Week 5: Collaborative Based RSs – Part I

CS3448: Recommender Systems /

CSX4207/ITX4207: Decision Support and Recommender Systems

Asst. Prof. Dr. Rachsuda Setthawong

Objectives

- To understand the concept of collaborative based filtering approach
- To be familiar with User-based Nearest Neighbor and Item-based Nearest Neighbor algorithms
- To introduce additional proximity measure
- To understand the problems of collaborative based filtering approach

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Main idea

- To exploit information about the past behavior/opinions of an existing user community.
- To predict which items the target user will most probably like.

Asst. Prof. Dr. Rachsuda Setthawong

How to Generate Recommendation Using Collaborative Based Filtering Approach

Target
User's Preferences

Recommendation Algorithm(s)

Community's Preferences (user-item rating matrix)

Prediction

Rank

Recommendation

Collaborative: "Tell me what's popular among my peers"

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Asst. Prof. Dr. Rachsuda Setthawong

Pure Collaborative Approaches

- Input:
  - A matrix of given user-item ratings

	Item 1	Item 2	Item 3	Item 4	...
User 1	3	1	2	3	...
User 2	4	3	4	3	...
User 3	3	3	1	5	...
...	...	...	...	...	...

- Outputs:
  - A (numerical) prediction of what degree the target user will like/dislike a certain (unseen) item
  - A list of *k* recommended items

Memory-based VS Model-based Approaches

Memory-based approach	Model-based approach

Memory-based approach

- Modeless (no model created)
- Directly applying rating matrix to find neighbors and to recommend items.
- Time consuming and not scalable
- Example, user-based NN

Model-based approach

- Offline process the raw data.
- At run time, require only the "learned" model to make prediction.
- Update/retrained the model periodically.
- Example, matrix factorization methods, association rule mining, etc.

Asst. Prof. Dr. Rachsuda Setthawong

User-based Nearest Neighbor (NN) Recommendation – **Main Idea**

- Given
  - A rating database containing:
    - The set of users: $\boldsymbol{U} = \{u_1, \ldots, u_n\}$
    - The set of products (items): $\boldsymbol{P} = \{p_1, \ldots, p_m\}$
    - $\boldsymbol{R}$ as $n \times m$ matrix of ratings $r_{i,j}$ with $i \in 1 \ldots n$, $j \in 1 \ldots m$
      - The rating values: 1 – 5 (1: *strongly dislike* and 5: *strongly like*)
  - The ID of the current (active) user
- Step 1: Find k-NN users that had similar preferences to those of the active user in the past.
- Step 2: Predict ratings of an unseen product p based on the ratings for p made by k-NNs.

Assumptions in Collaborative Approach

1. If users had similar tastes in the past, they will have similar tastes in the future.
2. User preferences remain stable and consistent over time.

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Asst. Prof. Dr. Rachsuda Setthawong

Example 1

- Goal: to determine whether Alice will *like* or *dislike* “Item 5”

Target User →

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

User-based Nearest Neighbor (NN) Recommendation

- Step 1: Select the value of k for nearest neighbored users to the target user.
- Step 2: Calculate Similarity between the target user and other users (*using Pearson's Correlation Coefficient (PCC)*)
- Step 3: Predicting Product Rating

Asst. Prof. Dr. Rachsuda Setthawong

Step 1: Select the value of *k* for nearest neighbored users to the target user.

- E.g.,
  - **k = 1 ← select the most similar user to the target user**
  - k = 10 ← select the top-10 most similar users to the target user
  - k = 20 ← select the top-20 most similar users to the target user

Step 2: Calculate Similarity between the target user and other users (*using Pearson's Correlation Coefficient (PCC)*)

- To determine similarity between user preferences.

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \bar{r_a})(r_{b,p} - \bar{r_b})}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r_a})^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r_b})^2}}$$

- where,
  - $r_{a,p}$, $r_{b,p}$ is a rating of user *a* (*or b*) on product *p*.
  - ** $\bar{r}_a$, $\bar{r}_b$ is the average rating of *the <u>common rated items</u>* of user *a* (*or b*).
  - *P* represents the set of *<u>common rated items</u>* by user *a* and *b*.
- Interpretation (in the range of -1 to 1):
  - -1 = strongly negative correlation
  - 1 = strongly positive correlation

Note: The common rated items =
The items that are both rated by users a and b.

