Collaborative Filtering Recommenders

David Yang
xdyang70@gmail.com

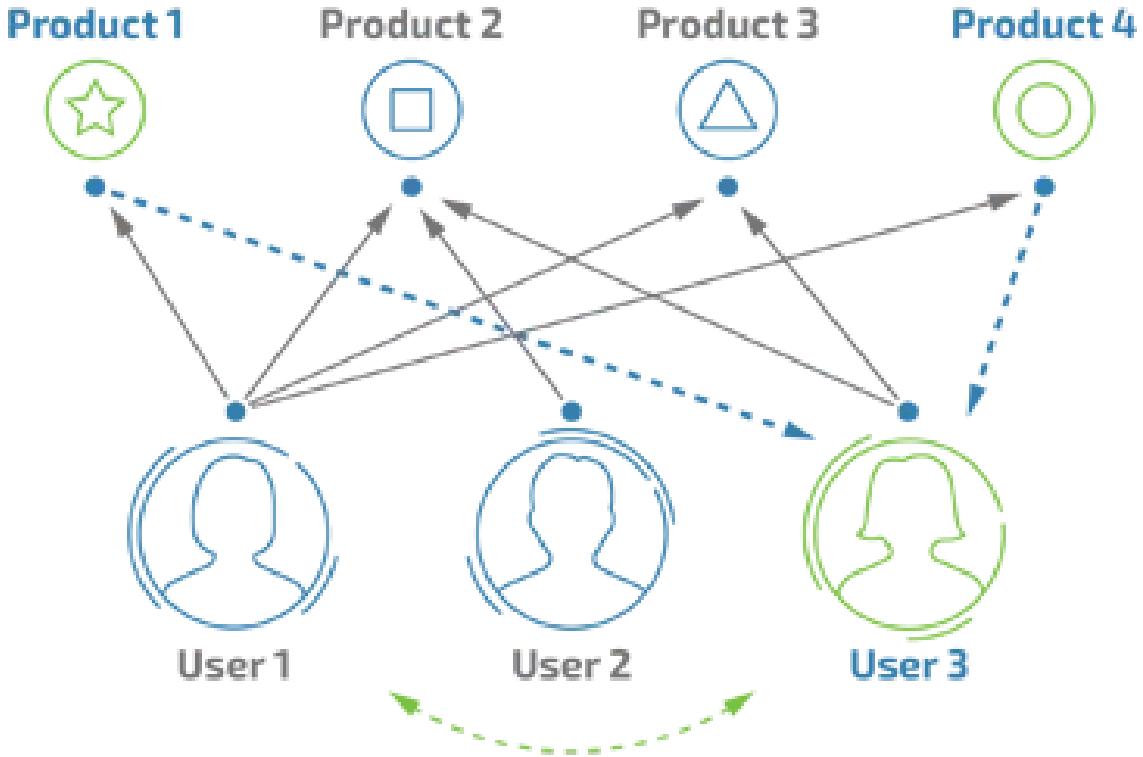
Key Idea of CF

- For Customer **X**
- Find a set **N** of other users whose historical ratings (behavior, tastes, …) are **similar** to X's ratings
- Recommend item **i** to **X**

  (1) Based on **i**'s **distance** to items viewed by X and users in **N**

  (2) By estimating **X**'s rating of **i** based on ratings of users in **N**

Approach 1. User-User Similarity



Approach 1. User-User Similarity

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
A	5			5	2	
B	5	5	4	5		
C				4	1	5
D		3		1	4	

