Effective Missing Data Prediction for Collaborative Filtering

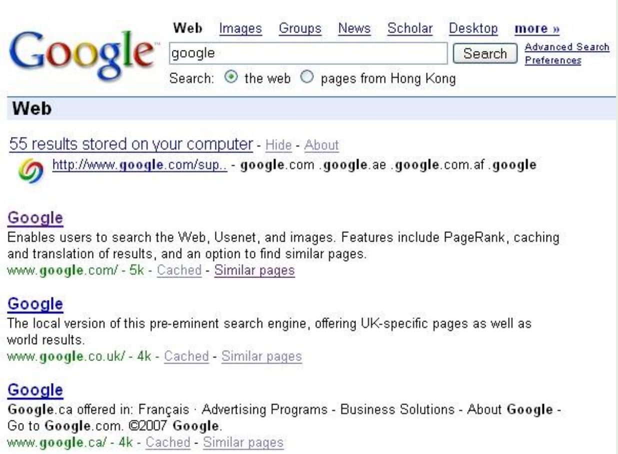
Hao Ma, Irwin King, and Michael R. Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong

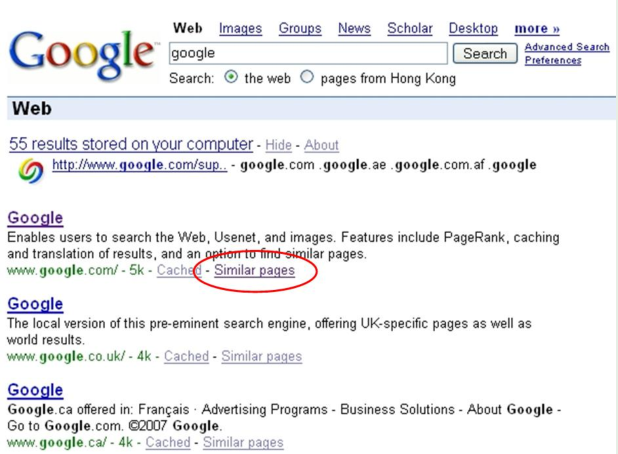
SIGIR 2007, Amsterdam, the Netherlands
July 24, 2007

1. Introduction
   - Simple Examples of Recommender System
   - Definitions of Some Concepts
   - A Simple CF Example
   - Pearson Correlation Coefficient
   - Significance Weighting
2. Missing Data Prediction
   - Collaborative Filtering Challenges
   - User-Item Matrix
   - Similar Neighbors Selection
   - Missing Data Prediction
   - Parameter Discussion
3. Empirical Analysis
   - Datasets
   - Metrics
   - Summary of Experiments
   - Comparisons
   - Impact of Parameters
4. Conclusions and Future Work
   - Conclusions and Future Work

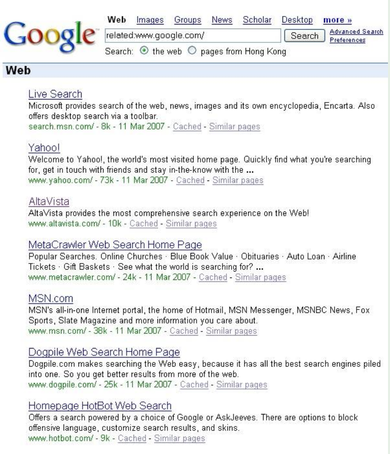
Search Using Google



Search Using Google



Search Using Google



Searching Products on Amazon.com

- If a user is viewing the palm Treo 750 Smartphone on Amazon.com, other related information will be recommended to user besides the specification of Treo 750

Searching Products on Amazon.com

- If a user is viewing the palm Treo 750 Smartphone on Amazon.com, other related information will be recommended to user besides the specification of Treo 750

Searching Products on Amazon.com

Customers who viewed this item also viewed

- Samsung i607 BlackJack Smartphone (Cingular) by Samsung
- BlackBerry 8100c Pearl (Cingular) by BlackBerry
- Cingular 8525 PDA Phone (Cingular) by HTC
- Sony Ericsson W810i Phone (Cingular) by Sony Ericsson

Customers who bought this item also bought

- PREMIUM RAPID CAR CHARGER for PALM TREO 650 / 680 / 700 / 700w / 700p / 700wx / 750 by Mybat
- Platinum Skin Case w/Swivel Clip --Treo 650 700w 700p
- OEM 2GB MINISD Mini Secure Digital (SD) Card 2 GB (Bulk Package) by OEM
- palm Treo 680 Smartphone (Cingular) by Palm

- These methods are very popular in many online recommendation systems

Searching Products on Amazon.com

Customers who viewed this item also viewed

- Samsung i607 BlackJack Smartphone (Cingular) by Samsung
- BlackBerry 8100c Pearl (Cingular) by BlackBerry
- Cingular 8525 PDA Phone (Cingular) by HTC
- Sony Ericsson W810i Phone (Cingular) by Sony Ericsson

