PURDUE UNIVERSITY

CS54701: Information Retrieval

Collaborative Filtering

18 February 2016

Prof. Chris Clifton

Indiana Center for Database Systems ™

Collaborative Filtering

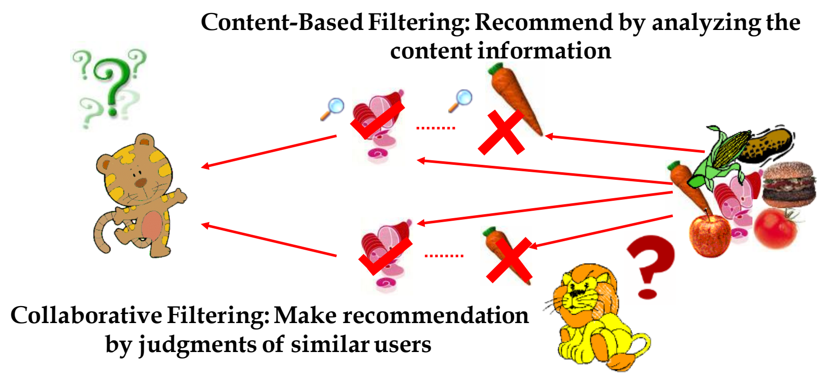
Outline

- Introduction to collaborative filtering
- Main framework
- Memory-based collaborative filtering approach
- Model-based collaborative filtering approach
  - Aspect model & Two-way clustering model
  - Flexible mixture model
  - Decouple model
- Unified filtering by combining content and collaborative filtering

What is Collaborative Filtering?

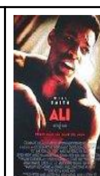
Collaborative Filtering (CF):

Making recommendation decisions for a specific user based on the judgments of users with similar tastes



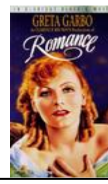
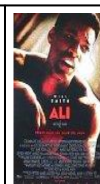
What is Collaborative Filtering?

Collaborative Filtering (CF): Making recommendation decisions for a specific user based on the judgments of users with similar tastes

					
Train_User 1	1	5	3	3	4
Train_User 2	4	1	5	3	2
Test User	1	?	3		4

What is Collaborative Filtering?

Collaborative Filtering (CF): Making recommendation decisions for a specific user based on the judgments of users with similar tastes

					
Train_User 1	1	5	3	3	4
Train_User 2	4	1	5	3	2
Test User	1	5	3		4

Why Collaborative Filtering?

- Advantages of Collaborative Filtering
  - Collaborative Filtering does not need content information as required by CBF
  - The contents of items belong to the third-party (not accessible or available)
  - The contents of items are difficult to index or analyze (e.g., multimedia information)
- Problems of Collaborative Filtering
  - Privacy issues, how to share one's interest without disclosing too much detailed information?

Why Collaborative Filtering?

- Applications Collaborative Filtering
  - E-Commerce
  - Email ranking: borrow email ranking from your office mates (be careful…)
  - Web search? (e.g., local search)

Formal Framework for Collaborative Filtering

Objects: O_m

		O_1	O_2	O_3 ……	O_j …………	O_M
Training Users: U_n	U_1	3	2	4		
	U_2		4	1		1
	⋮					
	U_i					
	⋮					
	U_N	5		2		2
Test User U_t		2	3			

$R_u^t(O_j)=$?

What we have:

- Assume there are some ratings by training users
- Test user provides some amount of additional training data

What we do:

- Predict test user's rating based training information

Memory-Based Approaches

- Memory-Based Approaches
  - Given a specific user *u*, find a set of similar users
  - Predict *u*'s rating based on ratings of similar users
- Issues
  - How to determine the similarity between users?
  - How to combine the ratings from similar users to make the predictions (how to weight different users)?

Memory-Based Approaches

- How to determine the similarity between users?
  - Measure the similarity in rating patterns between different users

Pearson Correlation Coefficient Similarity

$$w_{u,u^t} = \frac{\sum (R_{u^t}(o) - \bar{R}_{u^t})(R_u(o) - \bar{R}_u)}{\sqrt{\sum (R_{u^t}(o) - \bar{R}_{u^t})^2}\sqrt{\sum (R_u(o) - \bar{R}_u)^2}}$$

Average Ratings

Vector Space Similarity

$$w_{u,u^t} = \frac{\sum R_{u^t}(o) R_u(o)}{\sqrt{\sum R_{u^t}(o)^2}\sqrt{\sum R_u(o)^2}}$$

Prediction:

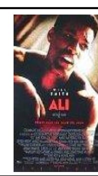
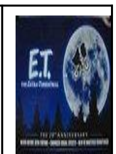
$$\hat{R}_{u^t}(o) = \bar{R}_{u^t} + \frac{\sum_u w_{u,u^t}(R_u(o) - \bar{R}_u)}{\sum_u |w_{u,u^t}|}$$

Memory-Based Approaches

- How to combine the ratings from similar users for predicting?
  - Weight similar users by their similarity with a specific user; use these weights to combine their ratings.

