International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.173.02

Collaborative Movie Recommendation usingGFM

Bharti Sharma
Assistant Professor, CSE Department
MSIT,New Delhi, India

Abstract:The authors of this paper propose a methodology to predict movie ratings a user might give to a particular movie based upon feature mapping of genres. The authors map each genre to a unique prime number and calculate feature values for movies based upon the genres to which they belong. This feature value is then used to identify probabilistic values of rating that a user might give. The methodology is evaluated over two movielens datasets, one of 100 K ratings and the other of 1 million ratings. The evaluation metrics that used are RMSE (Root Mean Square Error, MAE (Mean Absolute Error), and Precision. The results show that the proposed methodology is able to predict the movie ratings with higher accuracy than the state-of- the-art algorithms for movieprediction s.

Keywords: Recommendation Systems, Rating Prediction, Collaborative Filtering, Movielens.

I. INTRODUCTION

With the explosive rise of information on the internet, finding pertinent information from the internet has become a universal problem. The problem of choosing a single product from millions of products is being experienced by millions of users. Recommendation system helps to solve the user E- commerce problem by selecting the items which are of interest of the user from available stock of millions of items [1]. A recommender system (RS) is used to collect the useful information about the items which interest the user and recommends those particular items or products [2]. Today, a RS is almost being used in every E-commerce website which assists lot of users.E-commercesitessuchasAmazon[3]for Electronic products, books and many other products, Netflix and Movielens [4] for movies, Jester [5] for jokes and Entrée for restaurants, use RS in assisting the customer. The results are used for both users as well as an E-commerce organizations [6] i.e. apart from assisting the customers in selecting the preferred items, a RS helps to sell more products and thus increases the overall revenue of an organization. On the basis of how any recommendation is done, the recommendation system can be divided into three categories: collaborative filtering, content-based filtering, and hybrid filtering. Collaborative filtering is the most successful techniques and is used widely for recommending an item. It checks the ratings of the other users in the system and then based on that it recommends an item to a particular user. Content-based filtering checks the characteristics of the similar items from the past history and performs prediction, for example if a user likes action movies it will recommend all movies that have been categorized as “Action movies”. Hybrid filtering is achieved by the combination of both collaborative and content-based filtering techniques to improve performance

The memory-based collaborative filtering frame work for recommendation system is shown in Figure1. The framework shows the E-commerce sites are used by millions of users and the reviews regarding the products are stored in the server. The data is collected and preprocessed into the required format (user rating matrix) by the E-commerce organizations. This similarity rating is used between the users by making use of different algorithms such as cosine similarity, Pearson correlation coefficient, Jaccard’s similarity and adjusted cosine similarity etc. The active user is recommended the top-N predicted items.

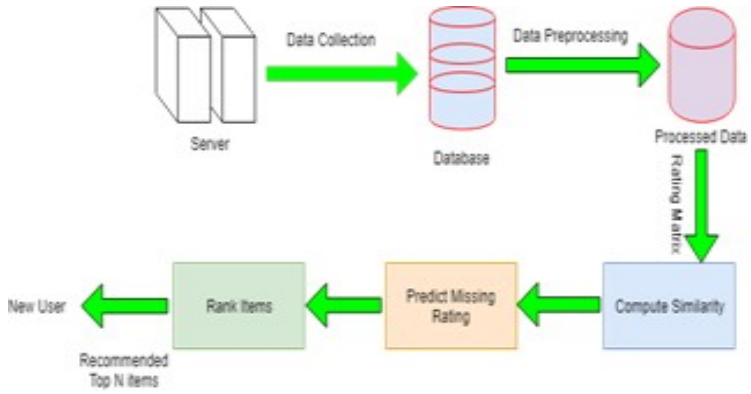


Fig. 1: Framework of memory based collaborative filtering based recommendation system

In Collaborative filtering (CF) technique, the ratings/opinions of the other users [7] are used to recommend an item to a particular user. A CF system builds a database of user preferences for items to perform recommendation. The users with similar interests/preferences are found by the system by computing similarities among the user profiles [8] and forms a group of similar users called neighborhood. Those products are recommended to the user which are not rated or purchased by him but ratings of his neighbors exist. Predictions or recommendations are performed by collaborative filtering, a prediction is just a numerical value while a recommendation is a set of top N items that are liked by the user the most. The classification two main broad categories of collaborative filtering technique are (a) Memory-based technique (b) Model-based technique[9].

