J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr
Copyright ⓒ 2024 KSII

Design of a Recommendation System for Improving Deep Neural Network Performance☆

Juhyoung Sung[1] Kiwon Kwon[1] Byoungchul Song[1*]

ABSTRACT

There have been emerging many use-cases applying recommendation systems especially in online platform. Although the performance of recommendation systems is affected by a variety of factors, selecting appropriate features is difficult since most of recommendation systems have sparse data. Conventional matrix factorization (MF) method is a basic way to handle with problems in the recommendation systems. However, the MF based scheme cannot reflect non-linearity characteristics well. As deep learning technology has been attracted widely, a deep neural network (DNN) framework based collaborative filtering (CF) was introduced to complement the non-linearity issue. However, there is still a problem related to feature embedding for use as input to the DNN. In this paper, we propose an effective method using singular value decomposition (SVD) based feature embedding for improving the DNN performance of recommendation algorithms. We evaluate the performance of recommendation systems using MovieLens dataset and show the proposed scheme outperforms the existing methods. Moreover, we analyze the performance according to the number of latent features in the proposed algorithm. We expect that the proposed scheme can be applied to the generalized recommendation systems.

☞ keyword : Collaborative Filtering, Deep Neural Network, Matrix Factorization, Recommendation System, Singular Value Decomposition

1. Introduction

Recommendation systems present content to users that they are likely to be interested in. These systems have been widely used in various fields such as multi-media streaming services and online shopping. The performance of the recommendation systems is affected by a variety of factors. Especially, how the recommendation algorithm processes the user preferences and the properties of the content is the most significant factor. Therefore, recommendation systems need to capture the elaborate features [1-4].

Content-based filtering (CBF) and collaborative filtering (CF) are representative algorithms in recommendation systems. CBF algorithms recommend other items to the users based on their previous experiences [5-6]. The algorithms are limited to recommend only similar items and the performance highly depends on the item's feature extraction. CF algorithms use the relationship between items and users. The algorithms can recommend items not similar to the past behavior unlike the CBF. There are several CF schemes and the most fundamental and powerful one is matrix factorization (MF) which decomposes the preference matrix between the users and the items into the user latent matrix and the item latent matrix [7-8]. Then, an interaction between a user and an item is represented as a result of the matrix multiplication of their components. However, MF is hard to reflect non-linearity properties between the users and the items since the matrix multiplication is a simple linear operation.

Recently, a variety of methods have been studied to improve the performance of recommendation algorithms. One of the most promising methods is to use deep learning. Deep learning has been widely applied to the various fields. Several studies have tried to combine recommendation systems and deep learning [9-12]. In [13], deep neural network (DNN) based CF was described. The authors introduced embedding layers to reflect user and item latent factor at the input. However, the embedding layers also need

1 Smart Network Research Center, Korea Electronics Technology Institute (KETI), Seoul, 03924, South Korea.
* Corresponding author (songbc@keti.re.kr)
[Received 15 November 2023, Reviewed 21 November 2023, Accepted 15 December 2023]
☆ This research was supported by Korea Institute of Marine Science & Technology (KIMST) funded by the Ministry of Oceans and Fisheries (20220210).

to be updated during training which can be burdensome as the number of components consisting of recommendation systems increases.

In this paper, we introduce the singular value decomposition (SVD) method to initialize latent features for CF rather than training the embedding layers. Moreover, the proposed method applies DNN structure to train the user preferences and the content characteristics more accurately. We evaluate the performance of the proposed scheme based recommendation system using MovieLens dataset generated on September 26, 2018 [14]. Since the proposed scheme does not have to train the embedding layers, it can reduce the unnecessary parameters while increasing the learning performance of recommendation systems and avoiding from overfitting problems.

The rest of this paper is organized as follows. In section 2, we introduce the proposed recommendation system design while comparing with the typical DNN based CF. In section 3, we evaluate the performance of recommendation systems and analyze the simulation results before concluding this paper in section 4.

