

## Recommender System

### Abstract:

Collaborative filtering Recommender Systems use a database about user preferences to predict additional topics a user might like. In this paper, we present the user based collaborative filtering technique to predict the user likes based on similar users' taste. We use Pearson's Correlation Coefficient algorithm and then compute them on weighted sum to predict the user rating.

### 1. Introduction

Given the data in the form of user, item and rating, we need to predict the value to those items, which the user has not rated. In our problem statements, users give ratings from 1 to 5, 1 being least liked and 5 being most liked by user. We used user based collaborative filtering as the dataset given has less number of records. Considering Sparsity of data, we used user-based algorithm as number of users are less than number of items in the dataset. In user-based, I used Pearson's Correlation algorithm and Weighted Sum to predict the user rating on an item.

### 2. Dataset

The number of users are 943 and number of items are 1682. Rating given in the range of 1 to 5, 1 being least and 5 being most liked item by user. Dataset has around 80,000 records.

### 3. Collaborative Filtering Algorithm

I used model-based in order to increase the performance of the algorithm. Program starts by reading the data from file and storing that in two dimensional array with first dimension as user and second dimension as item being rating the value. I calculated the user similarity using Pearson's Correlation Coefficient algorithm. The similarity threshold is set to 0.2. Value with equal or greater than threshold are the similar users. I stored them in Hash Map in order to retrieve the records faster. After that, I used weighted sum on similar taste's user to compute the rating for the item to whom user has not rated. Finally the output matrix is write to file.

### 4. Result

Prediction of 943 users and 1682 items on building the model for 80,000 records and computing the prediction for zeros rating roughly takes 30 seconds. Sparsity of data, accuracy and performance are achieved from this algorithm.

### References

1. J. Breese, D.Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proc. Conf. Uncertainty in Artificial Intelligence, (UAI98) 1998
2. [https://en.wikipedia.org/wiki/Collaborative\\_filtering](https://en.wikipedia.org/wiki/Collaborative_filtering)
3. <http://grouplens.org/datasets/movielens/100k/>
4. [http://www.cs.carleton.edu/cs\\_comps/0607/recommend/recommender/modelbased.html](http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/modelbased.html)