DATA ANALYTICS

Unit 4: Collaborative Filtering System

Jyothi R.
Department of Computer Science and Engineering

Basic Models of Recommender Systems

- The basic models for recommender systems work with two kinds of data, which are
- i. User-Item Interactions, such as ratings or buying behavior, and
- ii).The attribute information about the users and items such as textual profiles or relevant keywords.
- Content-based systems also use the ratings matrices in most cases, although the model is usually focused on the ratings of a single user rather than those of all users.
- In knowledge-based recommender systems, the recommendations are based on explicitly specified user requirements.
- Hybrid systems combine the strengths of various types of recommender systems to create techniques that can perform more robustly in a wide variety of settings.

Collaborative Filtering Models

- Collaborative filtering models use the collaborative power of the ratings by multiple users to make recommendations.
- The main challenge in designing collaborative filtering methods is that the underlying ratings matrices are sparse.
- Eg. Movie Recommendations.
- The basic idea of collaborative filtering methods is that these unspecified ratings can be imputed.
- Here the observed ratings are highly correlated across various users and items.
- Most of the models for collaborative filtering focus on leveraging either inter-item correlations or inter-user correlations for the prediction process. Some models also use both types of correlations.
-

Collaborative Filtering Models contd.

- There are two types of methods that are commonly used in collaborative filtering.

a). Memory- based methods: they are also referred to as neighborhood-based collaborative filtering algorithms. In which the ratings of user-item combinations are predicted on the basis of their neighborhoods.

These neighborhoods can be defined in one of two ways

i). User-based Collaborative filtering: The ratings provided by the like-minded users of a target user A are used in order to make the recommendations for A.

ii). Item-based collaborative filtering: To make the rating predictions for target item B by user A, the first step is to determine a set S of items that are most similar to target item B.

- The advantages of memory-based techniques are that they are simple to implement and the resulting recommendations are often easy to explain.

b). Model-based Methods:

Collaborative Filtering Models contd.

b). Model-based Methods: Here the machine learning and data mining methods are used in the context of predictive models.

- In cases where the model is parameterized, the parameters of this model are learned within the context of an optimization framework.
- Examples of model-based methods include Decision trees, Rule-based models, Bayesian methods and latent factor models.

- Collaborative filtering models are closely related to missing value analysis.
- It can be viewed as a special case of problems in which the data matrix is very large and sparse.
- It can also be viewed as generalizations of classification and regression modeling, here the class/dependent variable can be viewed as an attribute with missing values, other columns are treated as features/independent variables.

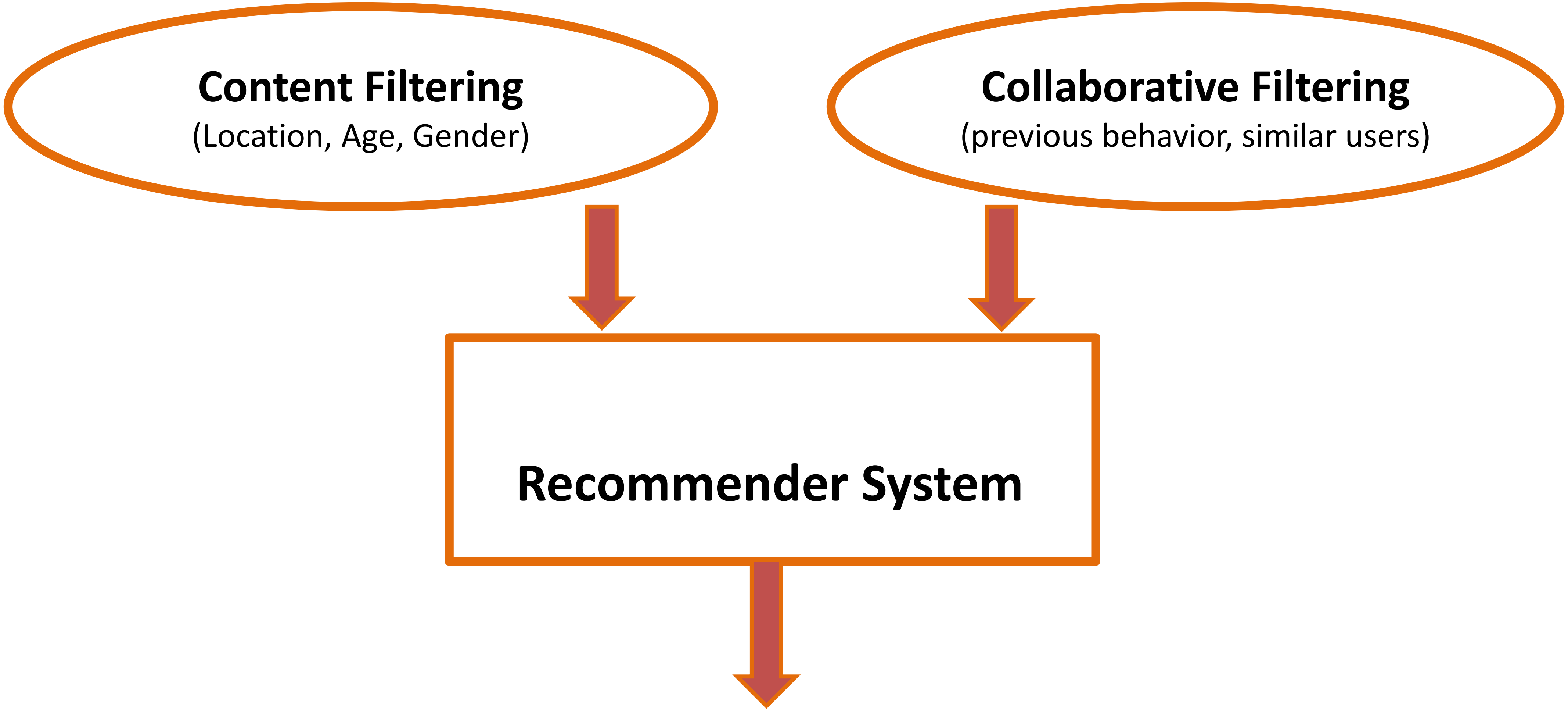
Neighborhood-Based Collaborative Filtering

