2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), 27-28 February, Dhaka

2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD) | 978-1-6654-1460-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICT4SD50815.2021.9397027

A Hybrid Collaborative Recommendation System Based On Matrix Factorization And Deep Neural Network

Md Rafidul Islam Sarker, Abdul Matin
Department of Electrical and Computer Engineering
Rajshahi University of Engineering and Technology
Rajshahi-6204, Bangladesh
Email: rafidulsarker@gmail.com, ammuaj.cseruet@gmail.com

***Abstract*— The paper explores a modified recommender system that is established based on the combination of matrix factorization and deep neural network that work on the implicit feedbacks of users and also auxiliary information of both users and items. Recent works show the effectiveness of deep neural network on recommendation systems. Proposed models aim at discovering additional relationships by using auxiliary information to explore the internal relationship between users and also the relationships of items among themselves. Experiments show 0.5556 and 0.8036 in NDCG and HR with the model which is an improvement compared to other popular collaborative filtering methods.**

Keywords*— *Collaborative filtering, Deep Neural Network, Hybrid Recommendation, Matrix factorization.

I. INTRODUCTION

The growth of the Internet of Things and the surge in web development produces roughly 2.5 quintillion bytes of data every day, with Ninety percent of the data produced in the last two years [1]. This vast amount of data available online creates information overload problems which resulted in incorrect decisions. This has contributed to the development of Recommenders systems. Recommendation Systems plays a major part in the engine of major e-commerce sites such as Facebook, Amazon, and Netflix, etc. Improving customer service and seeking to personalize the service is the cornerstone for all companies.

Recommendation systems (RS) tell consumers what they are most likely to be interested in (movies, music, books, etc.) Good RS can significantly increase the revenues of a company or retain customers. Eighty percent of the films viewed on Netflix, for example, come from the company's RS [2]. One effective way to construct this kind of algorithm is to anticipate how a consumer will rate an individual object. There are two main approaches to addressing this problem: Content based Filtering and Collaborative Filtering.

Content based Filtering uses information from the user or item to calculate a new ranking. User details may include, for example, age, gender, or location info, etc. Item details may include the type of film, a summary, or tags. CF, on the other hand, uses the history or rating interactions of items and users. History of interaction of a user with some items is compared with history of all other users with the items to estimate a new ranking. For example, when a user rates a few books, Collaborative Filtering (CF) would try to approximate the feedback he would have provided to other books by using the reviews of other users. CF is quite often favored to CBF because CF relies only on user feedback, while CBF needs sophisticated engineering on items to perform well [3].

Collaborative filtering is classified primarily into two forms- model based and memory based. CF with memory-based approach, the resemblance is measured using rating information for users or objects, and recommendations are made. However, in model-based collaborative filtering various ML models and mathematical models are used to learn from rating. In CF, the most efficient method is to extract possible latent components from the sparse rating matrix. Latent components of books are likely to fall within the category of books (action, thriller, fantasy, etc.) or other types of literature. Popular latent factorization techniques calculate a rating matrix with low-rank by applying Singular Value Decomposition [4] or Regularized Alternating Least Square algorithm [5]. Aforementioned techniques however cannot detect non-linear and subtle factors.

A hybrid recommender system merges the content and collaborative approach to further increase the accuracy of the predictions. Here, we define a model that uses the combination of matrix factorization to filter collaboratively and deep learning approach to perform collaborative filtering which is enhanced using the auxiliary info of items and users.

The arrangement of the remaining of the paper is such as, Section II describes the related works. Section III illustrates the model we proposed. Section III shows the experiment dataset and performance evaluation metrics. Section V discusses the result analysis. And finally, section VI presents the conclusion and future works.

II. RELATED WORKS

The user's predilection for products can be conveyed with the use of two forms of feedback –implicit feedback and explicit feedback. Feedbacks can be explicit, like ratings, or something like that. However, Implicit feedback is based on interaction, such as clicking, view or buy. Although explicit feedback provides an accurate representation of the interaction between users, the amount of data representing implicit feedbacks far exceeds the quantity of explicit feedback. The earlier works were focused on explicit feedbacks [6,7], Recent works focus on implicit feedbacks [8].

He et al. [9] have suggested a recommender system with implicit feedback as its input. The user interactions are mapped to objects using one hot encoding. A generalized model for collaborative filtering is implemented in this study

978-1-6654-1460-9/21/$31.00 ©2021 IEEE

which uses multilayer perceptron to model high non-linearity levels.

In the recommender system proposed by He, implicit feedbacks of the users to items were first transformed into smaller latent space and delivered into both multilayer perceptron layer and matrix factorization as input whose outputs were then fused to provide predicted ratings on items without ratings based on the interaction function.

