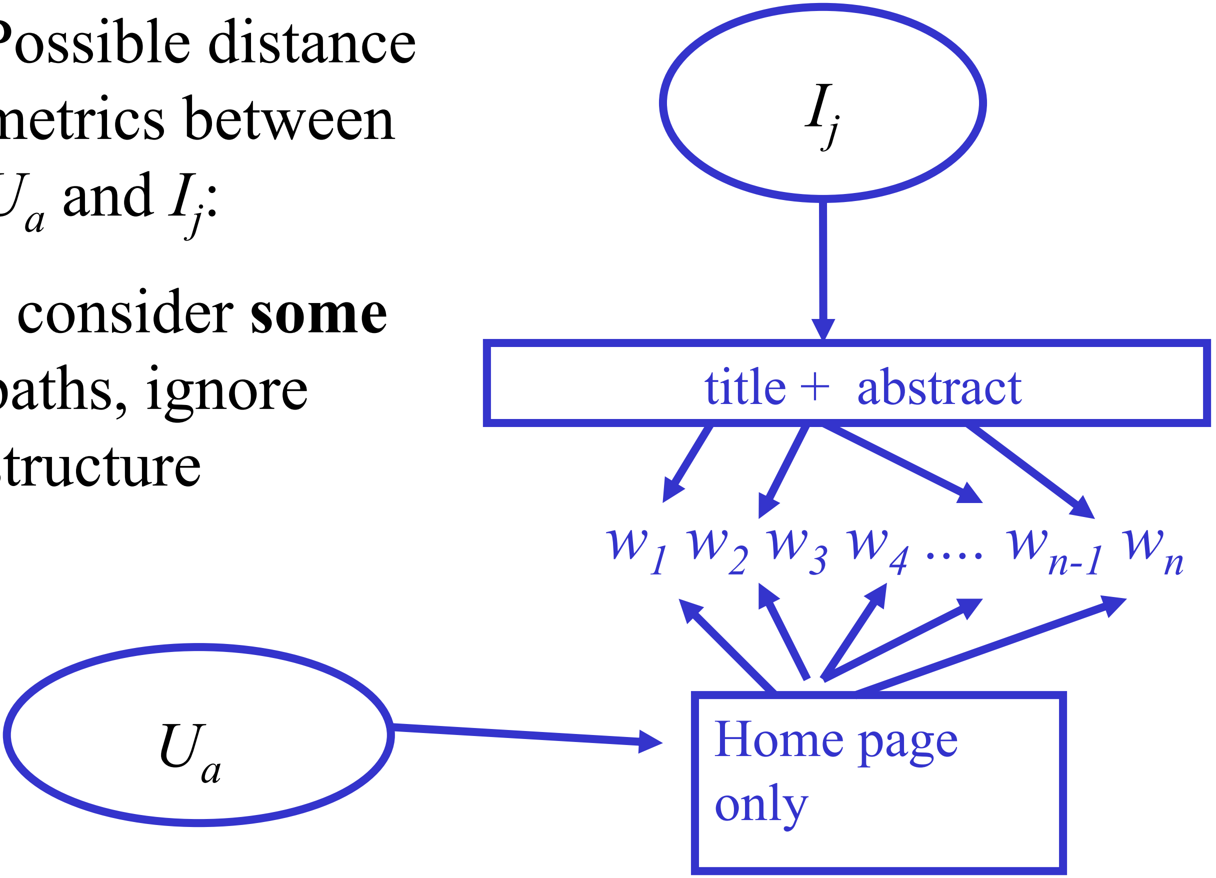
Home page + online papers

Technical Paper Recommendation

(Basu et al, JAIR 2001)

Possible distance metrics between U_a and I_j:

- consider **some** paths, ignore structure



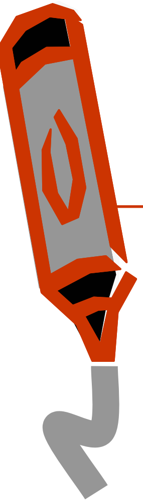
I_j

title + abstract

w_1 w_2 w_3 w_4 …. w_{n-1} w_n

U_a

Home page only



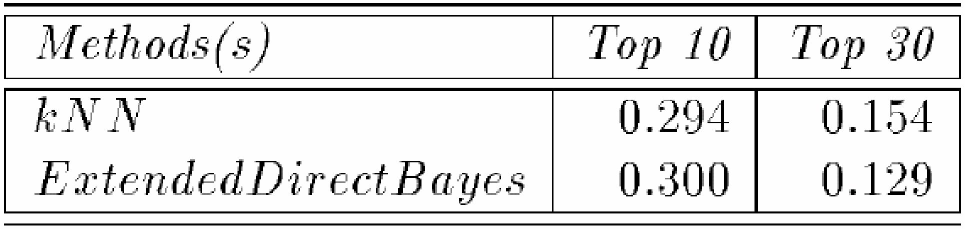
Technical Paper Recommendation

(Basu et al, JAIR 2001)

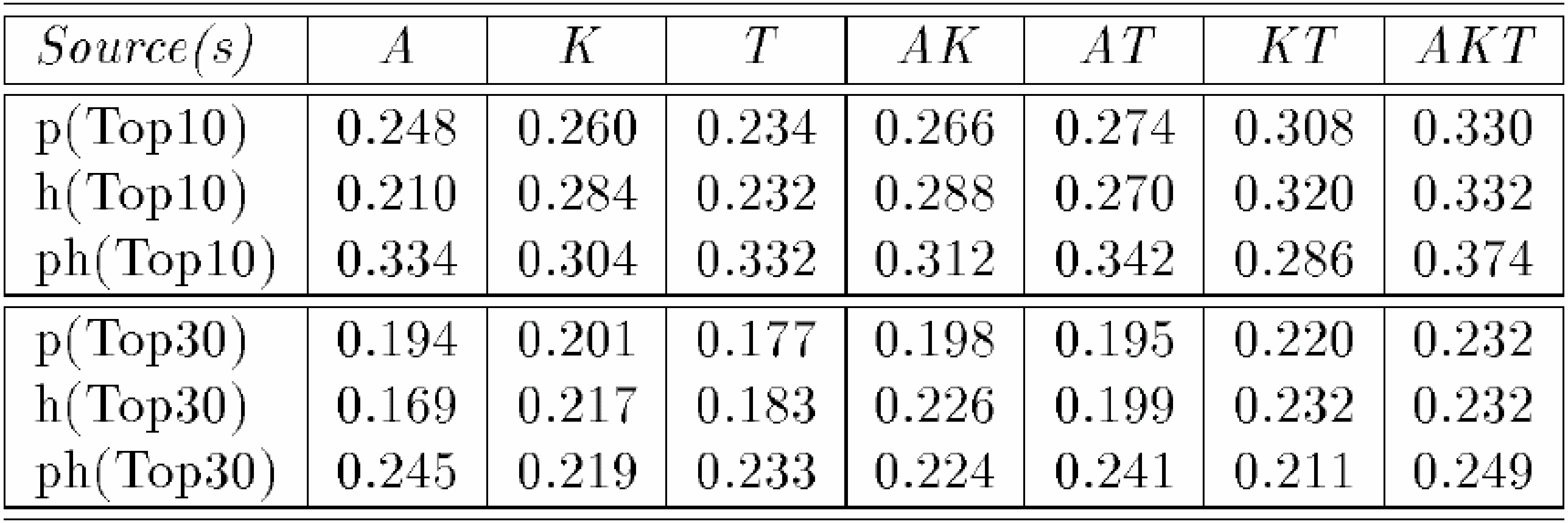
- Use WHIRL (Datalog + built-in cosine distances) to formulate structure similarity queries
  - Product of TFIDF-weighted cosine distances over each part of structure
- Evaluation
  - Try and predict stated reviewer preferences in AAAI self-selection process
    - Noisy, since not all reviewers examine all papers
  - Measure precision in top 10, and top 30

Technical Paper Recommendation

R 2001)



Methods(s)	*Top 10*	*Top 30*
kNN	0.294	0.154
ExtendedDirectBayes	0.300	0.129



Source(s)	*A*	*K*	*T*	*AK*	*AT*	*KT*	*AKT*
p(Top10)	0.248	0.260	0.234	0.266	0.274	0.308	0.330
h(Top10)	0.210	0.284	0.232	0.288	0.270	0.320	0.332
ph(Top10)	0.334	0.304	0.332	0.312	0.342	0.286	0.374
p(Top30)	0.194	0.201	0.177	0.198	0.195	0.220	0.232
h(Top30)	0.169	0.217	0.183	0.226	0.199	0.232	0.232
ph(Top30)	0.245	0.219	0.233	0.224	0.241	0.211	0.249

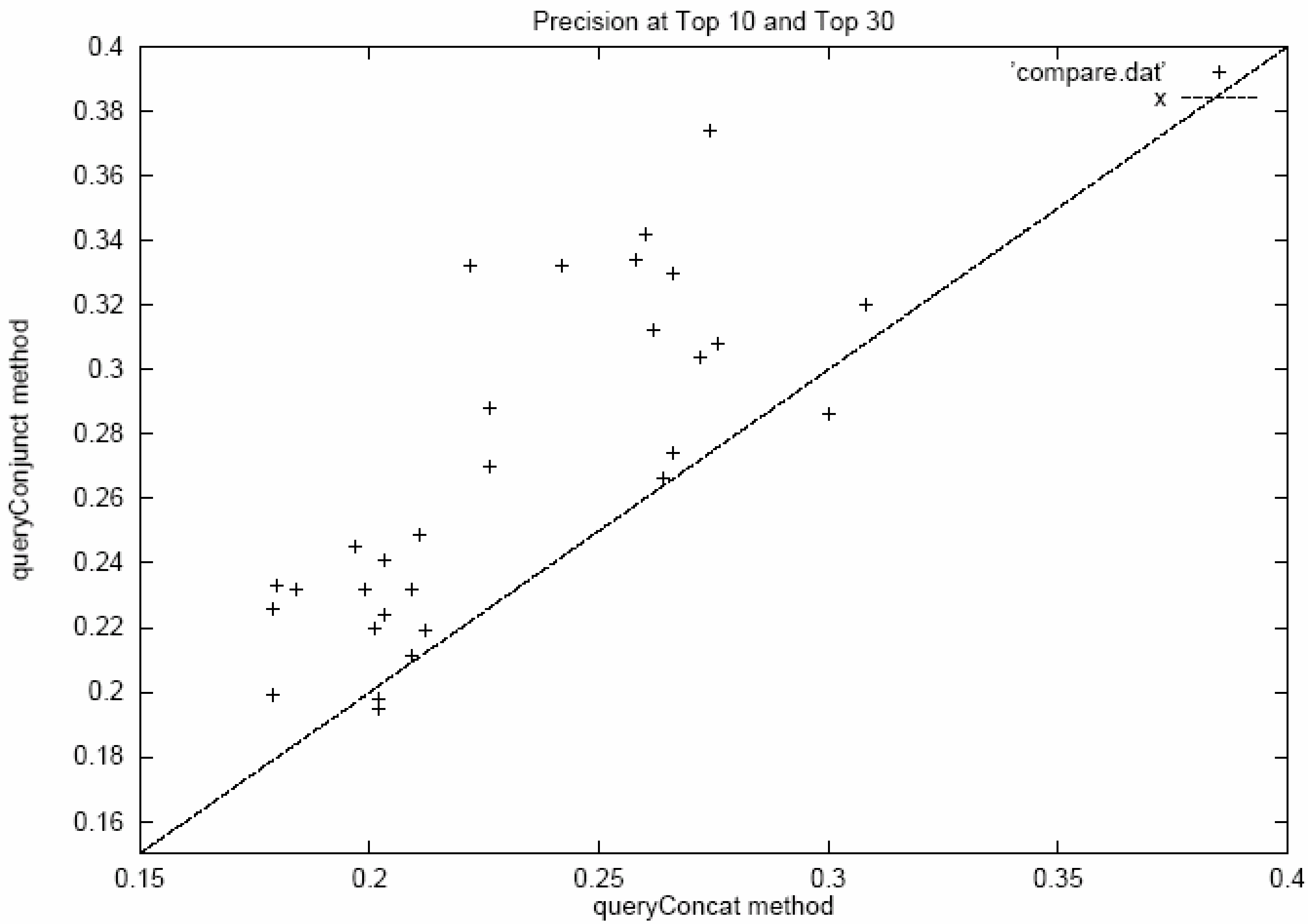
p=papers, h=homePage

A=abstract, K=keywords, T=title

structured similarity queries with WHIRL

Technical Paper Recommendation

(Basu et al., JAIR 2001)



Structure *vs* no structure

Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
  - Combining CF and content filtering
  - CF as matching content and user
- Algorithms for CF
  - Ranking-based CF
  - Probabilistic model-based CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings

Learning to Order

(Cohen,Schapire,Singer JAIR 99)

- *Ordering Example*: a pair *(x,y)* where
  - The “problem” *x* is a set of objects
  - The “solution” *y* is a partial order over *x*
- *Loss function: Loss(y,y*)* is number of incorrectly ordered pairs *a¡b*
- Learner uses ordering examples to improve performance.

- Outline of Cohen et al 99:
  - Learn a binary relation *PREFER(a,b)* = “*a* should precede *b*”
  - Given a new set *x* to order, construct the (possibly inconsistent) pairwise preferences, then find a (nearly) optimal total ordering given the pairs.
  - Formal guarantees on learning and ordering algorithm imply a performance guarantee for the whole system

Learning to Order

(Cohen et al JAIR 99)

- *Learning to Order Things,* Cohen, Schapire, Singer, JAIR 1999.

- Task: given a set of objects *X*, find a "good" ranking of *X*
- Inputs:
  - – On each run, a set of candidate (partial) orderings over *X*, to choose among and/or combine
  - – As training data triples $(X_1,F_1,\Phi_1),\ldots,(X_m,F_m,\Phi_m)$, where each *X* is set of objects to order; *F* is set of "feature" orderings $f_1,\ldots,f_n$, and Φ is the desired ordering of *X*.

Learning to Order

(Cohen et al JAIR 99)

- Outline:
  - – Approach for constructing linear combinations of “feature” orderings
    - Result is “preference” relation PREFER(*x*,*x*’*)*
  - – Approach for learning linear combinations
  - – Approach for converting PREFER to approximately optimal mapping
  - – Formal results of (nearly) optimal combination-learner and bounds on overall performance.

Learning to Order

(Cohen et al JAIR 99)

- Ranking functions are graphs with edge weights in [0,1].
- Weighted combination of two ordering

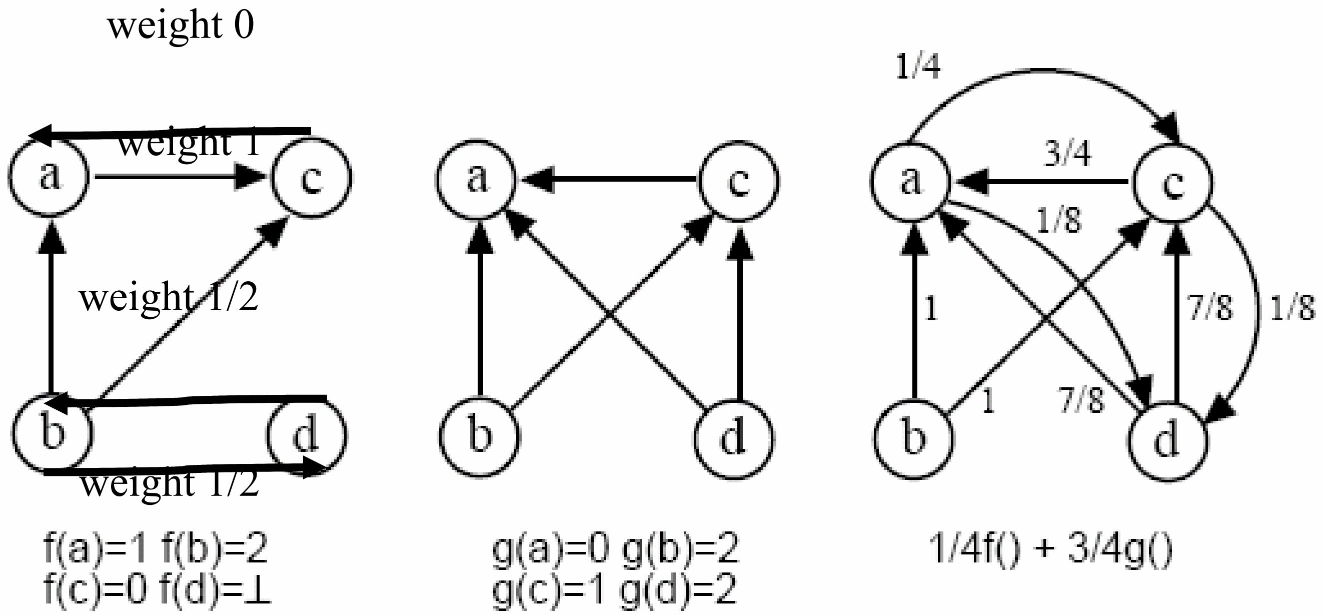


Figure 1: Left and middle: Two ordering functions and their graph representation. Right: The graph representation of the preference function created by a weighted ($\frac{1}{4}$ and $\frac{3}{4}$) combination of the two functions. Edges with weight of $\frac{1}{2}$ or 0 are omitted.

