

Item-Based or Model-Based Filtering

Hina Arora

References:

Item-Based Collaborative Filtering Recommendation Algorithms – Sarwar, Karypis, et al
Slope One Predictors for Online Rating-Based Collaborative Filtering – Lemire and Maclachlan
Guide to Data Mining Chapter 3 - Zacharski

Issues with User-Based Filtering

In user-based filtering, we first create the user-user similarity matrix based on items that have been rated by the user pairs. Then, we predict an item's rating for a user by taking a weighted average of the item ratings of the k most similar users.

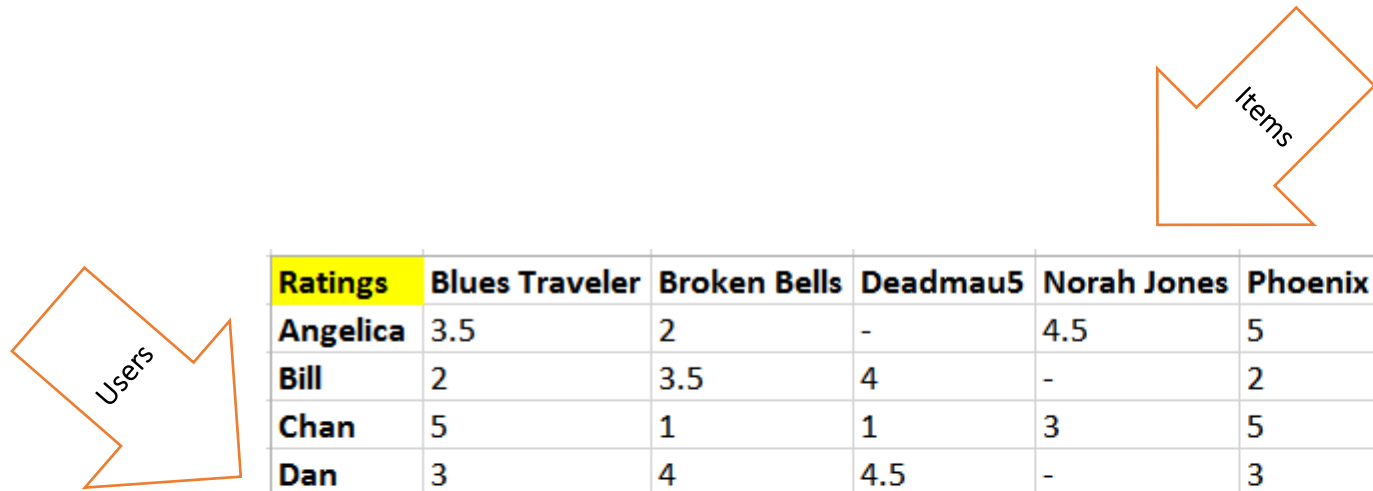
- Sparsity Issues – A lot of users can end up with no recommendations since most users haven't rated anything in common
- Performance Issues – need to update nearest neighbors every time a new user added to the system, and need to update similarities every time an existing user updates/add ratings.

Item-Based or Model-Based Filtering

- What if we turned the problem on its head and considered Item-Item similarities instead of User-User similarities, and used those to make recommendations?
- In item-based filtering, we first create the item-item similarity matrix based on users who have rated the item pairs. Then, we predict an item's rating for a user by taking a weighted average of the k most similar items that the user has rated.
- Also called Model-Based Filtering because we don't need to store all the ratings. We just build a model representing how close every item is to every other item.

- Item-Based Filtering also addresses the two issues we had with User-Based Filtering:
 - Typically, number of users in systems is much larger than the number of items. So sparsity is lower because instead of user ratings being considered across a larger number of users, these are now considered across a fewer number of items.
 - Also, item relationships change less rapidly than user relationships. So we can pre-compute item-item similarity without having to update it too often.
- The disadvantage with item-based filtering is that it generally leads to more conservative recommendations (since recommended items are essentially similar to what user has already purchased); whereas user-based filtering can lead to novel and surprising recommendations

Running Example for this Section



Ratings	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.5	2	-	4.5	5	1.5	2.5	2
Bill	2	3.5	4	-	2	3.5	-	3
Chan	5	1	1	3	5	1	-	-
Dan	3	4	4.5	-	3	4.5	4	2
Hailey	-	4	1	4	-	-	4	1
Jordyn	-	4.5	4	5	5	4.5	4	4
Sam	5	2	-	3	5	4	5	-
Veronica	3	-	-	5	4	2.5	3	-