**: Ref: http://www.recommenderbook.net/media/corrigenda.pdf

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Calculating PCC

- To determine similarity between user preferences.

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	?
User 1	3	1	2	3	3

$$sim(Alice, User1) = \frac{\sum_{p \in P}(r_{Alice,p} - \overline{r_{Alice}})(r_{User1,p} - \overline{r_{User1}})}{\sqrt{\sum_{p \in P}(r_{Alice,p} - \overline{r_{Alice}})^2}\sqrt{\sum_{p \in P}(r_{User1,p} - \overline{r_{User1}})^2}}$$

$$\overline{r_{Alice}} = \frac{5+3+4+4}{4} = 4 \qquad \overline{r_{User1}} = \frac{3+1+2+3}{4} = 2.25$$

$$\sqrt{\sum_{p \in P}(r_{Alice,p} - \overline{r_{Alice}})^2} = 1.4142 \qquad \sqrt{\sum_{p \in P}(r_{User1,p} - \overline{r_{User1}})^2} = 1.6583$$

$$sim(Alice, User1) = \frac{(5-4)(3-2.25)+(3-4)(1-2.25)+ \ldots +(4-4)(3-2.25)}{(1.4142)(1.6583)} = 0.85$$

Additional Resource: http://recommender-systems.org/collaborative-filtering/

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Calculating PCC:

(The similarity between Alice and other users using PCC)

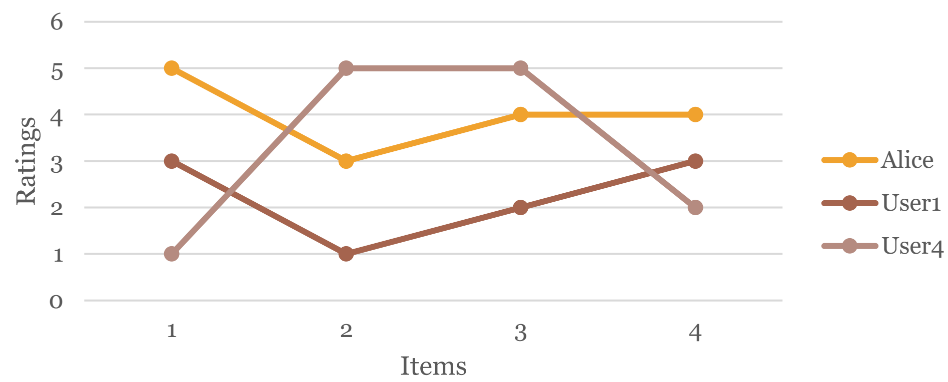
	Pearson Correlation Coefficient
sim(Alice,User1)	0.85
sim(Alice,User2)	0.7
sim(Alice,User3)	0
sim(Alice,User4)	-0.79

Asst. Prof. Dr. Rachsuda Setthawong

Comparing Alice with Two Other Users Using PCC

The original ratings of common rated items of Alice and other two users:

	Item1	Item2	Item3	Item4
Alice	5	3	4	4
User1	3	1	2	3
User4	1	5	5	2



PCC of Alice and other two users:

	PCC
sim(Alice,User1)	**0.85**
sim(Alice,User4)	**-0.79**

Strongly positive correlation

Strongly negative correlation

Asst. Prof. Dr. Rachsuda Setthawong

Step 3: Predicting Product Rating

- The prediction is made by adjusting the average rating of the target user with the similarity and the opinion of *k*-NN.
- The formula *pred*(*a*,*p*) for computing a prediction for the rating of the target user *a* for item *p* considers
  - **$\overline{r_a}$: the average rating of all rated items of *a*
  - $r_{b,p}$: *k*-NNs' opinion (deviation from their (individual) means)
  - *sim*(*a*,*b*): relative proximity of the *k* nearest neighbors

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b\in k} sim(a,b)(r_{b,p} - \overline{r_b})}{\sum_{b\in k} sim(a,b)}$$

**: In the above eq., all known ratings of the target user are meant by $\overline{r_a}$ as we are interested in the user's global rating bias.
Ref: http://www.recommenderbook.net/media/corrigenda.pdf

