Item Based Collaborative Filtering with User Features

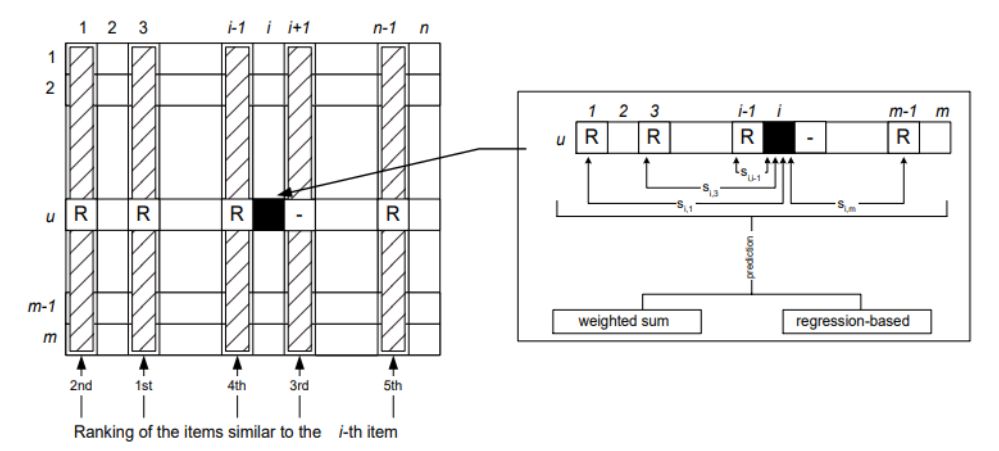
Project Report submitted by Krishna Sirisha Motamarry

**Abstract**

Recommender systems play a significant role in most of the online applications like Amazon, Netflix, eBay, Best Buy. Collaborative Filtering algorithms make personalized recommendations in which Item Based Collaborative Filtering produce high quality recommendations by computing the item item similarities. Though the algorithm performs better, it does not consider any of the user features. Demographic information of the user is also very important as they contribute a lot to the recommendations. The proposed approach takes in the user demographic data along with the ratings, groups users based on their features, calculates the user similarities within the group in order to find the most similar users. Item based Collaborative Filtering is then applied to calculate the predictions. In this report, we look into various similarity measures, prediction computation and comparison of the proposed approach to the Item Based Collaborative Filtering.

1. **Overview of Item Based Collaborative Filtering Algorithm:**

The Item Based Collaborative Filtering approach looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items. Adjusted Cosine Similarity measure and weighted sum prediction method are employed to generate good quality predictions.



1. **Issue with the Item Based Collaborative Filtering Algorithm:**

The user features are completely isolated and only the ratings on the items are taken into consideration. This may lead to low quality recommendations. Consider a case where the database has 80% of users who are of age 50-90. Their ratings on items may dominate the other 20% of the group and may lead to recommendations which are not appropriate. In order to overcome this issue, the user’s demographic information are utilized to assist in generating quality recommendations.

1. **Item Based Collaborative Filtering with User Features:**

In the present approach, instead of directly computing the predictions from item similarities, the users are grouped based on selected features, most similar users in a group are determined and with the most similar items, the rating value is predicted. The similarity computation and prediction generation methods are listed in the later sections.

**3.1 Feature Selection and Grouping of users:**

In order to group the users, first the most important feature needs to be selected on which the users would be grouped. Random Forest algorithm is used for deriving the feature importances. The user demographic data is fed to the random forest algorithm and the most important feature is determined.

**3.1.1 Label Encoding:**

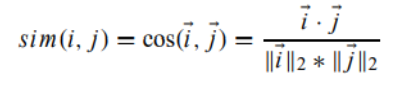
Label Encoder encode labels with values between 0 and n\_classes-1, where n\_classes are the n categories the feature has. For example, gender feature has values Male and Female. So wherever value “male” is found the method encodes it to 0 whereas, whenever there is a value “female” it encodes it to 1. Before deriving feature importances, all the categorical features which are not of integer type need to be label encoded to transform it to an integer encoded value.

**3.2 Similarity Computation:**

In the present approach, two similarity measures are needed, one for computing the most similar users within the group formed and other for computing the most similar items. The following are the similarity measures used for User Similarity and Item Similarity computations.

