Computing, Telecommunication and Control, 2022, Vol. 15, No. 1, Pp. 64–72.
Информатика, телекоммуникации и управление. 2022. Том 15, № 1. С. 64–72.

Intellectual Systems and Technologies
Интеллектуальные системы и технологии

Research article
DOI: https://doi.org/10.18721/JCSTCS.15106
UDC 004.852

COMPARISON OF RECOMMENDATION SYSTEMS BASED ON MACHINE LEARNING METHODS

V. Van[1], *A.S. Gruzdev*[2] ✉, *Q.T. Nguyen*[3], *N.T. Nguyen*[4]

[1] Ho Chi Minh University of Education, Ho Chi Minh City, Vietnam;

[2,4] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation;

[3] University of Phan Thiet, Bình Thuận, Vietnam

✉ gruzdev_spb@mail.ru

Abstract. Embedding-based models have been used in collaborative filtering over a decade. According to traditional collaborative filtering, the researchers used dot product or similarity measure to combine two or more embeddings. Typically, matrix factorization is the simplest example of an embedding-based model. In recent years, it has been proposed to replace the dot product with deep learning methods, for example, using multi-layer perceptron (MLP) algorithm. This approach is often referred to as neural collaborative filtering (NCF). In this paper, we used NCF in our research, specifically predicting item ratings results and displaying recommendations to users on e-commerce websites. We have applied NCF to the recommender system by using a deep learning model. The article used Olist's dataset to serve our experiment. We have successfully built a NCF-based recommender system with a large and sparse dataset. We have obtained better results than those produced by other methods.

Keywords: recommender system, deep learning, multi-layer perceptron, neural collaborative filtering, metric

Citation: Van V., Gruzdev A.S., Nguyen Q.T., Nguyen N.T. Comparison of recommendation systems based on machine learning methods. Computing, Telecommunications and Control, 2022, Vol. 15, No. 1, Pp. 64–72. DOI: 10.18721/JCSTCS.15106

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.15106
УДК 004.852

СРАВНЕНИЕ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ, ОСНОВАННЫХ НА МЕТОДАХ ГЛУБОКОГО МАШИННОГО ОБУЧЕНИЯ

В. Ван[1], А.С. Груздев[2] ✉, К.Т. Нгуен[3], Н.Т. Нгуен[4]

[1] Педагогический университет Хо Ши Мина, Хо Ши Мин, Вьетнам;
[2,4] Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация;
[3] Университет г. Фантхьет, Вьетнам
✉ gruzdev_spb@mail.ru

Аннотация. Нейросетевые модели испытывают сложности при необходимости работы с разреженными категориальными признаками. Вложения являются способом уменьшения размерности таких признаков ради повышения производительности модели. Согласно традиционной совместной фильтрации, используется скалярное произведение или мера сходства для объединения двух или более вложений. Как правило, матричная факторизация является простейшим примером модели вложения. В статье рассмотрена нейронная совместная фильтрация (NCF) для прогнозирования результатов оценки товаров и отображения рекомендаций пользователям на электронных коммерческих площадках. Алгоритм нейронной совместной фильтрации на основе линейной и квадратичной метрики показывает преимущество перед другими методами. Можно применять алгоритм NCF в рекомендательной системе, использующей модель глубокого обучения.

Ключевые слова: машинное обучение, нейронная сеть, система рекомендации, глубокое обучение, нейронная совместная фильтрация

Для цитирования: Van V., Gruzdev A.S., Nguyen Q.T., Nguyen N.T. Comparison of recommendation systems based on machine learning methods // Computing, Telecommunications and Control. 2022. Т. 15, № 1. С. 64–72. DOI: 10.18721/JCSTCS.15106

Introduction

Recommender Systems (RSs) were developed for the internet trading with the purpose to build the automatic systems that can provide valuable information or items for users. For example, Ebay, Amazon, MovieLens have a recommender system for their business. In general, there are two main approaches for the traditional RS: content-based and collaborative filtering. Besides, hybrid approach is also used in order to bring the effective results for RSs.

The content-based (CB) approach [1, 2] as its name suggests, is a method mainly based on content and characteristic of items. We can calculate the similarity between two items based on feature vectors of items. When a user u gives a rating for an item i_j, the system will find the items i_k, i_h, ... that have a feature vectors similarity with item i_j, in order to recommend them for user u. The advantage of CB is the users' possibility to receive fitting recommendation about items by calculating the similarity of items with each other, rather than equating similar preferences of all users. The disadvantage lies in the limited content to base the recommendations for users on.