Learning to Order

(Cohen et al JAIR 99)

- Outline:
  - Approach for constructing linear combinations of “feature” orderings
    - Result is “preference” relation PREF(*x,x’)*
  - Approach for learning linear combinations
    - Natural extension of existing learning methods
  - Approach for converting PREFER to approximately optimal mapping: total order ρ that minimizes

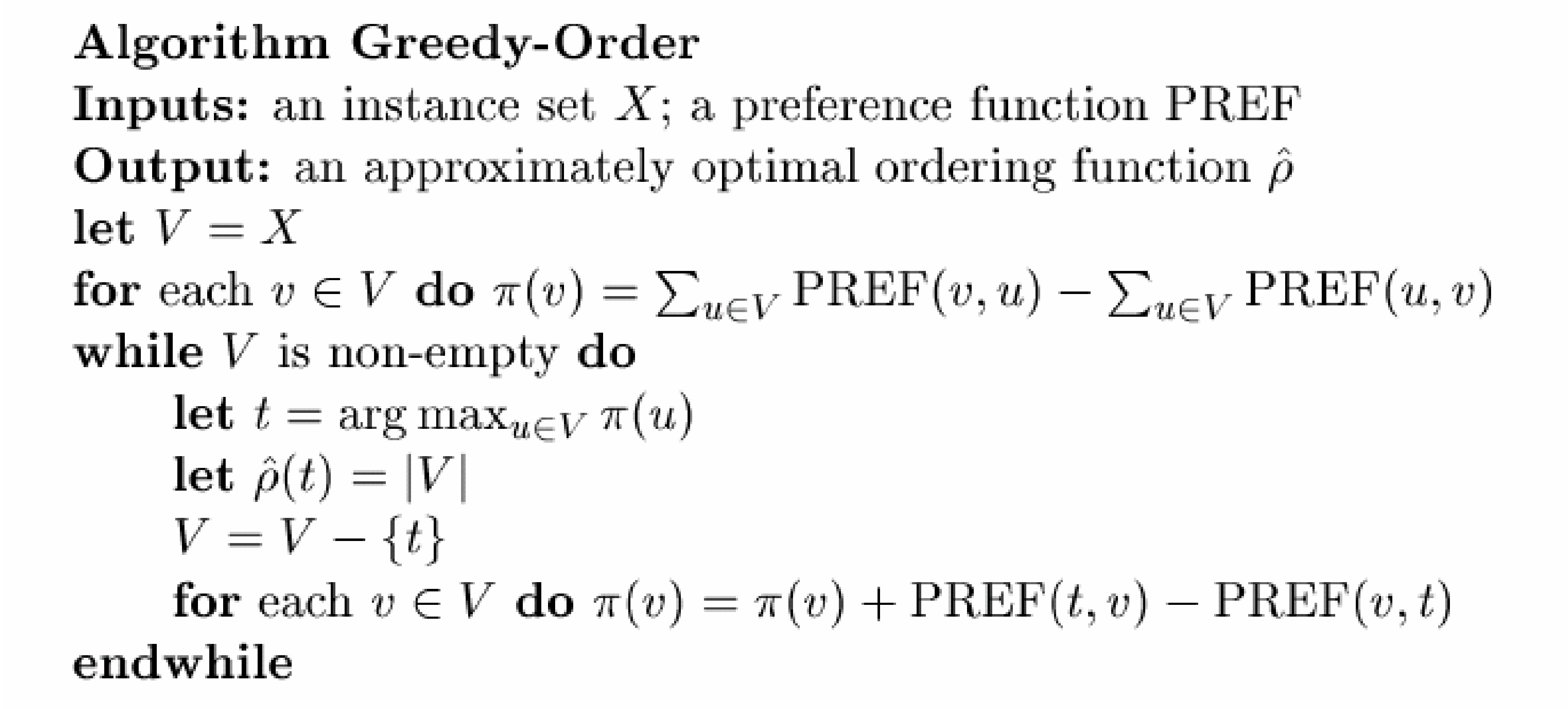
$$\mathrm{DISAGREE}(\rho, \mathrm{PREF}) = \sum_{u,v:\rho(u)>\rho(v)} (1 - \mathrm{PREF}(u,v))$$

- Unfortunately this is NP-Hard...

Learning to Order

(Cohen et al JAIR 99)

Fortunately, a “potential-greedy” algorithm obtains good results
(within factor of 2x the optimal agreement weight, which is tight)



```
Algorithm Greedy-Order
Inputs: an instance set $X$; a preference function PREF
Output: an approximately optimal ordering function $\hat{\rho}$
let $V = X$
for each $v \in V$ do $\pi(v) = \sum_{u \in V} \mathrm{PREF}(v,u) - \sum_{u \in V} \mathrm{PREF}(u,v)$
while $V$ is non-empty do
    let $t = \arg\max_{u \in V} \pi(u)$
    let $\hat{\rho}(t) = |V|$
    $V = V - \{t\}$
    for each $v \in V$ do $\pi(v) = \pi(v) + \mathrm{PREF}(t,v) - \mathrm{PREF}(v,t)$
endwhile
```

Learning to Order

(Cohen et al JAIR 99)

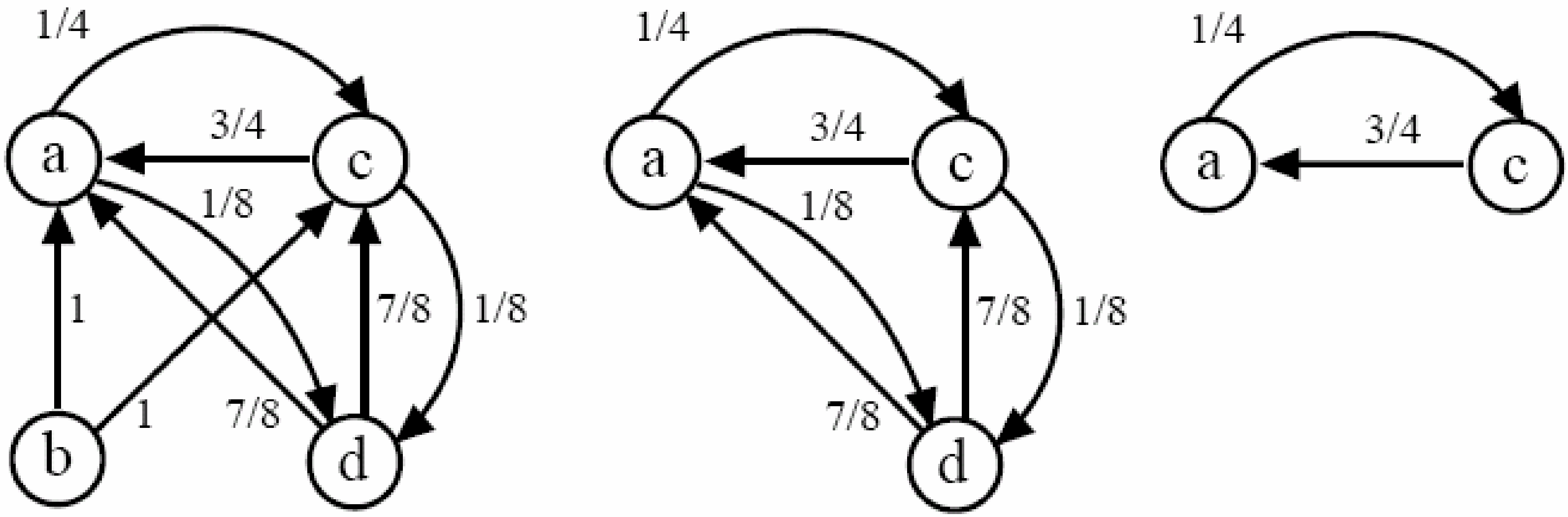
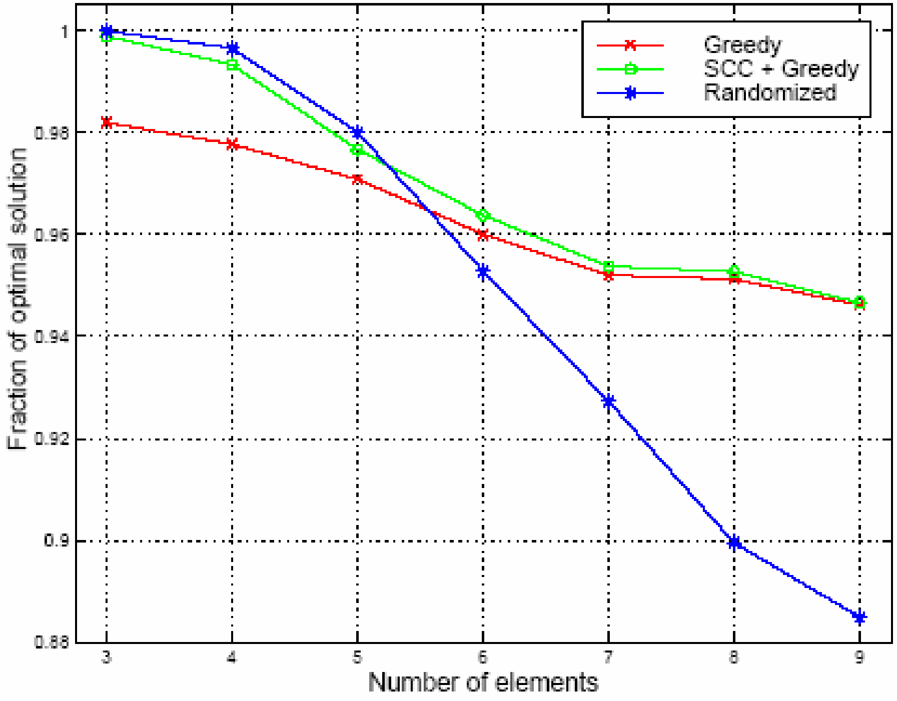


Figure 4: Behavior of the greedy ordering algorithm. The leftmost graph is the original input. From this graph, node b will be assigned maximal rank and deleted, leading to the middle graph; from this graph, node d will deleted, leading to the rightmost graph. In the rightmost graph, node c will be ranked ahead of node a, leading the total ordering $b > d > c > a$.

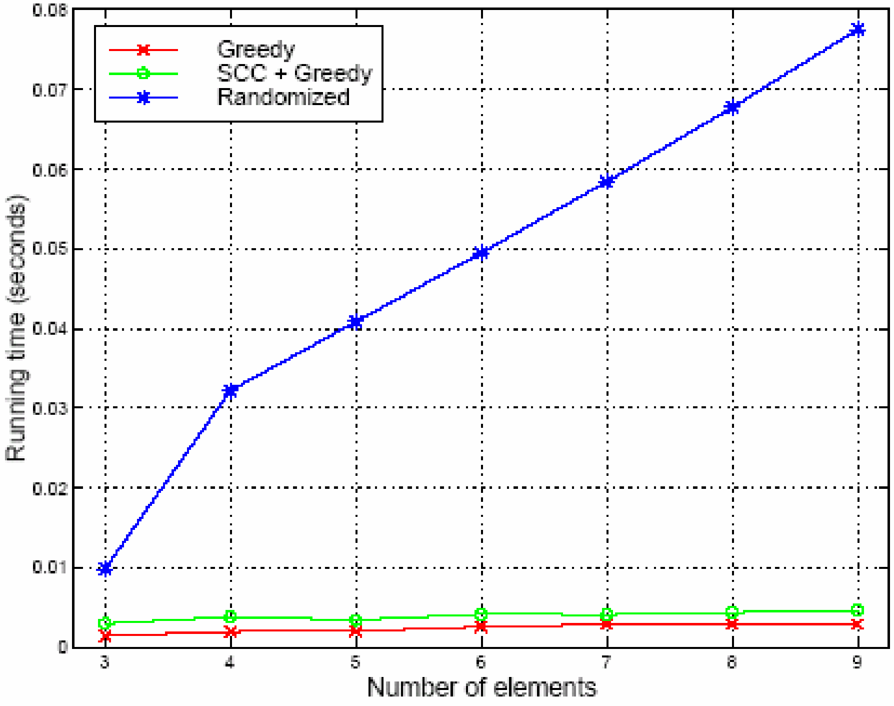
Learning to Order

(Cohen et al JAIR 99)

goodness *vs* optimal



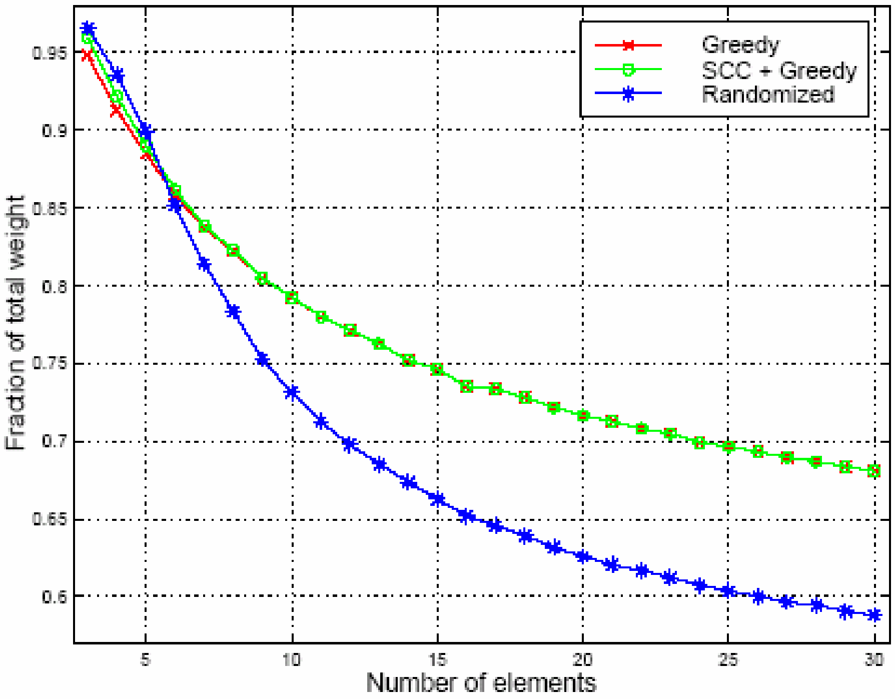
run-time



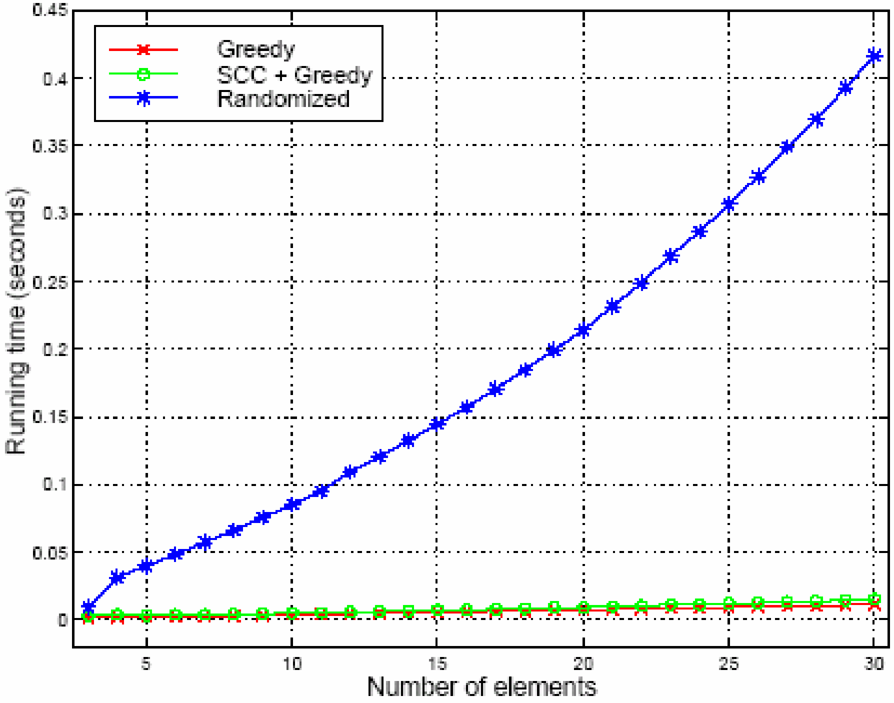
Learning to Order

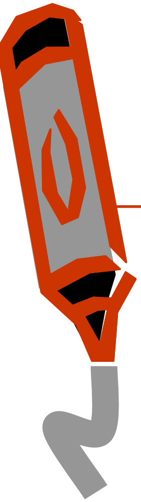
(Cohen et al JAIR 99)

goodness *vs* total



run-time





Learning to Order for CF

(Freund,Iyer,Schapire,Singer JMLR 01)

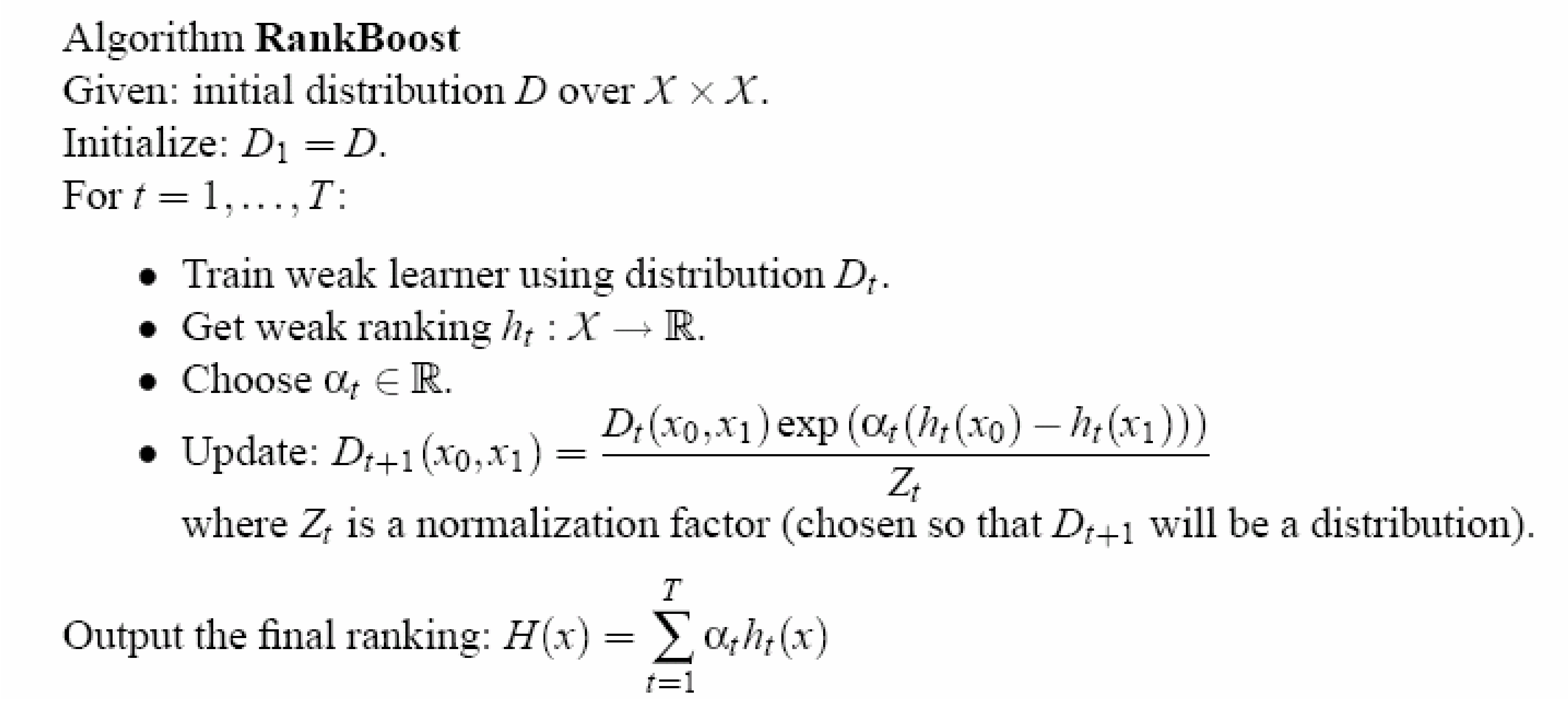
- A flaw in rating-based CF data
  - users tend to rate on different scales
  - this makes ratings hard to aggregate and transfer
- A solution:
  - disbelieve (ignore) a user's *absolute* ratings
  - believe (use in training) *relative* values
    - e.g., if user rates item j_1 at “5” and item j_2 as “8” then believe j_1 is preferred to j_2 .
  - i.e., treat CF as a problem of *learning to rank items.*

Learning to Order for CF

- The formal model:
  - objects to rank (e.g. movies) are in set *X*
  - *features* of object are ranking functions *f_1,f_2,..*
    - if *f(x) ¿ f(x')* then *x* is preferred to *x'*
    - *f(x)* can be undefined (*x* is unrated)
  - *training data* is a partial function *Φ(x, x')*
    - positive iff *x* should be preferred to *x'*
  - ranking loss: *D(x,x')* is distribution over pairs *x,x'* where *x* is preferred to *x'*, and *$rloss_D(H)$* is $\Pr_{x\sim D(x,x')}[H(x)\leq H(x')]$