Content-based (CB) filtering recommends those items which have similar features to the items that have already been used by the users in the past. The attribute of the item is analyzed more by the content- based approach to produce recommendations. CB filtering technique is majorly successful in publications, web pages and news recommendation. Personalized profiles of the users are automatically created by the CBF system based on the feedback and the types of items a user likes. The user collected information is compared against the features of the item examined in order to generate useful recommendations.

Hybrid filtering is the merge of two or more recommendation systems which helps in achieving improved performance over content-based filtering and collaborative filtering. To obtain the hybrid filtering System, the CF and CBF techniques can be combined in a distinct way, which may generate different outputs. The process can be defined into seven different ways such as (1) Switching (2) Weighted (3) Mixed approach (4) Cascade(5)
Feature augmentation (6) Feature combination (7) Meta-level.

II. RELATED WORK

According to Gu et al. [10], due to the unprecedented advancement of the users and items in E-commerce site, the problem of data sparsity occurs in the simple collaborative filtering technique. In this technique, firstly the similarity is found between a user and the items and then to find the impact of user and items similarity, a weight controlling method is suggested. So, this method performs better in various situations of data sparsity.

Ji et al. [11] introduced CF algorithm based on matrix factorization, which performs prediction based on user- category matrix and a user- keyword matrix in place of using an individual user-item rating matrix. The real dataset is used for implementing the algorithm and the scalability for new items is good according to theresults.

Koohi et al. [12] Collaborative filtering majorly experienced from high dimensionality and data sparsity problems. In this paper, the author tries to solve these problems using subspace clustering algorithm to find the neighbor user. The author forms the various subsets of a given rated matrix namely Interested (I), Uninterested (U) and Neither Interested Nor Uninterested (NIU). For finding the neighbors of an active user, the three levels of a tree are formed based on these subsets. This method efficiently deals with sparse data.

Lee et al. [13] introduced Predictive Clustering-based Collaborative Filtering (PCCF) This approach tracks the modifications in user preferences and bridges the gap between the dynamic model and static model for solving issues of unstable performance and of reduced coverage.

Kumar et al. [14] introduced a hybrid collaborative filtering which solves the sparsity issues and scalability and provides further incorporate. Recommendations. The approach executes in two phases, in the first phase Case based reasoning (CBR) followed by average filling is used to resolve sparsity and in the second phase clustering into groups by using self-organizing maps which are optimized with a genetic algorithm is used to resolve scalability.

Kim et al. [15] introduced a recommender system which uses GA K-means clustering for online shopping. This system check the buying behavior of a user and then segment the online shopper accordingly. GA helps in resolving the problem of local optima which is found in K-means clustering and provides an efficient method of finding the related groups.

III. PROPOSED METHODOLOGY

The proposed methodology is divided into three parts: (i) feature generation, (ii) training, and (iii) prediction. The methodology uses prime numbers as to identify each movie type and genre uniquely. Each genre type from the movie dataset is mapped to a unique prime number which makes it easier to identify the genre that might be present in a movie. We calculate and rather generate feature for a movie based on the genres that it belongs to. The feature is a product of all the mapped prime numbers to the genres that a movie might have. That is, for example if a movie M has genre G_1, G_2, and G_3, and; G_1 is mapped with prime number P_1, G_2 is mapped with prime number P_2, and G_3 is mapped with prime number P_3; then feature value of movie is $P_1*P_2*P_3$. This gives a unique number to each movie and movies with same feature are and can be clubbed as same in terms of their genre type.

GENERATE_FEATURE

```
Map each genre to a unique prime number forEach movie Mi
    forEach genre Gj in Mi
        Feature = Feature * genreMap[Gi] EndOfforEach
EndOfforEach
```

TRAINING

```
Calculate average Rating forEach Movie forEach User Ui
    forEach Movie Mj in Ui
        forEachfeatureFactorfk in MjratingArray[k] += Mj.ratingcountArray[k]++
        EndOfforEachEndOfforEach
EndOfforEach
```

PREDICTION

```
forEach UserUi
  forEach Movie Mj in Ui
    forEachfeatureFactorfk in Mj predictedValue1 = USER_PREDICTED() if Collaborative_Rating
                predictedValue2 =
                        Collaborative_Rating
        else
                predictedValue2 = predictedValue1 EndOfforEach
         endOfforEachendOfforEach
```

After generation of movie feature value, during training and testing, the average rating of a movie is calculated based upon the input of the other users that have watched it. And also is calculated the average rating that a particular user gives to a particular feature type. This data helps in taking into the account the personal preference of a user and also that of the other users in general, leading to final prediction that is based upon not only the history and likeness of a user but also the community as whole, because our decisions do reflect the general choice and behaviour of the people around us. And in movie ratings, we tend to watch movies that have been given higher ratings by other users and also that match ourpreference.