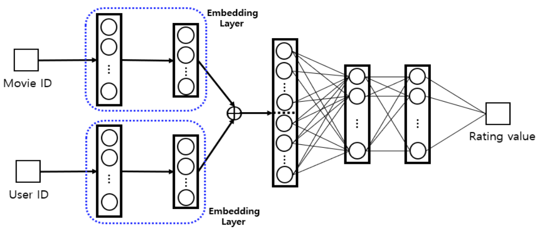
2. System Model

In this section, we describe the MovieLens dataset and how the dataset consists of. Next, we introduce the proposed DNN-aided recommendation system scheme comparing with the existing method.

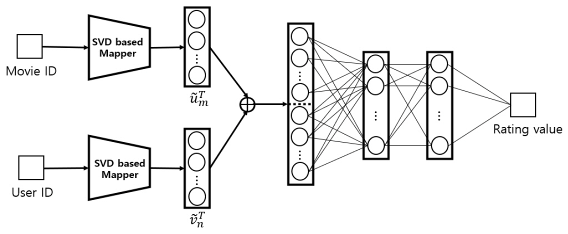
2.1 MovieLens Dataset

We utilize the MovieLens dataset which has approximately 100000 ratings with 9724 movies and 610 users. The rating values account for less than 2 % when considering the size of the matrix. Thus, this matrix consists of sparse values similar to other recommendation systems. Each user and each movie are projected onto the unique user ID and the unique movie ID. In addition, all rating scores range from 0.5 to 5 in increments of 0.5 points.

In order to apply CF, we process the MovieLens data to have $M \times N$ matrix containing user ratings where the row size M is the number of movies and the column size N is the number of users. Each row represents the user rating values according to the movie ID. On the other hand, each column means the specific user's ratings for all kinds of movie IDs.



(Figure 1) Conventional DNN based CF structure for recommendation systems.



(Figure 2) Proposed CF structure for recommendation systems.

2.2 Proposed Recommendation System Structure

Conventional DNN based CF utilizes embedding layers for vectorizing a movie ID and a user ID represented as scalar values. The embedding layer is a kind of a dictionary, so the layer converts the user ID and the movie ID into unique vectors. In order to pass the embedding layer, one-hot encoding is used to handle with categorical IDs [15]. After concatenating the output of each embedding layer, the output vector goes through the multi layers to get the rating inference. During the training, the embedding layers are also updated to fit the model. An overall architecture of the typical DNN aided CF is illustrated in Figure 1.

The proposed scheme in this paper utilizes the fixed mapper which converts ID to the vector rather than training the embedding layer. In order for the mapper to reflect initial features related to the preference well, we consider a concept of the matrix factorization (MF) based CF [8]. The MF based

(Table 1) ID conversion method comparison between typical DNN based CF and proposed CF

	ID Conversion method
Conventional DNN based CF	Embedding: One-hot encoding + hidden layer(s)
Proposed CF	SVD based vectorization

scheme extracts latent features using the singular value decomposition (SVD). In case of the sparse matrix in the recommendation systems, the truncated SVD is applicable to reduce the computation complexity. When S is an $M \times N$ original matrix, the truncated SVD of S with the number of latent features L can be expressed as

$$\tilde{S} = \tilde{U}\tilde{\Sigma}\tilde{V}^T \quad (1)$$

where $\tilde{U}$ is an $M \times L$ unitary matrix, $\tilde{\Sigma}$ is an $L \times L$ diagonal matrix, and $\tilde{V}$ is an $N \times L$ unitary matrix. In (1), $\tilde{S}$ is not equivalent to S since the truncated SVD does not fully recover the original matrix. Each row of $\tilde{U}$ is $\tilde{u}_m$ with $m \in \{1, 2, ..., M\}$ and each row of $\tilde{V}$ is $\tilde{v}_n$ with $n \in \{1, 2, ..., N\}$, which are the feature vectors of the m's movie ID and the n's user ID, respectively. The typical recommendation systems using MF based CF just compares the feature vectors to analyze similarity or utilizes $\tilde{S}$ to calculate the rating scores. The conventional MF based method is simple and intuitive but the results of the SVD are linear operation so the MF only scheme is hard to reflect non-linearity characteristics.

