Recommender Systems

Recommender System attributes

- · Need explicit or implicit ratings by user
 - Purchase is 0/1 rating
 - · Movie tickets
 - Books
- Have focused category
 - examples: music, courses, restaurants
 - hard to cross categories with content-based
 - easier to cross categories with collaborative-based
 - users share tastes across categories?

3

1

Recommender Systems

- · Look at classic model and techniques
 - Items
 - Users
 - Recommend Items to Users
- · Recommend new items based on:
 - similarity to items user liked in past: individual history "Content Filtering"
 - Liked by other users similar to this user: collaborative history

2

4

- "Collaborative Filtering"
- Liked by other users: crowd history
 - easier case

Content Filtering

- · Items must have characteristics
- · user values item
 - \Rightarrow values characteristics of item
- model each item as vector of weights of characteristics
 - much like vector-based IR
- user can give explicit preferences for certain characteristics

Buy/no buy prediction method: similarity with centroid

- Average vectors of items user bought

 user's centroid
- · Find similarity of new items to user's centroid
- Decide threshold for "buy" recommendation

5

7

Example

- user bought book 1 and book 2
- Average books bought = (0, 1, 0.5, 0)
- Score new books
 - dot product gives: score(A) = 0.5; score (B)= 1
- · decide threshold for recommendation

	1 st person	romance	mystery	sci-fi
book 1	0	1	1	0
book 2	0	1	0	0
new book A	1	.5	0	0
new book B	0	1	0	.2

Method issues

- · Centroid best way to build a preference vector?
- What metric use for similarity between new items and preference vector?
 Normalization?
- What if users give ratings?
 - Centroid per rating value?
- how include explicit user preferences
- How determine threshold?

Example with explicit user preferences

How use scores of books bought?

Try: preference vector p where component k = user pref for characteristic k if ≠ 0 avg. comp. k of books bought when user pref =0 0 pref for user = "don't care"

		1 st per	rom	mys	SCI-TI	
p =(0, 1, 0.5, -5)	user pref	0	1	0	-5	
New scores? p •A = 0.5	book 1	0	1	1	0	
p •B = 0	book 2	0	1	0	0	
	new A	1	.5	0	0	
	new B	0	1	0	.2	8

Other methods: machine learning

- Major alternatives based on classifiers
 - Training set: items bought and not bought

9

11

- Train classifier many algorithms
- Classify new item as buy/no buy
- Observations
 - Uses books not bought. Problems?
 - Multiple rating value
 Can use multiple classes

Limitations of Content Filtering

- Can only recommend items similar to those user rated highly
- New users – Insufficient number of rated items
- Only consider features explicitly associated with items
 - Do not include attributes of user

Applying content filtering methods to search

- Characterize documents (info. objects)
 - topic analysis?
 - other properties, e.g.:
 - Domain of source
 - Date of publication/update
- Characterize individuals
 - deduce from properties of objects interact with
 - user provided preferences

Applying content filtering methods to search, cont.

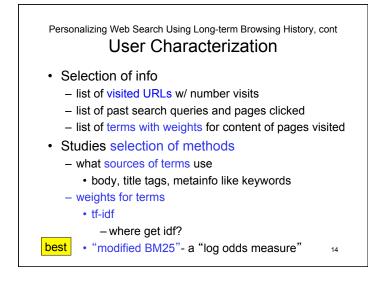
10

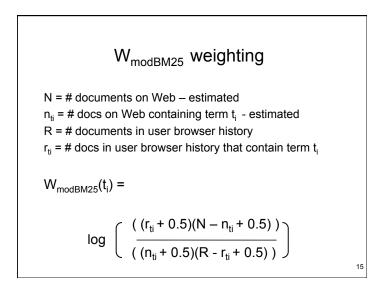
- Query filters documents to consider
 - Convert query to topic-based?
 - Too error prone?
 - Modify query to bias towards user's preferred topics?
- Ranking is recommendation
 Use similarity to user's characterization

Example study: Personalizing Web Search Using Long-term Browsing History (in *WSDM11*)

- Goal: rerank
 top 50 results from Google query
- · Query is initial filter to get results from Google
- · Strategy:
 - score snippets from search result against user profile
 - rerank based on snippet score

