Ekapope V., Deborah K., Anne K., Yenchun L., Majid M.

Group Assignment Recommendation Tools

Introduction

The purpose of this assignment is to get an in-depth understanding on recommendation systems,understand some of the different types there are and to see which systems predicts ratings and preferences of users more effectively. For this assignment, the LastFM data set was used. This data set contains music streaming data as below:

- 17,632 artists
- 1,892 users
- 92,834 user-listened artist relations
- 11,946 tags
- 186,479 tag assignments

The report covers three main parts:

- Reading in data.
- Creation of new functions.
- Computing and evaluating recommendation systems.

Reading In the data

1. **Creating base matrices for chosen recommendation systems**

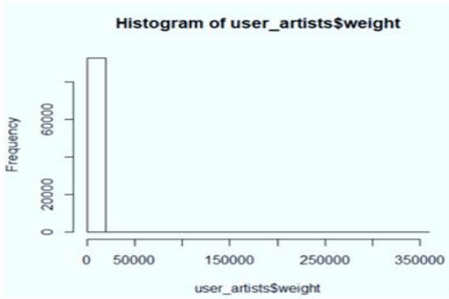
To start with our recommender systems, two base matrices were to be created from the available data sets

Content-based recommendation system: For this, two data sets were merged. They are:‘user_taggedartists.dat' and 'tags.dat' data. Merging was done using the tagID variable, all unused columns were dropped (day,month,year,tagID). An additional dummy variable value was added with a value of 1 to allow changing of the data frame to a wide data. Final data frame after merging had three variables (artistID,tagValue,value). The data frame is transposed using the dcast Reshape2 r package. Other pre-processing:

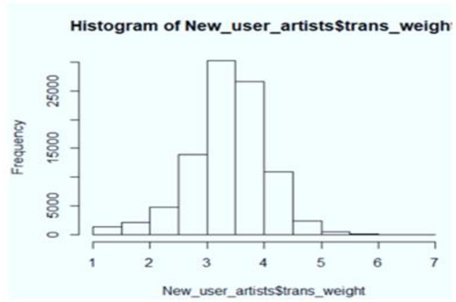
- Changing of the column names formats by adding the length to 5 and an ‘a’ for example ‘a00001’
- Creation of more dummies
- Summation of column sums for all the tags based on the values
- Data is subset after the tagvalues are arranged in descending order and the top names picked per artistID. Data frame is then coerced into a matrix.
- 12133 artists

Collaborative filtering: For this, the 'user_artists.dat' data set was used. The dataset had three variables (*Users* which would in this case be considered as users in recommender terms, *artists* who would be considered as items and the *weights* as the artist ratings from the different LastFM users). We first started by inspection of the data to check on the distribution. The weights indicated that the data was highly skewed i.e. about 45 skewness. The impact of this is that skewed data tends to have a disproportionate effect on parameter estimates which are based on minimization of squared error. A log transformation on the weight’s variable was done to remove the skewness and transform the weights into a normal distribution. See figure below.

Original right skewed data



Log10 transformed data



Once the weights variable was transformed through the log10, the 'user_artists.dat' dataset was then changed into a wide data using the dplyr separate function to have the user's variable as the rows, artist variable as the columns and the transformed weights as the behaviour values. This data frame was then changed to a matrix to act as a data input for all our subsequent recommendation systems building.

Other preprocessing tasks done before the spread function was applied to the data frame include:

- Rounding off the transformed weights values to 2 decimal places
- Adding 'a' to every artist value and having all have a maximum length of 5 for example 'a00001', same was done for the users i.e. add u and a maximum length of 4 u0001 for user number 1 in this instance. This was done for better labelling purposes
- Replacing all NA's in the matrix with zeros.

Creation of new functions

Collaborative Filtering

Collaborative filtering recommendation systems operate based on the assumption that users give ratings to catalogue items either explicitly(actual preference given by a user e.g. star rating) or implicitly(no explicit indication of user preference but other measures such as number of clicks done by a user etc.) and secondly users that have had similar tastes in the past and likely to have similar tastes in the future. For our task we concentrated on memory-based methods which include the user-based collaborative filtering (UBCF) and Item-based collaborative filtering (IBCF). The base matrix from the previous steps is split into train and test data (70/30 respectively). K-nearest neighbour approach is used for both methods that are covered in this assignment.

UBCF function

UBCF assumes that users are similar to each other based on similar items ratings. It ideally works well when: -

- The number of items is more than the number of users.
- Items change less frequently than the users

Actual function…

First an empty similarity matrix is initialized for pair of users available in our base matrix. Although there are several methods to calculate similarity Pearson correlation is used in this case for both collaborative filtering methods. Assuming we have two users x and y ,the functions works' by, letting $r_{x,i}$ be the rating of the ith item under user x, $\overline{r_x}$ be the average rating of user x, and I_{xy} be the set of items rated by both user x and user y. i is the item in set . The same applies to user y. Therefore, similarity between user x and y is given by the below formula.

$$sim_{(x,y)} = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \overline{r_x})(r_{y,i} - \overline{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \overline{r_x})^2 \sum_{i \in I_{xy}} (r_{y,i} - \overline{r_y})^2}}$$

Once the the similarity value between user x and user y is obtained we then used the KNN method to find the k closest users to a target user. Assuming that our nearest neighbour value is set to 10 this means that our algorithm searches for 10 users closest to our target user.A critical value was also obtained to allow the splitting of the clusters.Afterwards the predictions were done on the test data.The predictions of the users per item was obtained based on artists not previewed by the users before and the TopN in our case Top3 artist were recommended for ever user.