To calculate the predicted value of rating that a user might give to a movie, two parameters are used: *predictedValue1*, the rating that a user might give to a movie if only his personal preference is considered, based upon his history of movie watching *predictedValue2*, the average rating that other users have given to a particular movie, this incorporates collaborative sense involved in decision making.

To further determine how each of these parameters affects the final rating two variables, θ_1 and θ_2 are used. The sum of θ_1 and θ_2 is always 1 and by varying the values of θ_1 and θ_2, the difference between the predicted rating and the actual rating as given by the user can be minimized i.e. such a value of θ_1 and θ_2 was such determined such that our predicted rating was closest to that of the actual rating as given by theuser.

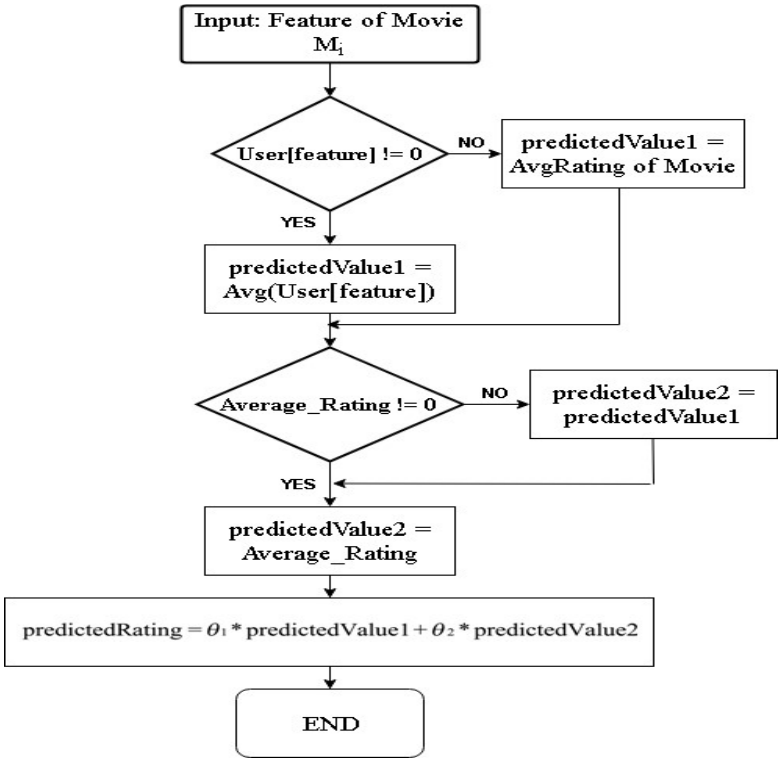


Fig 2: Flow Chart of Prediction Experiment andResults

IV. EXPERIMENT AND RESULTS

For experimentation and testing purpose of our algorithm two movie-lens datasets were utilized. The values of θ_1 and $\theta_2$were varied between the range of 0 to 1 such that θ_1 and $\theta_2$was always equal to 1, to determine the best predicted rating value as per the formula,

$$\text{P}erdictionRating = \theta_1 * PerdictedValue1 + \theta_2 * PerdictedValue2$$

The values θ_1 and $\theta_2$also help to determine how and to what extent individual preference and the average rating that a movie gets affect the final rating that a user is bound to give to a movie.

A. Datasets

Movielens 100K- 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users. This dataset is perfect for early training and testing purposes, with a collective data of approximately 700 users with their respective ratings to particular movies, it forms a basic database abundant enough to help with testing of system on small scale. Also, with presence of 9000 movies with their average rating, it also helps in scaling quite a big movie related data. Ratings are in the range of 1-5.

Movielens 1M- Stable benchmark dataset. 1 million ratings from 6000 users on 4000 movies. This dataset is a standard benchmark for testing and training purposes with data of approximately 6000 user and their respective ratings to the movies they have watched. It compromises of 1 million ratings by these 6000 users for about 4000 movies. Ratings are in the range of 1-5.