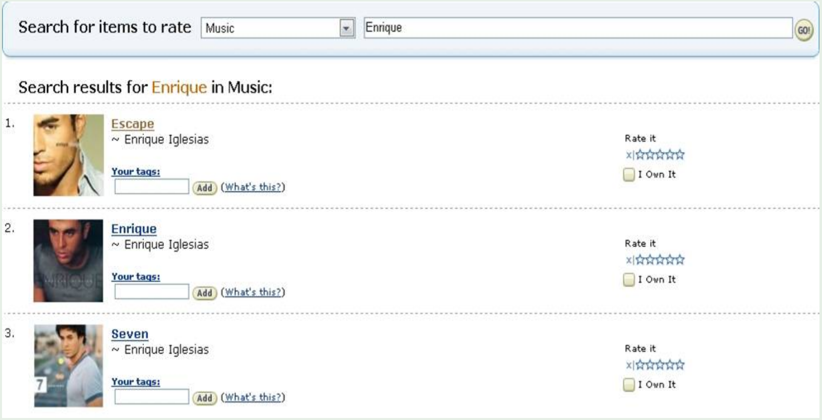
Customers who bought this item also bought

- PREMIUM RAPID CAR CHARGER for PALM TREO 650 / 680 / 700 / 700w / 700p / 700wx / 750 by Mybat
- Platinum Skin Case w/Swivel Clip --Treo 650 700w 700p
- OEM 2GB MINISD Mini Secure Digital (SD) Card 2 GB (Bulk Package) by OEM
- palm Treo 680 Smartphone (Cingular) by Palm

- These methods are very popular in many online recommendation systems

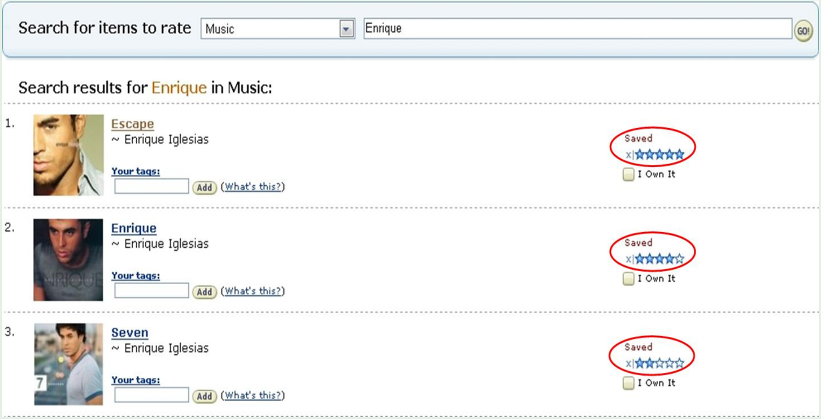
More Complicated Recommendations

More Complicated Recommendations

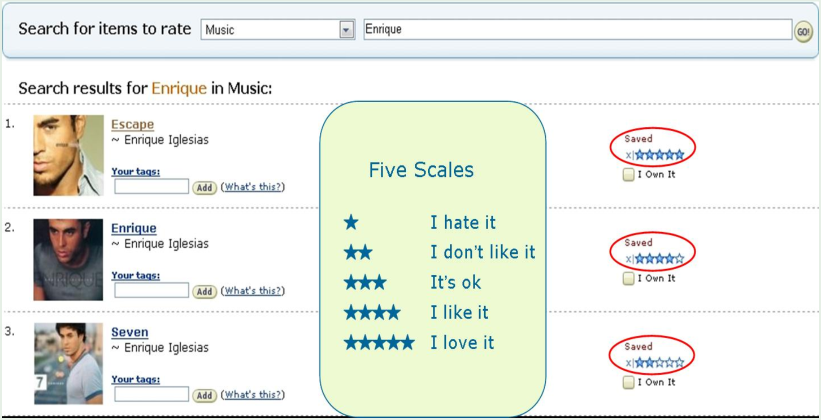


More Complicated Recommendations

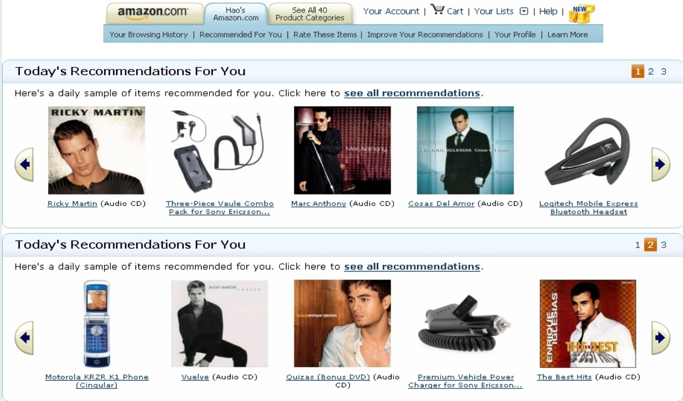
More Complicated Recommendations



More Complicated Recommendations

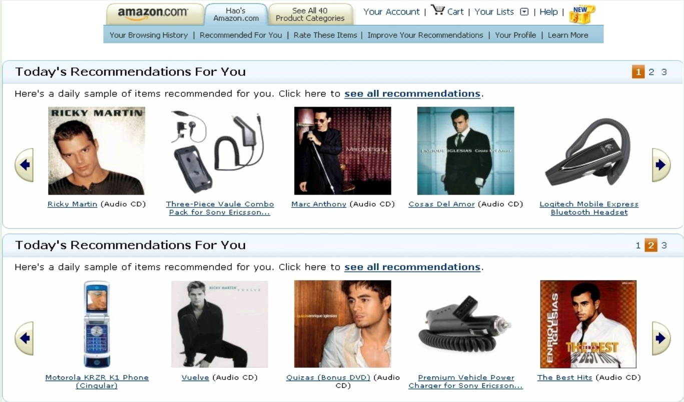


More Complicated Recommendations



- The technique Amazon.com adopts is called Collaborative Filtering!

More Complicated Recommendations



- The technique Amazon.com adopts is called Collaborative Filtering!