Let's say we were trying to get the predicted rating of Broken Bells for Veronica

User-Based Filtering

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

- Find k closest users to Veronica based on user-user similarity;
- Then take a (user-user similarity-based) weighted avg of the ratings of Broken Bells by the k closest users (to Veronica)

Item-Based Filtering

Ratings	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.5	2	-	4.5	5	1.5	2.5	2
Bill	2	3.5	4	-	2	3.5	-	3
Chan	5	1	1	3	5	1	-	-
Dan	3	4	4.5	-	3	4.5	4	2
Hailey	-	4	1	4	-	-	4	1
Jordyn	-	4.5	4	5	5	4.5	4	4
Sam	5	2	-	3	5	4	5	-
Veronica	3	-	-	5	4	2.5	3	-

- Find k closest items to Broken Bells based on item-item similarity;
- Take a (item-item similarity-based) weighted sum of the ratings of the k closest items (to Broken Bells) rated by Veronica

What measures of similarity can we use between two items?

- 1) Adjusted Cosine Similarity
- 2) Weighted Slope One

(1) Adjusted Cosine Similarity

Let's say we want to predict how well user v will like item k .

(i) First compute an Item-Item Similarity Matrix based on the mean-adjusted cosine similarity measure of items ratings:

$$S_{i,j} = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_u})^2}} \quad \text{for each } i, j$$

Where

$S_{i,j}$ is the similarity between items i and j

$U_{i,j}$ is set of all users who rated both items i and j

$R_{u,i}$ is the rating user u gives to item i

$\overline{R_u}$ is the average rating of all items rated by user u

(note: subtracting corresponding user averages from each co-rated pair accounts for differences in rating scales between different users)

ii) Then calculate normalized ratings of user v for all items i (to lie between -1 and 1):

$$NR_{v,i} = \frac{2(R_{v,i} - Min_R) - (Max_R - Min_R)}{(Max_R - Min_R)} \text{ for each } i$$

(note: we do this since the similarity lies between -1 and 1, and so normalizing the ratings from -1 to 1 make the weighted-sum predictions in step (iii) work better)

(iii) Then predict rating of how well user v will like item k

[Note: This predicted rating will also be in normalized scale of -1 to 1]

$$NR'_{v,k} = \frac{\sum_{j \in \{S_{k,j} \text{ exists}\} - \{k\}} (S_{k,j} * NR_{v,j})}{\sum_{j \in \{S_{k,j} \text{ exists}\} - \{k\}} (|S_{k,j}|)}$$

(iv) Then de-normalize the prediction score

[Note: This predicted rating will now be in original de-normalized scale of Min_R to Max_R]

$$R'_{v,k} = \frac{1}{2} \left((NR'_{v,k} + 1) * (Max_R - Min_R) \right) + Min_R$$

(i) First Compute Item-Item Similarity Matrix

User-Item data with unknown ratings (?) that we'd potentially like to predict

Ratings	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend	$\overline{R_u}$ userAvgRatings
Angelica	3.50	2.00	?	4.50	5.00	1.50	2.50	2.00	3.00
Bill	2.00	3.50	4.00	?	2.00	3.50	?	3.00	3.00
Chan	5.00	1.00	1.00	3.00	5.00	1.00	?	?	2.67
Dan	3.00	4.00	4.50	?	3.00	4.50	4.00	2.00	3.57
Hailey	?	4.00	1.00	4.00	?	?	4.00	1.00	2.80
Jordyn	?	4.50	4.00	5.00	5.00	4.50	4.00	4.00	4.43
Sam	5.00	2.00	?	3.00	5.00	4.00	5.00	?	4.00
Veronica	3.00	?	?	5.00	4.00	2.50	3.00	?	3.50

$$\frac{5 + 1 + 1 + 3 + 5 + 1}{6}$$

Item-Item Adjusted-Cosine Similarity ($S_{i,j}$) Matrix: lies between -1 and 1

similarityMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	-0.88	-0.97	-0.04	0.85	-0.67	0.43	0.17
Broken Bells	-0.88	-	0.24	0.21	-0.86	0.69	0.02	-0.42
Deadmau5	-0.97	0.24	-	-0.87	-0.97	0.95	-0.57	0.35
Norah Jones	-0.04	0.21	-0.87	-	0.48	-0.71	-0.27	-0.93
Phoenix	0.85	-0.86	-0.97	0.48	-	-0.89	-0.22	-0.3
Slightly Stoopid	-0.67	0.69	0.95	-0.71	-0.89	-	0.58	0
The Strokes	0.43	0.02	-0.57	-0.27	-0.22	0.58	-	-0.57
Vampire Weekend	0.17	-0.42	0.35	-0.93	-0.3	0	-0.57	-