B. TestingMeasures

i. RMSE - Root Mean SquareValue

Root Mean Square Error (RMSE) is used to measure the differences between the values as predicted by a methodology and the values that are actually observed. It is the square root of the average of squared errors. In our methodology the error is was the difference between predicted rating and the actual rating as given by the user.

$$RMSE = \sqrt{\sum^{N} (p_i - A_i)} / N$$

where, P_i is the predicted rating and A_i is the actual rating as given by the user, N is the total number of rating predictions

ii. MAE - Mean AbsoluteValue

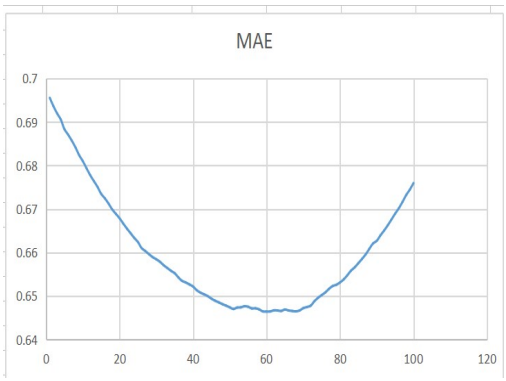
In statistics, mean absolute error (MAE) is a measure of difference between two continuous variables. The MAE is the average of all absolute errors and absolute error is the amount of error in your measurements. It is the difference between the measured value and "true"value.

$$MAE = \frac{1}{N}\sum^{n} |P_i - A_i|$$

iii. Precision

Precision in general recommender systems is the amount of relevant recommendations found in the retrieved set of recommendations, for our experimentation it is the exact match of ratings as predicted by us and that given by the user, that is all those rating where P_i-A_i equal to zero.

$$\text{P}erdictionRating = \theta_1 * PerdictedValue1 + \theta_2 * PerdictedValue2$$



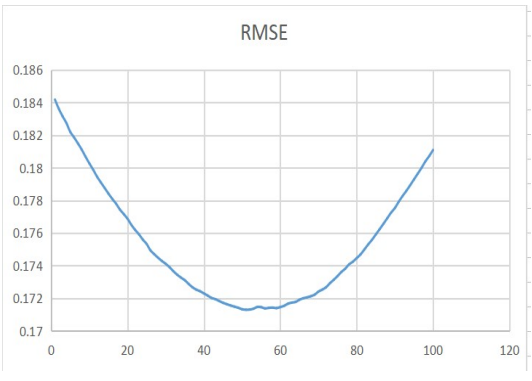


Fig 3 : Result of MAE with **Movielens 100K**

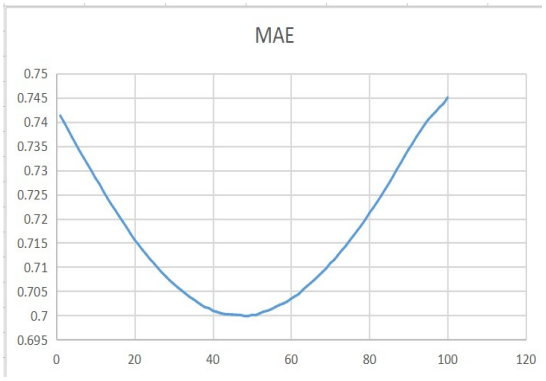


Fig 4: Result of RMSE with **Movielens 100K**

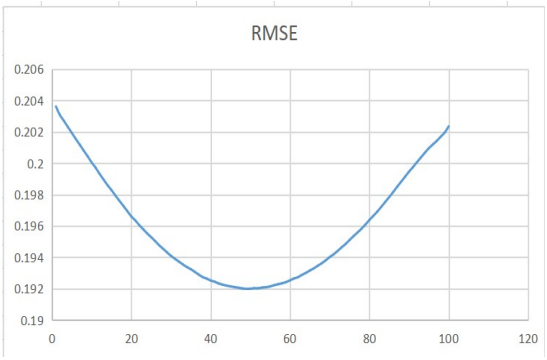
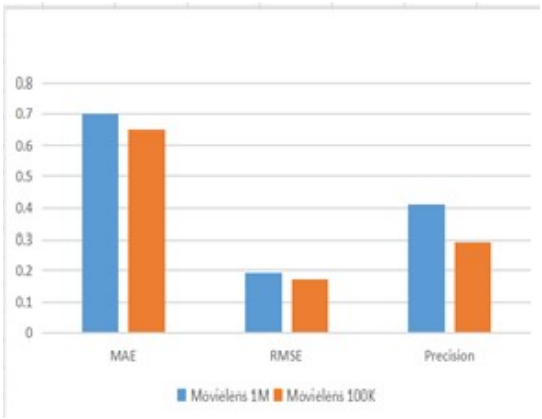


Fig 5: Result of MAE with Movielens 1M

Fig 6: Result of RMSE with Movielens1M

The findings indicate that as the size of the data increases, the proposed solution learns better and can perform with better accuracy and precision. Figure 8 presents the comparison of precision value of proposed solution with that of state-of-the-art algorithms [16].