Learning to Order for CF

Assume a “weak learner”, which given a weighted set of examples *$\Phi(x,x')$* finds a better-than-useless *total* ranking function *h*



```
Algorithm RankBoost
Given: initial distribution $D$ over $X \times X$.
Initialize: $D_1 = D$.
For $t = 1, \ldots, T$:
    • Train weak learner using distribution $D_t$.
    • Get weak ranking $h_t : X \to \mathbb{R}$.
    • Choose $\alpha_t \in \mathbb{R}$.
    • Update: $D_{t+1}(x_0,x_1) = \frac{D_t(x_0,x_1)\exp(\alpha_t(h_t(x_0) - h_t(x_1)))}{Z_t}$
      where $Z_t$ is a normalization factor (chosen so that $D_{t+1}$ will be a distribution).
Output the final ranking: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
```

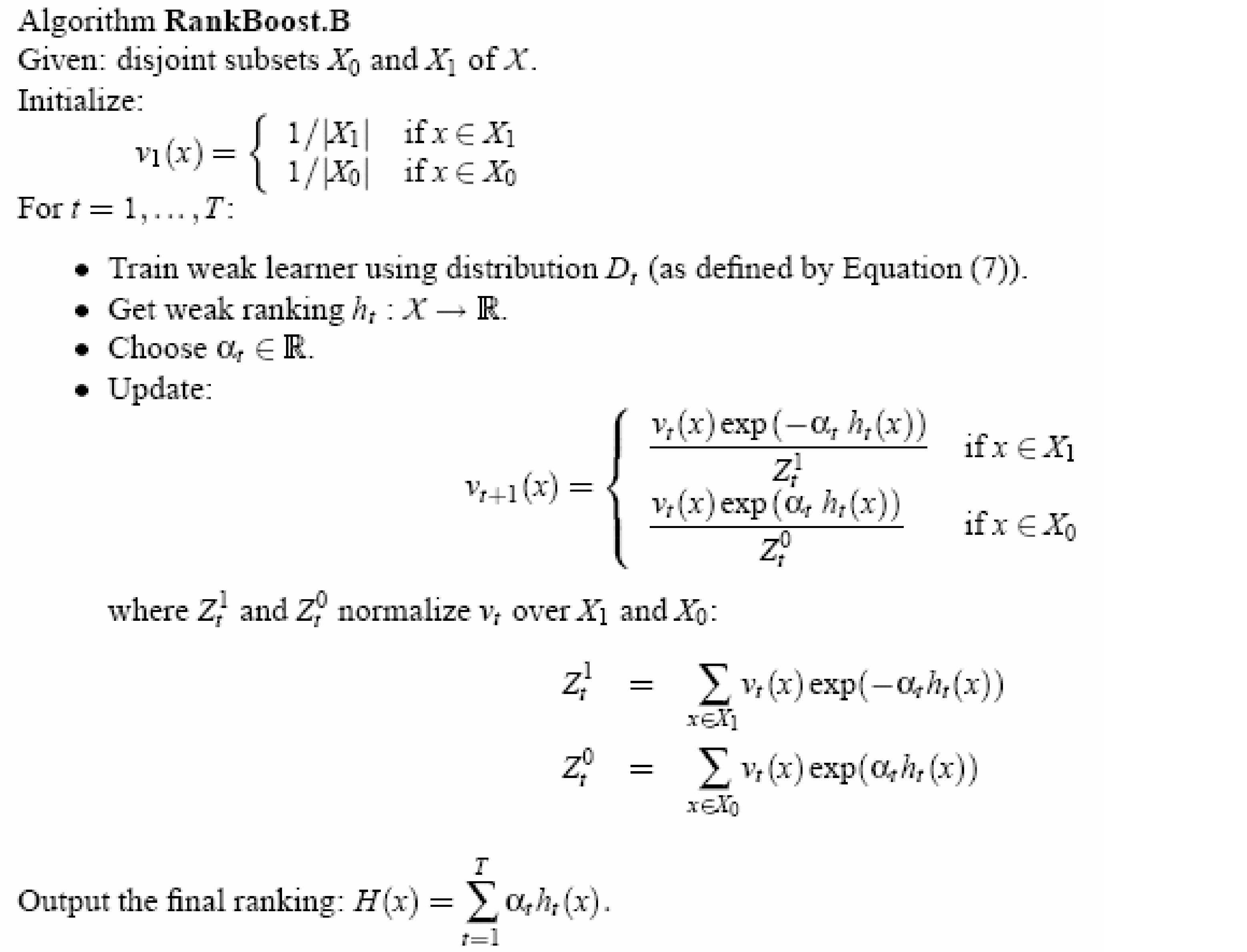
Learning to Order for CF

- Theorem: usual methods can be used to pick an optimal value for α
- Theorem: analogous to the usual case for boosting in classification, $rloss_D(H)$ is bounded by

$$rloss_D(H) \leq \prod_t Z_t$$

- Also: learning can be faster/simpler if Φ is "bipartite"—eg if target ratings are *like*, *don't like*, or *don't care*.
  - Don't need to maintain distribution over *pairs* of *x*'s.

Learning to Order for CF



Algorithm **RankBoost.B**

Given: disjoint subsets X_0 and X_1 of X.

Initialize:

$$v_1(x) = \begin{cases} 1/|X_1| & \text{if } x \in X_1 \\ 1/|X_0| & \text{if } x \in X_0 \end{cases}$$

For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t (as defined by Equation (7)).
- Get weak ranking $h_t : X \to \mathbb{R}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update:

$$v_{t+1}(x) = \begin{cases} \dfrac{v_t(x)\exp(-\alpha_t\, h_t(x))}{Z_t^1} & \text{if } x \in X_1 \\ \dfrac{v_t(x)\exp(\alpha_t\, h_t(x))}{Z_t^0} & \text{if } x \in X_0 \end{cases}$$

where Z_t^1 and Z_t^0 normalize v_t over X_1 and X_0:

$$Z_t^1 = \sum_{x \in X_1} v_t(x)\exp(-\alpha_t h_t(x))$$

$$Z_t^0 = \sum_{x \in X_0} v_t(x)\exp(\alpha_t h_t(x))$$

Output the final ranking: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$.

Learning to Order for CF

- Possible weak learners:
  - A feature function f_i—*i.e.*, ratings of some user
    - plus def. weight for unrated items to make *h* total
    - sensitive to actual values of *f*'s
  - Thresholded version of some f_i

$$h(x) = \begin{cases} 1 & \text{if } f_i(x) > \theta \\ 0 & \text{if } f_i(x) \leq \theta \\ q_{\text{def}} & \text{if } f_i(x) = \perp \end{cases}$$

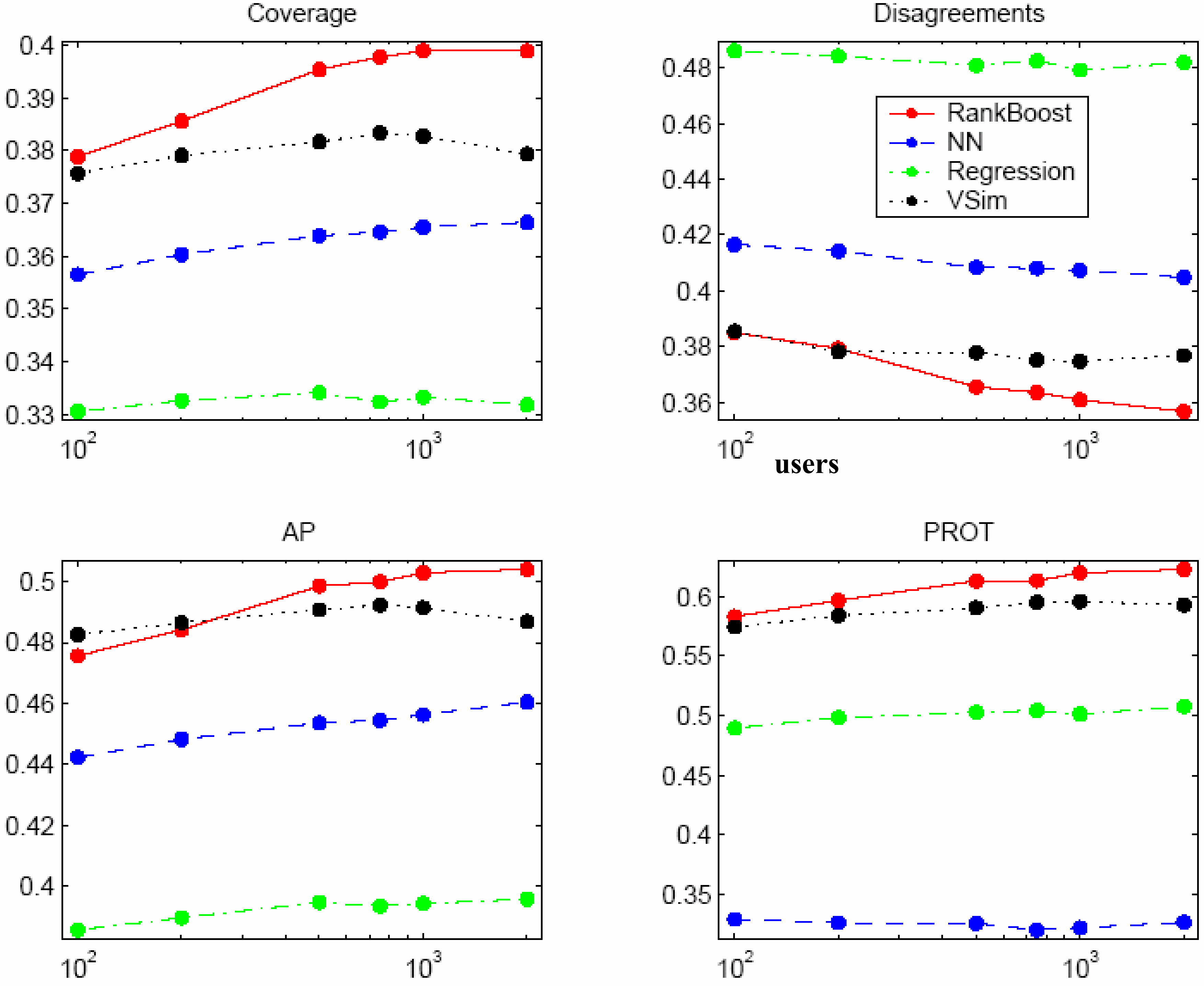
  - Values for θ, q_{def} can be found in linear time

Learning to Order for CF

- Evaluation:
  - EachMovie dataset
    - 60k users, 1.6k movies, 2.8M ratings
  - Measured, on test data:
    - Fraction of pairs mis-ordered by *H* relative to *Φ*
    - PROT (predicted rank of top-rated movie)
    - Average precision: $$\mathrm{AP} = \frac{1}{K}\sum_{k=1}^{K}\frac{k}{\mathrm{rank}(t_k)}$$
    - Coverage: $$\frac{1}{\mathrm{rank}(t_K)}\,,$$

Learning to Order for CF

- Evaluation: compared RankBoost with
  - VSIM (as in Breese et al)
  - 1-NN (predict using “closest” neighbor to U_a, using *rloss* on known ratings as distance)
  - Linear regression (as in Bellcore’s MovieRecommender)
  - Vary
    - number of *features* (aka users, community size, ...)
    - *feature density* (movies ranked per community member)
    - *feedback density* (movies ranked per target user)

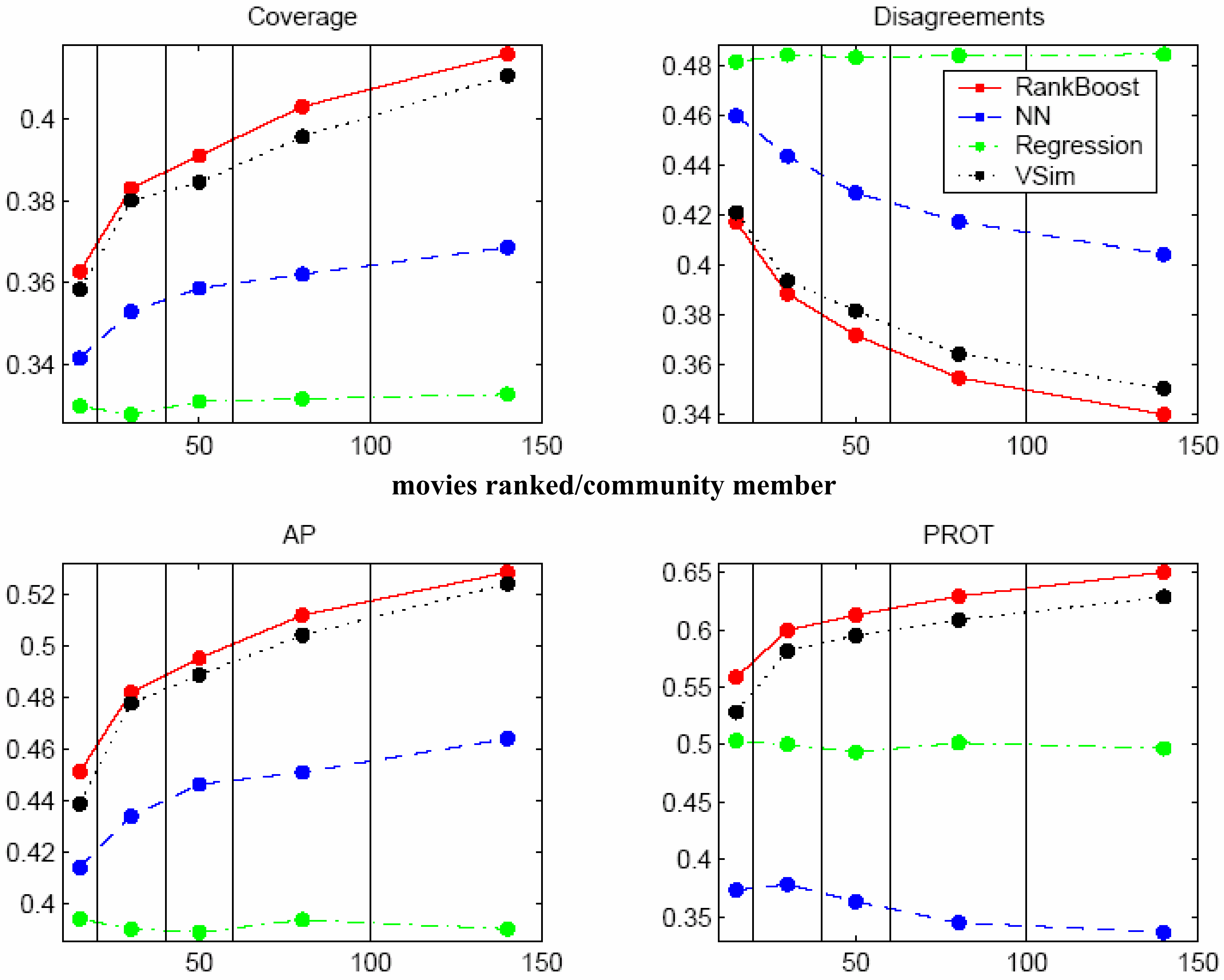


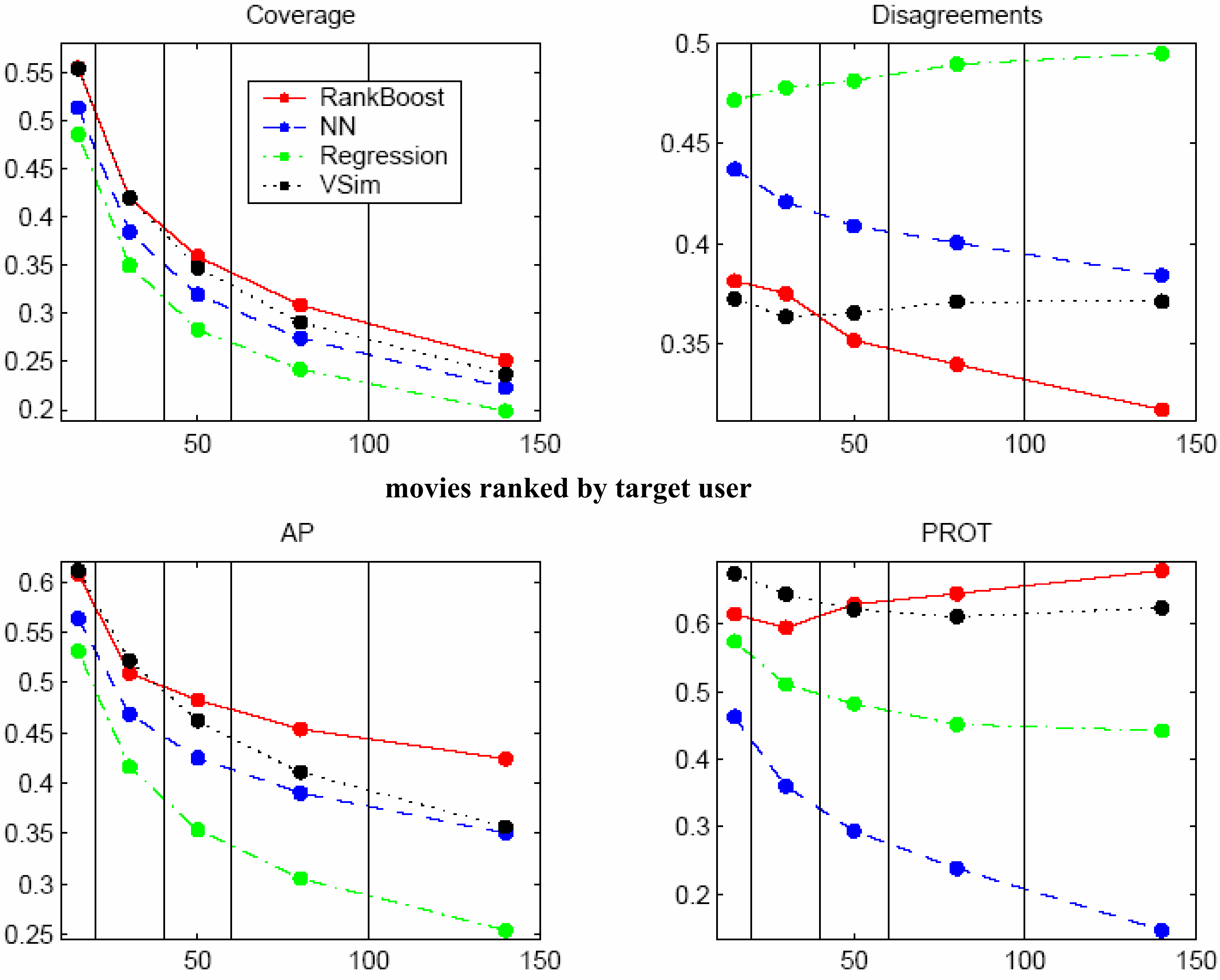
Coverage

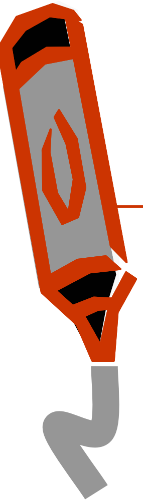
Disagreements

AP

PROT


Coverage
0.4
0.38
0.36
0.34
50
100
150
Disagreements
0.48
0.46
0.44
0.42
0.4
0.38
0.36
0.34
50
100
150
RankBoost
NN
Regression
VSim
movies ranked/community member
AP
0.52
0.5
0.48
0.46
0.44
0.42
0.4
50
100
150
PROT
0.65
0.6
0.55
0.5
0.45
0.4
0.35
50
100
150