Google

- Similarity calculation
- Link analysis

Amazon – Simple Example

- User-item matrix is consisted of lots of 0s and 1s
- Frequent pattern mining

Amazon – Complicated Example

- User-item matrix is consisted of lots of ratings which are rated by different users
- Predict other missing data as accurate as possible

Google

- Similarity calculation
- Link analysis

Amazon – Simple Example

- User-item matrix is consisted of lots of 0s and 1s
- Frequent pattern mining

Amazon – Complicated Example

- User-item matrix is consisted of lots of ratings which are rated by different users
- Predict other missing data as accurate as possible

Google

- Similarity calculation
- Link analysis

Amazon – Simple Example

- User-item matrix is consisted of lots of 0s and 1s
- Frequent pattern mining

Amazon – Complicated Example

- User-item matrix is consisted of lots of ratings which are rated by different users
- Predict other missing data as accurate as possible

Google

- Similarity calculation
- Link analysis

Amazon – Simple Example

- User-item matrix is consisted of lots of 0s and 1s
- Frequent pattern mining

Amazon – Complicated Example

- User-item matrix is consisted of lots of ratings which are rated by different users
- Predict other missing data as accurate as possible

Definition of Recommendation Systems

- Computer programs
- Predict items that a user may be interested in
- Items could be movies, music, books, news, web pages, etc.
- Given some information about the user's profile

Definition of Recommendation Systems

- Computer programs
- Predict items that a user may be interested in
- Items could be movies, music, books, news, web pages, etc.
- Given some information about the user's profile

Definition of Recommendation Systems

- Computer programs
- Predict items that a user may be interested in
- Items could be movies, music, books, news, web pages, etc.
- Given some information about the user's profile

Definition of Recommendation Systems

- Computer programs
- Predict items that a user may be interested in
- Items could be movies, music, books, news, web pages, etc.
- Given some information about the user's profile

Definition of Recommendation Systems

- Computer programs
- Predict items that a user may be interested in
- Items could be movies, music, books, news, web pages, etc.
- Given some information about the user's profile

Definition of Collaborative Filtering

- Making automatic predictions (filtering) about the interests of a user
- By collecting taste information from many other users (collaborating)

Definition of Collaborative Filtering

- Making automatic predictions (filtering) about the interests of a user
- By collecting taste information from many other users (collaborating)

Definition of Collaborative Filtering

- Making automatic predictions (filtering) about the interests of a user
- By collecting taste information from many other users (collaborating)

User-based Collaborative Filtering

		Items												
	u_1													
	u_2	1	3		4		2		5			3	4	
Users	u_3													
	u_4		3		4			3	4		3	4		4
	u_5													
	u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

Items

Users

u_1													
u_2	1	3		4		2		5			3	4	
u_3													
u_4		3		4			3	4		3	4		4
u_5													
u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

Items

Users

u_1													
u_2	1	3		4		2		5			3	4	
u_3													
u_4		3		4			3	4		3	4		4
u_5													
u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

Items

Users

u_1													
u_2	1	3		4		2		5			3	4	
u_3													
u_4		3		4			3	4		3	4		4
u_5													
u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

Items

Users

u_1													
u_2	1	3		4		2		5			3	4	
u_3													
u_4		3		4			3	4		3	4		4
u_5													
u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

Items

Users

u_1													
u_2	1	3		4		2		5			3	4	
u_3													
u_4		3		4			3	4		3	4		4
u_5													
u_6	1			3	5	2		4	1			3	

User-based Collaborative Filtering

- User-based collaborative filtering predicts the ratings of active users based on the ratings of similar users found in the user-item matrix
- The similarity between users could be defined as:

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a) \cdot (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

- $Sim(a,u)$ is ranging from $[-1,1]$, and a larger value means users a and u are more similar

User-based Collaborative Filtering

- User-based collaborative filtering predicts the ratings of active users based on the ratings of similar users found in the user-item matrix
- The similarity between users could be defined as:

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a) \cdot (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

- $Sim(a,u)$ is ranging from $[-1,1]$, and a larger value means users a and u are more similar

User-based Collaborative Filtering

- User-based collaborative filtering predicts the ratings of active users based on the ratings of similar users found in the user-item matrix
- The similarity between users could be defined as:

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a) \cdot (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

- $Sim(a,u)$ is ranging from $[-1,1]$, and a larger value means users a and u are more similar

User-based Collaborative Filtering

- User-based collaborative filtering predicts the ratings of active users based on the ratings of similar users found in the user-item matrix
- The similarity between users could be defined as:

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a) \cdot (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}$$

- $Sim(a,u)$ is ranging from $[-1, 1]$, and a larger value means users a and u are more similar

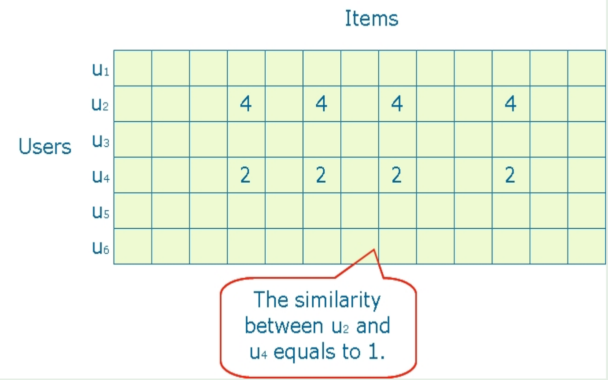
User-based Collaborative Filtering

Items

Users

u_1													
u_2				4		4		4			4		
u_3													
u_4				2		2		2			2		
u_5													
u_6													

