Report on Recommendation System Built for Last.fm

Bannem Anatol Clement Junior, Kalapati Devakanth, Kusterer Kurt Maxwell,

Razo Eduardo Alfonso, Xie Chenxin

1. Transformation of the data

1.1 User_artist file

The user_artist.dat file that contains the user-artist pairings with the listen counts was loaded. Then we obtained the calculation of the number of users who have listened to each artist listed in the file. The result allowed us to filter the user-artists data to the top 1000 artists according to the number of listeners.

Then the data was truncated by removing any artists that don't have a corresponding entry within the artists.dat file. The analysis indicates that 146 of the top 1000 artists do not have a corresponding entry within the artists.dat file. Since 146 artists were removed, this left us with 854 artists to be retained within our user_artists data.

1.2 Creation of the User-Artist matrix

Since it was necessary to create an item-based and a user-based collaborative filtering to generate recommendations to Last.fm users, we converted the user-artists data from the step above to a user-item matrix. This was done by using R's spread () function. Since the first column of the resulting matrix contains user ID's, that column was converted to index before using R's as.matrix () function to convert the data frame to a matrix object.

The row names in the matrix represent the unique Last.fm user ID's we extracted from the user_artists.dat file while the column names represent the unique Last.fm artist ID's. The values of each cell represent the number of times a user has listened to a given artist.

2. Collaborative filtering

2.1 IBCF (Item based collaborative filtering)

The function used to compute the predictions is based on item-based collaborative filtering and consists of the following components:

- Similarity matrix: Generation of the similarity matrix using "Pearson correlation" and calculation of this similarity considering train data set as main input.
- Nearest Neighbors: Calculation of Nearest Neighbors using the similarity matrix as main input.
- Prediction: Computation of the predictions using the test data set as main input.

In addition, the function allows configuring of different parameters like the number of recommendations (N) and the number of Nearest Neighbors (NN).

The output is a matrix with recommendations for all the users and a data frame showing the top recommendations also for each of the users.

2.2 UBCF (User based collaborative filtering)

The function used to compute the predictions based on user-based collaborative filtering consists of the following components:

- Similarity matrix: Generation of the similarity matrix using "Pearson correlation" and calculation of this similarity considering train data set as main input.
- Nearest Neighbors: Calculation of Nearest Neighbors using the similarity matrix as main input.
- Prediction: Computation of the predictions using the test data set as main input.

In addition, the function allows configuring different parameters like the number of recommendations (N) and the number of Nearest Neighbors (NN).

The output is a matrix with recommendations for all the users and a data frame showing the top recommendations also for each of the users.

3. Function evaluation

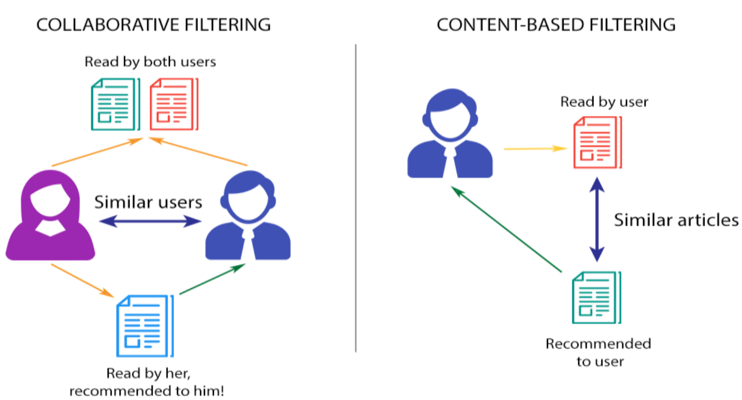
For both functions IBCF and UBCF, the metric used to compute the evaluation was Mean Absolute Error (MAE), utilizing the matrix containing the predictions and the test set, this means prediction vs real information

F1 score is often for classification. This is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric. In most real-life classification problems, imbalanced class distribution exists and thus F1-score is a better metric to evaluate our model on.

4. Argumentation

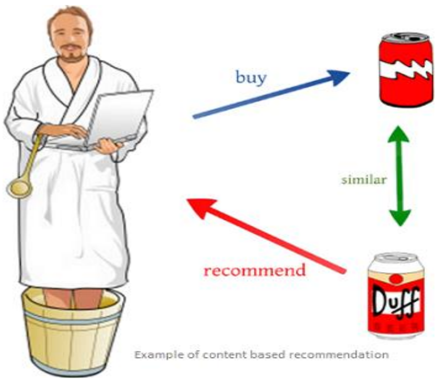
4.1 Why we used specific recommendation techniques

Recommender systems were originally built to target a specific group or user or a single user by recommending him a product based on his purchase history. They are typically 2 types of recommendation techniques which we can use i.e., collaborative filtering and content based filtering.



Content based filtering

Content based filtering is used when we want to recommend the user based on his purchase history, and trying to figure out similarities between the products he bought and the other products. The system finds out similarities between the products based on the context, description and other data point. For example if you liked the movie “300” then we can recommend movie movies of gerard buttler or movies with the genre “War” or “Action”.



It constructs and then compares user profile and item profile using the content of shared attribue space.

