**Item Based Collaborative Filtering**

The main aim of this project was to implement an item-based collaborative filtering recommender system for a given user.

**Item Based Collaborative Filtering**

**Format of data file for item**

The input data file is in the form of Tab separated value - ratings-dataset.tsv. The file consists of one rating event per line. Each rating event is of the form:

User\_id\tRating\tMovie\_title

The User\_id is a string that contains only alphanumeric characters with hyphens or spaces (no tabs). The rating is one of the float values 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, and 5.0 The Movie\_title is a string . The three fields -- User\_id, rating, and the Movie\_title -- are separated by a single tab character (\t).

**Running the code**

***python nayak\_nandan\_collabFilter.py <ratingsFileName> <user> <n> <k>***

The program takes 4 arguments:

1. ratingsFileName: The name of the ratings file.

2. user: The id of the user (string) for whom you should make recommendations.

3. n: The size of the neighborhood N to be used in the prediction equation.

4. k: The number of recommendations that should be made for the specified user.

**Output of the program**

The program outputs ‘k’ recommended movies for the user specified in the input parameter and the predicted rating (rounded up to 5 decimal places) for each of those movies. The output is sorted in descending order of the predicted ratings. In case the ratings are same for two movies, they are sorted ascendingly by name.

To test the code, you can run:

***python nayak\_nandan\_collabFilter.py ratings-dataset.tsv Kluver 20 5***

**Output:**

Eternal Sunshine of the Spotless Mind 4.19946

Kill Bill: Vol. 2 4.19182

Kill Bill: Vol. 1 4.17114

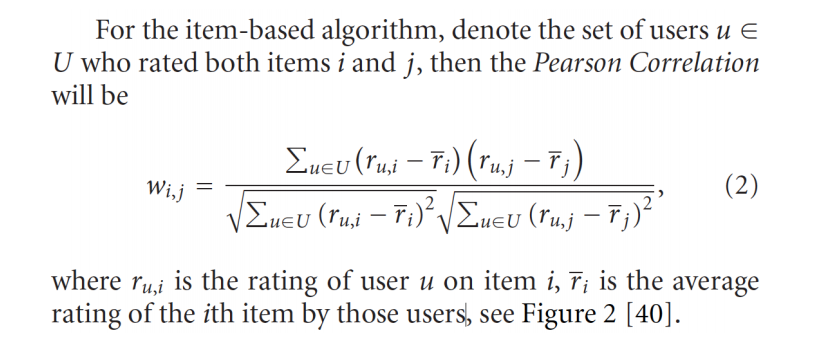
Erin Brockovich 4.16667

Sin City 4.16667

**Program Description**

**Finding Item Similarity**

The ratings file is read and the similarity between all pairs of items is computed using the Pearson correlation equation as mentioned below:

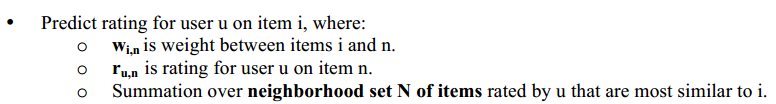


* Sum over set of users U who rated both items i, j.
*  is rating of user u on item i.
*  is average rating of i-th item by those users.

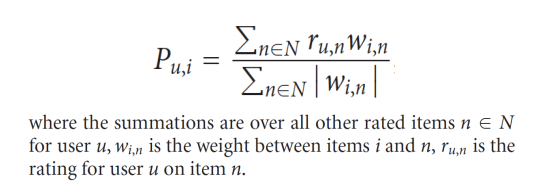
Sometimes there is not enough data to compute the similarity between two items. In the worst case -- for example, if only one user rated both items -- the denominator in the formula will be zero. In such cases, the similarity function just returns 0.0.

**Ratings Prediction**

Using the prediction equation, the rating of a user ‘U’ on an item ‘i’ that the user has not already rated is predicted:



If there are fewer than N items with nonzero similarity to item i, then the similarity function returns a prediction of 0.0.



The program outputs a list of ‘k’ items to recommend to the user specified as an input argument. The k items with the highest predicted rating for this user is identified.