,

Anime Recommender System

Amita Banchhor
Department of CSE
Texas A&M University
College Station, TX, USA
amitabanchhor@tamu.edu

Amulya Agarwal
Department of ECE
Texas A&M University
College Station, TX, USA
amulya123@tamu.edu

Rahul Bhagel
Department of CSE
Texas A&M University
College Station, TX, USA
rahul2313@tamu.edu

Venkata Satya Kavya Sree Bondapalli
Department of CSE
Texas A&M University
College Station, TX, USA
kavyasree.bvs@tamu.edu

ABSTRACT

Recommender systems utilize the knowledge discovery techniques to recommend new items to users and thus, are used in e-commerce platforms like Amazon and Netflix. Matrix Factorization is the traditional method, yet very effective, for collaborative filtering. However, relying just on the rating might not be sufficient to capture the user relation and interactions between the latent features and exploration of new features might yield a better recommendation. In this paper, we employ a key feature, number of episodes and item's rating by individual users to perform the user-user collaborative filtering. Empirical results show that when we combine the rating cosine similarity along with the cosine similarity calculated on average number of episodes yields an improvement in RMSE by 0.2-0.7 points when compared to current effective methods like Neural collaborative filtering and traditional matrix factorization.

KEYWORDS

Datasets, Neural Networks, Machine Learning, Collaborative Filtering, Recommender systems, Matrix Factorization

1 INTRODUCTION

Recommending new items to a user has been widely recognized as an important research topic in data mining area [1] and recommender systems have been extensively applied in various applications across different domains to recommend items such as book, movie, music, friend, scientific article, video, and restaurant [2]. Many e-commerce services such as Amazon and Netflix mainly depend on recommender systems to increase their profits by selling what consumers are interested in against overloaded information of products.

Collaborative filtering (CF) has been successfully adopted among diverse recommendation strategies due to its high quality performance and domain free property. For a query user, CF recommends items preferred by other users who present similar rating patterns to the query user. This strategy does not require domain knowledge for recommendation, since it relies on only user history such as item ratings or previous transactions. However, the simple user-item rating matrix is not always effective to capture the relations between the users. In order to address the problem it is vital to consider implicit and explicit feedback from the user to provide more information and generate a better recommendations.

In this paper, we propose a novel framework which integrates the user-item rating matrix along with other latent feature specifically the number of episodes. The rating matrix is the traditional method which yields recommendation based on cosine similarity while the number of episodes provide additional information if the two users are related to each other. For instance, in Figure 1, we can see that when the average number of episodes for the users 44 and 53 is high they tend to give bad rating, less than 5, to those TV series. Hence, exploiting such relations will definitely yield a better performance. Further, we introduce a weighting mechanism to combine the two matrixes.

Further, we analyze different item-based recommendation generation algorithms. We look into different techniques for computing item-item similarities (e.g., item-item correlation vs. cosine similarities between item vectors) and different techniques for obtaining recommendations from them. Results on Anime data, shows that the proposed architecture beats the sophisticated Neural Collaborative Filtering and Traditional Matrix Factorization by 0.2-0.7 points in RMSE.

2 RELATED WORK

The most common approach to CF is based on neighborhood models. Its original form, which was shared by virtually all earlier CF systems, is user-oriented. Such user-oriented methods estimate unknown ratings based on recorded ratings of like minded users. Later, an analogous item-oriented approach. became popular. In those methods, a rating is estimated using known ratings made by the same user on similar items. Better scalability and improved accuracy make the item-oriented approach more favorable in many cases.

In order to address the interaction of the user and items, there have been recommender systems which have integrated them into much more sophisticated interaction function. Neural collaborative filtering, uses the deep learning method which finds these interaction function using the non-linearity and integrating the embeddings together and then feeding into complicated neural

,

network consisting of dense layers. Each layer of the Neural Collaborative filtering layers [4] can be customized to discover certain latent structures of user and item interactions. The dimension of the last hidden layer X determines the capability of the model [5].