Coverage
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
50
100
150
RankBoost
NN
Regression
VSim
Disagreements
0.5
0.45
0.4
0.35
50
100
150
movies ranked by target user
AP
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
50
100
150
PROT
0.6
0.5
0.4
0.3
0.2
50
100
150



Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
  - Combining CF and content filtering
  - CF as matching content and user
- Algorithms for CF
  - Ranking-based CF
  - Probabilistic model-based CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings

CF as density estimation

(Breese et al, UAI98)

- Estimate *$Pr(R_{ij}=k)$* for each user *i*, movie *j*, and rating *k*
- Use all available data to build *model* for this estimator

R_{ij}	**Airplane**	**Matrix**	**Room with a View**	**...**	**Hidalgo**
Joe	9	7	2	…	7
Carol	8	?	9	…	?
…	…	…	…	…	…
Kumar	9	3	?	…	6

CF as density estimation

(Breese et al, UAI98)

- Estimate *Pr(R$_{ij}$=k)* for each user *i*, movie *j*, and rating *k*
- Use all available data to build *model* for this estimator
- A simple example:

$$\forall \text{ movies } j, \Pr(R_{ij} = k) = \frac{\#(\text{users } i : R_{ij} = k)}{\#(\text{users } i \text{ rating } j)}$$

Leads to this expected value for unknown R_{ij} :

$$\mathrm{E}[R_{ij}] = \sum_k k \cdot \Pr(R_{ij} = k) = \text{average rating of movie } j$$

CF as density estimation

(Breese et al, UAI98)

- Estimate *Pr(*R_{ij}*=k)* for each user *i*, movie *j*, and rating *k*
- Use all available data to build *model* for this estimator
- More complex example:
  - Group users into *M* “clusters”: *c(1), ..., c(M)*
  - For movie *j*,

$$\Pr(R_{ij}=k \mid i)=\sum_m \overbrace{\Pr(R_{ij}=k \mid i\in c(m))}^{\textbf{estimate by counts}}\Pr(i\in c(m))$$

$$\mathrm{E}[R_{ij}]=\sum_m \Pr(i\in c(m))\cdot(\text{average rating of } j \text{ in } c(m))$$

CF as density estimation: BC

(Breese et al, UAI98)

- Group users into clusters using Expectation-Maximization:
  - Randomly initialize $Pr(R_{m,j}=k)$ for each m

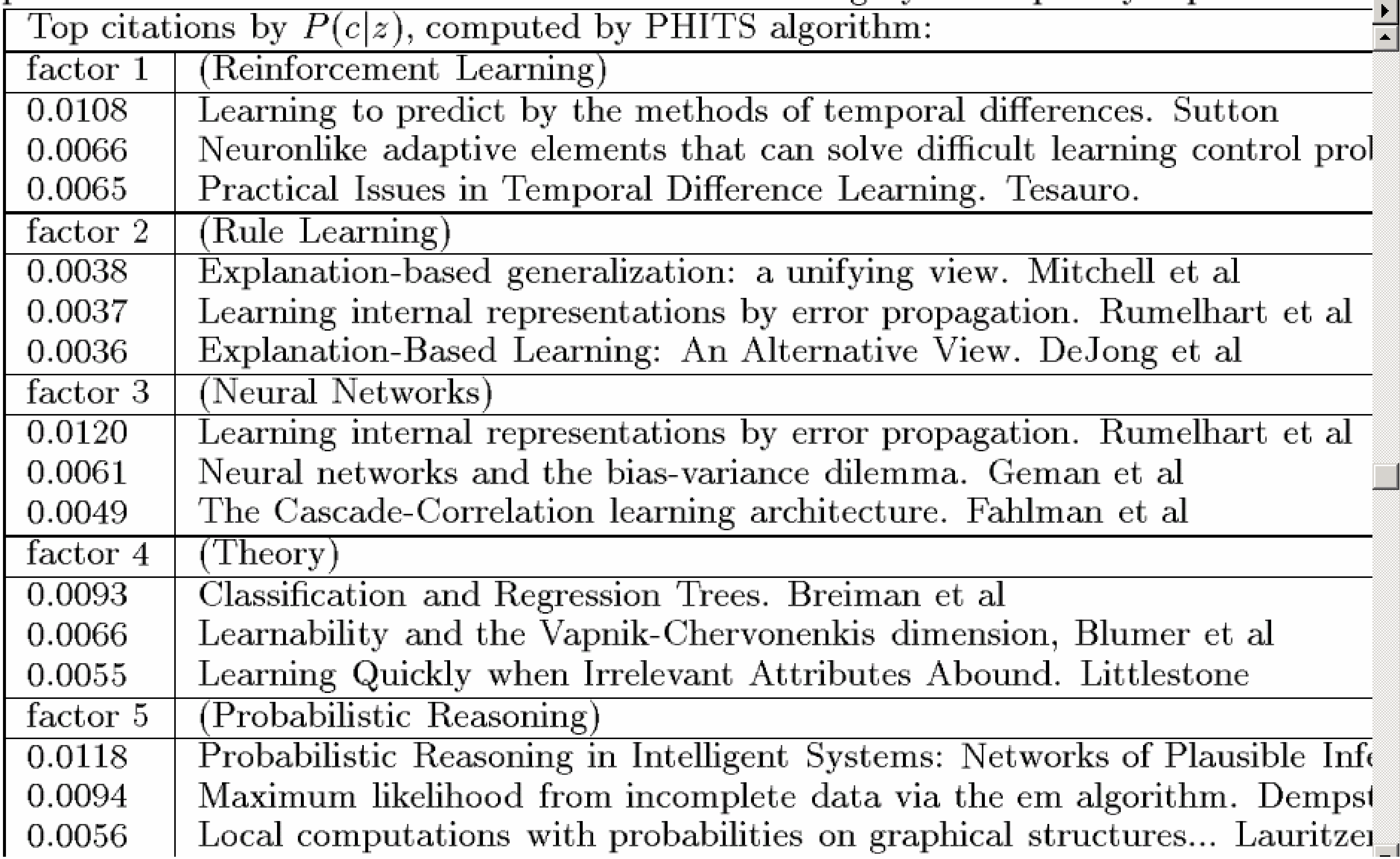
    (i.e., initialize the clusters differently somehow)
  - E-Step: Estimate Pr(user i in cluster m) for each i,m
  - M-Step: Find maximum likelihood (ML) estimator for R_{ij} within each cluster m
    - Use ratio of #(users i in cluster m with rating $R_{ij}=k$) to #(user i in cluster m), **weighted** by Pr(i in m) from E-step
  - Repeat E-step, M-step until convergence

CF as density estimation: BC

(Breese et al, UAI98)

- **Aside**: clustering-based density estimation is closely related to PageRank/HITS style web page recommendation.

- *Learning to Probabilistically Recognize Authoritative Documents,* Cohn & Chang, ICML-2000.
  - Let observed bibliographies be community “users”, and papers “items” to recommend
  - Cluster bibliographies into “factors” (subcommunities, user clusters)
  - Find top-ranked papers for each “factor” (top movies for each subcommunity/cluster)
    - These are “authoritative” (likely to be cited)

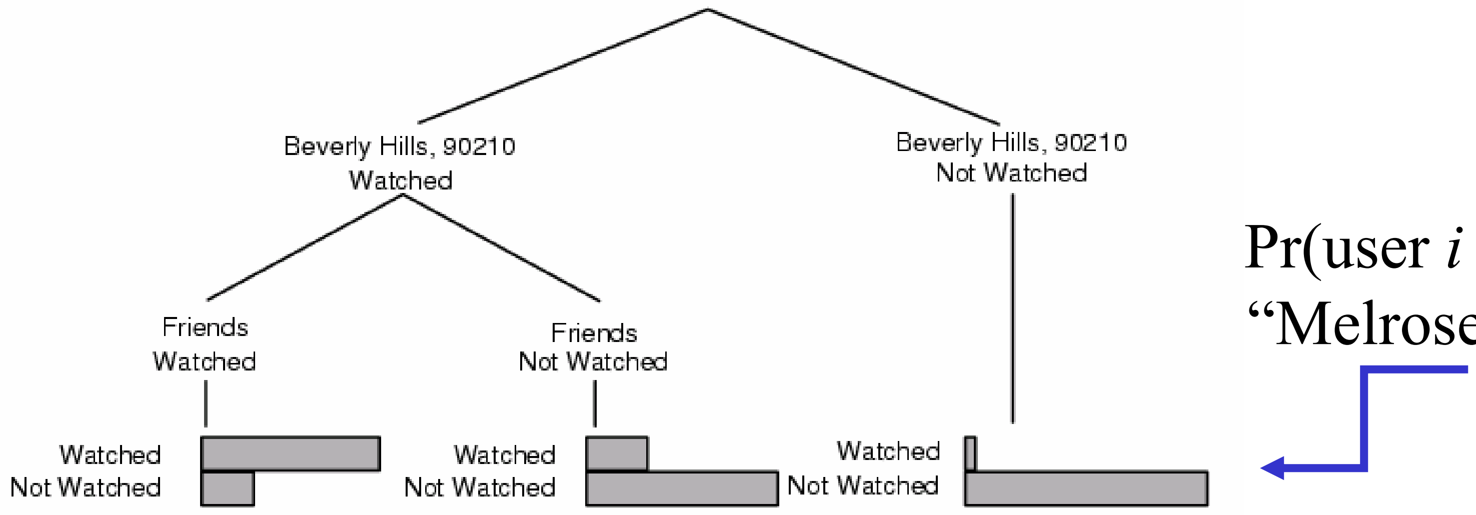


Top citations by $P(c\|z)$, computed by PHITS algorithm:	
factor 1	(Reinforcement Learning)
0.0108	Learning to predict by the methods of temporal differences. Sutton
0.0066	Neuronlike adaptive elements that can solve difficult learning control pro
0.0065	Practical Issues in Temporal Difference Learning. Tesauro.
factor 2	(Rule Learning)
0.0038	Explanation-based generalization: a unifying view. Mitchell et al
0.0037	Learning internal representations by error propagation. Rumelhart et al
0.0036	Explanation-Based Learning: An Alternative View. DeJong et al
factor 3	(Neural Networks)
0.0120	Learning internal representations by error propagation. Rumelhart et al
0.0061	Neural networks and the bias-variance dilemma. Geman et al
0.0049	The Cascade-Correlation learning architecture. Fahlman et al
factor 4	(Theory)
0.0093	Classification and Regression Trees. Breiman et al
0.0066	Learnability and the Vapnik-Chervonenkis dimension, Blumer et al
0.0055	Learning Quickly when Irrelevant Attributes Abound. Littlestone
factor 5	(Probabilistic Reasoning)
0.0118	Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inf
0.0094	Maximum likelihood from incomplete data via the em algorithm. Dempst
0.0056	Local computations with probabilities on graphical structures... Lauritze

CF as density estimation: BN

(Breese et al, UAI98)

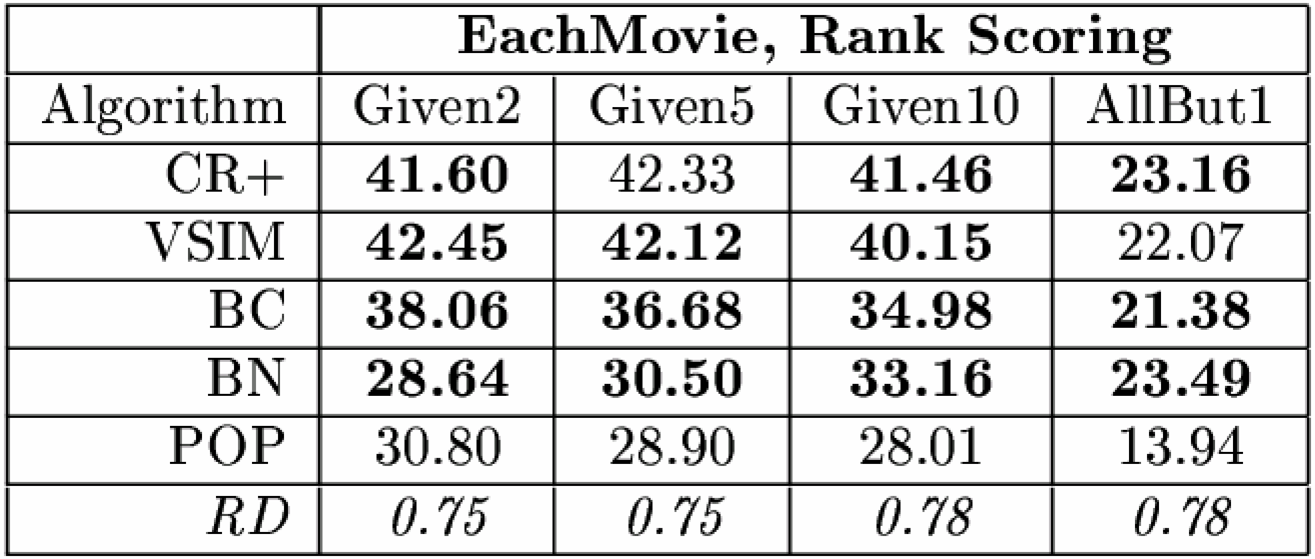
- BC assumes movie ratings within a cluster are **independent.**
- Bayes Network approach allows **dependencies** between ratings, but does not cluster. (Networks are constructed using greedy search.)



Pr(user *i* watched "Melrose Place")

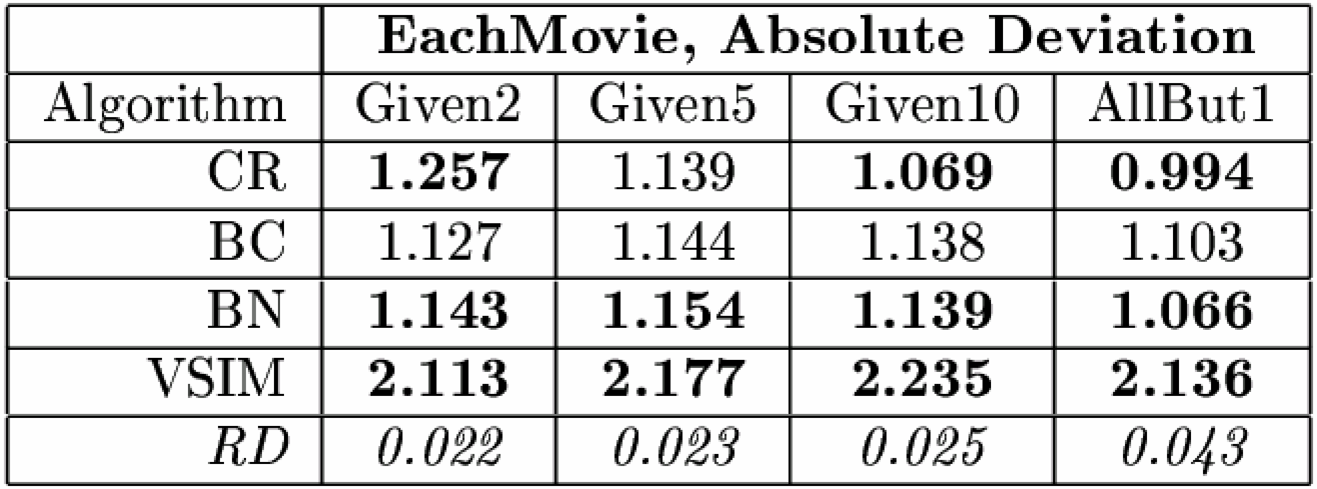
Algorithms for Collaborative Filtering 2: Memory-Based Algorithms (Breese et al, UAI98)

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
RD	*0.75*	*0.75*	*0.78*	*0.78*

golf score

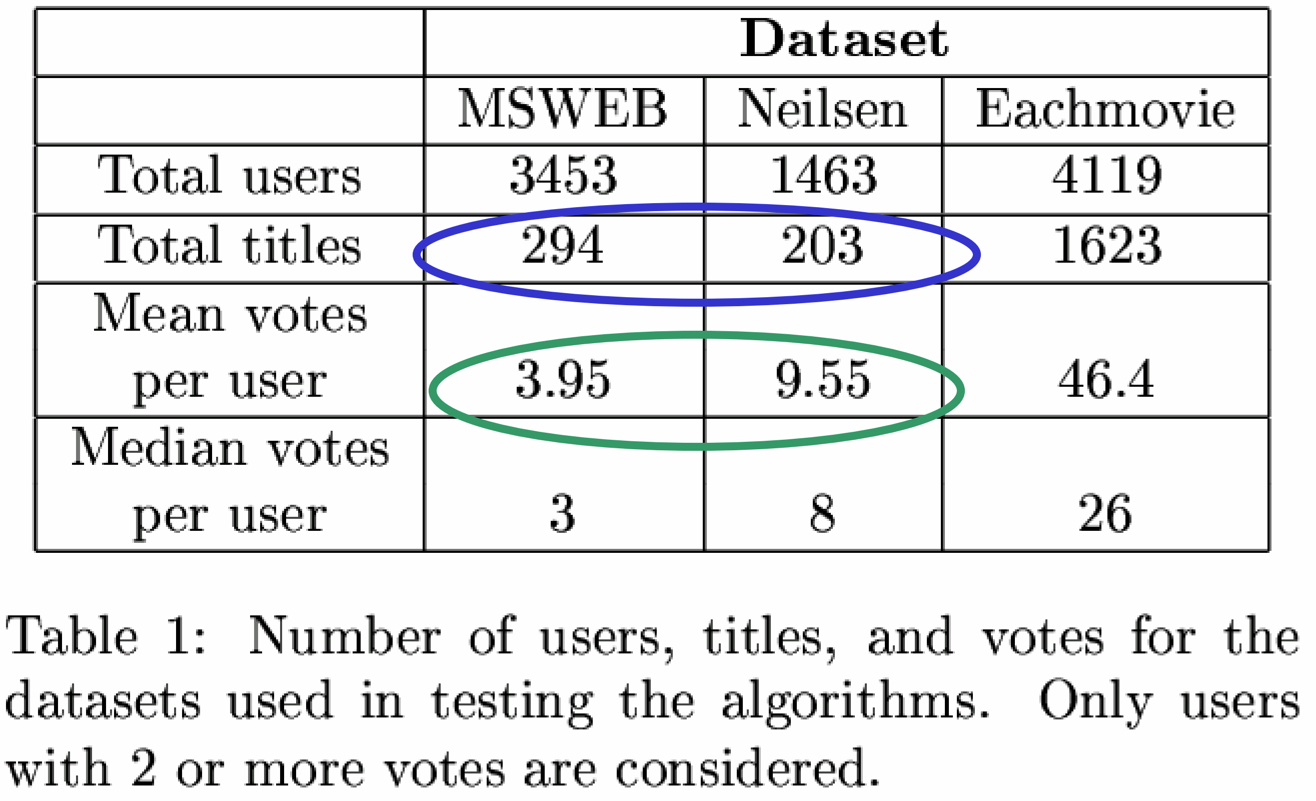


	EachMovie, Absolute Deviation			
Algorithm	Given2	Given5	Given10	AllBut1
CR	**1.257**	1.139	**1.069**	**0.994**
BC	1.127	1.144	1.138	1.103
BN	**1.143**	**1.154**	**1.139**	**1.066**
VSIM	**2.113**	**2.177**	**2.235**	**2.136**
RD	*0.022*	*0.023*	*0.025*	*0.043*

Datasets are different...

fewer items to recommend

fewer votes/user

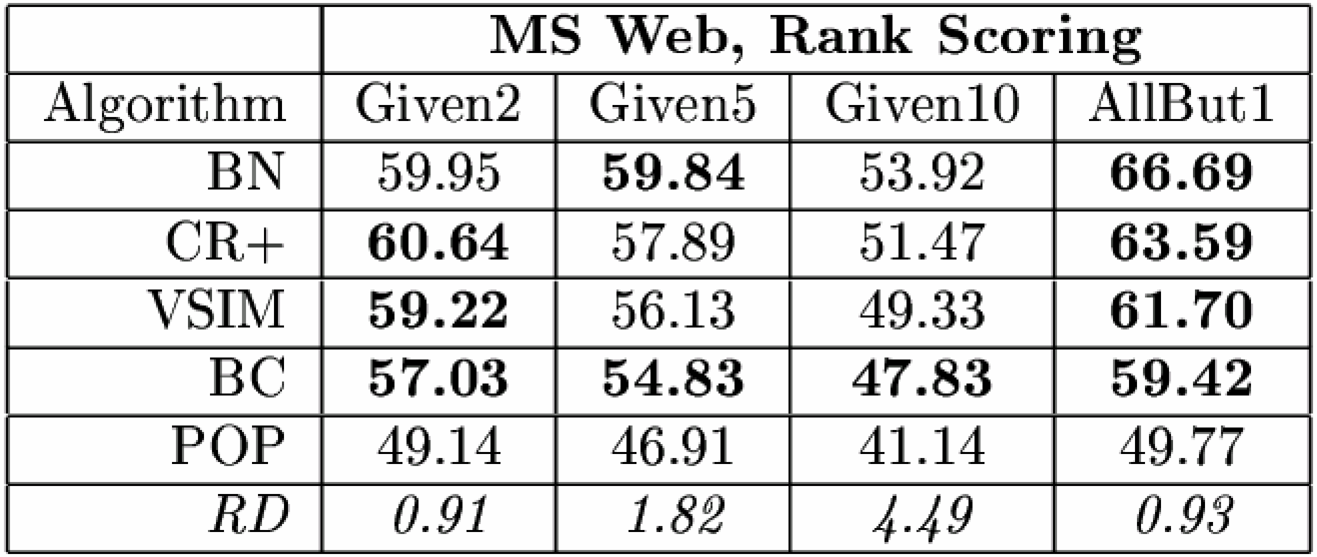


	Dataset		
	MSWEB	Neilsen	Eachmovie
Total users	3453	1463	4119
Total titles	294	203	1623
Mean votes per user	3.95	9.55	46.4
Median votes per user	3	8	26

Table 1: Number of users, titles, and votes for the datasets used in testing the algorithms. Only users with 2 or more votes are considered.