13







- · Characterization
 - words in snippet
 - original rank by Google search
- Scoring
 - best performing: language-based model
 - based on content (terms)
 - adjustments for
 - URLs previously visited
 - original rank of snippet in search

Scoring a snippet

N_{si}= # unique words in snippet s_i

- r_{si} = rank of snippet s_i in original search results
- n_i = # previous visits by user to web page with snippet s_i
- $w(t_k)$ = weigth of term t_k in user profile
- w_{total} = sum of all term weights in user profile

$$score_{lang.\;model}\;(s_i) = \Sigma_{k=0}^{N_{si}}\; \text{log}\;(\;(w(t_k)\;\text{+}1)/w_{total})$$

• modif. for URLs previously visited:

 $score_{w/URL}(s_i) = score(s_i)^*(1+\alpha^*n_i)$ parameter α

modif to acct. for orig. rank:

 $score_{w/orig}(s_i) = score(s_i)^*(1/(1+\log (r_{si})))$

17

- Personalizing Web Search Using Long-term Browsing History Evaluation
- "offline" evaluation:
 - relevance judgments by volunteers
 - used to select best of algorithmic variations
- online evaluation of best variations:
 - add-on to Browser by volunteers
 - interleave original results (no personalization) with results reranked by snippet score
 - record clicks by user which list from

18

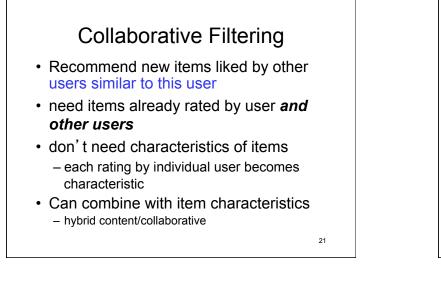
20

- Offline: normalized DCG, avg. of 72 queries
 - Google's ranking w/out personalization: 0.502
 - best-performing of variations for reranking: 0.573
- Online
 - 8% queries: # clicks from original and reranked same
 - of rest: 60.5% queries: more clicks from reranked
 39.5% queries: more clicks from original

Observation

 Reranking can be done completely in browser if enough space for data for user profile What we've just seen: Applying content filtering to search

Now back to recommender systems: Collaborative Filtering



Major method types

- Nearest neighbor
 - Use similarity function
 - Prediction based on previously rated items

22

- Matrix Factorization
 - "Latent factors"
 - Matrix decomposition
- Both use (user × item) matrix – vector similarity

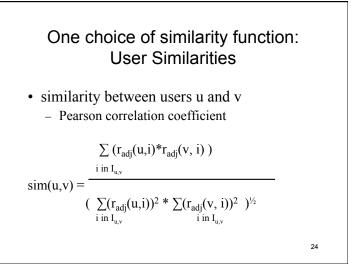
Example of nearest neighbor: Preliminaries

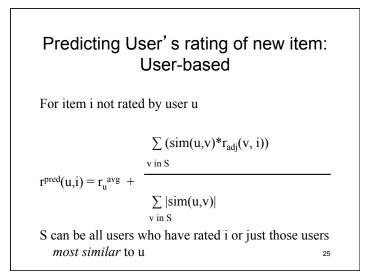
• Notation

- r(u,i) = rating of ith item by user u
- $-I_u =$ set of items rated by user u
- $I_{\boldsymbol{u},\boldsymbol{v}}$ = set of items rated by both users \boldsymbol{u} and \boldsymbol{v}
- $-U_{i,j}$ = set of users that rated items i and j
- Adjust scales for user differences

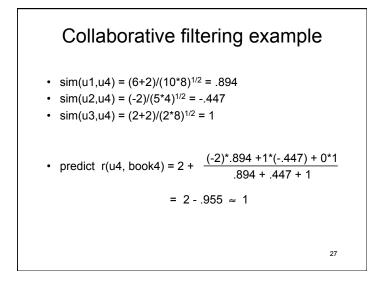
$$r_u^{avg} = (1/|I_u|) * \sum_{i=1}^{n} r(u,i)$$

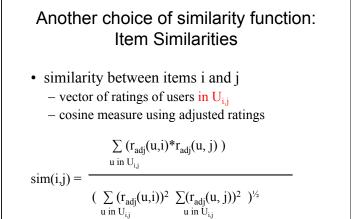
- Adjusted ratings:
$$r_{adi}(u,i) = r(u,i) - r_u^{avg}$$