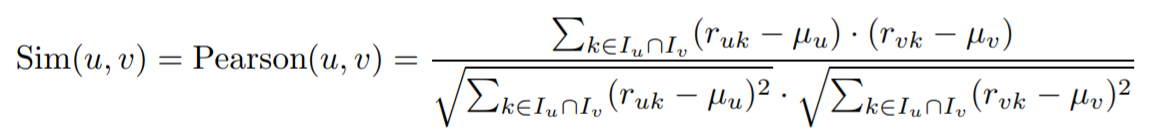
**3.2.1 User Similarity Computation:** A group matrix is constructed with users as rows and items as columns. Each user, item pair has a rating value specified in the cell for corresponding user row and item column

**Cosine-based Similarity:** One way of computing the similarity between two users is to treat each user as a vector and cosine between the vectors is computed as the measure of similarity.



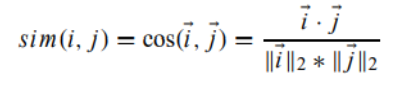
Where i and j are thought of user vectors

**Pearson Correlation Similarity:** The Pearson correlation coefficient between the user’s u and v is defined as follows



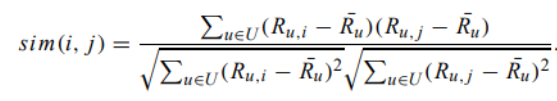
**3.2.2 Item Similarity Computation**: Once the most similar users in a group are identified. A user-item matrix is constructed with only the most similar users and all items. Similarity between the items are computed by one of the similarity measures listed below

**Cosine-based Similarity:** One way of computing the similarity between two items is to treat each item as a vector and cosine between the vectors is computed as the measure of similarity.



Where i and j are thought of item vectors

**Adjusted-Cosine Similarity:** In the adjusted cosine similarity, the user mean is subtracted from the raw rating value and makes the ratings mean centered. The adjusted cosine similarity between items i and j is computed as



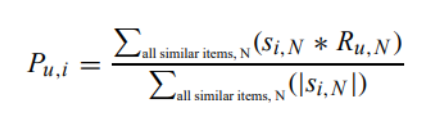
Where is the average of the uth user’s ratings

**3.3 Prediction Computation:**

Once the similar items are determined, the next step is to calculate the prediction. Here I have considered the weighted sum technique

**3.3.1 Weighted Sum:**

This method computes the prediction on an item i for a user u by computing the sum of ratings given by the user on the items similar to i. Ratings are weighted by their corresponding similarity. It is defined as follows



Where Si,j  is the similarity value of items i and j

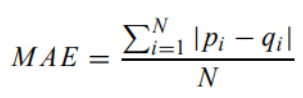
**4 Experimental Evaluation:**

**4.1 Data Set**

For this project, data was used from MovieLens Recommender System. It is a web based recommender system and consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has at least rated 20 movies. Simple demographic information for the users i.e age, gender, occupation and zipcode are included. The 100K rows are divided into a training and a test set.

**4.2 Evaluation Metrics:**

**Mean Absolute Error:** Mean Absolute Error is a widely used metric. For every predicted value of an item, the corresponding user specified true values are compared and absolute error between them is calculated. The MAE is computed by first summing these absolute errors of the N corresponding ratings-prediction pairs and then computing the average. The lower the MAE, the predicted values are more accurate.



Where pi, qi is the original rating, prediction pair

**4.3 Experimental Procedure:**

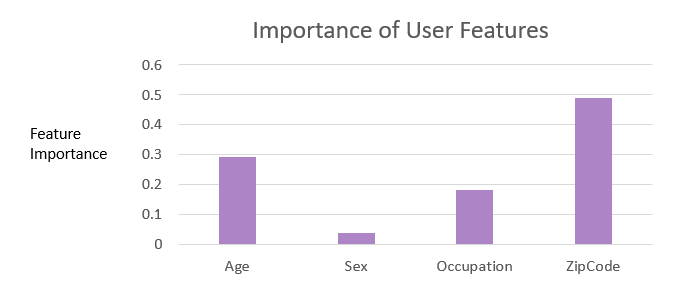
Initially the dataset is divided into training and test set (validation set). The training data set is further divided into training and test set for tuning the parameters. I have conducted a 5 fold cross validation on the experiments by randomly choosing training and test sets each time and taking the average of MAE values.