In the recommendation system suggested by He, the users' interaction with the items was converted into a latent space and then they were used as input for the matrix factorization and MLP layer. Then they were combined into a single layer to get produced the predicted rating.

Wu et al. [10] proposed a new similarity-based estimation method for CF that shows superior performance compared to the memory-based techniques. But the recommendation systems with deep learning that learn the interaction function shows better performance than any other user similarity-based methods. Both of these methods only use user and item interaction and doesn't use any side information of users and items. So Hybrid methods that use the deep learning system and make use of side information should be explored to gain more accurate predictions.

III. Proposed Method

A. Preparation

Users and items interaction can be simulated using the item ratings given by each user. The interaction can be counted as a favorability of users to specific items. Thus, an existing rating is considered as a positive interaction (as 1) and non-existing ratings can be considered as negative feedback (as 0).

TABLE I. EXAMPLE OF RATING MATRIX

Userid	Movieid	Rating
1	1	4
2	1	3
1	2	1
3	2	5

By using the demographic information of users one can understand the interest of the users and increase the accuracy of prediction. The user's demographic information is shown is the table 2:

TABLE II. EXAMPLE OF USER INFO

User	Age	Gender	Profession
1	15	M	***5***
2	25	F	***3***
3	20	M	***1***

The movies can be categorized according to users' tastes more accordingly by using the information of movies such that their genre and genome. The genre can be directly used using one-hot encode. The most popular 20 genome tags are considered within them top tags with high relevance are used by one-hot encoding.

TABLE III. EXAMPLE OF MOVIE INFO

Movieid	Genre
1	Adventure, Animation, Children, Comedy, Fantasy
2	Adventure, Children, Fantasy

TABLE IV. EXAMPLE OF GENOME INFO

Movieid	Tagid	Relevence
1	5	0.146
1	9	0.262

Multiple items can be rated by one user as the quantity of items is much larger than that of the quantity of users. So, the individual users and individual items are then converted into multi-dimensional space so that they can be converted into a vector to envision their relationship. After that users and items embeddings are articulated for individual users and items. The embeddings that represent the users and item in latent space than can be used input instead of the far bigger user-item rating matrix.

B. Procedure for the proposed method

The architecture of the proposed model for recommendation is illustrated in figure 1 with a schematic diagram. At first, the item embedding and user embedding are created for all then individual users and items found in the dataset. Then the embeddings are directly fed as input to the matrix factorization layer. The users' information is fed directly and item information is fed after converting them into one hot encoding alongside the user & item embedding to hybrid MLP layer. Then the vectors found as a result of the previous two layers are concatenated and fed into the last layer that calculates resultant output. Thus, the model learns all the patterns of user-item interaction to predict the unseen interactions.

The proposed model is divided into three parts as discussed below:

1) Matrix factorization layer:

In the matrix factorization layer, the user and item presented into latent space using the user embedding and item embedding. If the user and item embeddings denoted as Au and Bi and the quantity of latent features is denoted by k then MF can be calculated by following the equation:

$$\hat{Y} = \sum_{k=1}^{K} A_{uk} B_{ik}$$

As the equation performs linearly so this part only discovers then the linear relationship of items and users.

2) Hybrid MLP layer:

The hybrid MLP layer uses hidden layers to discover more intricate relationships that cannot be comprehended linearly. This layer not only takes the item and user embeddings but also takes the item and user demographic info as side info to further understand the interaction function between the users and items. In this, a towering structure is used, as it is seen that using a successive smaller number of nodes in a higher layer tends to capture more abstract features of data [11]. The function this layer performs can be shown in the simplified form if W, A, B represents weight, activation, the bias of each layer and S is the activation of the last layer.

$$\hat{Y} = S(\emptyset(Z_{L-1}))$$

Where, $\emptyset(Z_{L-1}) = A_L(W^T Z_{L-1} + B_L)$ is for layer 0 to L-1

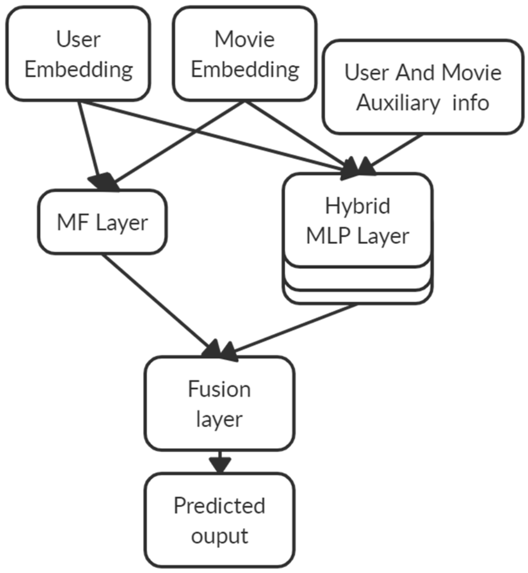

Fig. 1. Structure of the proposed model

3) Fusion MF and Hybrid MLP layer:

To merge the output and get the final result output from the former two-layer taken without their activation are concatenate and fed as input to the final layer. Finally, the sigmoid function is as last activation, and results are found in the range of [0,1]

IV. Experiment

A. Dataset

The dataset that is used in the experimentation of the recommender system is the Movielens dataset. This is dataset is one of the most popular datasets that is use the topic of movie recommendation. Among the sizes of the dataset, the ML1M dataset was used. In this dataset 972849 ratings are listed given by 6040 users for 3126 movies. The Ratings were ranged from 1 to 5. But as experimentation, they were converted to simulate implicit feedback so that ratings are converted to 1.

The dataset was also provided demographic information about users. The information is users' age, gender, profession. The information about the movies such as genre, the genome was included in the dataset for the unique id of the movies. The data set was then randomly shuffled and fragmented into two parts for testing and training purposes.

B. Experiment Setup

The model of the network is composed of three parts matrix factorization layers, a hybrid MLP layer, and a layer to fuse the outputs of the former two layers to get the result.

For the matrix factorization layer, thus embedding size were as 64 for both the users and the movie. Then they were flattened and multiplied. A single dense layer is used to get output during pretraining.

For the hybrid MLP layer, the embedding size for user and items were 32. Then 5 hidden layers with 256, 128, 64, 32, 16 nodes are used in the consecutive layers.

Then the output of these two layers is given two the last layer as input.

As the model deals with implicit feedback, then last activations before taking of the model use sigmoid activation so that output can get in the range of 0 to 1. Other activation of the model used RELU as the activation. Adam or Adaptive moment estimation is used when parts of the model during pretraining and also the full model was trained as it executes minor change on recurrent parameters and bigger update on uncommon parameter.

C. Evaluation metrics

For the evaluation of the performance, NDCG@k and HR@k used.

NDCG stands for Normalized Discounted Cumulative Gain. This metric is used as it brings in normalization. It has appropriate lower and upper bounds so that mean can be calculated all across the values of the scores resulting in a concluding score. NDCG is defined as the ratio of DCG of order recommendation to DCG of ideal order:

$$NDCG = \frac{DCG}{iDCG}$$

Where,

DCG= The recommendation order DCG

iDCG= The ideal order DCG

The value of ratio will at all times be within the range [0,1]. Evaluation of a recommender engine is done, by computing the average of NDCG for the recommendations test users.

HR@k or Hit ratio at k is a method of calculating the number of times you have ‘hits’ in a k-sized list of ranked items. A "hit" could be defined as something that the user has clicked on, purchased, or saved/favorited. The hit rate of the system is determined by dividing the hit counts with the test user counts. It measures how often a removed rating can be recommended. the higher HR means better performance.

V. Result Analysis

For comparing the proposed model, the performance of the other popular models such as MF, MLP, NeuMF is considered here. Also, the performance of the individual parts of the model is shown here. Table III presents the test results of the considered models for the dataset.

TABLE V. PERFORMANCE OF DIFFERENT MODELS

Model	NDCG	HR
MF	0.5329	0.7886
MLP	0.5268	0.7775
NeuMF	0.5461	0.8015
Hybrid MLP	0.5366	0.7912
NHF(proposed)	0.5556	0.8038

The table shows the different model's performance on the dataset on the performance criteria of NDCG and HR. From the table, we can see our model has a better performance compared to the other models. The cause of better performance is that our model can better identify the interaction between the items and users. The other models were only discussions rating but our model uses additional information about users and items to learn not only user-item relation but also item-item and user-user relation to further progress the accuracy of the model.

VI. CONCLUSION AND FUTURE WORKS

This paper delves into the recommendation systems based on deep neural networks for implicit feedback. The architecture of combining the matrix factorization method with neural network gives it a significant improvement over other popular method. It captures the additional patterns learned from using the users and movies demographic info which effectively improves the HR and NDCG. The proposed model provides suggestions that close to the real interaction between users and movies. It can provide a better user experience and sales growth. The models use user rating, user and item demographic information. In this only genre and the genome of the movies considered, the cast and other information are not used. This model is not tested in user or item cold-start situations. In the future, the model will be revised to tackle those situations.

REFERENCES

[1] B. Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read", May 2018, [online] Available:https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-veryone-should-read/#4c8deee860ba.

[2] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[3] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," *Recommender systems handbook*, pp. 73–105, 2011.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[5] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," in *International conference on algorithmic applications in management.* Springer, 2008, pp. 337– 348.

[6] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[8] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 549–558.

[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[10] X. Wu, Y. Huang, and S. Wang, "A new similarity computation method in collaborative filtering based recommendation system," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–5.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.