- It also referred as memory-based algorithms.
- Neighborhood-based filtering algorithms can be formulated in one of two ways:

1. Predicting the rating value of a user-item combination: In this case, the missing rating r_{uj} of the user u for item j is predicted.
2. Determining the top-k items or top-k users: The problem of determing the top-k items is more common than that of finding the top-k users.

- Item-based methods provide more relevant recommendations because of the fact that a user’s own ratings are used to perform the recommendation.
- In item-based methods, similar items are identified to a target item, and the user’s own ratings on those items are used to extrapolate the ratings of the target.
- Although item-based recommendations are often more likely to be accurate, the relative accuracy between item-based and user-based methods also depends on the data set at hand.

Types of Recommendation system



Collaborative Filtering

- Consider user x
- Find set N of other users whose ratings are “similar” to x’s ratings
- Estimate x’s ratings based on ratings of users in N.

Similar Users(1):

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- Consider users X and Y with rating vectors r_x and r_y
- We need a similarity metric sim(x, y)
- Capture intuition that sim(A,B) > sim (A,C)

Option 1:Jaccard Similarity:

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- $\text{Sim}(A, B) = |r_A \cap R_B| / |r_A \cup r_B|$
- Sim(A, B)<sim(A, C)
- Problem: Ignores rating values!

Option 2: Cosine similarity

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4	0	0	5	1	0	0
B	5	5	4	0	0	0	0
C	0	0	0	2	4	5	
D	0	3	0	0	0	0	3

- Sim(A, B) = cos(r_A,$r_{B)}$
- Sim(A, B) = 0.38, Sim(A, C)= 0.32
- sim(A,B)< sim(A,C), but not by much
- Problem: treats missing ratings as negative

Option 3: Centered cosine

- Normalize ratings by subtracting row mean

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4-10/3=2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

Option 3: Centered cosine similarity(2)

	HP1	HP2	HP3	KGF	BB1	BB2	BB3
A	4-10/3=2/3			5/3	-7/3		
B	1/3	1/3	-2/3				
C				-5/3	1/3	4/3	
D		0					0

- Sim(A, B) = cos($r_{A,}$ r_B) =0.09; Sim(A, C) = -0.56
- Sim(A, B) > sim(A, C)

- Captures intuition better
- Missing ratings treated as "average"
- Handles "tough raters" and " easy raters"
- Also known as Pearson Correlation

Rating Predictions

- Let r_x be the vector of users x's ratings
- Let N be the set of k users most similar to x who have also rated item I
- Prediction for user x and item I

- Option 1: $r_{xi} \quad = 1/k \sum_{y \epsilon N} r_{yi}$
- Option 1: $r_{xi} \quad = \sum_{y \epsilon N} S_{xy}\, r_{yi} \,/ \sum_{y \epsilon N} S_{xy}$

Item-Item Collaborative Filtering:

- So far: User-user Collaborative filtering

- Another view: Item-Item
- For item I, find other similar items
- Estimate rating for item I based on ratings for similar items
- Can use same similarity metrics and prediction functions as in user-user model.