User-based Collaborative Filtering



Item-based Collaborative Filtering

- Item-based collaborative filtering predicts the ratings of active users based on the information of similar items computed
- The similarity between items could be defined as:

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}$$

- Like user similarity, item similarity $Sim(i,j)$ is also ranging from $[-1,1]$

Item-based Collaborative Filtering

- Item-based collaborative filtering predicts the ratings of active users based on the information of similar items computed
- The similarity between items could be defined as:

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}$$

- Like user similarity, item similarity $Sim(i,j)$ is also ranging from $[-1, 1]$

Item-based Collaborative Filtering

- Item-based collaborative filtering predicts the ratings of active users based on the information of similar items computed
- The similarity between items could be defined as:

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}$$

- Like user similarity, item similarity $Sim(i,j)$ is also ranging from $[-1,1]$

Item-based Collaborative Filtering

- Item-based collaborative filtering predicts the ratings of active users based on the information of similar items computed
- The similarity between items could be defined as:

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}$$

- Like user similarity, item similarity $Sim(i,j)$ is also ranging from $[-1,1]$

An Example

Items

	1	3	2	5	3	2	3						
Users													
							3	2	1	5	4	1	4

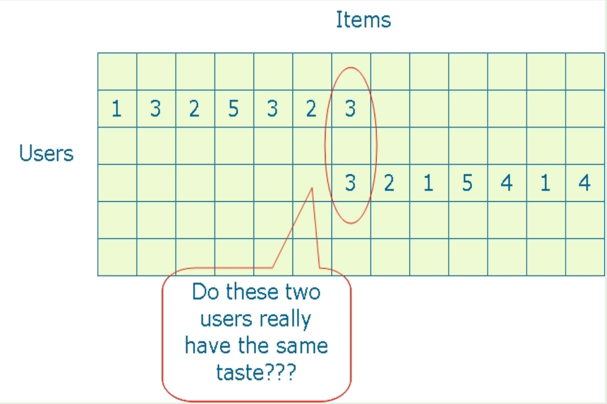
An Example

Items

Users

1	3	2	5	3	2	3						
						3	2	1	5	4	1	4

An Example


Items
Users
1 3 2 5 3 2 3
3 2 1 5 4 1 4
Do these two users really have the same taste???

Significance Weighting

- We use the following equation to solve this problem:

$$Sim'(a,u) = \frac{Min(|I_a \cap I_u|, \gamma)}{\gamma} \cdot Sim(a,u),$$

  where $|I_a \cap I_u|$ is the number of items which user a and user u rated in common

- Then the similarity between items could be defined as:

$$Sim'(i,j) = \frac{Min(|U_i \cap U_j|, \delta)}{\delta} \cdot Sim(i,j),$$

  where $|U_i \cap U_j|$ is the number of users who rated both item i and item j

Significance Weighting

- We use the following equation to solve this problem:

$$Sim'(a,u) = \frac{Min(|I_a \cap I_u|, \gamma)}{\gamma} \cdot Sim(a,u),$$

where $|I_a \cap I_u|$ is the number of items which user a and user u rated in common

- Then the similarity between items could be defined as:

$$Sim'(i,j) = \frac{Min(|U_i \cap U_j|, \delta)}{\delta} \cdot Sim(i,j),$$

where $|U_i \cap U_j|$ is the number of users who rated both item i and item j

Significance Weighting

- We use the following equation to solve this problem:

$$Sim'(a,u) = \frac{Min(|I_a \cap I_u|, \gamma)}{\gamma} \cdot Sim(a,u),$$

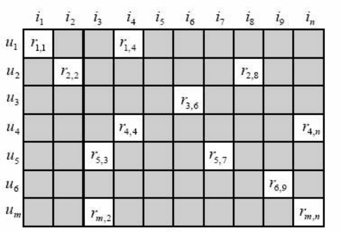
where $|I_a \cap I_u|$ is the number of items which user a and user u rated in common

- Then the similarity between items could be defined as:

$$Sim'(i,j) = \frac{Min(|U_i \cap U_j|, \delta)}{\delta} \cdot Sim(i,j),$$

where $|U_i \cap U_j|$ is the number of users who rated both item i and item j

User-Item Matrix

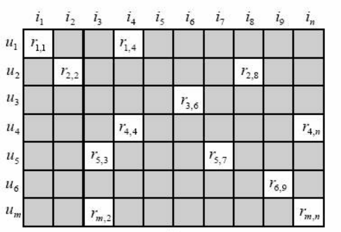


(a)

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

User-Item Matrix

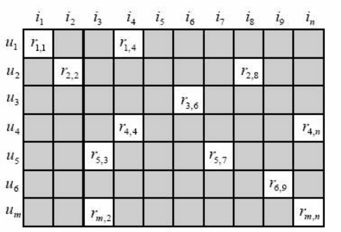


(a)

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

User-Item Matrix

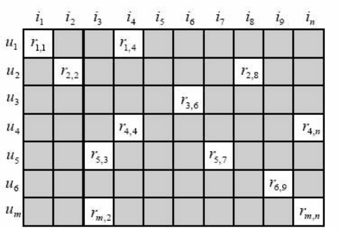


(a)

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

User-Item Matrix

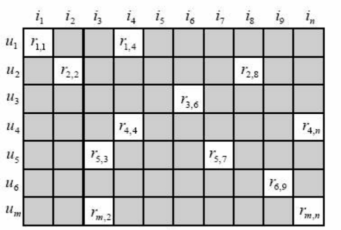


(a)