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Calculating the Predicted rating of Item 5 for Alice based on Ratings of 2-NNs (User1 and User2)

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	?
User 1	**3**	**1**	**2**	**3**	**3**
User 2	**4**	**3**	**4**	**3**	**5**

	Pearson Correlation Coefficient
Sim(Alice,User1)	0.85
Sim(Alice,User2)	0.7

the individual user's **average rating** of ALL rated items

$$\overline{r_{Alice}} = \frac{5+3+4+4}{4} = 4$$

$$\overline{r_{User1}} = \frac{3+1+2+3+3}{5} = 2.4$$

$$\overline{r_{User2}} = \frac{4+3+4+3+5}{5} = 3.8$$

$$pred(Alice, Item5) = \overline{r_{Alice}} + \frac{\sum_{b\in\{User1,User2\}} sim(Alice,b)(r_{b,Item5}-\overline{r_b})}{\sum_{b\in\{User1,User2\}} sim(Alice,b)}$$

$$pred(Alice, Item5) = 4 + \frac{0.85(3-2.4)+0.7(5-3.8)}{0.85+0.7} = 4.88$$

Other Measures to Determine Proximity between Users – 1/2

- Spearman's rank correlation coefficient
  - Case 1: no tied ranks
    - $\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$
    - d = the difference between the two ranks of each item
    - n = number of cases
  - Case 2: having tied ranks

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

    - i = paired score

User A score	User B score	Rank(User A's score)	Rank(User B's score)	d	d^2
50	30	1	2	1	1
30	20	3	3	0	0
40	10	2	4	2	4
20	40	4	1	3	9

$$\sum d_i^2 = 1 + 0 + 4 + 9 = 14$$

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

$$\rho = 1 - \frac{6 \times 14}{4(4^2-1)} = -0.4$$

Ref: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php

Other Measures to Determine Proximity between Users – 2/2

- Mean squared difference (*msd*) (Shardanand and Maes 1995)
  - Measure the degree of dissimilarity between two user profiles U_a and U_b by the mean squared difference between the two profiles $(U_a - U_b)^2$

$$msd(U_a, Ub) = \frac{\sum_{i=1}^{n}(r_{U_a,i} - r_{U_b,i})^2}{n}$$

- Adjusted cosine similarity
  - To be discussed in Item-based collaborative filtering

Ref: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.586&rep=rep1&type=pdf

Asst. Prof. Dr. Rachsuda Setthawong

Factors Not Addressed by Pearson Correlation Coefficient

- Different weighting of controversial items and a generally liked item
  - An agreement by two users on a more controversial item has more "value" than an agreement on a generally liked item
- Too few number of co-rated items
- Too few number of agreed items

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Asst. Prof. Dr. Rachsuda Setthawong

Guidelines on Neighborhood Selection

- Select neighbors who have
  - **Positive correlation** with the target user
  - **Rating** on the **predicted items**
- Reduce the size of the neighborhood by
  - Taking only the ***k*** **NNs**.
  - Defining **a minimum threshold** of user similarity.

Further reading: https://pdfs.semanticscholar.org/aeb6/3457163d83f46b9b365381adb4308a161723.pdf

Asst. Prof. Dr. Rachsuda Setthawong

Effects of Neighborhood Selection Using *k* NNs

- Too high *k*: **noise** is added to the prediction.
- Too small *k*: the **quality** of the prediction may be **negatively affected** ($k < 10$).

Asst. Prof. Dr. Rachsuda Setthawong

Effects of Neighborhood Selection Using a Min Threshold of User Similarity

- **Too high** threshold: *(get too few neighbors)*
  - May have a **coverage** problem *(**too few unseen items** to suggest)*

- **Too low** threshold: *(get too many neighbors)*
  - The **neighbor sizes** are **NOT significantly reduced**

    → *suggested **items** are **bias to global opinion**.*

Asst. Prof. Dr. Rachsuda Setthawong

Problems with CF Approaches

- High Computation time for real-time predictions of millions of users and millions of catalog items in large e-commerce sites.