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

- S_{ij} … Similarity of items I and j
- R_{xj}…. Rating of user x on item j
- N(I;x)… set items rated by x similar to i

Item-Item CF(|N|=2)

Users

Movies

	1	2	3	4	5	6	7	8	9	10	11	
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- Estimate rating of movie 1 by user 5

Unknown Rating

Rating between 1to 5

Item-Item CF(|N|=2)

Users

Movies	1	2	3	4	5	6	7	8	9	10	11		Sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Here we use Pearson correlation as similarity

1) Subtract mean rating m from each movie I

M = (1+3+5+5+4)/5 = 3.6

Row 1:[-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0, 0.4, 0]

2) Compute cosine similarities between rows

Item-Item CF(|N|=2)

Users

Movies

	1	2	3	4	5	6	7	8	9	10	11	
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Predict by taking weighted average

$$r_{15} = (0.41*2 + 0.59*3) / (0.41 + 0.59) = 2.6$$

Item-Item vs. User-User

1. In theory, user-user and item-item are dual approaches

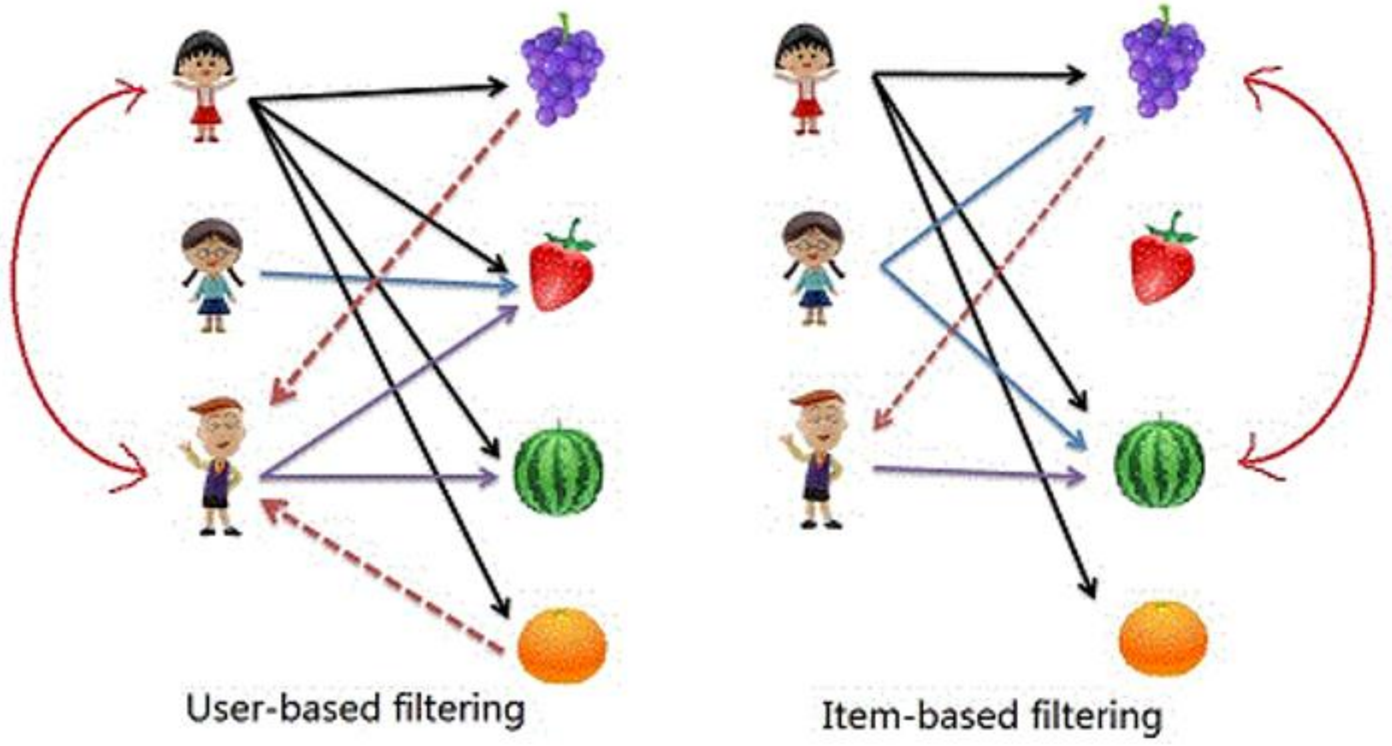
2. In practice, item-item outperforms user-user in many use cases.