We used this to get the similar artists which the user might like.

Collaborative Filtering

This technique is used when you want to recommend a product based on similar interests. The recommendations are solely based on the user’s behavior. If user A likes “John Wick”, “Harry Potter : chamber of secrets”, “Terminator” and user B likes “John Wick”,”Harry Potter prisoner of askaban”, “Terminator”. Now there is huge chance that user A may like ”,”Harry Potter prisoner of askaban” and User B may like “Harry Potter : chamber of secrets”. Other users behavior and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known.

They are 2 types of collaborative filtering Techniques:

1) User based Collaborative Filtering;
2) Item based Collaborative Filtering.

User based Collaborative Filtering: In this approach, users of the same neighborhood, with whom he/she shares common preferences, base Items that are recommended to a user on an evaluation of items. If the article was positively rated by the community, it will be recommended to the user. In the user-based approach, articles which are already rated by a user, play an important role in searching for a group that shares appreciations with him/her.

Item based Collaborative Filtering: Referring to the fact that the taste of users remains constant or change very slightly, similar articles build neighborhoods based on appreciations of users. Afterwards, the system generates recommendations with articles in the neighborhood that a user might prefer.

4.2 Why we used the specific evaluation metrics

4.2.1 MAE

Mean absolute error measures the average magnitude of error in a set of predictions. It's the average over the test sample of absolute differences between the predictions and the actual observation.

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

Since we are working with predictions, we use MAE because this is an accuracy metric.

4.2.2 F1

F1 is combination of precision and recall. We use this evaluation metric in order to evaluate a recommender system that propose a list of top N items that a user is expected to buy.

Concerning top-N recommendation, important metrics are recall-precision related measures. Data is first divided in a training set and a test set. The algorithm runs on the training set, giving a list of recommended items. The concept of 'hit set' is considered, containing only the recommended (top-N) items that are also in the test set.

Recall = Size of hit set / Size of test set
Precision = Size of hit set / N

F1 = (2 * *recall* * *precision*) / (*recall* + *precision*)

Computing F1 for each user and then taking the average gives the score of the top-N recommendation list.

4.3 Why we use the specific hybridization techniques

The strength of both implicit and explicit feedback can be combined in a hybrid model. In this way, we minimize the weaknesses and get the best performing system. This method uses a combination of different recommendation techniques in order to improve the system optimization. The idea here is that combining algorithms provides for a more accurate and effective recommendation model. In this way one algorithm weaknesses and covered by another algorithm. Ultimately, this would be the reason behind selecting a hybrid model. (F.O. Isinkaye, 2015)

In this case, we have used mixed hybridization, combing the recommendations results from two different techniques at the same time instead of having just one recommendation per item. (F.O. Isinkaye, 2015)

5. Discussion

5.1 Compare the recommendation above

In step 3, we have computed three type of collaborative filtering recommendation systems: user-based, item-based, cluster based and evaluated them with MAE and F1 score. We have computed the content-based recommendation system and evaluated it with MAE and F1 score. At last, we computed two different hybrid recommendation systems and evaluated them with MAE.

Recommendation algorithm		MAE (Mean Absolute Error)	F1 Score
Collaborative Filtering	User-based	1.259513	0.0299148
	Item-based	Item10: 5.002119 Item15: 5.070134	Item10: 0.06600496 Item15: 0.04436762
	Cluster-based	/	0.684
Content-based		4.977372	0.03576021
Hybrid	Content-based & Item-based	5.516052	/
	Content-based & User-based	2.361167	/

MAE is negatively-oriented scores, which means lower values are better. F1 score reaches its best value at 1 and worst value at 0. As for user-based collaborative filtering, MAE is relatively small and F1 score is very close to 1. It indicates that user-based collaborative filtering is making a high accuracy prediction.

Based on the recommendation systems' performance above, we believe the hybrid recommendation system combing content-based and user based collaborative filtering is the best one among the recommendation systems we have evaluated above.

Different algorithms have their strengths and weakness. For example, collaborative filtering works well when there are lots of users & items. Content-based filters often work without much user-item interaction data. In practice, hybrid techniques are used to take advantages of different recommender systems to produce better recommendations.

5.2 Advantages and disadvantages of the best recommendation system

Hybrid recommender systems combine both collaborative and content information. Depending on the hybridization approach, different types of systems can be found. There are two approaches' that can be taken, one method builds two different systems and combines the output of the two. The other approach consist of combining both the content and collaborative features. (Luis M.Campos, 2010)

The strengths of the hybrid models:

It has been empirically proven that combining both content and collaborative information helps to improve the accuracy of the model. This type of model is versatile, it can work on both content based or collaborative filtering, and it can be applied to a range of different tasks. (Luis M.Campos, 2010)

The limitations of hybrid models:

Problems such as data sparseness do exist and ideally, real time ranking should be done. (Luis M.Campos, 2010)

Areas for further research would include:

- Design of new feature selection methods
- Incorporation of relationships between features

References

F.O. Isinkaye, Y. F. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 261 - 273.

Luis M.Campos, J. M.-l.-M. (2010). Combining content-based and collaborative recommendations:A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 15.