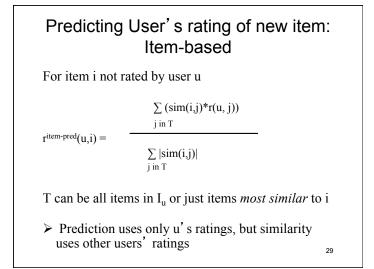




			ening d	examp	ле
user		book 1	book 2	book 3	book 4
ratings	user 1	5	1	2	0
	user 2	x	5	2	5
	user 3	3	1	x	2
	user 4	4	0	2	?
[book 1	book 2	book 3	book 4
adj.	user 1	3	-1	0	-2
user ratings	user 2	x	1	-2	1
	user 3	1	-1	x	0
	user 4	2	-2	0	?







Limitations May not have enough ratings for new users New items may not be rated by enough users Need "critical mass" of users All similarities based on user ratings But can take user "out of comfort zone"

Applying nearest-neighbor collab. filtering concepts to search

- · Collaborative histories
 - How determine user similarity?
 - Clicking URL = buying product?
 - · Behavior on only identical searches?
 - Exact URLs or general topic interests?
 - Hybrid content-based and behavior-based
 - Computational expense?
 - Argues for general topic-interest characterizations

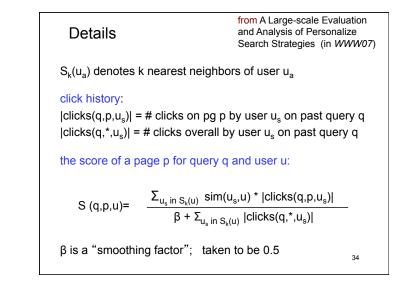
31

- How apply similarity?
 - Same search? or Same topic of search?
 - Bias ranking? or Bias topics of results?

Example from A Large-scale Evaluation and Analysis of Personalize Search Strategies (in *WWW07*)

- Goal: rerank search results
- Based on query log history clicks as ratings
- Also uses 67 pre-defined topic categories
- Strategy:
 - get similarity of users based on user history of visited pages
 - find K most similar users to user doing search K nearest neighbor; use K=50
 - calc. score for each result of search based on click history of K nearest neighbors
 - rerank results of search based on score

Details	from A Large-scale Evaluation and Analysis of Personalize Search Strategies (in WWW07)					
P(u) = collection	of Web pages visited by user u in the past					
P(p u) = # ti	# times u clicked on page p in past					
tota	total # times u clicked on a page in past					
w(p) = log(total	w(p) = log(total # users / # users visited page p)					
"impact we	"impact weight" - idf-like					
c(p) = "category	c(p) = "category vector" for page p					
do classification of page						
vector gives confidence # for top 6 categories (other entries 0)						
User profile	$\boldsymbol{c}_{\ell}(u) = \sum_{p \text{ in } P(u)} P(p u)w(p)\boldsymbol{c}(p)$ hybrid!					
User similarity	sim(u ₁ , u ₂) = $\frac{\boldsymbol{c}_{\ell}(u_1) \cdot \boldsymbol{c}_{\ell}(u_2)}{ \boldsymbol{c}_{\ell}(u_1) \boldsymbol{c}_{\ell}(u_2) }$ 33					



Experiments	from A Large-scale Evaluation and Analysis of Personalize Search Strategies (in WWW07)
 Data set: MSN query I sampled 10,000 distinct used 11 days for training ~ 4000 test queries 	tusers
and order given by pa • Evaluation: 2 metrics	using a "fusion" of original rank age scores S(q,p,u) h clicking indicating relevance
	35

from A Large-scale Evaluation and Analysis of Personalize Search Strategies (in WWW07)				
Good news: re-ranking improves over original ranking				
queries where there is room				
zation improves 3.68%				
q,p,u) s(q,*, u) 36				

Where are we?

- ✓ Refinement/Personalization of results
- Study techniques of
 - Recommender systems
 - ✓ Content filtering
 - Applying content filtering to search
 - Collaborative filtering
 - ✓Nearest neighbor methods
 - Applying nearest neighbor method to search
- **NEXT** Matrix factorization methods
- 37