We focus on that the SVD contains the latent features. Therefore, the mapper in our proposed method replaces the embedding layers with $\tilde{U}$ and $\tilde{V}$. The rest of the proposed model architecture is the same as the typical DNN based CF. Figure 2 shows the proposed recommendation system. A generalized procedure to get the input layer of DNN of proposed CF is described in Algorithm 1. The proposed structure can decrease the number of weights that need to be fitted by $(M+N) \times L$ compared to the conventional architecture when the same size of vector is used for initialization. This can mitigate both the consumed training time and feed-forwarding time. Table 1 summarizes the ID conversion method between typical DNN based CF and proposed scheme.

(Algorithm 1) SVD based ID vectorization and concatenation to generate input layer of DNN

```
Initialize:
    Item ID set: $\{1,2,...,M\}$
    User ID set: $\{1,2,...,N\}$
    $L$: Number of latent features
    Generate rating matrix $S$ of size $M \times N$
    Do truncated SVD to $S$ according to (1) and derive $\tilde{U}$ and $\tilde{V}$
Input: item $m$ and user $n$
Output:
    Get $m^{th}$ row of $\tilde{U}$ as $\tilde{u}_m$
    Get $n^{th}$ row of $\tilde{V}$ as $\tilde{v}_n$
    Return $[\tilde{u}_m \ \tilde{v}_n]^T$
```

3. Performance Evaluation

In this section, we evaluate the performance of the proposed recommendation system using MovieLens dataset. We describe the detailed simulation environment before analyzing the simulation results.

3.1 Simulation Environments

We consider the output of the recommendation system model is the rating value for the pair of the movie ID and user ID as input depicted in Figure 2. There are 100836 rating values in the MovieLens dataset and we divide the data into training and validation and test sets having a 70:20:10 ratio. We apply the truncated SVD to set the number of latent features for initialization to 15. This means that the size of each feature vector of a movie ID and a user ID is 15. Moreover, there are 2 hidden layers in the model. We train the model to minimize the mean squared error (MSE) loss between the estimated rating value and the ground truth. The detailed simulation environment including the hyper-parameters to train the model is summarized in Table 2.

(Table 2) Simulation environments for the proposed recommendation system

Parameter	Value
Number of movies	9724
Number of users	610
Number of latent features	15
Number of hidden layers	2, 64x16
Split ratio of training, validation, and test set	70:20:10
Number of epochs	100
Training batch size	100
Cost function	MSE loss
Learning rate	0.001
Optimizer	Adam [16]

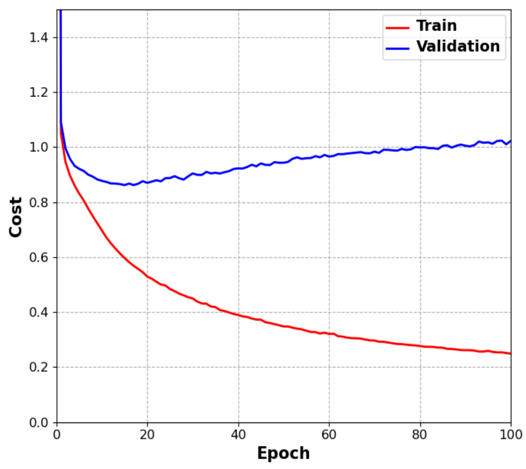
3.2 Cost Function Comparison during Training Phase

To compare with the proposed scheme, we consider the typical DNN based CF model. Without loss of generality, we consider each embedding layer converts the movie ID or the user ID to a vector of size 15 which is the same as the number of the latent features in the proposed scheme. The same hyper-parameters described in Table 2 are applied to train and evaluate the reference model. Moreover, we use the same training, validation dataset for both models for an equal comparison.