$$sim(A,B) = | r_A \cap r_B | / | r_A \cup r_B |$$

$$sim(A,B) = \cos(r_A, r_B)$$

User-User Similarity

	A	B	C	D
A	**-**	**0.8**	**0.6**	**-0.5**
B		**-**	**0.1**	**-0.2**
C			**-**	**-0.8**
D				**-**

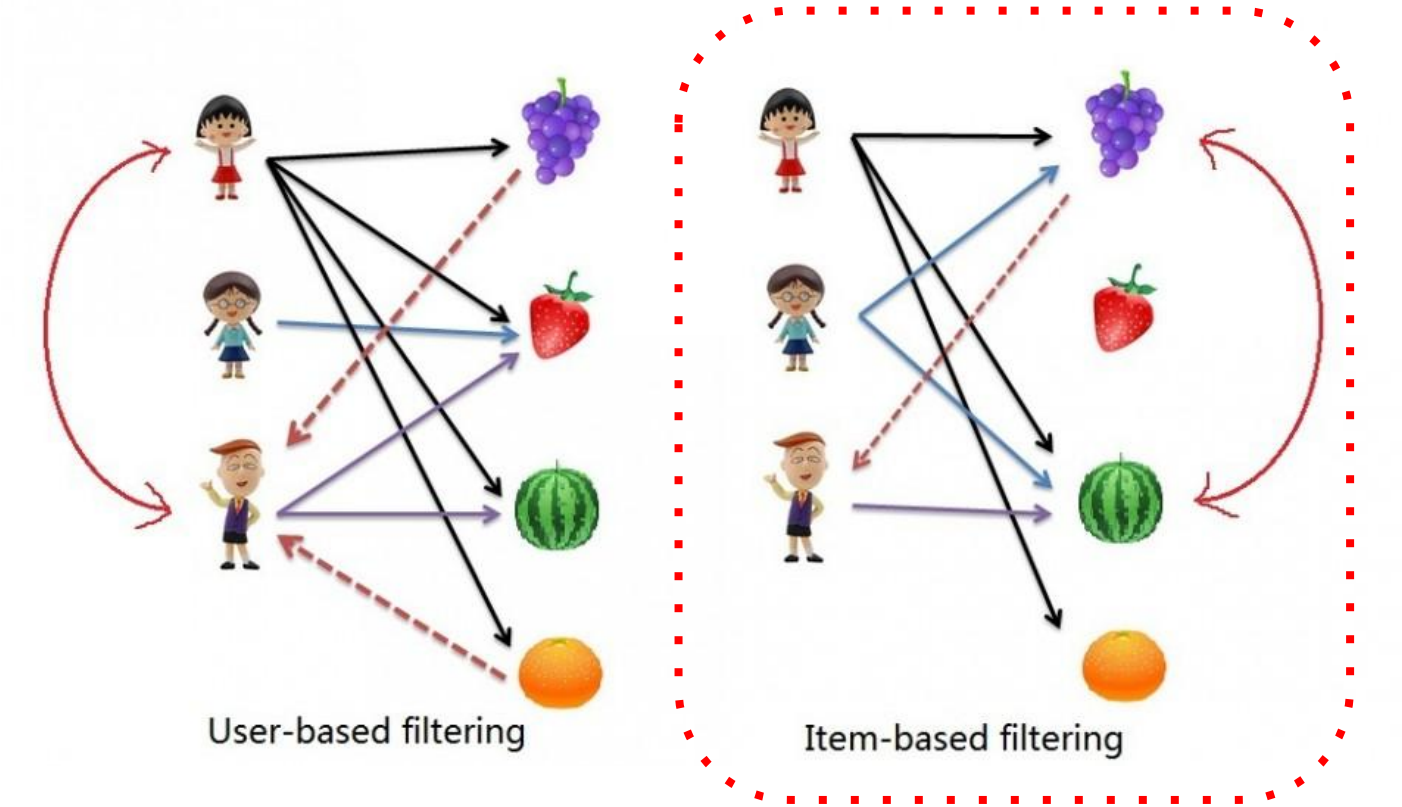
Rating Predictions

User **x**'s rating of the **i**-th item

$$r_{xi} = \sum_{y \in N} s_{xy} r_{yi} / \sum_{y \in N} s_{xy}$$

where $s_{xy} = \text{sim}(x,y)$

Approach 2. Item-Item Similarity



Approach 2. User-User Similarity

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
A	5			5	2	
B	5	5	4	5		
C				4	1	5
D		3		1	4	

Sim(*1, 4*) = cos(R_1, R_4)

Item-Item Similarity

	1	2	3	4	5	6
1	-	0.6	0.3	0.9	-0.1	na
2		-	0.3	0.9	0.1	na
3			-	0.6	na	na
4				-	-0.8	0.7
5					-	-0.5
6						-

Rating Predictions

User **x**'s rating of the **i**-th item

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

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***

Summary of CF Recommenders

Pros

- Usually more powerful than content-based recommenders
- Can handle cases where no features available
- Fully utilize information available in the user-item interaction matrix

Cons

- “Cold Start’: new user/item has no historical data to use.
- Require more computing effort: O(n*n) vs O(m*m).

Example 2

In this example, we will learn how to use Scala to implement KNN (K-Nearest Neighbours) method for CF with item-item similarity.

See “Example 2 Collaborative Filtering Recommender.scala”