$$S_{i,j} = \frac{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u \in U_{i,j}} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R_u})^2}}$$

$$\frac{(1 - 2.67)(3 - 2.67) + (1 - 2.8)(4 - 2.8) + (4 - 4.43)(5 - 4.43)}{\sqrt{(1 - 2.67)^2 + (1 - 2.8)^2 + (4 - 4.43)^2} \sqrt{(3 - 2.67)^2 + (4 - 2.8)^2 + (5 - 4.43)^2}}$$

(ii) Then calculate normalized ratings of user v for all items i (to lie between -1 and 1)

Original User-Item data ($R_{v,i}$): lies between $Min_R=1$ and $Max_R=5$

Ratings	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.50	2.00	?	4.50	5.00	1.50	2.50	2.00
Bill	2.00	3.50	4.00	?	2.00	3.50	?	3.00
Chan	5.00	1.00	1.00	3.00	5.00	1.00	?	?
Dan	3.00	4.00	4.50	?	3.00	4.50	4.00	2.00
Hailey	?	4.00	1.00	4.00	?	?	4.00	1.00
Jordyn	?	4.50	4.00	5.00	5.00	4.50	4.00	4.00
Sam	5.00	2.00	?	3.00	5.00	4.00	5.00	?
Veronica	3.00	?	?	5.00	4.00	2.50	3.00	?

Normalized User-Item data ($NR_{v,i}$): lies between -1 and 1

usersItemRatingsNorm	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	0.25	-0.50	?	0.75	1.00	-0.75	-0.25	-0.50
Bill	-0.50	0.25	0.50	?	-0.50	0.25	?	0.00
Chan	1.00	-1.00	-1.00	0.00	1.00	-1.00	?	?
Dan	0.00	0.50	0.75	?	0.00	0.75	0.50	-0.50
Hailey	?	0.50	-1.00	0.50	?	?	0.50	-1.00
Jordyn	?	0.75	0.50	1.00	1.00	0.75	0.50	0.50
Sam	1.00	-0.50	?	0.00	1.00	0.50	1.00	?
Veronica	0.00	?	?	1.00	0.50	-0.25	0.00	?

$$NR_{v,i} = \frac{2(R_{v,i} - Min_R) - (Max_R - Min_R)}{(Max_R - Min_R)}$$

$$\frac{2(4-1) - (5-1)}{(5-1)}$$

(iii) Then predict (normalized) rating of how well user v will like item k

Normalized
User-Item data
($NR_{v,j}$)

usersItemRatingsNorm	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	0.25	-0.50	?	0.75	1.00	-0.75	-0.25	-0.50
Bill	-0.50	0.25	0.50	?	-0.50	0.25	?	0.00
Chan	1.00	-1.00	-1.00	0.00	1.00	-1.00	?	?
Dan	0.00	0.50	0.75	?	0.00	0.75	0.50	-0.50
Hailey	?	0.50	-1.00	0.50	?	?	0.50	-1.00
Jordyn	?	0.75	0.50	1.00	1.00	0.75	0.50	0.50
Sam	1.00	-0.50	?	0.00	1.00	0.50	1.00	?
Veronica	0.00	?	?	1.00	0.50	-0.25	0.00	?

Similarity Matrix
($S_{i,j}$)

similarityMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	-0.88	-0.97	-0.04	0.85	-0.67	0.43	0.17
Broken Bells	-0.88	-	0.24	0.21	-0.86	0.69	0.02	-0.42
Deadmau5	-0.97	0.24	-	-0.87	-0.97	0.95	-0.57	0.35
Norah Jones	-0.04	0.21	-0.87	-	0.48	-0.71	-0.27	-0.93
Phoenix	0.85	-0.86	-0.97	0.48	-	-0.89	-0.22	-0.3
Slightly Stoopid	-0.67	0.69	0.95	-0.71	-0.89	-	0.58	0
The Strokes	0.43	0.02	-0.57	-0.27	-0.22	0.58	-	-0.57
Vampire Weekend	0.17	-0.42	0.35	-0.93	-0.3	0	-0.57	-