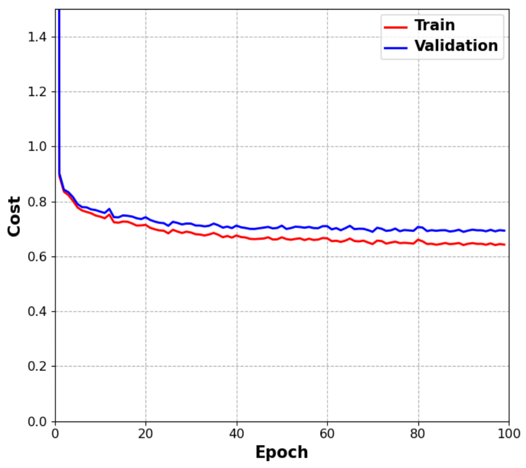
Figure 3 and Figure 4 represent the cost functions during the training phase for each of the reference model and the proposed model, respectively. For the conventional DNN based CF model, the training loss consistently decreases while the validation loss decreases up to a certain point but begins to increase again. In contrast, we can check both the training loss and validation loss steadily decrease as the training epoch increases in the proposed model. This means that the proposed model can avoid from the overfitting during the training phase and we can expect the proposed model can be trained with larger epochs for further reducing the loss. Even though the training loss of the proposed model is larger than that of the reference model, the validation loss is lower than the minimum validation loss of the reference model.

3.3 MSE Comparison of Test Dataset for Various Schemes

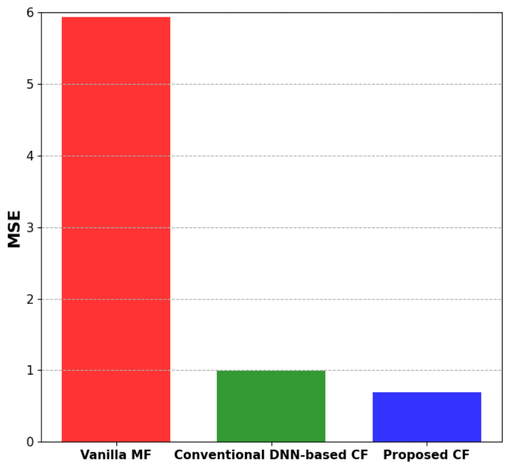
In this subsection, we evaluate the MSE performance using test set not involved in the training phase. In order to generate the best inference, the model weights of both the conventional DNN based CF and the proposed scheme are applied based on when the validation loss reached the minimum value in the Figure 2 and Figure 3. We consider



(Figure 3) Cost function of training and validation dataset for the conventional DNN based CF.



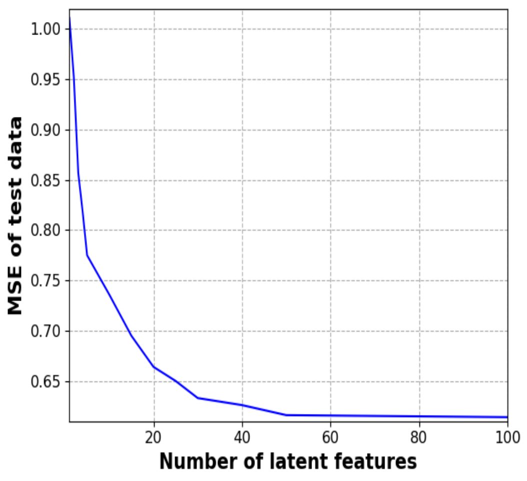
(Figure 4) Cost function of training and validation dataset for the proposed CF.



(Figure 5) MSE comparison.

one more reference method with the vanilla SVD based MF CF where the number of latent features is 15. In addition, all data for calculating MSE in the test dataset is identical for all cases.