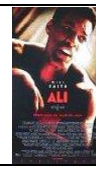
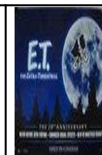
Prediction: $$\hat{R}_{u'}(o) = \bar{R}_{u'} + \frac{\sum_u w_{u,u'}(R_u(o) - \bar{R}_u)}{\sum_u |w_{u,u'}|}$$

Memory-Based Approaches

					
Train_User 1	1	5	3	3	4
Train_User 2	4	1	5	3	2
Test User	1	?	3		4

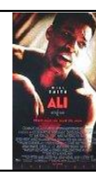
Remove User-specific Rating Bias

Memory-Based Approaches

					
Train_User 1	1	5	3	3	4
Sub Mean (Train1)	-2.2	1.8	-0.2	-0.2	0.8
Train_User 2	4	1	5	3	2
Sub Mean (Train2)	1	-2	2	0	-1
Test User	1	?	3		4
Sub Mean (Test)	-1.667		0.333		1.33

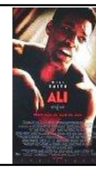
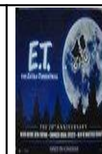
Normalize Rating

Memory-Based Approaches

					
Train_User 1	1	5	3	3	4
Sub Mean (Train1)	-2.2	1.8	-0.2	-0.2	0.8
Train_User 2	4	1	5	3	2
Sub Mean (Train2)	1	-2	2	0	-1
Test User	1	?	3		4
Sub Mean (Test)	-1.667		0.333		1.33

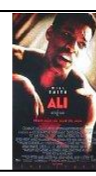
Calculate Similarity: Wtrn1_test=0.92; Wtrn2_test=-0.44;

Memory-Based Approaches

					
Train_User 1	1	5	3	3	4
Sub Mean (Train1)	-2.2	1.8	-0.2	-0.2	0.8
Train_User 2	4	1	5	3	2
Sub Mean (Train2)	1	-2	2	0	-1
Test User	1	?	3		4
Sub Mean (Test)	-1.667		0.333		1.33

Make Prediction: 2.67+(1.8*0.92+(-2)*(-0.44))/(0.92+0.44)=4.54

Memory-Based Approaches

					
Train_User 1	1	5	3	3	4
Sub Mean (Train1)	-2.2	1.8	-0.2	-0.2	0.8
Train_User 2	4	1	5	3	2
Sub Mean (Train2)	1	-2	2	0	-1
Test User	1	5	3		4
Sub Mean (Test)	-1.667		0.333		1.33

Make Prediction: 2.67+(1.8*0.92+(-2)*(-0.44))/(0.92+0.44)=4.54

Memory-Based Approaches

- Problems with memory-based approaches
  - Associated a large amount of computation online costs
    (have to go over all users, any fast indexing approach?)
  - Heuristic method to calculate user similarity and make user rating prediction
- Possible Solution
  - Cluster users/items in offline manner, save for online computation cost
  - Proposal more solid probabilistic modeling method

PURDUE
UNIVERSITY

CS54701: Information Retrieval

Collaborative Filtering

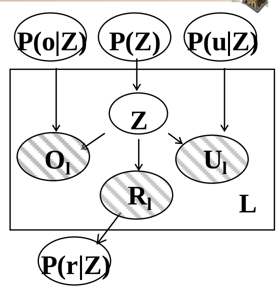
23 February 2016

Prof. Chris Clifton

Indiana Center for Database Systems ™

Collaborative Filtering

- Model-Based Approaches:
- **Aspect Model (Hofmann et al., 1999)**
  - Model individual ratings as convex combination of preference factors



$$P(o_{(l)},u_{(l)},r_{(l)})=\sum_{z\in Z}P(z)P(o_{(l)}\mid z)P(u_{(l)}\mid z)P(r_{(l)}\mid z)$$

Two-Sided Clustering Model (Hofmann et al., 1999)