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

User-Item Matrix



(a)

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

Challenges of Collaborative Filtering

- Data Sparsity
- Prediction Accuracy
- Scalability

Data Sparsity

- Propose an algorithm to increase the density of User-Item Matrix
- Only predict some of the missing data

Prediction Accuracy

- Adopt significance weighting
- Linearly combine user information with item information
- Predict the missing data with high confidence
- Our algorithm increases 6.24% of prediction accuracy over other state-of-the-art methods in average

User-Item Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_n
u_1	$r_{1,1}$			$r_{1,4}$						
u_2		$r_{2,2}$						$r_{2,8}$		
u_3						$r_{3,6}$				
u_4				$r_{4,4}$						$r_{4,n}$
u_5			$r_{5,3}$				$r_{5,7}$			
u_6									$r_{6,9}$	
u_m			$r_{m,2}$							$r_{m,n}$

(a)

Predicted User-Item Matrix

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_n
u_1	$r_{1,1}$	0	$\hat{r}_{1,3}$	$r_{1,4}$	0	$\hat{r}_{1,6}$	0	$\hat{r}_{1,8}$	$\hat{r}_{1,9}$	0
u_2	0	$r_{2,2}$	0	$\hat{r}_{2,4}$	$\hat{r}_{2,5}$	0	$\hat{r}_{2,7}$	$r_{2,8}$	0	$\hat{r}_{2,n}$
u_3	$\hat{r}_{3,1}$	0	$\hat{r}_{3,3}$	$\hat{r}_{3,4}$	$\hat{r}_{3,5}$	$r_{3,6}$	0	$\hat{r}_{3,8}$	$\hat{r}_{3,9}$	0
u_4	$\hat{r}_{4,1}$	$\hat{r}_{4,2}$	0	$r_{4,4}$	$\hat{r}_{4,5}$	$\hat{r}_{4,6}$	$\hat{r}_{4,7}$	0	$\hat{r}_{4,9}$	$r_{4,n}$
u_5	$\hat{r}_{5,1}$	$\hat{r}_{5,2}$	$r_{5,3}$	0	$\hat{r}_{5,5}$	0	$r_{5,7}$	$\hat{r}_{5,8}$	$\hat{r}_{5,9}$	$\hat{r}_{5,n}$
u_6	$\hat{r}_{6,1}$	$\hat{r}_{6,2}$	0	$\hat{r}_{6,4}$	$\hat{r}_{6,5}$	$\hat{r}_{6,6}$	$\hat{r}_{6,7}$	0	$r_{6,9}$	$\hat{r}_{6,n}$
u_m	$\hat{r}_{m,1}$	0	$r_{m,2}$	$\hat{r}_{m,4}$	0	$\hat{r}_{m,6}$	0	$\hat{r}_{m,8}$	$\hat{r}_{m,9}$	$r_{m,n}$

(b)

Similar Neighbors Selection

- For every missing data $r_{u,i}$, a set of similar users $S(u)$ towards user u can be generated according to:

$$S(u) = \{u_a | Sim'(u_a, u) > \eta, u_a \neq u\}$$

where $Sim'(u_a, u)$ is computed using Significance Weighting, and η is the user similarity threshold

- At the same time, for every missing data $r_{u,i}$, a set of similar items $S(i)$ towards item i can be generated according to:

$$S(i) = \{i_k | Sim'(i_k, i) > \theta, i_k \neq i\}$$

where θ is the item similarity threshold

Similar Neighbors Selection

- For every missing data $r_{u,i}$, a set of similar users $S(u)$ towards user u can be generated according to:

$$S(u) = \{u_a | Sim'(u_a, u) > \eta, u_a \neq u\}$$

where $Sim'(u_a, u)$ is computed using Significance Weighting, and η is the user similarity threshold

- At the same time, for every missing data $r_{u,i}$, a set of similar items $S(i)$ towards item i can be generated according to:

$$S(i) = \{i_k | Sim'(i_k, i) > \theta, i_k \neq i\}$$

where θ is the item similarity threshold

Similar Neighbors Selection

- For every missing data $r_{u,i}$, a set of similar users $S(u)$ towards user u can be generated according to:

$$S(u) = \{u_a | Sim'(u_a, u) > \eta, u_a \neq u\}$$

where $Sim'(u_a, u)$ is computed using Significance Weighting, and η is the user similarity threshold

- At the same time, for every missing data $r_{u,i}$, a set of similar items $S(i)$ towards item i can be generated according to:

$$S(i) = \{i_k | Sim'(i_k, i) > \theta, i_k \neq i\}$$

where θ is the item similarity threshold

Missing Data Prediction Algorithm

- Given the missing data $r_{u,i}$, if $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \lambda \times (\overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a, u)}) +$$

$$(1 - \lambda) \times (\overline{i} + \frac{\sum_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} Sim'(i_k, i)})$$

Missing Data Prediction Algorithm

- Given the missing data $r_{u,i}$, if $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \lambda \times (\overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a, u)}) +$$

$$(1 - \lambda) \times (\overline{i} + \frac{\sum_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} Sim'(i_k, i)})$$

Missing Data Prediction Algorithm

- If $S(u) \neq \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a, u)}$$

- If $S(u) = \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{i} + \frac{\sum_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} Sim'(i_k, i)}$$

Missing Data Prediction Algorithm

- If $S(u) \neq \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{u} + \frac{\sum\limits_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum\limits_{u_a \in S(u)} Sim'(u_a, u)}$$