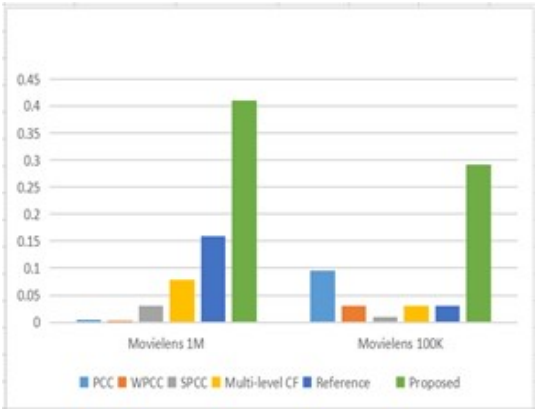


Fig 7: Performance of the proposedsolution

Fig 8: Comparison with state-of-the-art

V. CONCLUSION AND FUTURE WORK

The system could predict the ratings with great efficiency and good precision, with varied and huge dataset the performance was found to be better. Our precision in predicting the ratings were 29% for Movielens 100K dataset and 41% for Movielens 1M dataset. With better predictions of ratings we can generate better recommendation results for users.

In future we aim to make the system more dynamic by using clustering techniques to find the features on its own. And also be able to find features and other relations even if no definite input is given for the feature calculation. We only have till now predicted it for Movielens Dataset as the feature or the genres were predefined, to make it more dynamic we are planning to use clustering techniques to determine the features of a random dataset. To generate much better recommended results it is important to analyze the difference of the ratedvalue.

REFERENCES

[1] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of*

the 2nd ACM conference on Electronic commerce, 2000, pp.158–167.

[3] G.Linden,B.Smith,andJ.York,"Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[4] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "MovieLens unplugged: experiences with an occasionally connected recommender system," in *Proceedings of the 8th international conference on Intelligent user interfaces*, 2003, pp.263–266.

[5] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retr. Boston.*, vol. 4, no. 2, pp. 133–151,2001.

[6] C. M. Rodrigues, S. Rathi, and G. Patil, "An efficient system using item & user-based CF techniques to improve recommendation," in *Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on*, 2016, pp.569–574.

[7] B. Kumar and N. Sharma, "Approaches, Issues and Challenges in Recommender Systems: A Systematic Review," *Indian J. Sci. Technol.*, vol. 9, no. 47,2016.

[8] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Informatics J.*, vol. 16, no. 3, pp. 261–273, 2015.

[9] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: Issues, challenges, and research opportunities," in *Information Science and Applications (ICISA) 2016*, Springer, 2016, pp.1179–1189.

[10] L. Gu, P. Yang, and Y. Dong, "An dynamic- weighted collaborative filtering approach to address sparsity and adaptivity issues," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp.3044–3050.

[11] K. Ji and H. Shen, "Using Category and Keyword for Personalized Recommendation: A Scalable Collaborative Filtering Algorithm," in *Parallel Architectures, Algorithms and Programming (PAAP), 2014 Sixth International Symposium on*, 2014, pp. 197–202.

[12] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of Collaborative Filtering," *Expert Syst. Appl.*, vol. 83, pp. 30–39, 2017.

[13] O.-J. Lee, J. J. Jung, and Y. Eunsoon, "Predictive clustering for performance stability in collaborative filtering techniques," in Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on, 2015, pp. 48–55.

[14] N. P. Kumar and Z. Fan, "Hybrid user-item based collaborative filtering," ProcediaComput. Sci., vol. 60, pp. 1453–1461, 2015.

[15] K. Kim and H. Ahn, "A recommender system using GA K-means clustering in an online shopping market," Expert Syst. Appl., vol. 34, no. 2, pp. 1200–1209, 2008.

[16] R. Katarya and O. P. Verma, "Effectual recommendations using artificial algae algorithm and fuzzy c-mean," Swarm Evol. Comput., vol. 36, pp. 52–61, 2017