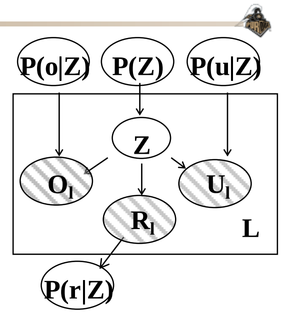
- Assume each user and item belong to one user and item group.

$$P(o_{(l)},u_{(l)},r_{(l)})=P(o_{(l)})P(u_{(l)})\sum_{v,u}I_{x(l)v}J_{y(l)u}C_{vu}$$

$I_{x(l)v}$, $J_{y(l)u}$: Indicator Variables C_{vu}: Associaion Parameter

Collaborative Filtering

- Model-Based Approaches:
- **Aspect Model (Hofmann et al., 1999)**
  - Model individual ratings as convex combination of preference factors



$$P(o_{(l)},u_{(l)},r_{(l)})=\sum_{z\in Z}P(z)P(o_{(l)}\mid z)P(u_{(l)}\mid z)P(r_{(l)}\mid z)$$

Brief description for the expectation maximization training process…

Collaborative Filtering

Thoughts:

- Previous algorithms all cluster users and objects either implicitly (memory-based) or explicitly (model-based)
  - Aspect model allows users and objects to belong to different classes, but cluster them together
  - Two-sided clustering model clusters users and objects separately, but only allows them to belong to one single class

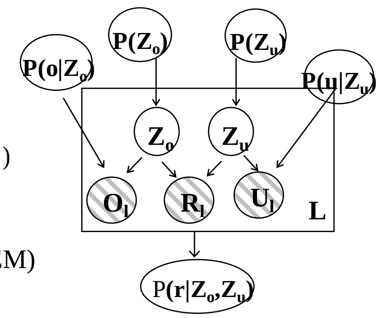
Previous Work: Thoughts

Cluster users and objects separately AND allow them to belong to different classes → **Flexible Mixture Model (FMM)**

Collaborative Filtering

- Flexible Mixture Model (FMM):

Cluster users and objects separately AND allow them to belong to different classes



$$P(o_{(l)},u_{(l)},r_{(l)}) = \sum_{Z_o,Z_u} P(Z_o)P(Z_u)P(o_{(l)}\,|\,Z_o)P(u_{(l)}\,|\,Z_u)P(r_{(l)}\,|\,Z_o,Z_u)$$

- Training Procedure:
  Annealed Expectation Maximization (AEM) algorithm

E-Step: Calculate Posterior Probabilities

$$P(z_o,z_u\,|\,o_{(l)},u_{(l)},r_{(l)}) = \frac{(P(Z_o)P(Z_u)P(o_{(l)}\,|\,Z_o)P(u_{(l)}\,|\,Z_u)P(r_{(l)}\,|\,Z_o,Z_u))^b}{\sum_{Z_o,Z_u}(P(Z_o)P(Z_u)P(o_{(l)}\,|\,Z_o)P(u_{(l)}\,|\,Z_u)P(r_{(l)}\,|\,Z_o,Z_u))^b}$$

Collaborative Filtering

$$P(Z_o);\ P(Z_u);\ P(o_{(l)}\,|\,Z_o);\ P(u_{(l)}\,|\,Z_u);\ P(r_{(l)}\,|\,Z_o,Z_u)$$

M-Step: Update Parameters

- Prediction Procedure:
  Fold-In process to calculate join probabilities

$$P(o,u^t,r_{(l)}) = \sum_{Z_o,Z_u} P(Z_o)P(Z_u)P(o\,|\,Z_o)P(u^t\,|\,Z_u)P(r\,|\,Z_o,Z_u)$$

Fold-in process by EM algorithm

Calculate expectation for prediction

$$\hat{R}_{u^t}(o) = \sum_r r \frac{P(o,u^t,r)}{\sum_{r'} P(o,u^t,r')}$$

"Flexible Mixture Model for Collaborative Filtering", ICML'03

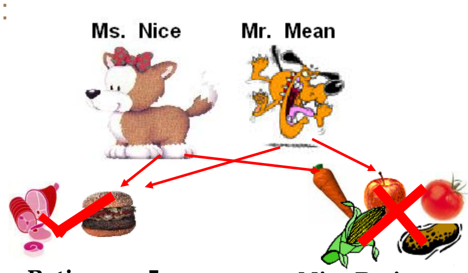
Collaborative Filtering

Thoughts:

- Previous algorithms address the problem that users with similar tastes may have different rating patterns implicitly (Normalize user rating)

Previous Work: Thoughts

•Thoughts:



Nice Rating: 5
Mean Rating: 2

Nice Rating: 3
Mean Rating: 1

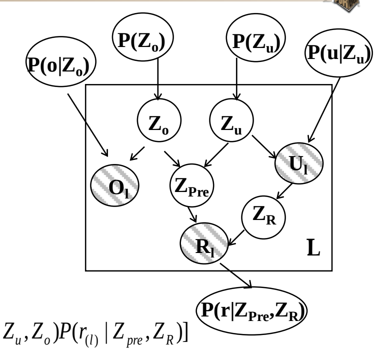
Explicitly decouple users preference values out of the rating values → **Decoupled Model (DM)**

Decoupled Model (DM)

- Decoupled Model (DM):
  Separate preference value

$Z_{pref} \in [1,\ldots,k]$ (1 disfavor, k favor)

from rating $r \in \{1,2,3,4,5\}$



Joint Probability:

$$P(o_{(l)},u_{(l)},r_{(l)}) = \sum_{Z_o,Z_u,Z_R} P(Z_o)P(Z_u)P(o_{(l)}|Z_o)P(u_{(l)}|Z_u)P(Z_R|u_{(l)})[\sum_{Z_{pre}} P(Z_{pre}|Z_u,Z_o)P(r_{(l)}|Z_{pre},Z_R)]$$

"Preference-Based Graphical Model for Collaborative Filtering", UAI'03

"A study of Mixture Model for Collaborative Filtering", Journal of IR

Experimental Data

Datasets:

MovieRating and EachMovie

	MovieRating	EachMovie
Number of Users	500	2000
Number of Movies	1000	1682
Avg. # of rated items/User	87.7	129.6
Scale of ratings	1,2,3,4,5	1,2,3,4,5,6

Evaluation:

MAE: average absolute deviation of the predicted ratings to the actual ratings on items.

$$MAE = \frac{1}{L_{Test}} \sum_{l} | r_{(l)} - \hat{R}_{o_{(l)}}(u_{(l)}) |$$

Collaborative Filtering

Vary Number of Training Users

Test behaviors of algorithms with different amount of training data

- For MovieRating
  100 and 200 training users
- For EachMovie
  200 and 400 training users

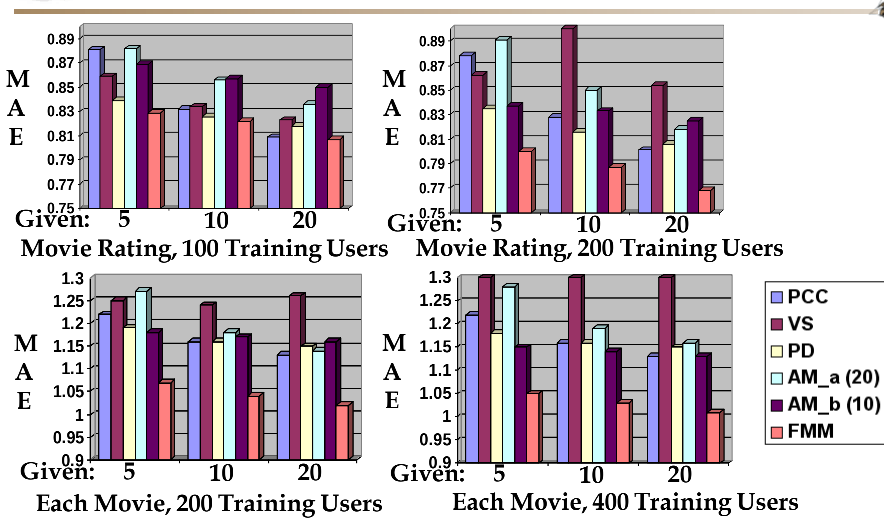
Vary Amount of Given Information from the Test User

Test behaviors of algorithms with different amount of given information from test user

- For both testbeds

Vary among given 5, 10, or 20 items

Results of Flexible Mixture Model



Experimental Results
Improved by Combing FMM and DM

Training Users Size	Algorithms	5 Items Given	10 Items Given	20 Items Given
100	FMM	0.829	0.822	0.807
	FMM+DM	0.792	0772	0.741
200	FMM	0.800	0.787	0.768
	FMM+DM	0.770	0.750	0.728

Results on Movie Rating

Training Users Size	Algorithms	5 Items Given	10 Items Given	20 Items Given
200	FMM	1.07	1.04	1.02
	FMM+DM	1.06	1.01	0.99
400	FMM	1.05	1.03	1.01
	FMM+DM	1.04	1.00	0.97

Results on Each Movie

Combine Collaborative Filtering and Content-Based Filtering

Content-Based Filtering (CBF): Recommend by analyzing the content information

Content information is very useful when few users have rated an object.