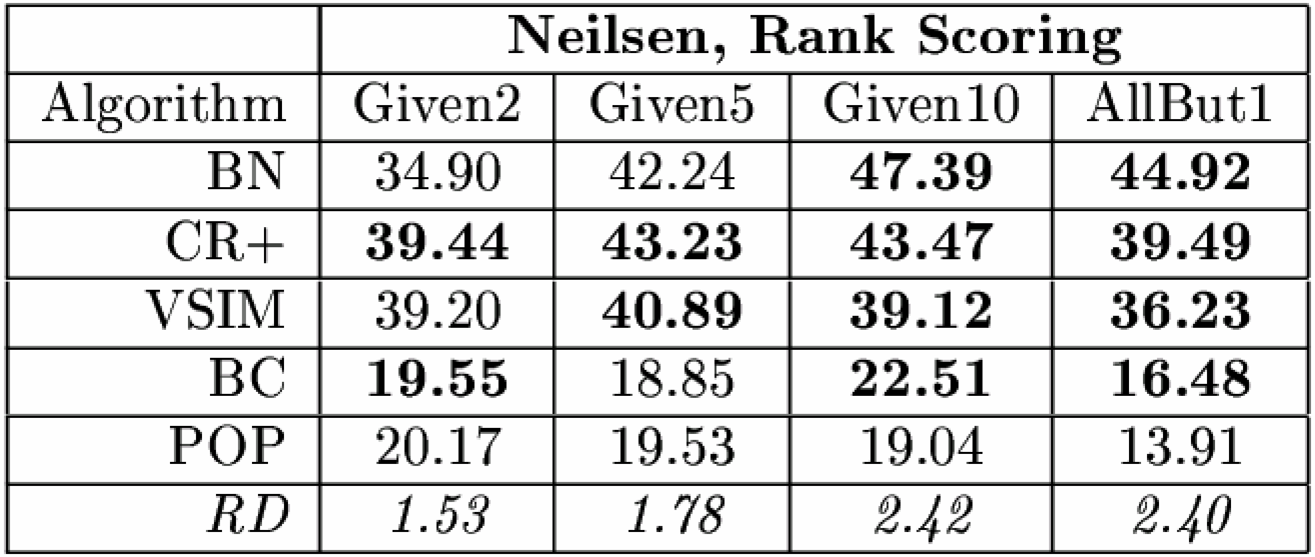
Results on MS Web & Nielson’s

soccer score



	MS Web, Rank Scoring			
Algorithm	Given2	Given5	Given10	AllBut1
BN	59.95	**59.84**	53.92	**66.69**
CR+	**60.64**	57.89	51.47	**63.59**
VSIM	**59.22**	56.13	49.33	**61.70**
BC	**57.03**	**54.83**	**47.83**	**59.42**
POP	49.14	46.91	41.14	49.77
RD	*0.91*	*1.82*	*4.49*	*0.93*

soccer score



	Neilsen, Rank Scoring			
Algorithm	Given2	Given5	Given10	AllBut1
BN	34.90	42.24	**47.39**	**44.92**
CR+	**39.44**	**43.23**	**43.47**	**39.49**
VSIM	39.20	**40.89**	**39.12**	**36.23**
BC	**19.55**	18.85	**22.51**	**16.48**
POP	20.17	19.53	19.04	13.91
RD	*1.53*	*1.78*	*2.42*	*2.40*

Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
  - Combining CF and content filtering
  - CF as matching content and user
- Algorithms for CF
  - Ranking-based CF
  - Probabilistic model-based CF
  - Probabilistic memory-based CF?
- CF with different inputs
  - true ratings
  - assumed/implicit ratings

Personality Diagnosis
(Pennock et al, UAI 2000)

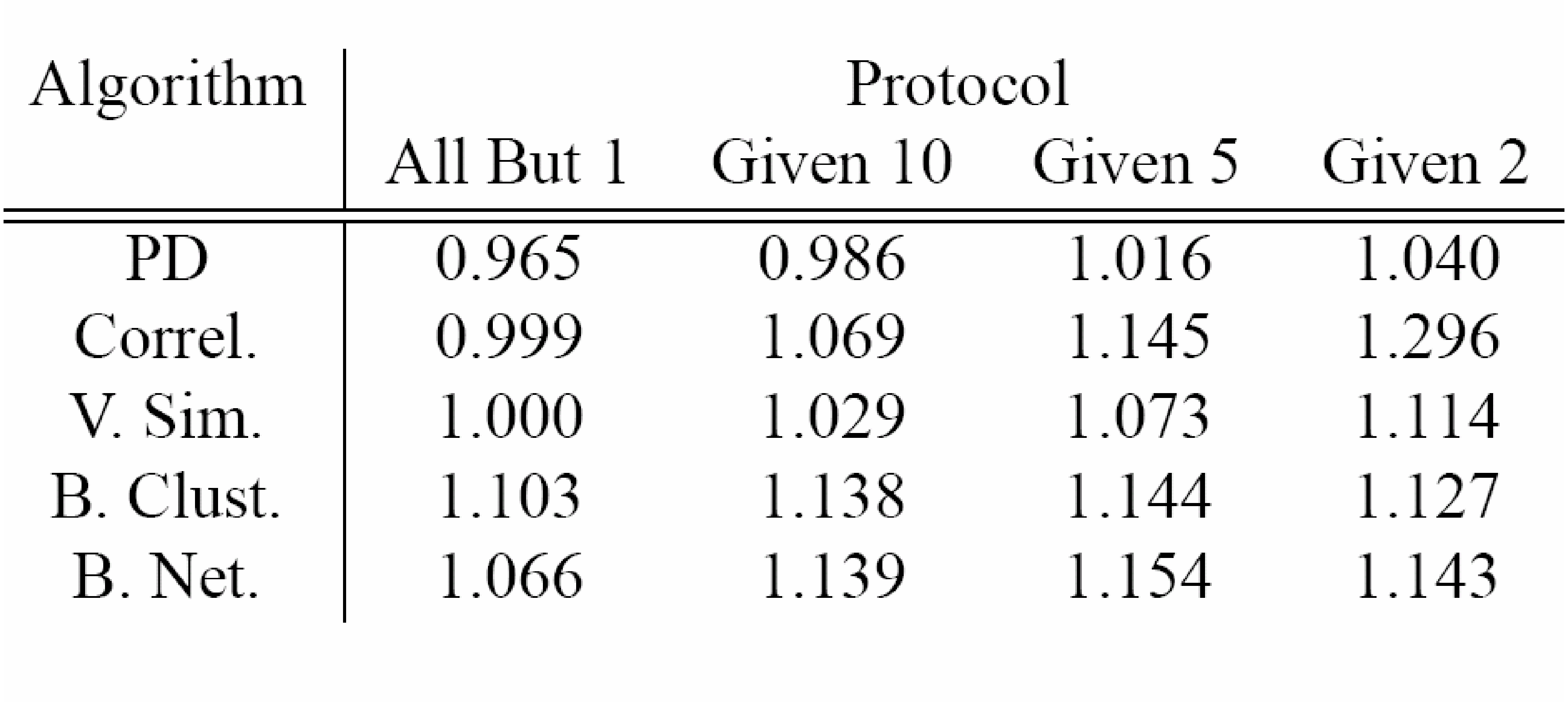
- *Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach,* Pennock, Horvitz, Lawrence & Giles, UAI 2000

- Basic ideas:
  - assume Gaussian noise applied to all ratings
  - treat each user as a separate cluster *m*
  - Pr(user *a* in cluster *i*) = *w(a,i)*

$$= \prod_j \Pr(R_{aj} \mid R_{ij}) = \prod_j \frac{1}{Z} e^{-(R_{aj}-R_{mj})/2\sigma^2}$$

Personality Diagnosis
(Pennock et al, UAI 2000)

- Evaluation (EachMovie, following Breese et al):

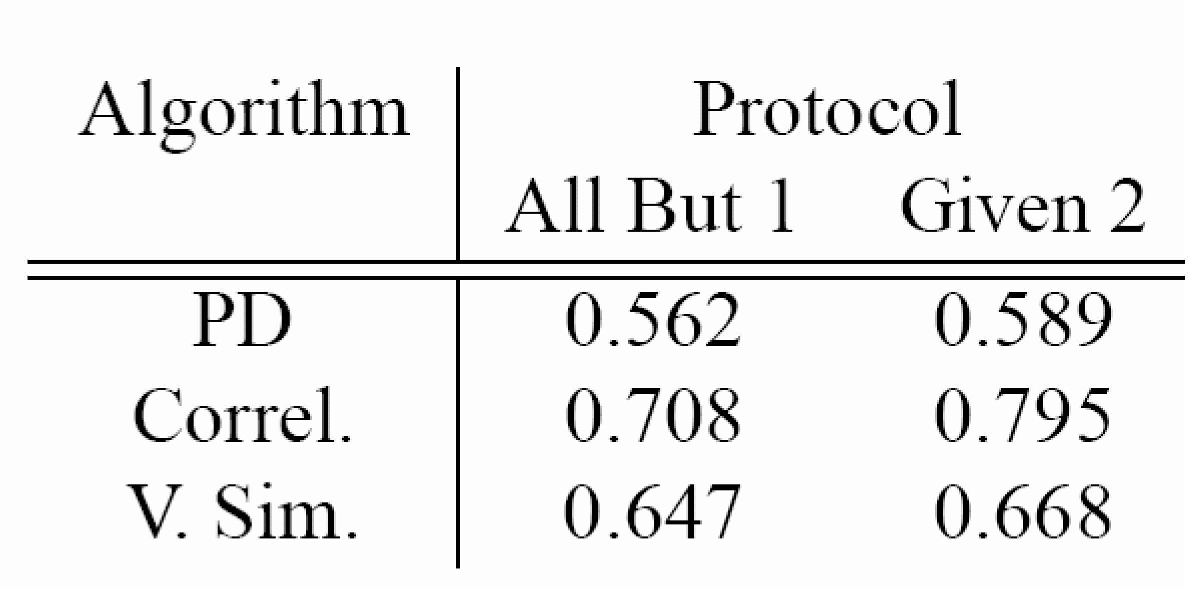


Algorithm	Protocol			
	All But 1	Given 10	Given 5	Given 2
PD	0.965	0.986	1.016	1.040
Correl.	0.999	1.069	1.145	1.296
V. Sim.	1.000	1.029	1.073	1.114
B. Clust.	1.103	1.138	1.144	1.127
B. Net.	1.066	1.139	1.154	1.143

Personality Diagnosis

(Pennock et al, UAI 2000)

- Evaluation (CiteSeer paper recommendation):



Algorithm	Protocol	
	All But 1	Given 2
PD	0.562	0.589
Correl.	0.708	0.795
V. Sim.	0.647	0.668

Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
  - Combining CF and content filtering
  - CF as matching content and user
- Algorithms for CF
  - Ranking-based CF
  - Probabilistic model-based CF
  - Probabilistic memory-based CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings
  - ratings inferred from Web pages

CF with pseudo-users

- *Web-Collaborative Filtering: Recommending Music by Crawling The Web*, Cohen and Fan, WWW-2000

- Goal: community filtering without a community
  - Approximate community with information automatically extracted from web pages.
- Outline:
  - problem & baseline CF system
  - creating “pseudo-users” from web pages
  - CF results with “pseudo-users”

A collaborative filtering task

- Data: server logs from a large digital music archive from June 1—Aug 24 1999 (*implicit*, not explicit, ratings)
- "Users" = IP addresses, some dynamic (sessions?)
- "Rating" = #downloads of artist

$$\text{rating}(u, a) = 1 \;\Leftrightarrow\; \#downloads(u, a) > 0$$

- Test data: all new IPs from Aug 1–Aug 24
- Train data: logs for remaining IPs
- 1017 artists, 1014 train users, 353 test users, 28,544 "ratings"

Proposed interface to recommender

A "smart" sound file player:

- Plays any file explicitly requested by the user
- If nothing is requested, smart player will choose a song and play it.
  - Plays song by artist "most likely to be liked by user" (strongest recommendation)
  - User can accept the song, or request something else (success, or failure)

Evaluating performance

```
Train recommendation system on “training” users.
For each test user $u_i$ simulate the “smart player”:
    For $j = 1, \ldots,$
        Recommend an artist $a_j$
        If $u_i$ listened to $a_j$ (according to log)
            Inform recommender that $u_i$ likes $a_j$
            Consider the trial a success
        Otherwise
            Pick some liked $a'_j$ (from log)
            Inform recommender that $u_i$ dislikes $a_j$, likes $a'_j$.
            Consider the trial an error
```

The baseline recommendation algorithm: K-NN

K-NN:

- Given: user u, set of artists A_u for which rating(u, a) is known.
- Pick K other most similar users $u_1, \ldots, u_K$:

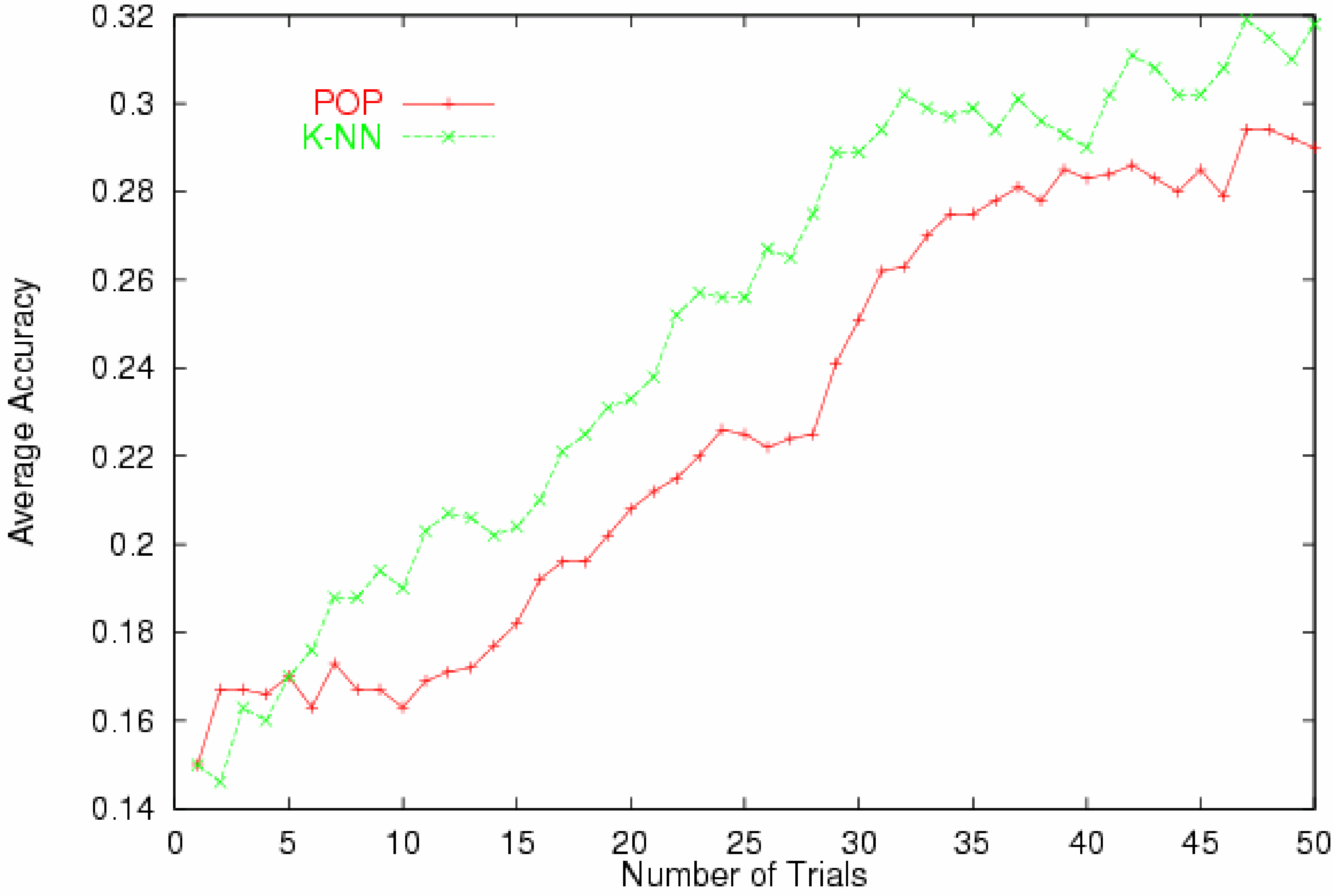
$$\mathrm{DIST}(u, u') = \sum_{a_i \in A_u} |\mathrm{rating}(u, a_i) - \mathrm{rating}(u', a_i)|$$

- Score other artists a by popularity with the "similar" u_i's:

$$\mathrm{SCORE}(a) = \sum_{i=1}^{K} rating(u_i, a)$$

- Recommend the top-scoring new artist.

