To reduce overfitting, improve performance and ease implementation, the Slope One family of easily implemented Item-based Rating-Based collaborative filtering algorithms is used.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Item P** | **Item Q** | **Item R** |
| **User A** | **5** | **3** | **2** |
| **User B** | **3** | **4** | **?** |
| **User C** | **?** | **2** | **5** |

The above table shows ratings table by users to items…

**Example**:

1. User A gave rating 5 to Item P , rating 3 to Item Q and rating 2 to Item R
2. User B gave rating 3 to Item P, rating 4 to Item Q
3. User C gave rating 2 to Item Q, rating 5 to Item R

Now the question is how user B will rate Item R and how User C will rate Item P according to slope one algorithm

**SLOPE ONE METHODOLOGY:- HERE we will find rating by User C to item P**

1. Slope one first calculate the average difference in ratings between two items..

In this case,

The average difference in ratings between item Q and P is

**((5-3) + (3-4)) /2 = (2+(-1))/2 = 0.5. ( this will calculate for only those users that has given ratings to both the items)**

Hence, on average, item P is rated above item Q by 0.5.

Similarly, The average difference between item R and P is

**(5-2)/1 = 3. (as there is only user A that has rated both the items R and P)**

Hence, Hence, on average, item P is rated above item R by 3.

1. **If we attempt to predict the rating of user C for item P using his rating for item Q,**

**we get 2+0.5 = 2.5**.

Similarly**, if we try to predict his rating for item P using his rating of item R, we get 5+3=8.**

1. If a user rated several items, the predictions are simply combined using a weighted average where a good choice for the weight is the number of users having rated both items. In the above example, we would predict rating for User C on item P as follows:

= **(2\*2.5+1\*8)/2+1 = 13/3 = 4.33**

= ( (no of users that has rated both items P and Q \* rating of item P by user C using his rating for item Q) **+ (**no of users that has rated both items P and R \* rating of item P by user C using his rating for itemR ) **)** / (no of users that has rated both items P and Q + no of users that has rated both items P and R)

Hence, given ***n*** items, to implement Slope One, all that is needed is to compute and store the average differences and the number of common ratings for each of the ***n*2** pairs of items.