	Science Fiction?
A group of **aliens** visit earth…………………….. kind of friendship in which **E.T** learns…………	Yes
Young Harry is in love and wants to marry an actress, much to the displeasure of his family….	No

Unified Filtering (UF): Combining both the content-based information and the collaborative rating information for more accurate recommendation

Content-Based Filtering and Unified Filtering

Content-Based Filtering (CF):

- Generative Methods (e.g. Naïve Bayes)
- Discriminative Methods (e.g. SVM, Logistic Regression)
  - **Usually more accurate**
  - **Can be used to combine features (e.g., actors for movies)**

Unified Filtering by combining CF and CBF:

- Linearly combine the scores from CF and CBF
- Personalized linear combination of the scores
- Bayesian combination with collaborative ensemble learning

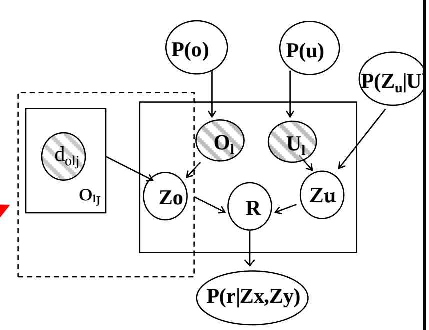
Unified Filtering by flexible mixture model and exponential model

- Unified Filtering with mixture model and exponential model (UFME):

Mixture model for rating information:

$$P(o_{(l)}, u_{(l)}, r_{(l)})$$
$$= \sum_{Z_o, Z_u} P(Z_o \mid \overrightarrow{d_{ol}}) P(Z_u \mid u) P(o_{(l)}) P(u_{(l)}) P(r_{(l)} \mid Z_o, Z_u)$$

Exponential model for content information

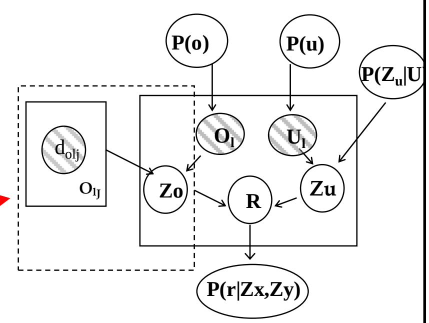


Unified Filtering by flexible mixture model and exponential model

- Unified Filtering with mixture model and exponential model (UFME):

Mixture model for rating information:

Exponential model for content information:



$$P_\theta(Z_o \mid \vec{d}_{ol}) = \frac{\exp(\sum_j \theta_{z_o j} * d_{olj})}{\sum_{Z_o'} (\exp(\sum_j \theta_{z_o j} * d_{olj}))}$$

Specific word

Unified Filtering by flexible mixture model and exponential model

- Training Procedure:

E-Step: Calculate posterior probabilities

Expectation Step of EM

M-Step: Update parameters

Second, refine the object cluster distribution with content information by maximizing

Iterative Scaling Training

"Unified Filtering by Combining Collaborative Filtering and Content-Based Filtering via Mixture Model and Exponential Model ", CIKM'04

Experiment Results

Table. MAE results for four filtering algorithms on EachMovie testbed. Four algorithms are pure content-based filtering (CBF), pure collaborative filtering (CF), unified filtering by combining mixture model and exponential model (UFME)

Training Users Size	Algorithms	0 Items Given	5 Items Given	10 Items Given	20 Items Given
50	CBF	1.43	1.21	1.24	1.19
	CF	1.21	1.14	1.13	1.12
	UFME	1.19	1.11	1.10	1.09
100	CBF	1.43	1.23	1.21	1.19
	CF	1.17	1.08	1.07	1.05
	UFME	1.17	1.08	1.06	1.05

Experiment Results

$P_\theta(Z_o \mid w)$

Table. Five most indicative words (with highest values) for 5 movie clusters, sorted by

Each column corresponds to a different movie cluster. All listed words are stemmed.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
forev	previou	mad	inhabit	custom
depress	passion	hang	dress	hang
mate	court	rape	relat	forev
broken	forget	finish	door	water
abandon	sea	arrest	younger	food

Summary

What we talked about so far?

- Proposed the flexible mixture model
  - Demonstrates the power of clustering users and objects separately AND allowing them to belong to different classes
- Proposed the decoupled model
  - Demonstrates the power of extracting preference values from the surface rating values
- Proposed the unified probabilistic model for unified filtering
  - Demonstrates the power of taking advantage of content information with limited rating information