CF with user data



Creating “Pseudo-Users” from the Web

1. Look for Web pages containing lists of artists
2. Extract lists from the pages
3. Treat each list of artists as a user in K-NN

Assumption: many of these artist-lists will be related in some useful way.

Creating “Pseudo-Users” from the Web

Extracting lists:

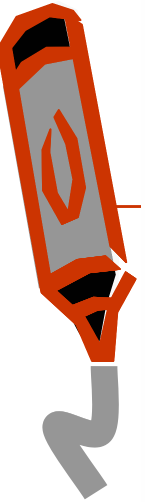
1. Parse the HTML markup
2. Associate each short marked-up section with its “position”

   (x_1,p_1),(x_2,p_2), $\ldots$
3. Find all triples (a_j,x_i,p_i) such that artist a_j’s name is highly similar to x.

   (Cosine similarity at least 0.9—WHIRL similarity join).
4. Each p_i is a “pseudo-user” that rates the associated a_j’s as positive. Collect all p_i’s with $\geq$4 positive ratings.

Creating pseudo-users: an example

```
<html><head>Biff's Home Page</head>
<body>
<h1>K00L Band Links</h1>
<table> <tr>
    <td>Metallica
    <td>Nine Inch Nails (new!)
  </tr><tr>
    <td>Barry Manilow
    . . .
```



Parsing and creating pairs

```
html(head(. . . ),
       body(
         h1(K00L Band Links),
         table(
           tr(td(Metallica),
               td(Nine Inch Nails (new!))),
           tr(td(Barry Manilow),
       . . .
```

(“K00L Band Links”, html_body_h1)

(“Metallica”, html_body_table_tr_td)

(“Nine Inch Nails (new!)”, html_body_table_tr_td)

(“Barry Manilow”, html_body_table_tr_td)

Normalizing and creating lists

(“K00L Band Links”, html_body_h1)

(“Metallica”, html_body_table_tr_td)

(“Nine Inch Nails (new!)”, html_body_table_tr_td)

(“Barry Manilow”, html_body_table_tr_td)

...

(“Metallica”, “Metallica”, html_body_table_tr_td)

(“Nine Inch Nails”, “Nine Inch Nails (new!)”, html_body_table_tr_td

(“Barry Manilow”, “Barry Manilow”, html_body_table_tr_td)

...

html_body_table_tr_td: Metallica, Nine Inch Nails, Barry Manilow,

...,

Creating “Pseudo-Users” from the Web

1. Look for Web pages containing lists of artists
2. Extract lists from the pages
3. Treat each list of artist as a user

Creating “Pseudo-Users” from the Web

Finding Web pages with lists (Phase 1):

- Search on each artist name, and take top 100 URLs.
- Extract lists from all URLs that appear more than once in a top-100 listing (5000+ URLs).

  Many of these are lists, but the statistics are very skewed.

Creating "Pseudo-Users" from the Web

Finding Web pages with lists (Phase 2):

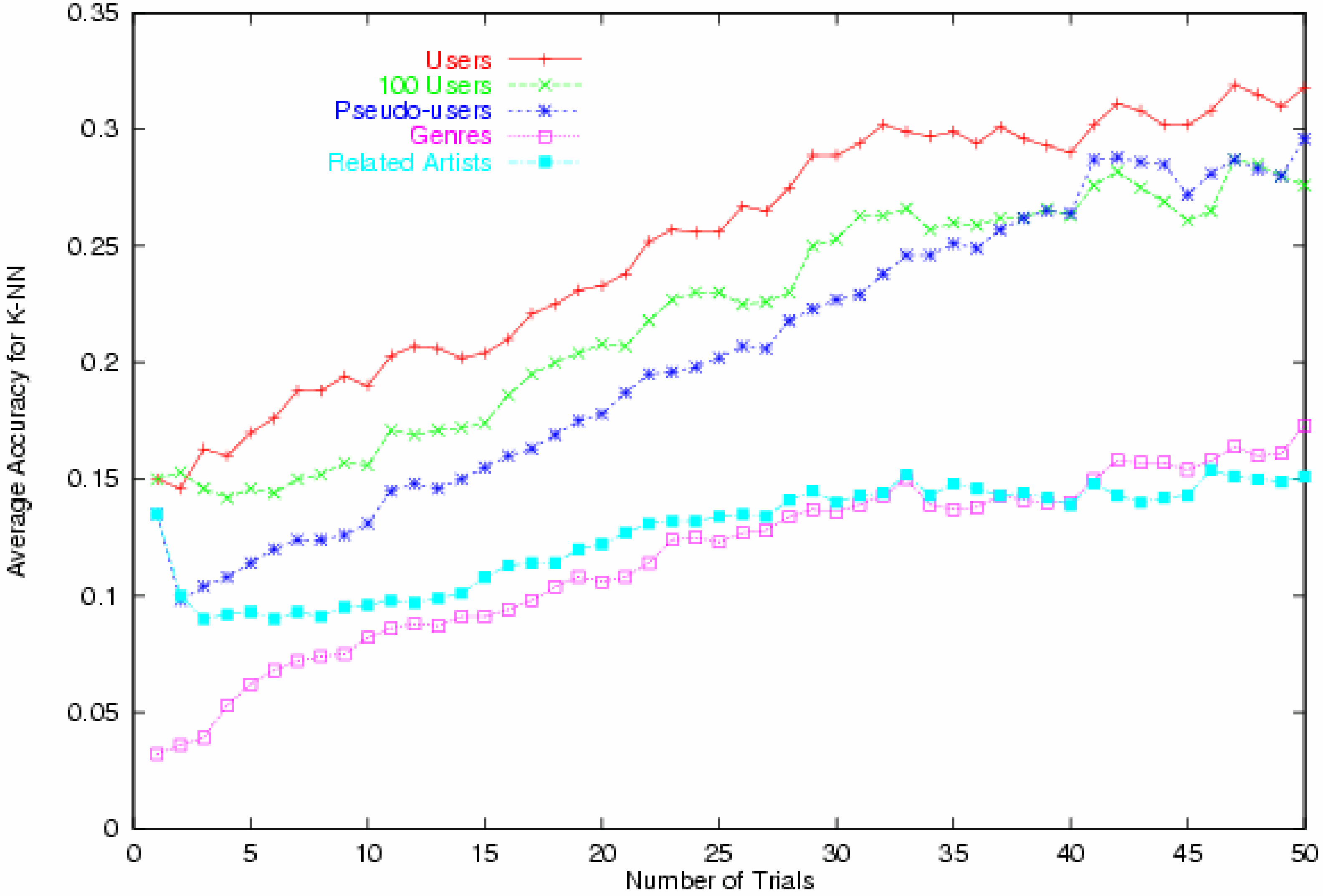
- Find pairs of artists that co-occur frequently in the phase-1 lists (1000 artists).
- Search on each* artist pair, and take the top 10 URLs (4000+ URLs).
- Extract lists from these URLs (1800+ pseudo-users with 48,000+ positive ratings).

A different sort of “pseudo-user”

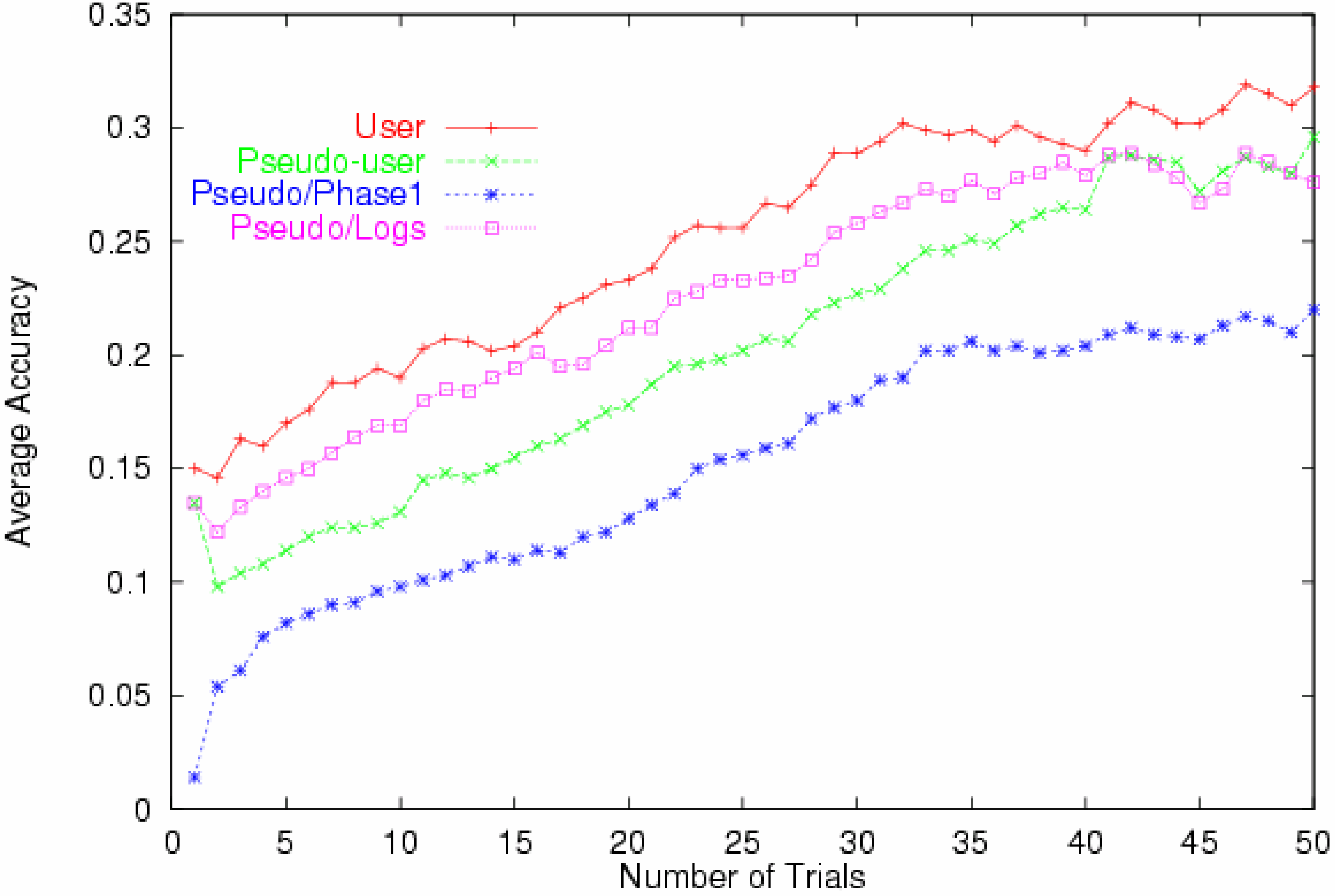
We also programmed a spider to crawl allmusic.com and collect

- Genres/styles
- Sets of “related artists”
    - $S_a = \{a\} \cup$ all artists “related to” a
- Again, treat each artist-set as a “user” in K-NN

K-NN with pseudo-users



Variant sets of pseudo-users



Variant collaborative filtering algorithms

- POP: recommend the globally most popular artist.
- WM: Weighted majority (following Abe et al, ICML'98)

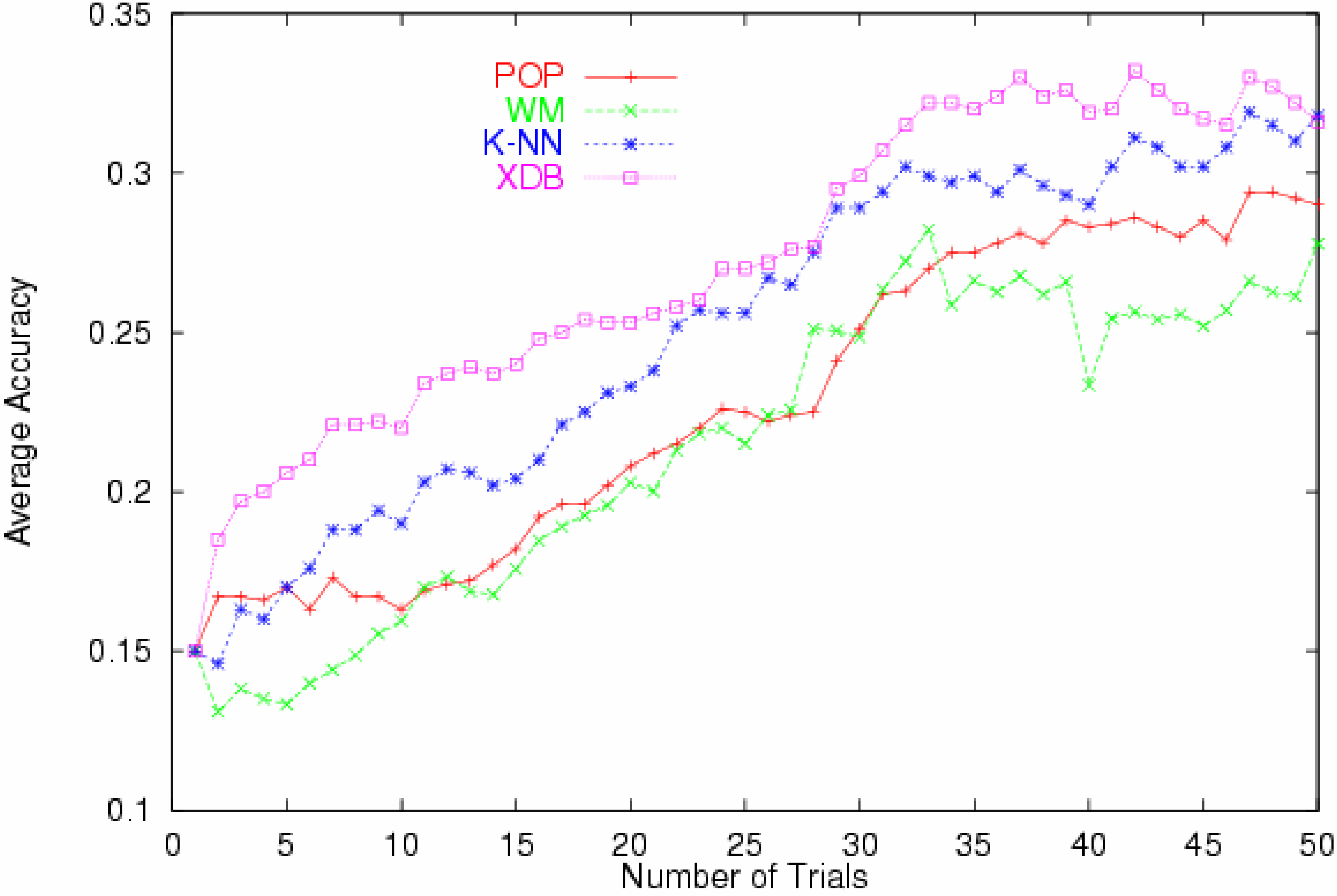
  Weighted combination of many very simple "experts" of the form "if you like (hate) artist a_j you'll like (hate) a'_j."
- XDB: if user u likes one artist a_1, score a according to

  $$R(a, a_1) = Prob_{u'}(likes(u', a) | likes(u', a_1))$$

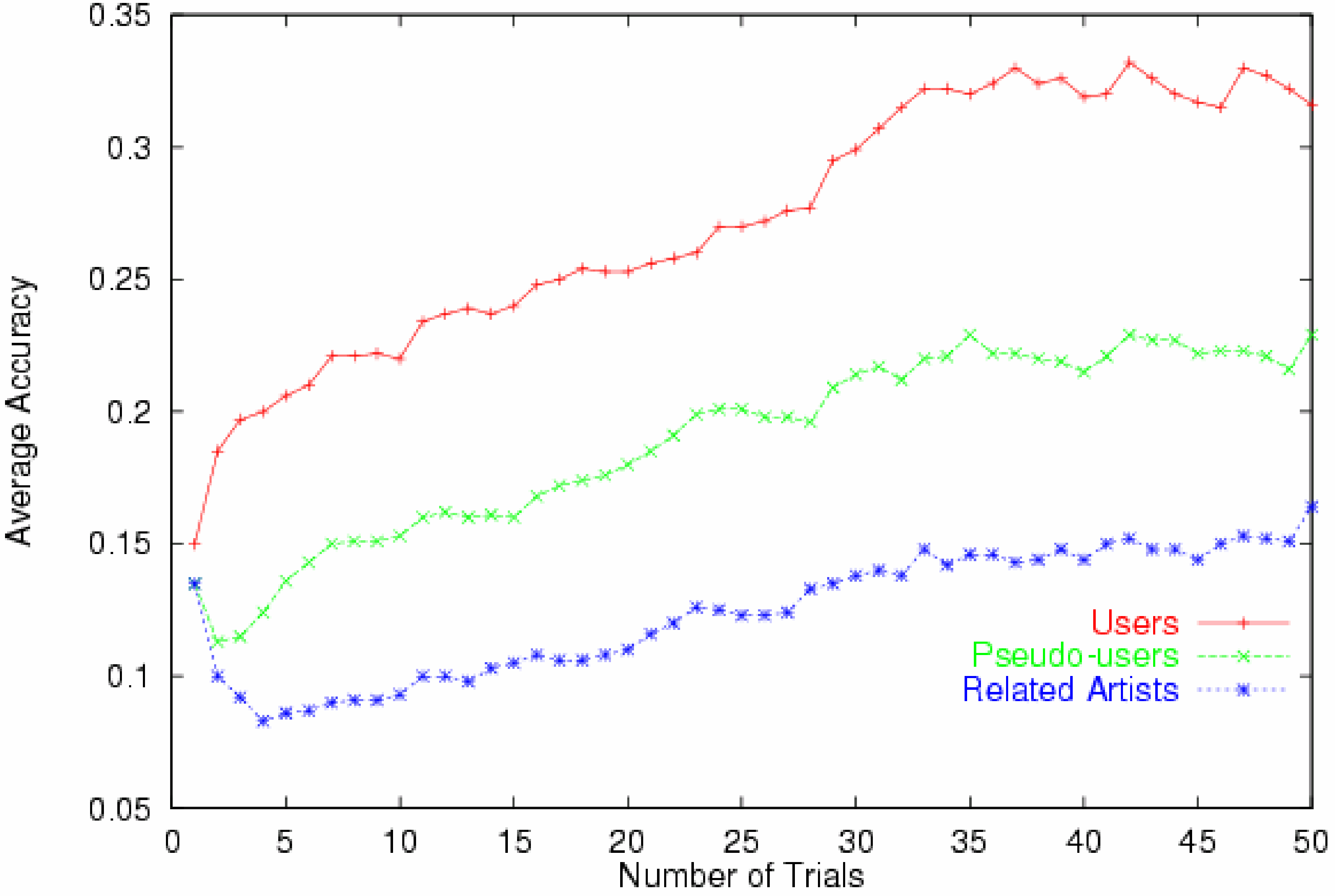
  If user u likes $a_1, \ldots, a_n$, score a with