Figure 5 illustrates the MSE performance of the proposed method with the reference schemes. A red bar and a green bar and a blue bar represent the MSE of the vanilla MF based CF and the conventional DNN based CF and the proposed scheme, respectively. As expected, the MSE of vanilla MF based CF is the largest since it can deal with only linear interactions. We figure out the MSE can be highly improved when DNN is applied. This is because the DNN architecture can take advantage of fitting non-linear conditions. Interestingly, we can check that the MSE performance is improved by approximately 30 % in the proposed scheme compared to the typical DNN based CF. This result is even more meaningful in that the proposed scheme utilizes a much smaller number of weights for model training. We speculate that the proposed scheme combines the benefits of the other methods. In other words, our model can represent the interaction between the users and the items in early stage from the SVD based initial feature extraction. After that, the model exploits the advantage of the DNN structure, which enables to learn non-linear properties.



(Figure 6) MSE according to the number of latent features.

3.4 Effect of The Number of Latent Features

We carry out additional experiments to check the effect of the number of the latent features for the proposed recommendation system model. In perspective of the computation cost, the fact that the complexity increases proportionally to the number of the latent features is explicit. However, we expect that the accuracy of the model also can be improved with the increased number of the parameters. In order for clarifying that, we vary the number of the latent features from 1 to 100 while training the proposed model. For an equal comparison, all simulation conditions depicted in Table 2 are the same for fitting the model except the number of latent features. Figure 6 describes the MSE of test dataset according to the number of the latent features. When the number of the latent features is smaller than 10, the inference performance can be highly accelerated as the number of the latent features increases. However, we can figure out the performance almost reaches saturation when the number of the latent features becomes approximately 50. Therefore, selecting a proper number of the latent features is one of the significant factor to determine the performance and design the model.

What's more interesting point is that the proposed scheme starts to generate better performance compared to the

〈Table 3〉 Relative performance comparison of proposed CF according to the number of latent features with the conventional DNN based CF

Number of latent features	Total number of weights in DNN	MSE
1	0.74%	101.1%
2	0.82%	95.2%
5	1.06%	77.5%
10	1.47%	73.6%
15	1.87%	69.5%
20	2.28%	66.4%
30	3.09%	63.3%
40	3.90%	62.6%
50	4.71%	61.6%
100	8.76%	61.4%

conventional DNN based CF described in the previous subsection, when the number of the latent features becomes only 2. Thus, we can verify the superiority of our proposed recommendation system from the result of Figure 6. Table 3 shows the performance comparison of the total number of weights in DNN and the MSE of the proposed scheme according to the number of latent features when the relative value of the conventional DNN based CF described in this section is 100%.

4. Conclusion and Future Works

In this paper, we proposed the recommendation system structure to enhance the performance of CF. We applied the SVD based initialization rather than training the embedding layers. This can reduce the number of model weights which leads to a decrease in the computation complexity. We referred to MovieLens dataset to evaluate our proposed method. From the simulation results, we checked the proposed scheme can take advantage of avoiding from overfitting while training the model. Furthermore, the improved MSE performance was obtainable compared to the typical DNN aided recommendation system. Lastly, we also analyzed the performance according to the number of the latent features. The result showed that when the number of the latent features exceeds a certain level, the MSE performance becomes to be saturated.

In the future, we will further inspect the performance of the proposed recommendation model for various scenarios. We expect the proposed scheme could be extended to other pragmatic use-cases including a design data platform.

References

[1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data. Eng.*, vol. 17, no. 6, pp. 734-749, Jun. 2005. https://doi.org/10.1109/TKDE.2005.99

[2] P. Kumar, R. S. Thakur, “Recommendation system techniques and related issues: A survey,” *Int. J. Inf. Tecnol.*, vol. 10, no. 4, pp. 495–501, Dec. 2018. https://doi.org/10.1007/s41870-018-0138-8

[3] P. B. Thorat, R. M. Goudar, and S. Barve, “Survey on collaborative filtering, content-based filtering and hybrid recommendation system,” *Int. J. Comput. Appl.*, vol. 110, no. 4, pp. 31-36, Jan. 2015. https://doi.org/10.5120/19308-0760

