**Recommender System**

**Introduction**

Recommender System applies knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. In our problem statement, users give ratings ranging from 1 to 5 to items. We have to predict rating of those items, which the user has not rated. We can use item based similarity or user based similarity for this purpose. Scalability and Sparsity are two important challenges faced by user based system. Also, cold start problem is an important problem faced by recommender systems. Hence, I used item based similarity. In that I use adjusted cosine similarity to compute similarity, since it has the minimum Mean Absolute Error (MAE) amongst other similarity techniques. I compute prediction using weighted sum technique, since regression technique is good only for a very sparse dataset.

**Code Flow:**

Code execution starts with Driver.java. Input is read from an input file (train\_all\_txt.txt) line by line and a 2D matrix is a [] [] is created using the data from the input file. I calculate user average and store the result in a float array Ru [].I calculate Item similarity using the adjusted cosine similarity. The similarity threshold is set to 0.3. This means that, the items whose adjusted cosine similarity is greater than or equal to 0.3, are similar. Hence, I use HashMap data structure to store the item and a list of similar items, similar to that item. I use HashMap, since while computing prediction, I can retrieve list of similar items in time complexity O (1). The adjusted cosine similarity value is added to a 2D matrix of items, so that, I can retrieve the item similarity in time complexity O(1).The column matrix, colmatrix, has 1682 rows and 1682 columns. For final step of prediction computation, I use weighted sum technique. In this technique, I compute the rating given by the user on items similar to the item, whose value is to be predicted. The predicted value is rounded to the nearest integer. If this value is less than or equal to 0, rating is set to 1. If the rounded predicted value is greater than 5, the rating is set to 5. Also, there may be items that are not similar to any of the 1682 items. If a user has 0 rating for such an item, the rating is set to 1. Finally, entire output data is in output matrix b [] [], which is then, written to the output.txt file.

**Evaluation Results**

Prediction computation of 943 users and 1682 items takes roughly 1 minute to produce results. Cold start problem is handled internally by the item based algorithm. High performance and improved quality are salient features of item based similarity.

**References**

[1] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. *Item –based Collaborative Filtering Recommendation Algorithms*

[3] John S. Breese, David Heckerman Carl Kadie. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*

[4] http://stackoverflow.com/questions/7899525/how-to-split-a-string-by-space

[5]http://stackoverflow.com/questions/1746501/can-someone-give-an-example-of-cosine-similarity-in-a-very-simple-graphical-wa

[6] http://grouplens.org/datasets/movielens/