  $$1 - (1 - R(a, a_1)) \cdot \ldots \cdot (1 - R(a, a_n))$$

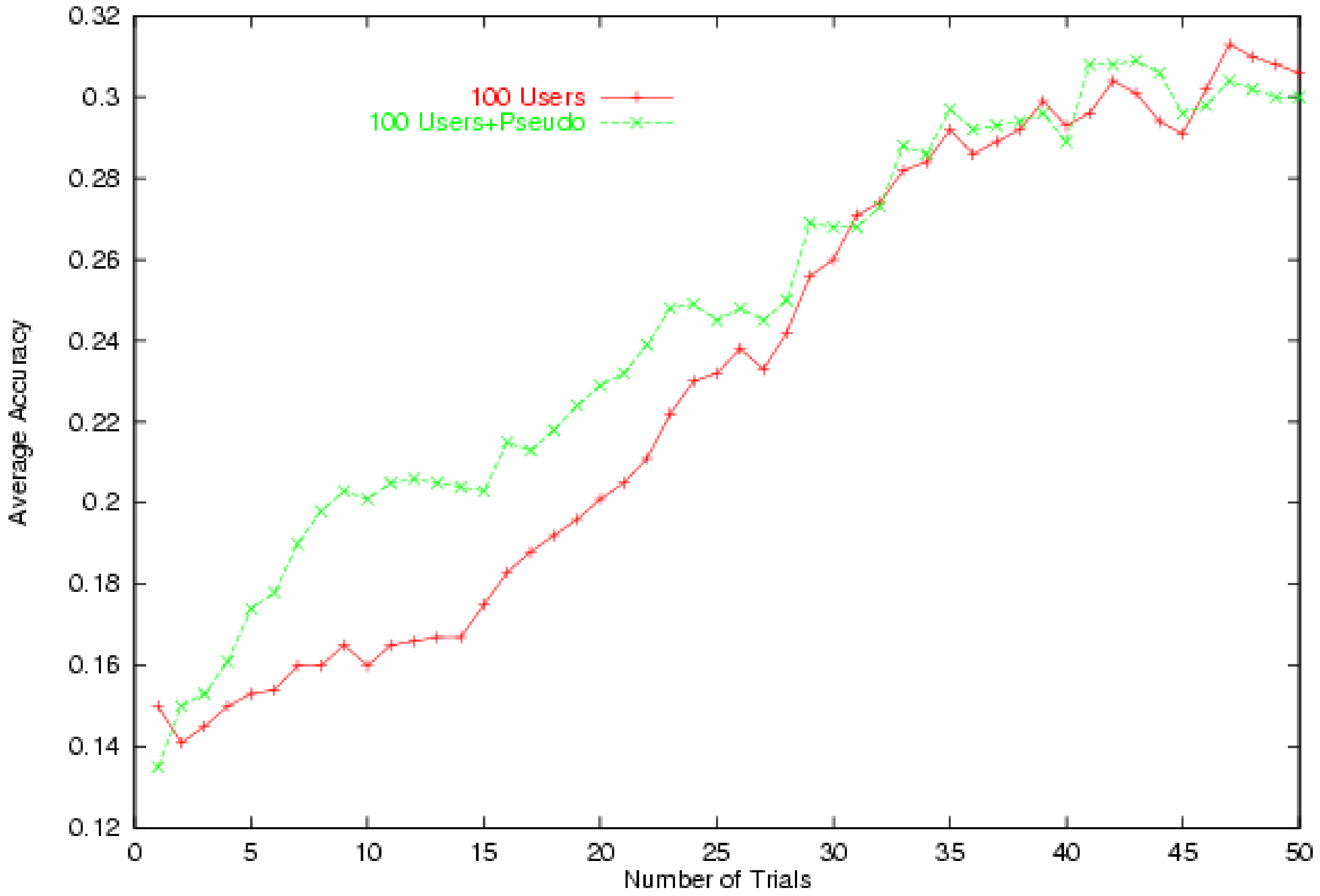
Baseline results for variant CF methods



Using pseudo-users with XDB



Adding pseudo-users to an undertrained XDB



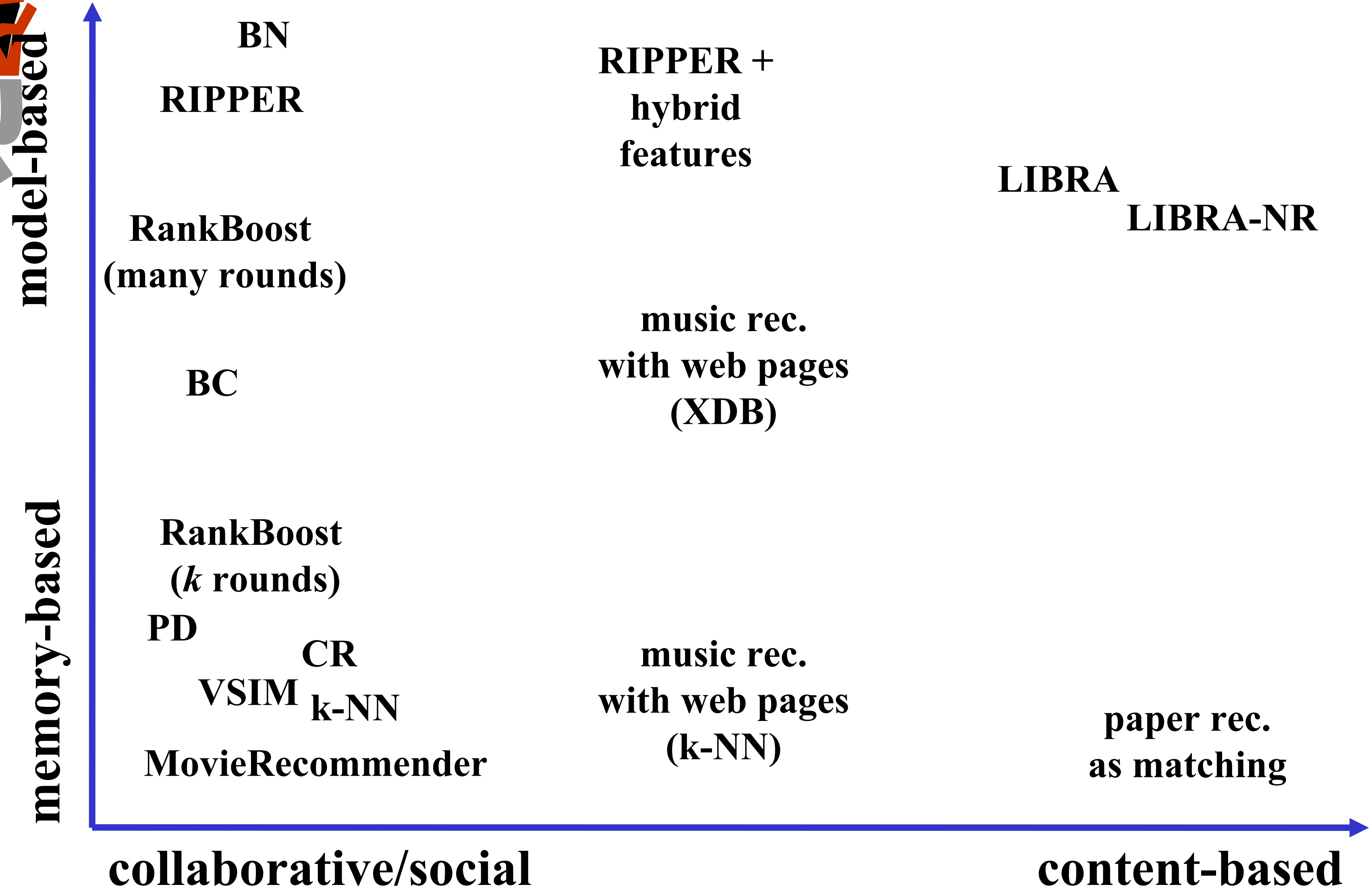
Outline

- Non-systematic survey of some CF systems
  - CF as basis for a virtual community
  - memory-based recommendation algorithms
  - visualizing user-user *via* item distances
  - CF versus content filtering
- Algorithms for CF
- CF with different inputs
  - true ratings
  - assumed/implicit ratings
- Conclusions/Summary

Tools for CF

- Memory-based (CR, VSIM, k-NN, PD,matching)
- Model-based (rules, BC, BN, boosting)
  - Social *vs* content
  - Hybrid social/content features
- Probabilistic (PD, BN, BC, PLSA, LDA, …)
  - Independence assumptions made
- Distance-based (matching, VSIM, k-NN, CR, PageRank)
  - Features used
  - Structures exploited
- Ranking based
  - RankBoost

Summary



model-based

memory-based

BN

RIPPER

RankBoost
(many rounds)

BC

RankBoost
(*k* rounds)

PD

CR

VSIM

k-NN

MovieRecommender

RIPPER +
hybrid
features

music rec.
with web pages
(XDB)

music rec.
with web pages
(k-NN)

LIBRA

LIBRA-NR

paper rec.
as matching

collaborative/social

content-based

Other issues, not addressed much

- Combining and weighting different types of information sources
  - How much is a web page link worth *vs* a link in a newsgroup?
- Spamming—how to prevent vendors from biasing results?
- Efficiency issues—how to handle a large community?
- What do we measure when we evaluate CF?
  - Predicting *actual* rating may be useless!
  - Example: music recommendations:
    - Beatles, Eric Clapton, Stones, Elton John, Led Zep, the Who, …
  - What's useful *and new?* for this need model of user's *prior knowledge*, not just his tastes.
    - Subjectively better recs result from "poor" distance metrics

Final Comments

- CF is one of a handful of learning-related tools that have had broadly *visible* impact:
  - Google, TIVO, Amazon, personal radio stations, …
- Critical tool for finding “consensus information” present in a large community (or large corpus of web pages, or large DB of purchase records, ….)
  - Similar in some respects to Q/A with corpora
- Science is relatively-well established
  - in certain narrow directions, on a few datasets
- Set of applications still being expanded
- Some resources:
  - http://www.sims.berkeley.edu/resources/collab/
  - http://www.cs.umn.edu/Research/GroupLens/
  - http://www.cis.upenn.edu/˜ungar/CF/

Social Networks

Instructor: Rada Mihalcea

Class web page:
http://www.cs.unt.edu/~rada/CSCE5200

(some of these slides were adapted from Jen Golbeck's talk slides)

What is a Social Network

- People and their connections to other people

Every aspect of our daily life is embedded
in a web of complex interactions:

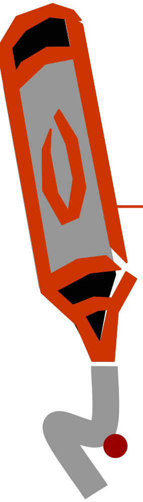
- social
- communication
- business
- …

Web-Based Social Networks (WBSNs)

- Social Networking on the Web
- Websites that allow users to maintain profiles, lists of friends
- Examples

Criteria

- It is accessible over the web with a web browser.
- Users must explicitly state their relationship with other people qua stating a relationship.
- Relationships must be visible and browsable by other users in the system.
- The website or other web-based

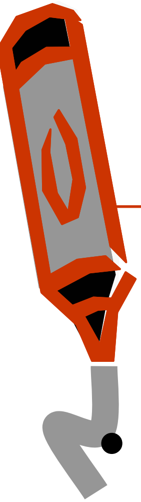


Numbers

- 141 Social Networks
- ¿200,000,000 user accounts
- Top Five

1.	My Space	56,000,000
2.	Adult Friend Finder	21,000,000
3.	Friendster	21,000,000
4.	Tickle	20,000,000
5.	Black Planet	17,000,000

Types / Categories



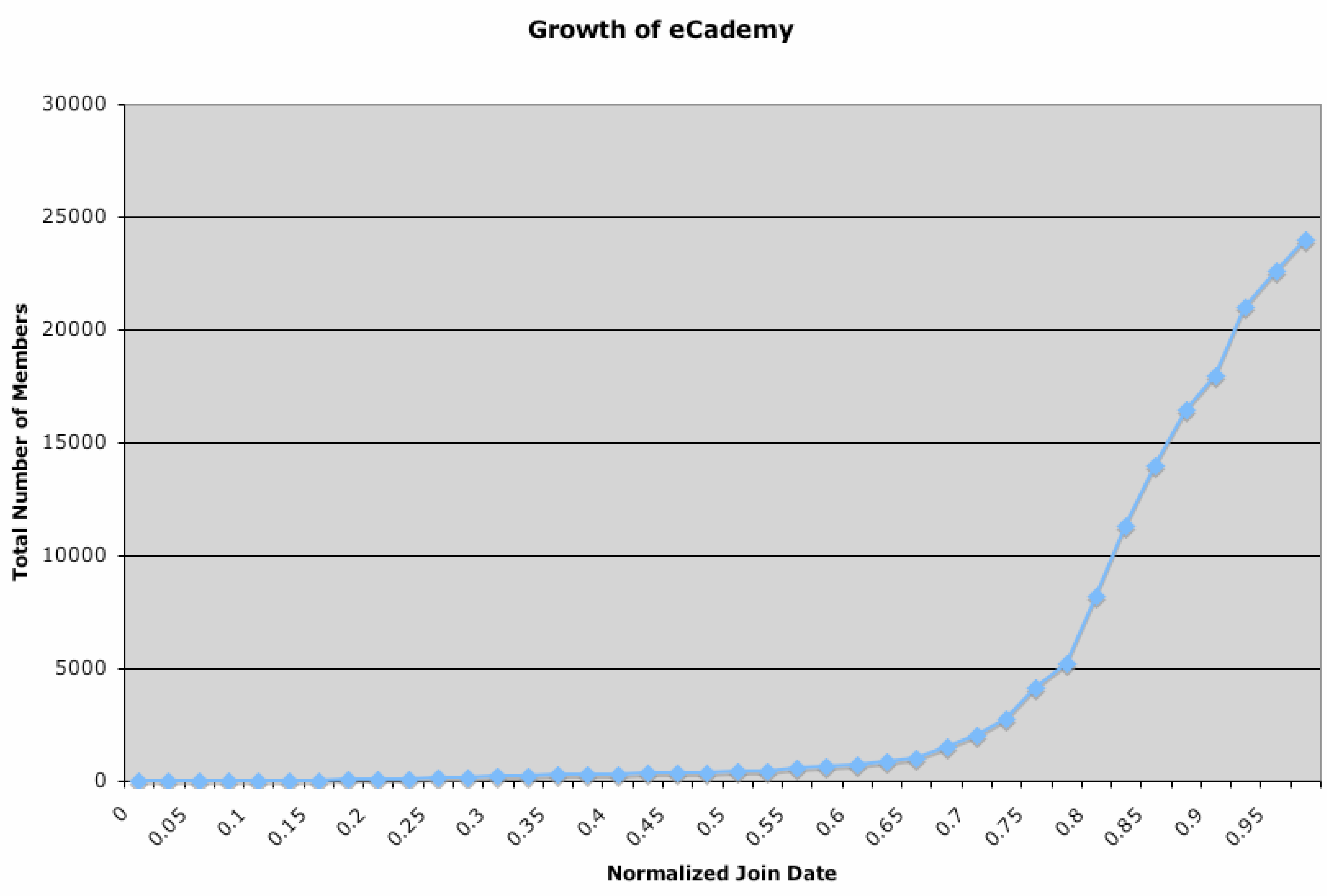
- Blogging
- Business
- Dating
- Pets
- Photos
- Religious
- Social/Entertainment

Relationships in WBSNs

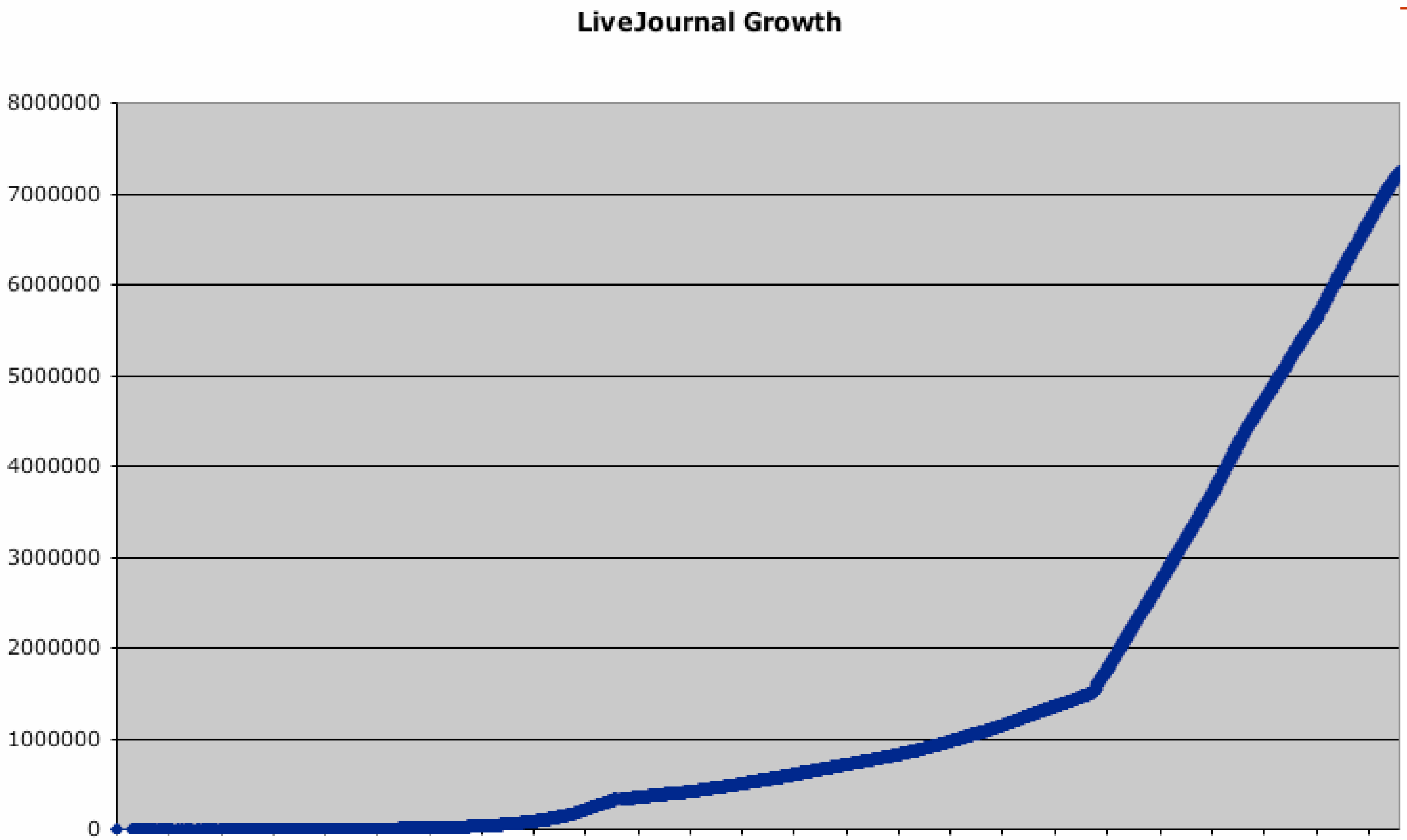
- Users can say things about the types of relationships they have
- Some networks provide some relationship annotation feature
- Free-text (e.g. testimonials)
- Fixed options (e.g. Lived Together, Worked Together, From and organization or team, Took a course together, From a summer/study abroad program, Went to school together, Traveled together, In my family, Through a friend, Through Facebook, Met randomly, We hooked up, We dated, I don't even know this person.)