**Item Based Collaborative Filtering**: To compare the performance of the present approach, the original Item Based Collaborative Filtering method was employed. The parameters were tuned to generate high quality predictions.

**5 Experimental Results:**

**5.1 Feature Selection:**

Grouping of the users requires to select the most important user feature. For this purpose, random forest algorithm for feature selection was applied. The features in the given data set are the user demographic information like age, gender, occupation and zipcode where in the gender, occupation and zipcode are of object type. Through Label Encoding all the object categorical variables are converted to integer encoded data. All the features are then fed into random forest algorithm and the most important feature is determined. Below are the results obtained



As depicted from the above representation, zip code has higher importance, followed by age and occupation. For the experiments, I have considered occupation as the feature to be grouped on, as there were 795 unique zip codes in the dataset for a total of 943 users. Grouping the users by zip code would result in 1 or 2 users per group which would be of no use. The next important feature was age but it was not considered because the range of age should be predefined manually. Hence the results vary when the ranges change and cannot be relied upon.

**5.2 Learning the parameters:**

The training set is further divided to training and a test set to learn the effect of item neighborhood size, user neighborhood size and the similarity measures which are listed in the next section

**5.2.1 Effect of Item Neighborhood Size:**

To determine the effect of Item Neighborhood size, the experiment was conducted on the whole group of users present in the training set by varying the item sizes from 100 to 600 with an increment of 100. Also, for determining the similar users, cosine and pearson correlation measures were employed and for determining the similar items cosine and adjusted cosine were employed. The combination of these metrics for user-user and item-item similarities at different item neighborhood sizes are presented in the results below. It can observed that irrespective of the similarity measure used, the MAE value decreases at 400 and starts to increase as the number of items increase. Hence the optimum value for Item neighborhood of 400 is selected.

MAE Error

**5.2.2 Effect of User Neighborhood Size:**

The number of similar users has a significant impact on the MAE value. Considering the dissimilar users may lead to high error in predicted value. To determine the effect of this parameter, item neighborhood size was fixed at the value 400 and experiments were done by varying the number of users i.e 25%, 50%, 75% of the most similar users in a group with all the four combinations of similarity measures as displayed below. It can be observed from the results that 50% of most similar users in a group lead to a lower MAE value. Hence the optimum value for the User Neighborhood size is considered as 50% of the most similar users in a group.

MAE Error

**5.2.3 Effect of Similarity Measures:**

To determine the effect of similarity measures, 50% of the most similar users in a group are considered and the values of item neighborhood size is taken as 400. Experiments were conducted on all four combinations of similarity measures and results are displayed below. It can be observed that Cosine and Pearson Correlation for computing the user similarities and Adjusted Cosine for computing the item similarities were having almost same MAE values while the other two combinations have the higher MAE values. Hence, Cosine measure for computing the user similarities and Adjusted Cosine for computing the item similarities are considered for further experimentation.

MAE Error

**5.2.4 Evaluation Results of Proposed Approach vs Item Based Collaborative Filtering:**

Once the parameters are learned and derived for the proposed approach, the experiments are conducted with learned parameters on the validation data set (10K rows). The approach is trained on the whole training set which has 90K rows. The results observed are as follows

MAE Error

Results depict that the proposed approach has MAE value of 0.8 while the item based collaborative filtering has an error of 0.79. The results are almost same as that of original method and there is no increase in accuracy.

**Conclusion:**

In this project, a different approach of evaluating item based collaborative filtering with user features is implemented. Though, there is no much change in accuracy, this approach would work with data sets that are dominated by a large group of users belonging to the same group, which in turn effects the quality of recommendations generated to the small group of users. The proposed approach also leads to good quality predictions even in normal datasets as observed in the results presented above.

**References:**

[1] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In WWW10

[2] MovieLens - https://grouplens.org/datasets/movielens/100k/

[3] Neighborhood Based Collaborative Filtering - https://link.springer.com/chapter/10.1007/978-3-319-29659-3\_2

[4] Understanding variable importances in forests of randomized trees - Gilles Louppe, Louis Wehenkel, Antonio Sutera and Pierre Geurts