3 DATA SET

For this project, the Anime Database from https://www.kaggle.com/CooperUnion/anime-recommendations-database on Kaggle has been used. The data set comprises of 73,516 users and 12,294 animes. Because of time complexity and computational restrictions, a subset of data has been used to design the proposed framework with 1000 users and 2000 anime. The focused data has been further split into training and testing data with a ratio of 3:1.

The data been processed from rating.csv and anime.csv files. The rating.csv file contains 3 columns depicting user id, anime id and rating. The rating ranges from 1 to 10. -1 has been assigned if the user has watched it but didn't assign a rating. For ease of processing, the entries containing -1 ratings have been dropped as they were of no significance. The anime.csv file contains details of each anime against anime id like name, genre, type of anime (movie, TV, OVA, etc.). number of episodes and average rating.

Using Figures 1, 2 and 3, the distribution of data can be visualized. It can be noticed how the number of users that have rated less items is very high. Similarly, the number of items that have been rated is very high for less number of ratings. It can be understand that the data is highly sparse from these histograms. Figure 3 shows a distribution of ratings in the data set.

Further data analysis yielded the following conclusions. An item on average has 14.59 ratings. An user on average has rated 21.13 items.

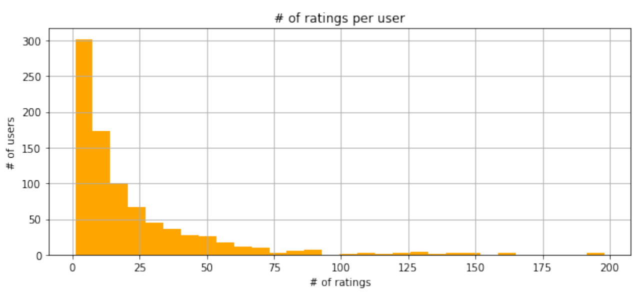

Figure 1: Histogram depicting number of ratings per user

Figure 4 shows a pattern in the dataset samples when we compared the average number of episodes of Animes a user rated good (>5) vs the average number of episodes that user has rated bad (<5).We found that there is a prominent difference in the average and also that different set of users are having similarity in obtaining this difference. This particular pattern and observation led us to use this factor for finding the similarity between users based on this and also to use this factor in the rating prediction.

4 PROPOSED SOLUTION

Matrix Factorization associates the user and items with a vector of latent features. Conventional Matrix Factorization method use

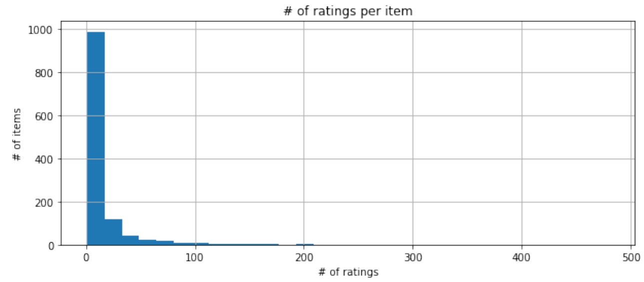

Figure 2: Histogram depicting number of ratings per item

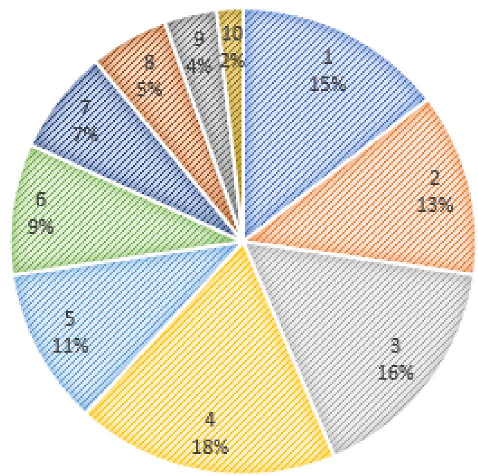

Figure 3: Ratings distribution in the dataset.