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

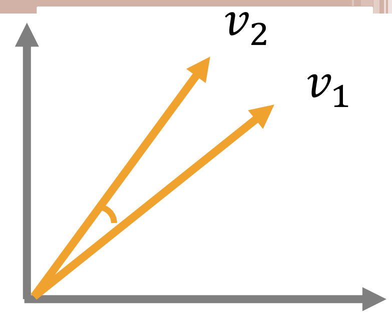
Asst. Prof. Dr. Rachsuda Setthawong

Alternative Approach -- Item-based Nearest Neighbor Recommendation

- Offline preprocessing
- Main idea:
  - Compute predictions using the similarity **between items**

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

Asst. Prof. Dr. Rachsuda Setthawong



The Cosine Similarity Measure

- Measures the similarity between two *n*-dimensional vectors based on the angle between them.
- $cosine\_sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|}$
- *Example 3,*

Interpretation:
- Range of possible value is -1 and 1.
- The closer to 1 → the more similar

$$cosine\_sim(Item1, Item5) = \frac{3*3+5*4+4*3+1*1}{\sqrt{3^2+5^2+4^2+1^2}*\sqrt{3^2+4^2+3^2+1^2}} = 0.9941$$

Item 1	Item 5
3	3
4	5
3	4
1	1

- *cosine_sim(**Item1**, Item5) = **0.9941***
- *cosine_sim(Item2, Item5) = 0.7389*
- *cosine_sim(Item3, Item5) = 0.7226*
- *cosine_sim(**Item4**, Item5) = **0.9396***

Asst. Prof. Dr. Rachsuda Setthawong

Example 2

- Goal: Predict Alice's rating on Item 5 from 2-NN items

  (e.g., 2-NN = Items 1 and 4)

	Item 1	Item 4	Item 5
Alice	5	4	?
User 1	3	3	3
User 2	4	3	5
User 3	3	5	4
User 4	1	2	1

= a weighted average of ratings 4 and 5

Asst. Prof. Dr. Rachsuda Setthawong

The Problem with Basic Cosine Measure

- Problem: it does not take into account the differences in the average rating behavior of the users.
- Solution: using the *adjusted cosine measure to calculate similarity between items a, b*

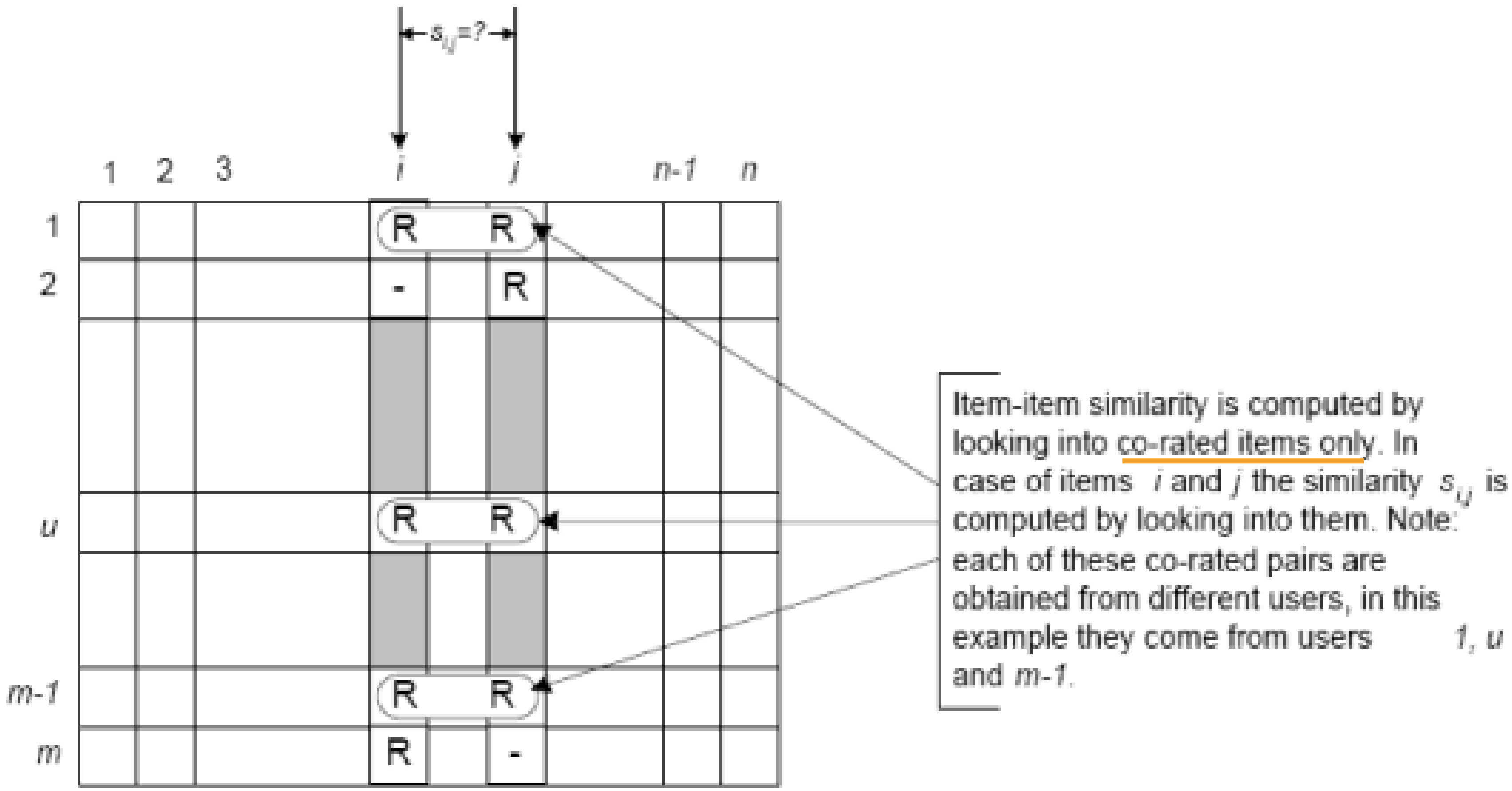
$$sim(a,b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2} \sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