IBCF function

IBCF assumes that items are similar to each other. It ideally works well when: -

- The number of users is more than the number of items.

Actual function…

First an empty similarity matrix is initialized for pair of items available in our base matrix. Assume similarity between item i and item j.The functions works' by, letting $r_{u,i}$ be the rating of the ith item under item i, $\overline{r_i}$ be the average rating of item i, and U_{ij} be the set of users rated both item i and j. i is the item in set . The same applies to item j. Therefore, similarity between item i and j is given by the below formula.

$$sim_{(i,j)} = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{i \in U_{ij}} (r_{u,i} - \overline{r_i})^2 \sum_{u \in U_{ij}} (r_{u,i} - \overline{r_j})^2}}$$

Once the the similarity value between item i and item j is obtained we then used the KNN method to find the k closest items to a target item. If our nearest neighbor value is set to 10 this means that our algorithm searches for 10 items closest to our target item. A critical value was also obtained to allow the splitting of the clusters. Afterwards the predictions were done on the test data. The predictions of the users per item was obtained and the TopN in our case Top3 artist were recommended for ever user.

Cluster-based

Clustering is a method to take several groups of similar data. It divides data into groups or clusters such that similar points are in the same cluster and dissimilar points are in different cluster.

We are using Cluster based collaborative filtering using k-means method. We made a function to predict top 5 items for every user. This function is first calculating the mean of every column and filling the missing values with their respective column mean.

Then we divided users into 100 clusters using k-means function in R. After that we calculated average rating for every cluster. And applied prediction function to get top 5 predictions for every user in a cluster.

Evaluations

- **Prediction accuracy: MAE**

Mean absolute error is the mean of absolute errors where y_i is the prediction and ŷi is the true value. In other words, MAE is the absolute difference between prediction and true value. MAE and Root mean squared error (RMSE) are similar in a sense that they both measure the mean magnitude of the error. However, MAE does not require the use of square and square root. Based on MAE formula below, we created a function and used it to calculate how our predictions made by user-based recommendation are different from real values. We first used UserBasedCF function and divided the dataset into 10 clusters and set TopN equal to 3, so in that way, each user gets top 3 artist recommendations. After this, we compared our prediction value acquired through user-based CF with the *test_data* (real value). Our MAE for user-based collaborative filtering is 0.022.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

- **Classification accuracy: F1**

F1 score is a measure of accuracy, which provides a weighted average of precision and recall. F1 score is useful when choosing a best model out of models that have higher scores in precision but lower scores in recall and vice versa. Our goal is to choose a recommendation system that has a highest F1 score (high precision and high recall). In order to obtain F1 score, we first created a classification function and a F1 function given the following information:

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned}$$

When the prediction and the real value are equal to threshold, we defined it as True Positive (TP). when the prediction is smaller than threshold, but the real value is greater or equal to threshold, we defined as False Negative (FN). We divided our users into 10 and used binary classification (threshold=2). Our precision recall and F1 measure for user-based collaborative filtering are 0.799, 0.024 and 0.047 respectively.

Hybridization Technique

In order to improve predictive performance and complement the weaknesses of each recommendation system, we created the hybrid recommendation system (combination of UBCF and Cluster-based) and took the mean and highest value of UBCF and Cluster-based.

Combined Evaluation

	UBCF	Cluster-based	IBCF[1]	Content-based	Hybrid[2]	
					Mean	Max
MAE	0.030	0.56	0.064	0.030	0.295	0.569
Recall	0.022	1	0.349	0.106	0.345	1
Precision	0.800	0.022	0.735	0.936	0.121	0.022
F1	0.042	0.042	0.474	0.191	0.179	0.041

Which recommendation techniques performed best?

Based on F1 measure, we believe that Item-Based Collaborative Filtering (IBCF) is a best performing recommendation system out of five. We used F1 measure because of two reasons:1) to compare a model with high recall and low precision with a model with high precision but low recall 2) to take both false positives and false negatives into consideration

[1] 2,000 items were used for Item-Based Collaborative Filtering due to computation limits.
[2] Combination of User-Based and Cluster-Based

Advantages and Disadvantages of the best recommendation technique

Item-Based Collaborative Filtering (IBCF)

Pros	Cons
• Item features are not required • Good Enough Results • Relevant Recommendations • Stable when there are changes in the ratings matrix • Allows the recommendations to be given alongside an explanation	• Sparsity (it requires ratings of users on certain items) • New Item Issue (it cannot recommend an item that has not been previously rated)

Improvements

Throughout the report, we have seen different recommendation systems all which have shown difficulty in providing recommendations that have high accuracy and precision at the same time. Although evaluation metrics vary from business to business depending on the objective, in this context we used F1 to gauge the recommender systems performance and IBCF came first hence improvement suggestions will only cover this resys.

One of the easiest ways to improve IBCF is deal with the issue of sensitivity by manipulating sensitive parameters such as nearest neighbors, ratio of the training and test sets and the similarity measure used. There's no bench mark for the ideal figures for the sensitive parameters and its entirely based on analyst gut feeling and experience to manipulate to achieve highest performance.

Another method of improving IBCF is improving scalability through the creation of hybrid feature through the Genre Interestingness Measure (GIM) which utilizes user created tags.

Conclusion

In order to find the best performing recommendation system for LastFM users,we explored 5 different recommendation systems: UBCF, IBCF, Cluster-based, Content-based and Hybrid. Every recommendation system has its own pros and cons. However, based on our F1 score, we recommend IBCF for a best performing recommendation system. Since IBCF requires ratings from the users regarding songs they listened, the results may rely heavily on user engagement.