- If $S(u) = \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{i} + \frac{\sum\limits_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum\limits_{i_k \in S(i)} Sim'(i_k, i)}$$

Missing Data Prediction Algorithm

- If $S(u) \neq \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{u} + \frac{\sum\limits_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum\limits_{u_a \in S(u)} Sim'(u_a, u)}$$

- If $S(u) = \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{i} + \frac{\sum\limits_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum\limits_{i_k \in S(i)} Sim'(i_k, i)}$$

Missing Data Prediction Algorithm

- If $S(u) = \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = 0$$

- This consideration is different from all other existing prediction or smoothing methods – they always try to predict all the missing data in the user-item matrix, which will predict some missing data with bad quality

Missing Data Prediction Algorithm

- If $S(u) = \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = 0$$

- This consideration is different from all other existing prediction or smoothing methods – they always try to predict all the missing data in the user-item matrix, which will predict some missing data with bad quality

Missing Data Prediction Algorithm

- If $S(u) = \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = 0$$

- This consideration is different from all other existing prediction or smoothing methods – they always try to predict all the missing data in the user-item matrix, which will predict some missing data with bad quality

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on γ and δ

- Employed to avoid overestimating the user similarities and item similarities
- Too high $\Longrightarrow$ users or items do not have enough neighbors $\Longrightarrow$ decrease of prediction accuracy
- Too low $\Longrightarrow$ overestimate problem still exists $\Longrightarrow$ decrease of prediction accuracy

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on γ and δ

- Employed to avoid overestimating the user similarities and item similarities
- Too high $\Longrightarrow$ users or items do not have enough neighbors $\Longrightarrow$ decrease of prediction accuracy
- Too low $\Longrightarrow$ overestimate problem still exists $\Longrightarrow$ decrease of prediction accuracy

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on γ and δ

- Employed to avoid overestimating the user similarities and item similarities
- Too high $\Longrightarrow$ users or items do not have enough neighbors $\Longrightarrow$ decrease of prediction accuracy
- Too low $\Longrightarrow$ overestimate problem still exists $\Longrightarrow$ decrease of prediction accuracy

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on γ and δ

- Employed to avoid overestimating the user similarities and item similarities
- Too high $\Longrightarrow$ users or items do not have enough neighbors $\Longrightarrow$ decrease of prediction accuracy
- Too low $\Longrightarrow$ overestimate problem still exists $\Longrightarrow$ decrease of prediction accuracy

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on η and θ

- Thresholds to select neighbors
- Too high $\Longrightarrow$ few missing data need to be predicted $\Longrightarrow$ user-item matrix is very sparse
- Too low $\Longrightarrow$ almost all the missing data need to be predicted $\Longrightarrow$ user-item matrix is very dense

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on η and θ

- Thresholds to select neighbors
- Too high $\Longrightarrow$ few missing data need to be predicted$\Longrightarrow$ user-item matrix is very sparse
- Too low $\Longrightarrow$ almost all the missing data need to be predicted $\Longrightarrow$ user-item matrix is very dense

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on η and θ

- Thresholds to select neighbors
- Too high $\Longrightarrow$ few missing data need to be predicted$\Longrightarrow$ user-item matrix is very sparse
- Too low $\Longrightarrow$ almost all the missing data need to be predicted $\Longrightarrow$ user-item matrix is very dense

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on η and θ

- Thresholds to select neighbors
- Too high $\Longrightarrow$ few missing data need to be predicted $\Longrightarrow$ user-item matrix is very sparse
- Too low $\Longrightarrow$ almost all the missing data need to be predicted $\Longrightarrow$ user-item matrix is very dense

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on λ

- Determines how closely the rating prediction relies on user information or item information
- $\lambda = 1 \Longrightarrow$ prediction depends completely upon user-based information
- $\lambda = 0 \Longrightarrow$ prediction depends completely upon item-based information

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on λ

- Determines how closely the rating prediction relies on user information or item information
- $\lambda = 1 \Longrightarrow$ prediction depends completely upon user-based information
- $\lambda = 0 \Longrightarrow$ prediction depends completely upon item-based information

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on λ

- Determines how closely the rating prediction relies on user information or item information
- $\lambda = 1 \Longrightarrow$ prediction depends completely upon user-based information
- $\lambda = 0 \Longrightarrow$ prediction depends completely upon item-based information

Parameter

- γ
- δ
- η
- θ
- λ

Discussion on λ

- Determines how closely the rating prediction relies on user information or item information
- $\lambda = 1 \Longrightarrow$ prediction depends completely upon user-based information
- $\lambda = 0 \Longrightarrow$ prediction depends completely upon item-based information

Parameter Discussion

Table: The relationship between parameters with other CF approaches (MDP: Mission Data Predicted)

λ	η	θ	**Related CF Approaches**
1	1	1	User-based CF without MDP
0	1	1	Item-based CF without MDP
1	0	0	User-based CF with full MDP
0	0	0	Item-based CF with full MDP

Parameter Discussion

Table: The relationship between parameters with other CF approaches (MDP: Mission Data Predicted)

λ	η	θ	**Related CF Approaches**
1	1	1	User-based CF without MDP
0	1	1	Item-based CF without MDP
1	0	0	User-based CF with full MDP
0	0	0	Item-based CF with full MDP

Movielens

- It contains 100,000 ratings (1-5 scales) rated by 943 users on 1,682 movies, and each user at least rated 20 movies. The density of the user-item matrix is:

$$\frac{100000}{943 \times 1682} = 6.30\%$$

- The statistics of dataset MovieLens is summarized in the following table:

Table: Statistics of Dataset MovieLens

Statistics	User	Item
Min. Num. of Ratings	20	1
Max. Num. of Ratings	737	583
Avg. Num. of Ratings	106.04	59.45

Movielens

- It contains 100,000 ratings (1-5 scales) rated by 943 users on 1,682 movies, and each user at least rated 20 movies. The density of the user-item matrix is:

$$\frac{100000}{943 \times 1682} = 6.30\%$$

- The statistics of dataset MovieLens is summarized in the following table:

Table: Statistics of Dataset MovieLens

Statistics	User	Item
Min. Num. of Ratings	20	1
Max. Num. of Ratings	737	583
Avg. Num. of Ratings	106.04	59.45

Movielens

- It contains 100,000 ratings (1-5 scales) rated by 943 users on 1,682 movies, and each user at least rated 20 movies. The density of the user-item matrix is:

$$\frac{100000}{943 \times 1682} = 6.30\%$$

- The statistics of dataset MovieLens is summarized in the following table:

Table: Statistics of Dataset MovieLens

Statistics	User	Item
Min. Num. of Ratings	20	1
Max. Num. of Ratings	737	583
Avg. Num. of Ratings	106.04	59.45

Mean Absolute Errors

- We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach with other collaborative filtering methods
- MAE is defined as:

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \widehat{r}_{u,i}|}{N},$$

where $r_{u,i}$ denotes the rating that user u gave to item i, and $\widehat{r}_{u,i}$ denotes the rating that user u gave to item i which is predicted by our approach, and N denotes the number of tested ratings

Mean Absolute Errors

- We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach with other collaborative filtering methods
- MAE is defined as:

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \widehat{r}_{u,i}|}{N},$$

where $r_{u,i}$ denotes the rating that user u gave to item i, and $\widehat{r}_{u,i}$ denotes the rating that user u gave to item i which is predicted by our approach, and N denotes the number of tested ratings

Mean Absolute Errors

- We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach with other collaborative filtering methods
- MAE is defined as:

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \widehat{r}_{u,i}|}{N},$$

where $r_{u,i}$ denotes the rating that user u gave to item i, and $\widehat{r}_{u,i}$ denotes the rating that user u gave to item i which is predicted by our approach, and N denotes the number of tested ratings

Summary of Experiments

- Comparisons with Traditional PCC Methods
- Comparisons with State-of-the-Art Algorithms
- Impact of Missing Data Prediction
- Impact of γ and δ
- Impact of λ
- Impact of η and θ

Summary of Experiments

- Comparisons with Traditional PCC Methods
- Comparisons with State-of-the-Art Algorithms
- Impact of Missing Data Prediction
- Impact of γ and δ
- Impact of λ
- Impact of η and θ

Comparisons with Traditional PCC Methods

- User-based collaborative filtering using Pearson Correlation Coefficient
- Item-based collaborative filtering using Pearson Correlation Coefficient

Summary of Experiments

- Comparisons with Traditional PCC Methods
- Comparisons with State-of-the-Art Algorithms
- Impact of Missing Data Prediction
- Impact of γ and δ
- Impact of λ
- Impact of η and θ

Comparisons with State-of-the-Art Algorithms

- Similarity Fusion (SF) [J. Wang, et al., SIGIR 2006]
- Smoothing and Cluster-Based PCC (SCBPCC) [G. Xue, et al., SIGIR 2005]
- Aspect Model (AM) [T. Hofmann, TOIS 2004]
- Personality Diagnosis (PD) [D. M. Pennock, et al., UAI 2000]

Summary of Experiments

- Comparisons with Traditional PCC Methods
- Comparisons with State-of-the-Art Algorithms
- Impact of Missing Data Prediction
- Impact of γ and δ
- Impact of λ
- Impact of η and θ

Impact of Missing Data Prediction

- Effective Missing Data Prediction (EMDP)
- Predict Every Missing Data (PEMD)

Summary of Experiments

- Comparisons with Traditional PCC Methods
- Comparisons with State-of-the-Art Algorithms
- Impact of Missing Data Prediction
- Impact of γ and δ
- Impact of λ
- Impact of η and θ

Impact of Parameters

- Impact of each parameter

MAE Comparisons with PCC Methods

Table: MAE comparison with other approaches (A smaller MAE value means a better performance)

Training Users	Methods	Given5	Given10	Given20
MovieLens 300	EMDP	**0.784**	**0.765**	**0.755**
	UPCC	0.838	0.814	0.802
	IPCC	0.870	0.838	0.813
MovieLens 200	EMDP	**0.796**	**0.770**	**0.761**
	UPCC	0.843	0.822	0.807
	IPCC	0.855	0.834	0.812
MovieLens 100	EMDP	**0.811**	**0.778**	**0.769**
	UPCC	0.876	0.847	0.811
	IPCC	0.890	0.850	0.824

MAE Comparisons with PCC Methods

Table: MAE comparison with other approaches (A smaller MAE value means a better performance)

Training Users	Methods	Given5	Given10	Given20
MovieLens 300	EMDP	**0.784**	**0.765**	**0.755**
	UPCC	0.838	0.814	0.802
	IPCC	0.870	0.838	0.813
MovieLens 200	EMDP	**0.796**	**0.770**	**0.761**
	UPCC	0.843	0.822	0.807
	IPCC	0.855	0.834	0.812
MovieLens 100	EMDP	**0.811**	**0.778**	**0.769**
	UPCC	0.876	0.847	0.811
	IPCC	0.890	0.850	0.824