where,

U is the subset of users that rated BOTH items *a* and *b*.

Asst. Prof. Dr. Rachsuda Setthawong

Similarities between Items



Ref. : http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html

Asst. Prof. Dr. Rachsuda Setthawong

Preprocessing Step: Adjusting Ratings
(Later Used by Adjusted Cosine Measure)

Original ratings database

$(r_{u,a})$

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

the individual user's **average rating** of ALL rated items

$(\bar{r_u})$

	Avg. Rating
Alice	4
User 1	2.4
User 2	3.8
User 3	3.2
User 4	2.8

Subtract the original ratings by the avg. rating

Mean-adjusted ratings database

$(r_{u,a} - \bar{r_u})$

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice	1.0	-1.0	0	0	?
User 1	0.6	-1.4	-0.4	-0.6	0.6
User 2	0.2	-0.8	0.2	-0.8	1.2
User 3	-0.2	-0.2	-2.2	2.8	0.8
User 4	-1.8	2.2	2.2	-0.8	-1.8

Revised Example 3

Mean-adjusted ratings

	$(r_{u,item1} - \bar{r_u})$	$(r_{u,item5} - \bar{r_u})$:
User1	0.6	0.6
User2	0.2	1.2
User3	-0.2	0.8
User4	-1.8	-1.8

- The adjusted cosine similarity of items *a* and *b*:

$$sim(a,b) = \frac{\sum_{u \in U}(r_{u,a} - \bar{r_u})(r_{u,b} - \bar{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \bar{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,b} - \bar{r_u})^2}}$$

U is the subset of users that rated BOTH items *a* and *b*.

$$sim(Item1, Item5) = \frac{0.6*0.6 + 0.2*1.2 + (-0.2)*0.8 + (-1.8)*(-1.8)}{\sqrt{0.6^2+0.2^2+(-0.2)^2+(-1.8)^2}*\sqrt{0.6^2+1.5^2+0.8^2+(-1.8)^2}} = 0.8049$$

- *sim(Item1, Item5) =* ***0.8049***
- *sim(Item2, Item5) = -0.9082*
- *sim(Item3, Item5) = -0.7636*
- *sim(Item4, Item5) =* ***0.4331***

Further reading: http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html

Asst. Prof. Dr. Rachsuda Setthawong

Predicting Product Rating

- **Use** the **previous rated items** of the **target user** by **adjusting** it **with** the similarity between its co-rated items to the *unseen* *(need to be predicted)* item.
- A formula for computing a prediction for the rating of the target user u for item p

$$pred(u,p) = \frac{\sum_{i \in ratedItems(u)} \overbrace{sim(i,p)}^{\text{Pre-calculated}} \times r_{u,i}}{\sum_{i \in ratedItems(u)} sim(i,p)}$$

where i are the items rated by the target user u.