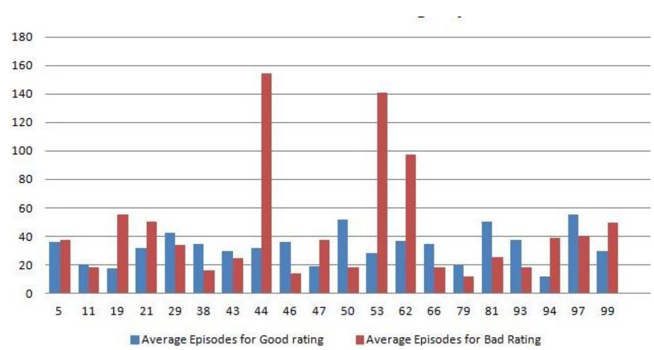

Figure 4: Distribution based on average episodes.

these vectors, assuming each dimension to be independent of each other, to calculate the cosine similarity and linearly combining them. Since, the data set contained extra information about the anime and TV series with the users, we used an extra feature, namely, the number of episodes rated as bad and good by every user. As we can see from Figure 1, that the users with very high number of episodes have rated that TV series to be bad and hence are very similar to each other.

,

Table 1: The MAE and RMSE for the baselines and proposed solution

Design	MAE	RMSE
Matrix Factorization	0.93	2.15
Neural Model	1.02	2.00
Rating Cosine Similarity	1.08	1.50
Cosine Similarity(Rating + # episodes) Model 1	1.51	2.43
Cosine Similarity (0.8* rating + 0.2*Episode count) Model 2	1.24	1.83
Cosine Similarity (rating X # episodes) Model 3	1.02	1.39
Cosine Rating Similarity with closeness of episodes in prediction Model 4	1.18	1.66

4.1 Predict Rating using Similarity of Episodes

The utilization of similarity of users with the number of episodes is two fold. First, it gives the recommender system an advantage to recommend those items to users which have been rated good on the basis of the number of episodes which can be of different genre. Secondly, the system exploits the user-user similarity based on the number of episodes. Hence, if the number of episodes is high and two users are similar with bad rating, such anime will not be recommended to the users.

To calculate episode based similarity we first calculate the average number of episodes per user for good rated episodes and bad rated episodes. For an anime to be good rated we have considered rating of 5 or more as good and for bad less than 5. Then we create a user x episode matrix which has one column for average no of episodes for good rated anime one for bad rated animes and one overall average.We then first calculate the cosine similarity for every users.

After this first we tried to incorporate this similarity by take a summation of a fraction of cosine similarity based on rating and based on episodes. The range of k in the below formula is from 0 to 1.The value we tried was 0.8.

$$S_o = k \times S_r + (1 - k) \times S_e$$

In order to incorporate the episode based similarity, we calculated the cosine similarity based on the number of episode. Then, we used the weighting mechanism to calculate the overall user-user similarity. After experimenting, we found that when the similarities are combined together by dot product, we yield a better performance.

$$S_o = S_r \times S_e$$

where S_o is the overall similarity, S_r is the rating cosine similarity and S_e is the episode cosine similarity This might be because when we multiply the rating and episode based similarity, the overall similarity is more pronounced for users which are similar in both aspects and hence the RMSE and MAE is improved.

4.2 Closeness of number of episode in prediction

The other way in which we tried to use the average episodes for a user that they rated good/bad is by using this average episode in the prediction formula. The formula depends on the closeness of number of episode of Anime for which we are predicting rating with the average number of episode that the user has good/bad. Out of the various formulae for predicting rating [6], he base formula used for prediction is following :

$$r_{c,s} = \bar{r}_c + k \sum_{c' \in \hat{C}} sim(c, c') \times (r_{c',s} - \bar{r}_{c'})$$

where $r_{c,s}$ is the rating to be predicted for user c for item (Anime) s.

c' is the user belonging to set C that has all neighbours of c which have rated the item s. $\overline{r_c}$ and $\overline{r_{c'}}$ are the average rating of users c and c'.