[4] F. Ricci, L. Rokach and B. Shapira, "Recommender systems: Introduction and challenges" in *Recommender Systems Handbook*, Boston, MA: Springer, pp. 1-34, 2015. https://doi.org/10.1007/978-1-4899-7637-6_1

[5] P. Lops, M. De Gemmis and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*, Boston, MA: Springer, pp. 73-105, 2011. https://doi.org/10.1007/978-0-387-85820-3_3

[6] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web*, Berlin, Germany: Springer. pp. 325-341, 2007. https://doi.org/10.1007/978-3-540-72079-9_10

[7] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, “Evaluating collaborative filtering recommender algorithms: A survey,” *IEEE Access.*, vol. 6, pp. 74003-74024, 2018. https://doi.org/10.1109/ACCESS.2018.2883742

[8] M. Vozalis and K. Margaritis, ‘‘Using SVD and demographic data for the enhancement of generalized collaborative filtering," *Inf. Sci.*, vol. 177, no. 15, pp. 3017-3037, Aug. 2007.

https://doi.org/10.1016/j.ins.2007.02.036

[9] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1-38, Feb. 2019. https://doi.org/10.1145/3285029

[10] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: A systematic review and new directions," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2709-2748, 2020. https://doi.org/10.1007/s10462-019-09744-1

[11] Z. Batmaz, A. Yurekli, A. Bilge and C. Kaleli, "A review on deep learning for recommender systems: Challenges and remedies," *Artif. Intell. Rev.*, vol. 52, pp. 1-37, 2018. https://doi.org/10.1007/s10462-018-9654-y

[12] Y. Deldjoo, F. Nazary, A. Ramisa, J. Mcauley, G. Pellegrini, A. Bellogin, et al., "A review of modern fashion recommender systems," in arXiv:2022.02757, 2022. https://doi.org/10.48550/arXiv.2202.02757

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T. Chua, "Neural collaborative filtering," in *Proc. 26th International World Wide Web Conference (WWW)*, Apr. 2017, pp. 173-182. https://doi.org/10.1145/3038912.3052569

[14] F. Harper and J. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst. (TiiS)*, vol. 5, no. 4, pp. 1-19, Dec. 2015. https://doi.org/10.1145/2827872

[15] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," in arXiv:1604.06737, 2016. https://doi.org/10.48550/arXiv.1604.06737

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in arXiv:1412.698, 2014. https://doi.org/10.48550/arXiv.1412.6980

◐ 저 자 소 개 ◑

성 주 형 (Juhyoung Sung)
2015년 고려대학교 컴퓨터통신공학부 (공학사)
2019년 한국과학기술원(KAIST) 전기및전자공학과 (공학석사)
2015년~2017년 대한민국 육군 정보통신장교
2019년~2021년 삼성전자 네트워크사업부 연구원
2021년~현재 한국전자기술연구원 스마트네트워크연구센터 선임연구원
관심분야: 무선 통신, 시스템 최적화, 강화학습
E-mail: jh.sung@keti.re.kr

권 기 원 (Kiwon Kwon)
1997년 광운대학교 컴퓨터공학과 (공학사)
1999년 광운대학교 컴퓨터공학과 (공학석사)
2011년 중앙대학교 전자전기공학부 (공학박사)
1999년~현재 한국전자기술연구원 스마트네트워크연구센터 센터장
관심분야: 디지털트윈, 유무선디지털통신시스템, 해양수산 ICT융합
E-mail: kwonkw@keti.re.kr

송 병 철 (Byoungchul Song)
1994년 명지대학교 전자공학과 (공학사)
1996년 명지대학교 전자공학과 (공학석사)
1999년~현재 한국전자기술연구원 스마트네트워크연구센터 수석연구원
관심분야: 네트워크 시스템, ICT융합 플랫폼
E-mail: kwonkw@keti.re.kr