Asst. Prof. Dr. Rachsuda Setthawong

An Example of calculating the predicted rating of Item for Alice based on 2-NN *(Item1 and Item4)*

	Item 1	Item 4	Item 5
Alice	5	4	?

cosine_sim(Item1, Item5) =	***0.8049***
cosine_sim(Item4, Item5) =	***0.4331***

$$pred(Alice, Item5) = \frac{\sum_{i \in ratedItems(Alice)} \overbrace{sim(i, Item5)}^{\text{Pre-calculated}} \times r_{Alice,i}}{\sum_{i \in ratedItems(Alice)} sim(i, Item5)}$$

$$= \frac{(0.8049 \times 5) + (0.4331 \times 4)}{0.8049 + 0.4331}$$

$$= 4.65$$

Asst. Prof. Dr. Rachsuda Setthawong

Preprocessing Data for Item-based Filtering

	Item1	Item2	Item3	Item4	Item5
Item1	1	...	...	...	***0.8049***
Item2	...	1	...	...	*-0.9082*
Item3	...	...	1	...	*-0.7636*
Item4	...	...	...	1	***0.4331***
Item5	...	...	...	...	1

- *Offline* precompute of the *item similarity matrix*

$$sim(a,b) = \frac{\sum_{u \in U}(r_{u,a} - \bar{r_u})(r_{u,b} - \bar{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \bar{r_u})^2}\ \sqrt{\sum_{u \in U}(r_{u,b} - \bar{r_u})^2}}$$

- *Online* predict ratings of a product *p* and user *u*,
  1. Determine the set of items *X* that are most similar to *p*.
  2. Build the ***weighted sum*** of *u*'s ratings for these items *X* in the neighborhood.

	Item 1	Item 4	Item 5
Alice	1.0	0	?

$$pred(u,p) = \frac{\sum_{i \in ratedItems(u)} sim(i,p) \times r_{u,i}}{\sum_{i \in ratedItems(u)} sim(i,p)}$$

- Perform subsampling
  - Randomly choose a subset of the data.
  - Ignore customer records with few rating.

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Asst. Prof. Dr. Rachsuda Setthawong

Pros and Cons of CF Based Approach

Pros	Cons
• Not require details of items and users in order to generate recommendations. • Capable of recommending serendipitous items based on similar users' behaviors.	• Requires explicit ratings/opinions from users. • Has a cold start problem (new user/ item). • Has poor accuracy if too few ratings exist. • Is slow (memory-based CF)

Outlines

- Collaborative Recommendation
- User-based Nearest Neighbor (NN) Recommendation
- Measures to Determine Proximity between Users
- Neighborhood Selection
- Item-based Nearest Neighbor (NN) Recommendation
- Pros and cons of CF based approach
- Problems with CF based approach

Problems with Rating Scales

- Explicit rating: user might not be willing to provide ratings.
- Implicit rating: user behavior is not easily interpreted as ratings.

Data Sparsity and the Cold-Start Problems

- Data sparsity problem: data the rating matrices tend to be very sparse.
  - Solution: Default voting
- Cold-start problem: hard to generate recommendations for new items.

Suggested Reading Articles

- "Deconstructing Recommender Systems. How Amazon and Netflix predict your preferences and prod you to purchase" By Joseph A. Konstan and John Riedl, IEEE Spectrum, Sep. 2012. http://spectrum.ieee.org/computing/software/deconstructing-recommender-systems
- The Big Promise of Recommender Systems https://www.researchgate.net/publication/220604928_The_Big_Promise_of_Recommender_Systems
- Recommender Systems An Overview https://www.researchgate.net/publication/220604600_Recommender_Systems_An_Overview