3. Items are “simpler” than users

- items belong to a small set of “genres”, users have varied tastes.
- Item similarity is more meaningful than user similarity

Amazon’s Item-to-Item CF

Difference with User-to-User CF



Amazon’s Item-to-Item CF

Similarity of item i with item 17

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
,1	,3	,6	,1	,3	,4	,3	,3	,2	,6	,2	,5	,4	,5	,5	,3	1	,3	,5	,4	,2	,4	,4	,5

Items

Users

	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
a			1		4	5			4		3					2			4		2				
b			4							3							5	1		3					
c		5		4			4						3		5					4		5			
d								3				5				3			4		2			3	
e		3					5			4	5				5					1			5	4	
f			4				1		3	5		4	1		5	4	4		4				3		
g	2	4				4		2			5		1	4	5		4	2	4		5			4	
h			2		1		4		3	5		4	2		5	4	5						5		
i		1					3			5				5		4	4		5			4		3	
j			4			4				5			1		5		4		4				4		
k		5				4			2		5		1	5		4		2		4				2	
l					3			3				4	1		4		4	2	4					3	
m	5		3					5	3		5	4		5	5	3			4	4	5	4		4	
n			1		4	5				4	5		1	5		4		3		4		4	3		
o			4			4				5		4		5			4	2		5		5		3	
p				4			5								5	4		2	4	4	5	4		2	
q					3			3					1	5		4	4		4			4		3	
r		4			1	4		2					2		5		4				5	4		4	
s			2		4		4			5			1			4		2	4		4		5		
t		1		4			3					4		5	5		4			4				3	
u			2		1		4		3				1		5	4		2	4		5	4			
v					4	5				4	3		5			2					2			5	
w				2			2		3			5			4	5		4	2		3	4			
x	4			5				3		3				4	5					1					
y			1			3				2	3						3	3		5		4			

Amazon’s Item-to-Item CF

- How It Works
- Matches each of the user’s purchased and rated items to similar items
- Combines those similar items into a recommendation list

An iterative algorithm:

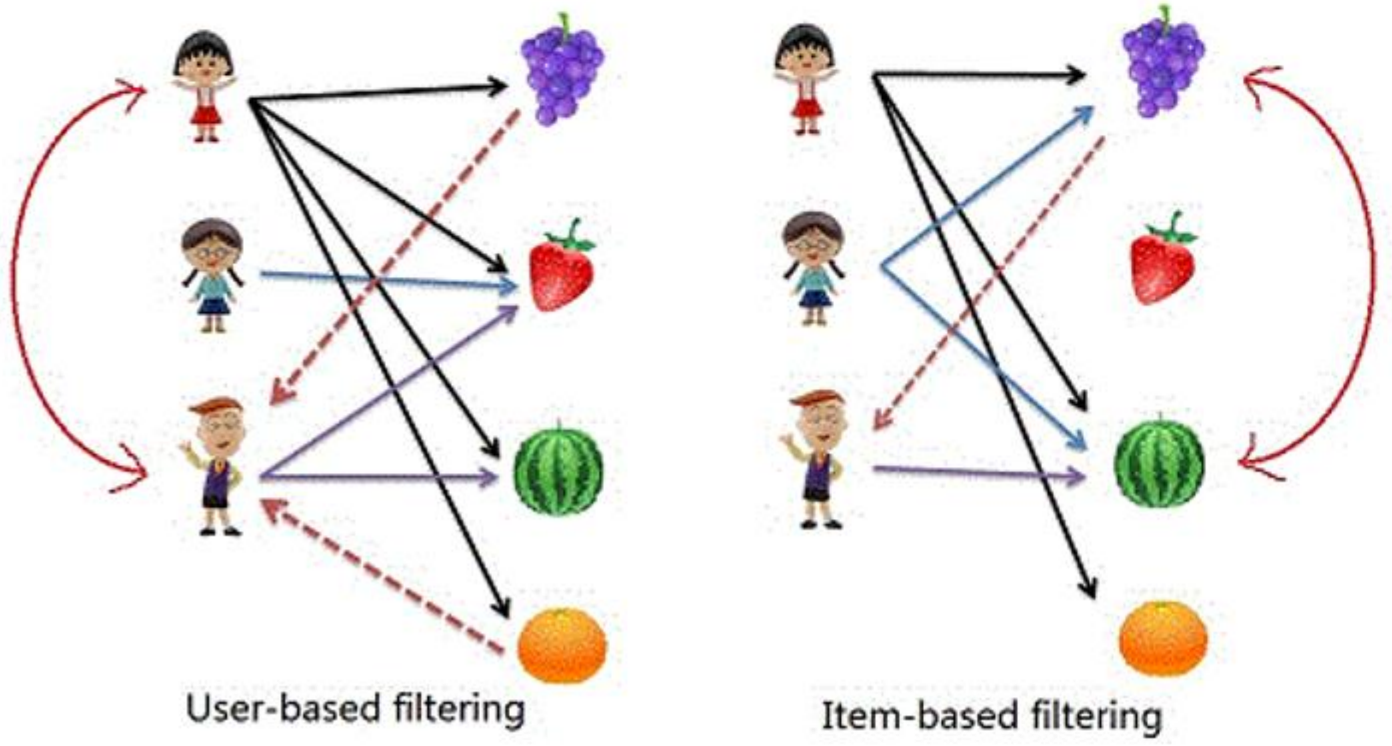
- Builds a similar-items table by finding items that customers tend to purchase together
- Provides a better approach by calculating the similarity between a single product and all related products:

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by customer C
      Record that a customer purchased I1 and I2
  For each item I2
    Compute the similarity between I1 and I2
```

- The similarity between two items uses the cosine measure
- Each vector corresponds to an item rather than a customer and
- Vector’s M dimensions correspond to customers who have purchased that item

Amazon’s Item-to-Item CF

Difference with User-to-User CF



Item-Item vs. User-User Scalability and Quality: Comparison

User Based collaborative filtering:

- little or no offline computation
- impractical on large data sets, unless it uses dimensionality reduction, sampling, or partitioning
- dimensionality reduction, sampling, or partitioning reduces recommendation quality

Cluster models:

- can perform much of the computation offline,
- but recommendation quality is relatively poor

Item-to-Item collaborative filtering:

- scalability and performance are achieved by creating the expensive similar-items table offline
- online component "looking up similar items" scales independently of the catalog size or the number of customers
- fast for extremely large data sets
- recommendation quality is excellent since it recommends highly correlated similar items
- unlike traditional collaborative filtering,
  - the algorithm performs well with limited user data,
  - producing high-quality recommendations based on as few as two or three items

Results:

- The MovieLens dataset contains 1 million ratings from 6,040 users on 3,900 movies.
- The best overall results are reached by the item-by-item based approach. It needs 170 seconds to construct the model and 3 seconds to predict 100,021 ratings.

	User Based	Model Based	Item Based
Model Construction Time (sec.)	730	254	170
Prediction Time (sec.)	31	1	3
MAE	0.6688	0.6736	0.6382

Case Study

- Suppose you set up a system, where a guided visual interface is used in order to determine the product of interest to a customer. What category of recommender system does this case fall into?

- Discuss a scenario in which location plays an important role in the recommendation process.

- The chapter mentions the fact that collaborative filtering can be viewed as a generalization of the classification problem. Discuss a simple method to generalize classification algorithms to collaborative filtering. Explain why it is difficult to use such methods in the context of sparse ratings matrices.

References

Text Book:

“Business Analytics, The Science of Data-Driven Making”, U. Dinesh Kumar, Wiley 2017

“Recommender Systems, The text book, Charu C. Aggarwal, Springer 2016 Section 1.and Section 2.

Image Courtesy

http://www.mmds.org/mmds/v2.1/ch09-recsys1.pptx

https://www.researchgate.net/publication/287952023_Collaborative_Filtering_Recommender_Systems

http://cs229.stanford.edu/proj2014/Rahul%20Makhijani,%20Saleh%20Samaneh,%20Megh%20Mehta,%20Collaborative%20Filtering%20Recommender%20Systems.pdf

https://www.scribd.com/presentation/414445910/CS548S15-Showcase-Web-Mining

https://towardsdatascience.com/image-recommendation-engine-leverage-transfert-learning-ec9af32f5239

http://elico.rapid-i.com/recommender-extension.html

https://www.youtube.com/watch?v=h9gpufJFF-0

THANK YOU

Jyothi R.
Assistant Professor,
Department of Computer Science
jyothir@pes.edu