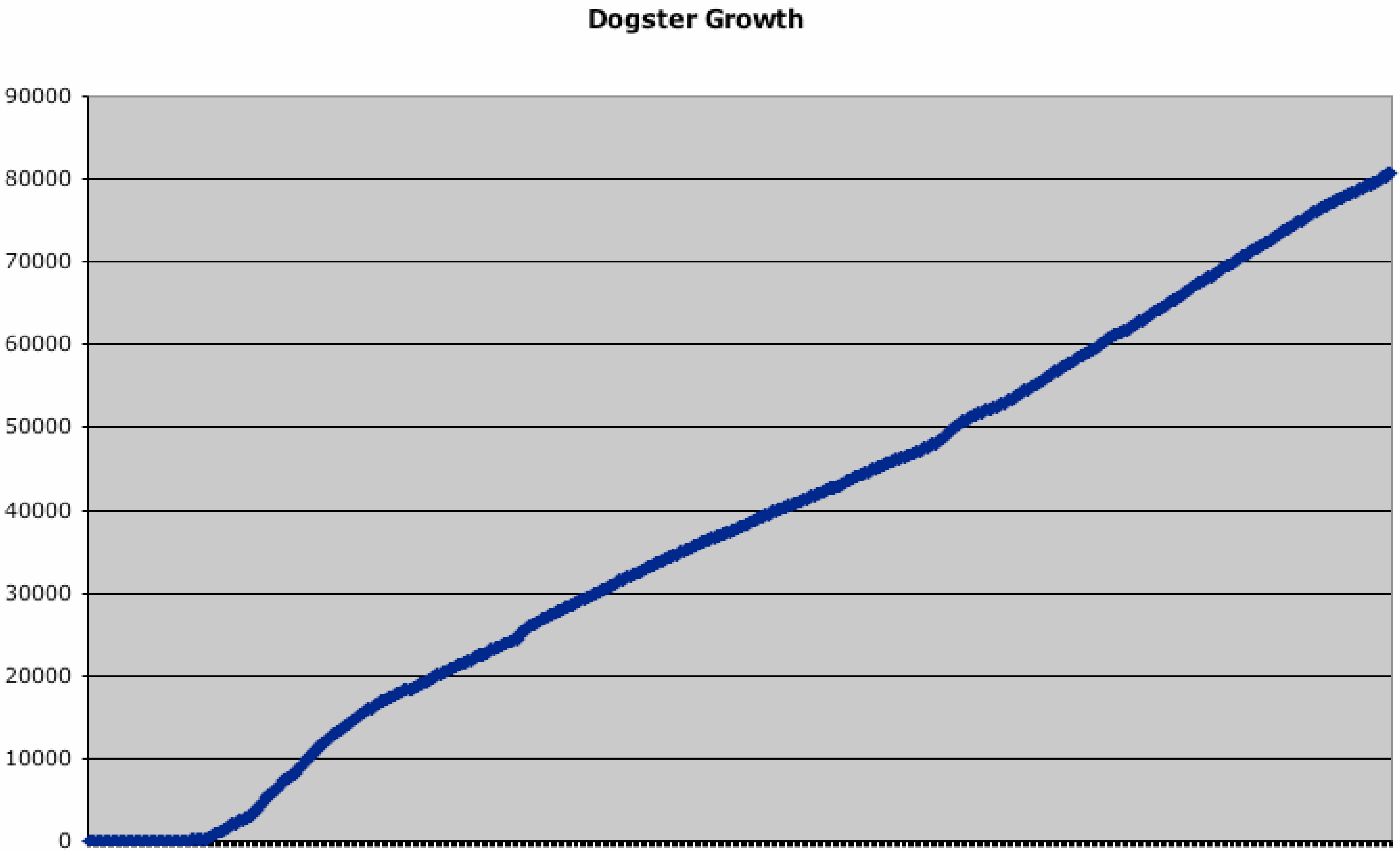
Growth Patterns

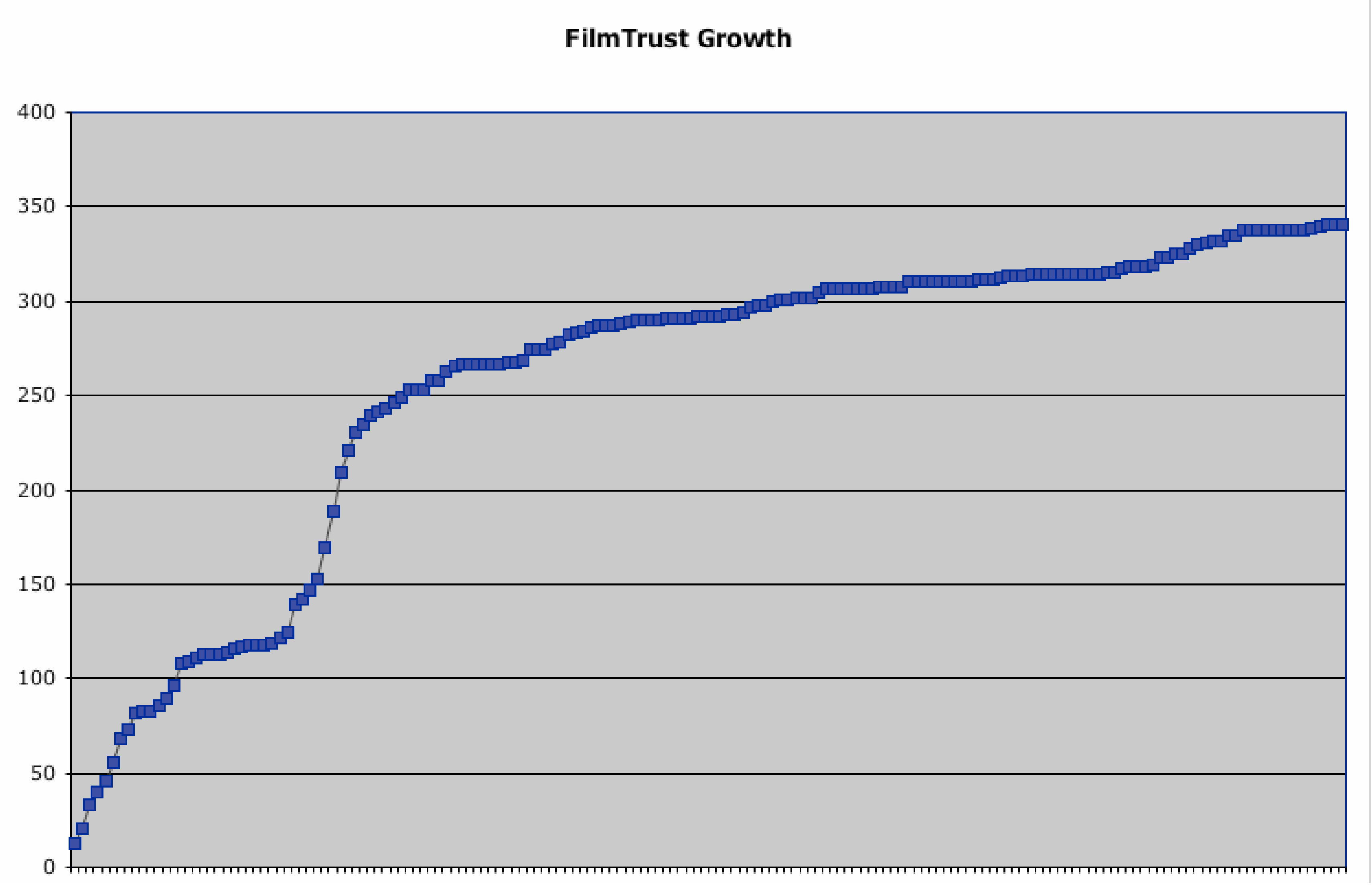
- Networks Grow in recognizable patterns
  - Exponential
  - Linear
  - Logarithmic



Growth of eCademy


LiveJournal Growth
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0


Dogster Growth
90000
80000
70000
60000
50000
40000
30000
20000
10000
0

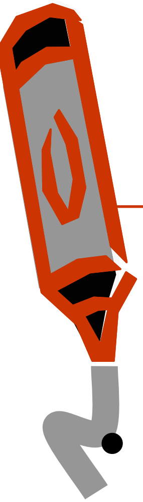

FilmTrust Growth
400
350
300
250
200
150
100
50
0

Public WBSNs: FOAF

- Friend of a Friend (FOAF): a vocabulary in OWL for sharing personal and social network information on the Semantic Web
- Over 10,000,000 FOAF profiles from 8 social networks

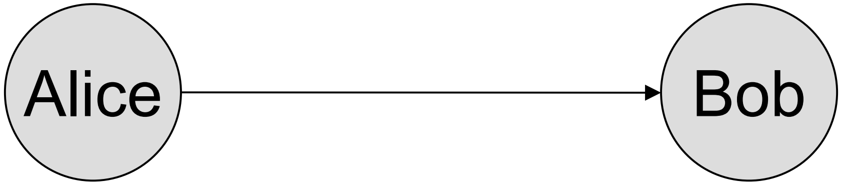
Social Networks as Graphs

(i.e. the math)



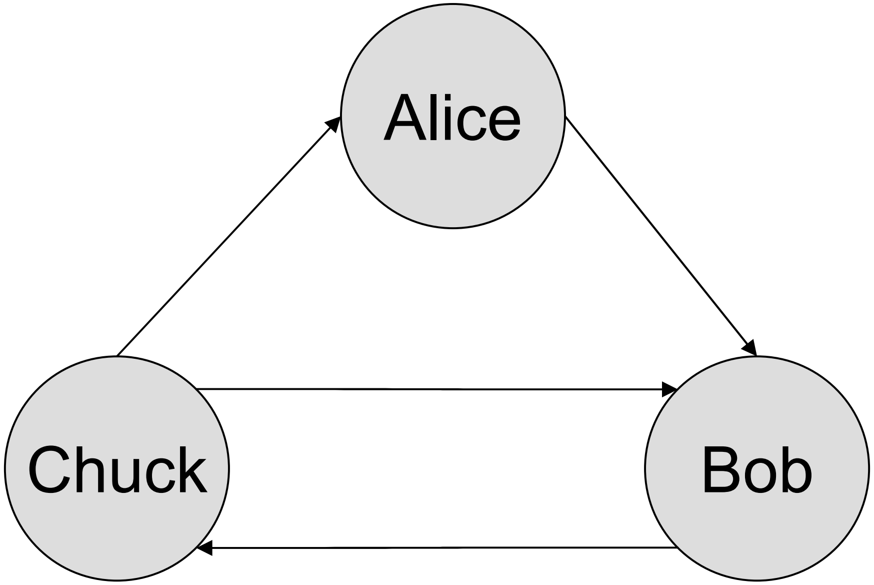
Building the Graph

- Each person is a node
- Each relationship between people is an edge
- E.g. Alice knows Bob

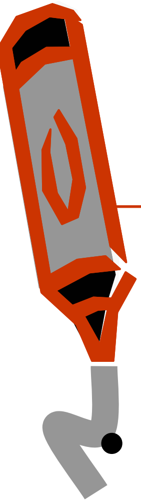


Graph Properties

- Edges can be directed or undirected
- Graphs will have cycles



Graph Properties



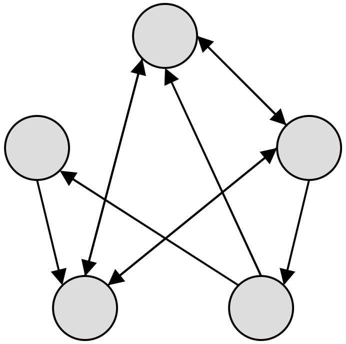
- Centrality
  - Degree
  - Closeness
  - Eigenvector centrality
- Clustering Coefficient (connectance)

Small Worlds

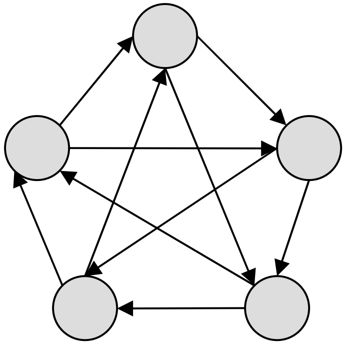
- Watts & Strogatz
- Small World networks have short average path length and high clustering coefficients
- Social Networks are almost always small world networks

Making Small World Networks

- Short Average path length
  - Like what we find in random graphs



- High connectance
  - Like what we find in lattices or other regular graphs



Origins SF

Scale free networks

(1) The number of nodes (N) is NOT fixed.

Networks continuously expand by the addition of new nodes

Examples:
WWW : addition of new documents
Business : new companies emerge

(2) The attachment is NOT uniform.

A node is linked with higher probability to a node that already has a large number of links.

Examples :
WWW : new documents link to well known sites (CNN, YAHOO, NewYork Times, etc)

Business: collaboration with well established partners

Scale-free model

BA model

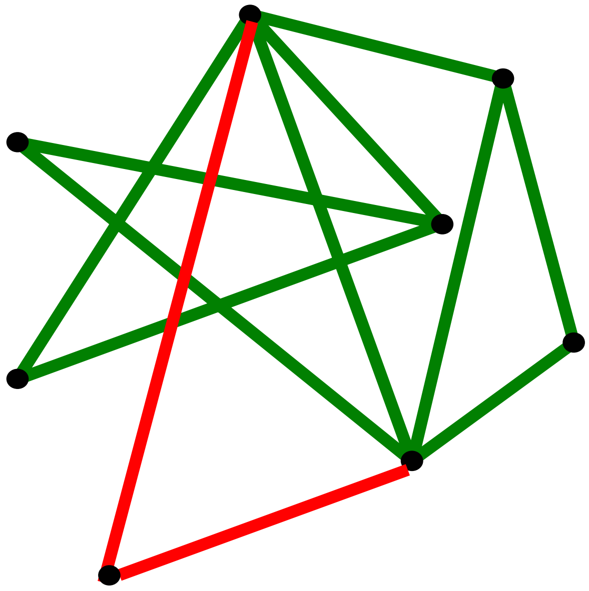
(1) GROWTH :

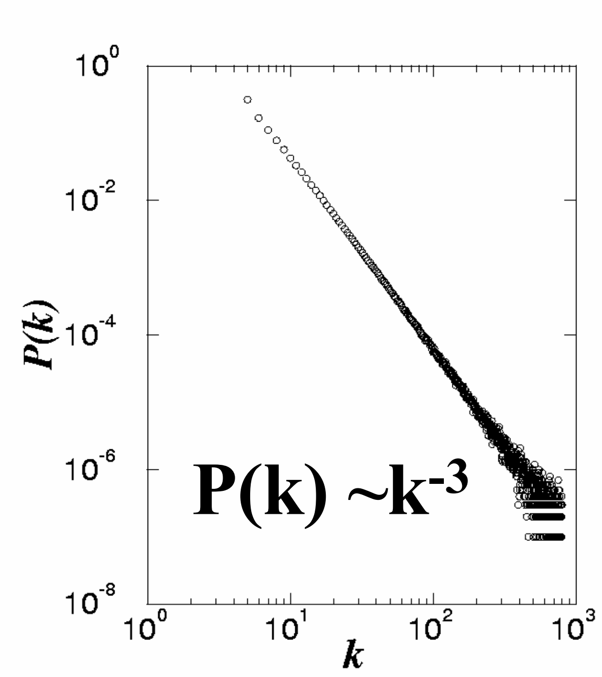
At every timestep we add a new node with *m* edges (connected to the nodes already present in the system).

(2) PREFERENTIAL ATTACHMENT :

The probability **Π** that a new node will be connected to node *i* depends on the connectivity k_i of that node

$$\Pi(k_i) = \frac{k_i}{\Sigma_j k_j}$$



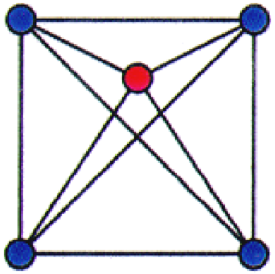


$P(k) \sim k^{-3}$

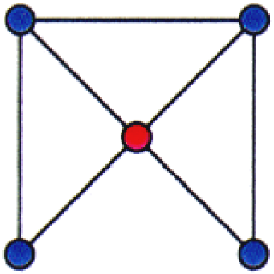
A.-L.Barabási, R. Albert, Science **286,** 509 (1999)

FEATURE: Local clustering

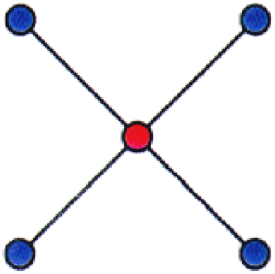
Clustering: My friends will likely know each other!



C=1



C=1/2



C=0

Real life networks are clustered [large C]

FEATURE: Small worlds

Although the networks are considerably huge, mutual distances remain small.

Social networks: ***6*** degrees of separation
WWW: ***19*** clicks to reach every web site

Real life networks have a small diameter.

FEATURE: Hubs

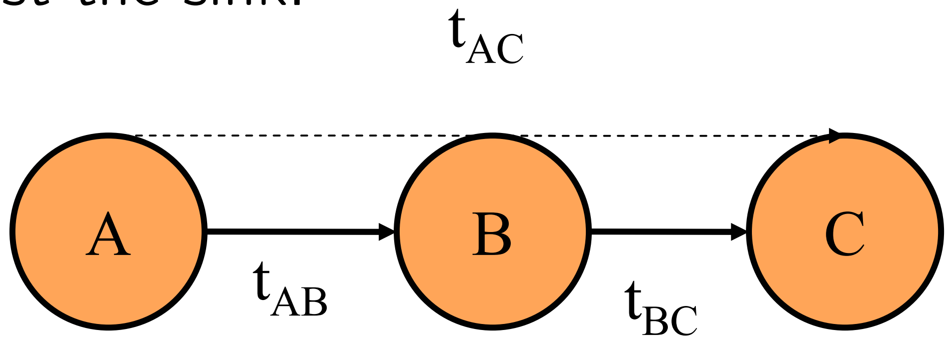
Real life networks are governed by a small number of highly linked nodes which appeared early in the network's emergence process ('first-get-rich')

Real life networks are robust against random attack

But vulnerable upon targeted attack of their hubs

Application: Inferring Trust

- Given a network with trust ratings, we can *infer* how much two people that don't know each other may trust one another
- The Goal: Select two individuals - the *source* (node A) and *sink* (node C) - and recommend to the source how much to trust the sink.



Using Computations

- More email: TrustMail
- Recommender Systems: FilmTrust
- Browsing Support: SocialBrowsing

Application: Information Diffusion

- Authoritative sources
- Small sources

Application: “Collaborations”

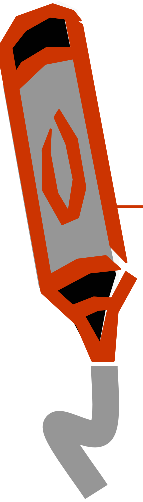
- Recommendations
- Annotations

END OF LECTURE

- text
- text

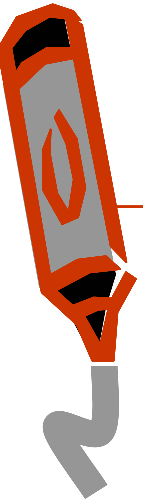
slide

- text
- text



END OF LECTURE

- text
- text



slide

- text
- text