MAE Comparisons with State-of-the-Art Algorithms

Table: MAE comparison with state-of-the-art algorithms (A smaller MAE value means a better performance)

Num. of Training Users	100			200			300		
Ratings Given	5	10	20	5	10	20	5	10	20
EMDP	0.807	0.769	0.765	0.793	0.760	0.751	0.788	0.754	0.746
SF	0.847	0.774	0.792	0.827	0.773	0.783	0.804	0.761	0.769
SCBPCC	0.848	0.819	0.789	0.831	0.813	0.784	0.822	0.810	0.778
AM	0.963	0.922	0.887	0.849	0.837	0.815	0.820	0.822	0.796
PD	0.849	0.817	0.808	0.836	0.815	0.792	0.827	0.815	0.789
PCC	0.874	0.836	0.818	0.859	0.829	0.813	0.849	0.841	0.820

MAE Comparisons with State-of-the-Art Algorithms

Table: MAE comparison with state-of-the-art algorithms (A smaller MAE value means a better performance)

Num. of Training Users	100			200			300		
Ratings Given	5	10	20	5	10	20	5	10	20
EMDP	**0.807**	**0.769**	**0.765**	**0.793**	**0.760**	**0.751**	**0.788**	**0.754**	**0.746**
SF	0.847	0.774	0.792	0.827	0.773	0.783	0.804	0.761	0.769
SCBPCC	0.848	0.819	0.789	0.831	0.813	0.784	0.822	0.810	0.778
AM	0.963	0.922	0.887	0.849	0.837	0.815	0.820	0.822	0.796
PD	0.849	0.817	0.808	0.836	0.815	0.792	0.827	0.815	0.789
PCC	0.874	0.836	0.818	0.859	0.829	0.813	0.849	0.841	0.820

Impact of Missing Data Prediction

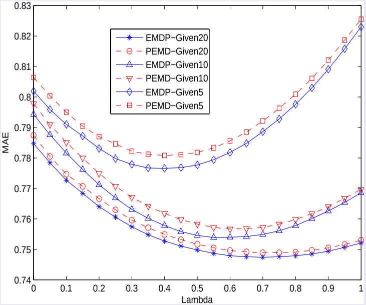


Figure: MAE Comparison of EMDP and PEMD (A smaller MAE value means a better performance)

Impact of γ and δ

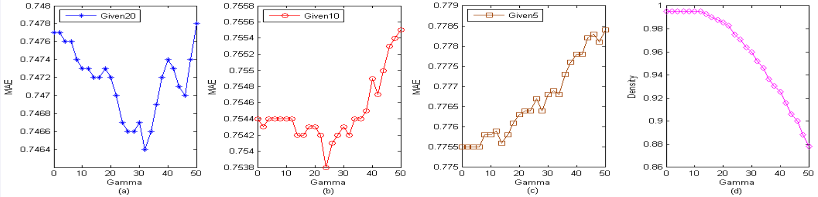


Figure: Impact of γ and δ on MAE and Matrix Density

Impact of λ

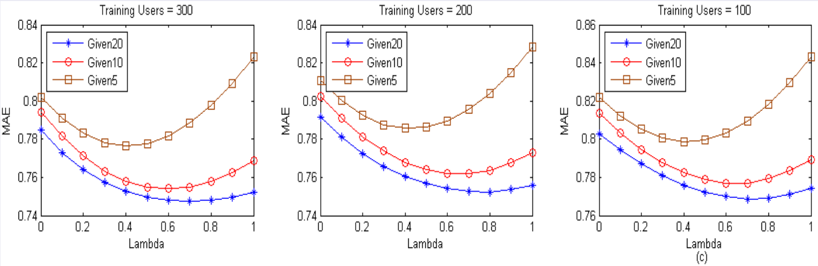


Figure: Impact of λ on MAE

Impact of η and θ

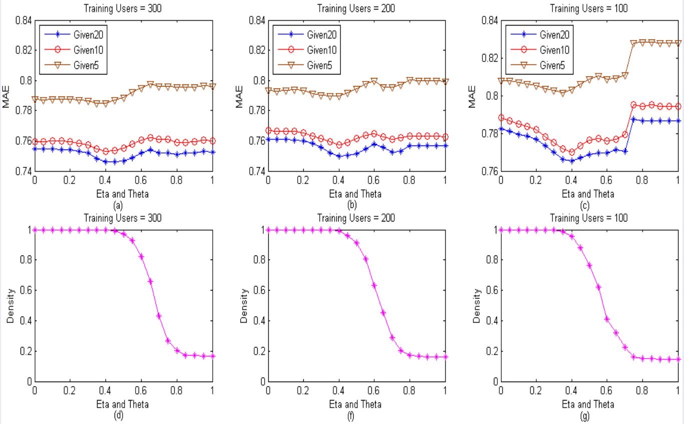


Figure: Impact of η and θ on MAE and Density

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Conclusions

- Proposes an effective missing data prediction algorithm for Collaborative Filtering
- Combines users information and items information together
- Outperforms other state-of-the-art collaborative filtering approaches

Future Work

- Explore the relationship between user information and item information
- Scalability analysis and improvement of our algorithm
- Employ more metrics to measure our algorithm

Q & A

- Home Page: http://www.cse.cuhk.edu.hk/~hma
- Email: hma@cse.cuhk.edu.hk
- Thanks!