Now let's say we're trying to predict (normalized) rating of Deadmau5 for Angelica

$$NR_{v,k} = \frac{\sum_{j \in \{S_{k,j} \text{ exists}\} - \{k\}} (S_{k,j} * NR_{v,j})}{\sum_{j \in \{S_{k,j} \text{ exists}\} - \{k\}} (|S_{k,j}|)} =$$

$$\frac{(0.25)(-0.97) + (-0.50)(0.24) + (0.75)(-0.87) + (1.00)(-0.97) + (-0.75)(0.95) + (-0.25)(-0.57) + (-0.50)(0.35)}{|-0.97| + |0.24| + |-0.87| + |-0.97| + |0.95| + |-0.57| + |0.35|}$$

= -0.56

Note: similarly, can predict ALL unknown (normalized) ratings ($\hat{R}_{v,k}$): these lie between -1 and 1

usersItemRatingsNormPred	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	0.25	-0.50	-0.56	0.75	1.00	-0.75	-0.25	-0.50
Bill	-0.50	0.25	0.50	-0.24	-0.50	0.25	-0.10	0.00
Chan	1.00	-1.00	-1.00	0.00	1.00	-1.00	0.09	-0.03
Dan	0.00	0.50	0.75	-0.21	0.00	0.75	0.50	-0.50
Hailey	0.22	0.50	-1.00	0.50	0.34	-0.23	0.50	-1.00
Jordyn	-0.13	0.75	0.50	1.00	1.00	0.75	0.50	0.50
Sam	1.00	-0.50	-0.47	0.00	1.00	0.50	1.00	-0.21
Veronica	0.00	-0.15	-0.37	1.00	0.50	-0.25	0.00	-0.55

(iv) Then de-normalize the predicted ratings

Recall from step (iii):

Predicted (normalized) rating of Deadmau5 for Angelica = **-0.56**

Therefore predicted (de-normalized) rating is

$$R'_{v,k} = \frac{1}{2} \left((NR'_{v,k} + 1) * (Max_R - Min_R) \right) + Min_R$$

$$= \frac{1}{2} ((-0.56 + 1)(5 - 1)) + 1$$

$$= \mathbf{1.89}$$

Note: similarly, can compute ALL predicted (de-normalized) ratings ($R'_{v,k}$): these lie between $Min_R=1$ and $Max_R=5$

usersItemRatingsDeNormPred	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.50	2.00	1.89	4.50	5.00	1.50	2.50	2.00
Bill	2.00	3.50	4.00	2.52	2.00	3.50	2.80	3.00
Chan	5.00	1.00	1.00	3.00	5.00	1.00	3.17	2.94
Dan	3.00	4.00	4.50	2.57	3.00	4.50	4.00	2.00
Hailey	3.45	4.00	1.00	4.00	3.69	2.54	4.00	1.00
Jordyn	2.73	4.50	4.00	5.00	5.00	4.50	4.00	4.00
Sam	5.00	2.00	2.06	3.00	5.00	4.00	5.00	2.59
Veronica	3.00	2.70	2.26	5.00	4.00	2.50	3.00	1.90

(2) Weighted Slope One

Let's say we want to predict how well user v will like item k . It's a 2-step process:

i) First compute deviations between every pair of items

$$D_{i,j} = \sum_{u \in U_{i,j}} \frac{(R_{u,i} - R_{u,j})}{\text{card}(U_{i,j})}$$

Where

$D_{i,j}$ is the average deviation of item i wrt item j

$U_{i,j}$ is the set of all users who rated both items i and j

$\text{card}(U_{i,j})$ is the number of users who have rated both items i and j

$(R_{u,i} - R_{u,j})$ is the difference in ratings of items i and j by user u

ii) Then predict rating of how well user v will like item k

$$R'_{v,k} = \frac{\sum_{j \in \{R_{v,j} \text{ exists}\} - \{k\}} (D_{k,j} + R_{v,j}) * \text{card}(U_{k,j})}{\sum_{j \in \{R_{v,j} \text{ exists}\} - \{k\}} (\text{card}(U_{k,j}))}$$

Let's say we're trying to find the predicted rating of Deadmau5 for Angelica

i)

Ratings	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.50	2.00	?	4.50	5.00	1.50	2.50	2.00
Bill	2.00	3.50	4.00	?	2.00	3.50	?	3.00
Chan	5.00	1.00	1.00	3.00	5.00	1.00	?	?
Dan	3.00	4.00	4.50	?	3.00	4.50	4.00	2.00
Hailey	?	4.00	1.00	4.00	?	?	4.00	1.00
Jordyn	?	4.50	4.00	5.00	5.00	4.50	4.00	4.00
Sam	5.00	2.00	?	3.00	5.00	4.00	5.00	?
Veronica	3.00	?	?	5.00	4.00	2.50	3.00	?

frequencyMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	5	3	4	6	6	4	3
Broken Bells	5	-	5	5	6	6	5	5
Deadmau5	3	5	-	3	4	4	3	4
Norah Jones	4	5	3	-	5	5	5	3
Phoenix	6	6	4	5	-	7	5	4
Slightly Stoopid	6	6	4	5	7	-	5	4
The Strokes	4	5	3	5	5	5	-	4
Vampire Weekend	3	5	4	3	4	4	4	-

deviationMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	1.2	0.17	0.25	-0.42	0.75	0.00	0.5
Broken Bells	-1.2	-	0.5	-1.2	-1.33	-0.33	-0.6	1.2
Deadmau5	-0.17	-0.5	-	-2	-0.38	0	-0.83	0.88
Norah Jones	-0.25	1.2	2	-	-0.7	1.4	0.6	2.17
Phoenix	0.42	1.33	0.38	0.7	-	1.07	0.7	1
Slightly Stoopid	-0.75	0.33	0	-1.4	-1.07	-	-0.3	0.75
The Strokes	0	0.6	0.83	-0.6	-0.7	0.3	-	1.38
Vampire Weekend	-0.5	-1.2	-0.88	-2.17	-1	-0.75	-1.38	-

$$D_{i,j} = \sum_{u \in U_{i,j}} \frac{(R_{u,i} - R_{u,j})}{\text{card}(U_{i,j})}$$

$$\frac{1-3}{3} + \frac{1-4}{3} + \frac{4-5}{3}$$

Let's say we're trying to find the predicted rating of Deadmau5 for Angelica

ii)

deviationMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	1.2	0.17	0.25	-0.42	0.75	0.00	0.5
Broken Bells	-1.2	-	0.5	-1.2	-1.33	-0.33	-0.6	1.2
Deadmau5	-0.17	-0.5	-	-2	-0.38	0	-0.83	0.88
Norah Jones	-0.25	1.2	2	-	-0.7	1.4	0.6	2.17
Phoenix	0.42	1.33	0.38	0.7	-	1.07	0.7	1
Slightly Stoopid	-0.75	0.33	0	-1.4	-1.07	-	-0.3	0.75
The Strokes	0	0.6	0.83	-0.6	-0.7	0.3	-	1.38
Vampire Weekend	-0.5	-1.2	-0.88	-2.17	-1	-0.75	-1.38	-

frequencyMatrix	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Blues Traveler	-	5	3	4	6	6	4	3
Broken Bells	5	-	5	5	6	6	5	5
Deadmau5	3	5	-	3	4	4	3	4
Norah Jones	4	5	3	-	5	5	5	3
Phoenix	6	6	4	5	-	7	5	4
Slightly Stoopid	6	6	4	5	7	-	5	4
The Strokes	4	5	3	5	5	5	-	4
Vampire Weekend	3	5	4	3	4	4	4	-

usersItemRatingsPred	Blues Traveler	Broken Bells	Deadmau5	Norah Jones	Phoenix	Slightly Stoopid	The Strokes	Vampire Weekend
Angelica	3.50	2.00	2.54	4.50	5.00	1.50	2.50	2.00
Bill	2.00	3.50	4.00	3.80	2.00	3.50	3.31	3.00
Chan	5.00	1.00	1.00	3.00	5.00	1.00	2.72	1.41
Dan	3.00	4.00	4.50	4.43	3.00	4.50	4.00	2.00
Hailey	3.53	4.00	1.00	4.00	3.85	2.85	4.00	1.00
Jordyn	4.86	4.50	4.00	5.00	5.00	4.50	4.00	4.00
Sam	5.00	2.00	3.27	3.00	5.00	4.00	5.00	2.76
Veronica	3.00	2.56	2.85	5.00	4.00	2.50	3.00	2.30

$$R'_{v,k} = \frac{\sum_{j \in \{R_{v,j} \text{ exists}\} - \{k\}} (D_{k,j} + R_{v,j}) * \text{card}(U_{k,j})}{\sum_{j \in \{R_{v,j} \text{ exists}\} - \{k\}} (\text{card}(U_{k,j}))}$$

$$\frac{(-0.17 + 3.50)3 + (-0.5 + 2.00)5 + (-2 + 4.50)3 + (-0.38 + 5)4 + (0 + 1.50)4 + (-0.83 + 2.50)3 + (0.88 + 2.00)4}{3 + 5 + 3 + 4 + 4 + 3 + 4}$$