If the number of episode is closer to the average number of good rated Animes, the prediction formula we used was :

$$r'_{c,s} = r_{c,s} + k \times |num\_of\_episode - avg\_good\_epi| / average\_good\_epi$$

whereas, if the number of episode is closer to the average number of bad rated Animes, the prediction formula used was :

$$r'_{c,s} = r_{c,s} - k \times |num\_of\_episode - avg\_bad\_epi| / average\_bad\_epi$$

where k is taken as a constant that shows the factor by which episode count similarity is affecting the rating.For our dataset and model, we found the best result with k=0.2.

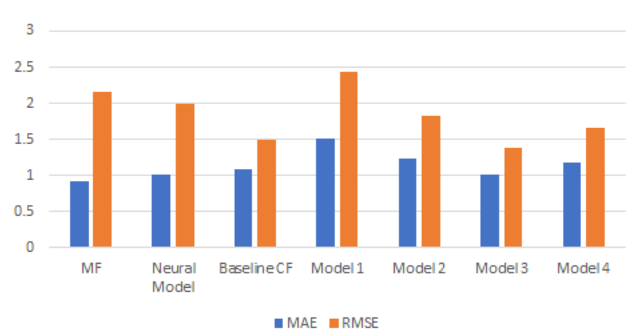

Figure 5: MAE and RMSE scores for all models tried.

5 RESULTS AND DISCUSSIONS

To evaluate the performance, we used RMSE (root mean square error) and MAE (mean absolute error) on the test data for different users' ratings. With different approaches that we tried, we found the RMSE and MAE and will rate the recommendations of system inversely with the scores on test data. These metrics have been used by the models previously and hence are being used for discussion as well.

As we can see from Table 1 and Figure 5, the proposed solution (Model 3) has the best overall performance with a balance of RMSE and MAE. It is very interesting to note that RMSE gives higher weights to larger errors. Hence, in the other baselines models we can see that the RMSE is very high. For instance, the predictions based on the simple similarity matrix and Neural Collaborative Filtering might recommend completely off topic genre anime and still have lower MAE as they don't take into account the number of episode which is relevant for recommending the TV series. But, when we take into account the number of episodes similarity matrix the RMSE decreases which implies that the recommendations don't have large errors and hence the overall system is better.

6 CONCLUSIONS

We understood the ideology behind building a recommendation system and predicted ratings starting with base models. We worked in analysis of dataset and finding patterns in the samples specially the number of episodes which we then chose to take further in our model implementations and prediction formula. The use of this feature in user similarity calculation and rating prediction helped us to understand how to choose a feature and to analyze its impact. Also the feature selection and then ways of using it helped us to see how we can incorporate neural model or other random factor attempt methods in base collaborative filtering to play around with the impact of features.

7 FUTURE WORK

We can extend our work further to try more combinations of calculated similarity and incorporate them into neural network for collaborative filtering. Another major challenge we faced was the cold start problem which could be overcome with content based approach. The content based approach can be designed in such a way that it could also use more features of animes like genre of anime and type of anime to evaluate if it could recommend better. Current work just deals with few combinations of rating and episode similarity. We would further want to use neural models [3] for factor tuning of contribution of these similarities.

REFERENCES

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD '98)*. ACM, New York, NY, USA, 94–105. https://doi.org/10.1145/276304.276314

[2] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD Rec.* 27, 2 (June 1998), 94–105. https://doi.org/10.1145/276305.276314

[3] Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition* 48 (1993), 71–99.

[4] Jonathan L. Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*. Association for Computing Machinery, Inc, 230–237. https://doi.org/10.1145/312624.312682

[5] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *In IEEE International Conference on Data Mining (ICDM 2008*. 263–272.

[6] Amar Saric, Mirsad Hadzikadic, and David Wilson. 2009. Alternative Formulas for Rating Prediction Using Collaborative Filtering. In *Foundations of Intelligent Systems*, Jan Rauch, Zbigniew W. Raś, Petr Berka, and Tapio Elomaa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 301–310.