The collaborative filtering (CF) [3, 4] approach is mainly based on the similarity of the users themselves. When a user u_i provides rating for an item i in a rating matrix $\boldsymbol{R}$, for each u_i the system will define a community of users u_j, u_k, ... so that they similar to user u_i, based on the feature vectors of users. After determining the community for user u_i, the system will give the recommendation about the items this community gives high ratings to. Recently, researchers tend to work with collaborative filtering method.

In addition, following the collaborative filtering-based approach, there are two main research directions: memory based and model based. The memory based direction [5] collects rating data in the system and uses it to calculate the ratings for new items. This direction can be implemented in two ways: user based or item based. However, the memory based direction is limited by several disadvantages. The model based direction [6] sets up a model that trains and predicts users' unknown ratings.

Previous studies focused on applying other methods, such as *Support Vector Machine*, *Singular Value Decomposition* [7], *Matrix factorization* [8], *Neural network* [9], etc.

The target of the work is comparison of recommendation systems based on machine learning methods. Comparison of algorithms will be made on the developed metrics.

Related works

Recently, researchers tended to use deep learning for RSs. In Neural Collaborative Filtering (NCF) method, fully connected embedding layers project the sparse representation to a dense vector. These embedding vectors are the input of a multi-layer neural network (neural collaborative filtering), while NCF maps these embedding vectors and ratings. Each layer of NCF can adjust to explore the latent structure between users and items.

Let y_{ui} be a target variable (y is true) and $\hat{y}_{ui}$ is a prediction variable (y is pre) of the model.

The prediction model can be presented in the form [9]:

$$\hat{y}_{ui} = f\left(P^T v_u^U,\ Q^T v_i^I \vee P,\ Q,\ \theta_f\right), \tag{1}$$

where $P \in R^{M*K}$ and $Q \in R^{N*K}$ denote latent matrices of users and items respectively.

With u being the user, and i the item, θ_f denotes the parameters of the model in the interaction function f. Because function f is defined as a multi-layer network, f can be formed as follows:

$$f\left(P^T v_u^U,\ Q^T v_i^I\right) = \varnothing_{out}\left(\varnothing_X\left(\dots\ \varnothing_2\left(\varnothing_1\left(P^T v_u^U,\ Q^T v_i^I\right)\right)\dots\right)\right), \tag{2}$$

where v_u^U and v_i^I are feature vectors that describe user u and item i, respectively; $\varnothing_{out}$ and $\varnothing_X$ respectively denote the mapping function for the output layer and x^{th} neural collaborative filtering (CF) layer, and there are X neural CF layers in total [9].

In NCF, the model tries to learn user-item interactions through a multi-layer perceptron (MLP). For MLP, such activation functions as Sigmoid, Hyperbolic tangent (tanh), Rectified linear unit (ReLU), etc. are used. The activation function simulates the rate of impulse transmission across the axon of a neuron. In an artificial neural network, the activation function acts as the linear component at the output of the neurons [10].

For MLP model, NCF uses two vectors to model users and items, then combines them into one vector via the concatenation. This structure was also widely used in multi-model deep learning [11, 12]. If we use additional hidden layers in the concatenated vector, the MLP model in NCF is defined as [9]:

$$z_1 = \varnothing_1\left(p_u,\ q_i\right) = \begin{bmatrix} p_u \\ q_i \end{bmatrix}, \tag{3}$$

$$\varnothing_2\left(z_1\right)=a_2\left(W_2^T z_1+b_2\right), \tag{4}$$

$$\varnothing_L\left(z_{L-1}\right)=a_L\left(W_L^T z_{L-1}+b_L\right), \tag{5}$$

$$\hat{y}_{ui}=f\left(h^T\varnothing_L\left(Z_{L-1}\right)\right), \tag{6}$$

where W_x, b_x and a_x denote the weight of matrix, bias vector, and activation function for x^{th} layer's perceptron.

Proposed NCF model for recommender systems

In this paper, we choose the activation function ReLU $f(x) = \max(0, x)$. The ReLU function simply filters the values under 0. Looking at the formula, we easily understand how it works (see Fig. 1).

Fig. 2 represents the architecture of NCF that we used in this paper as shown below.

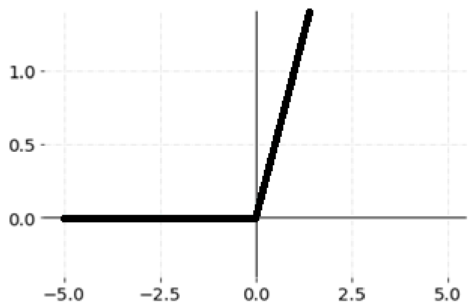

Fig. 1. Graph of ReLU function

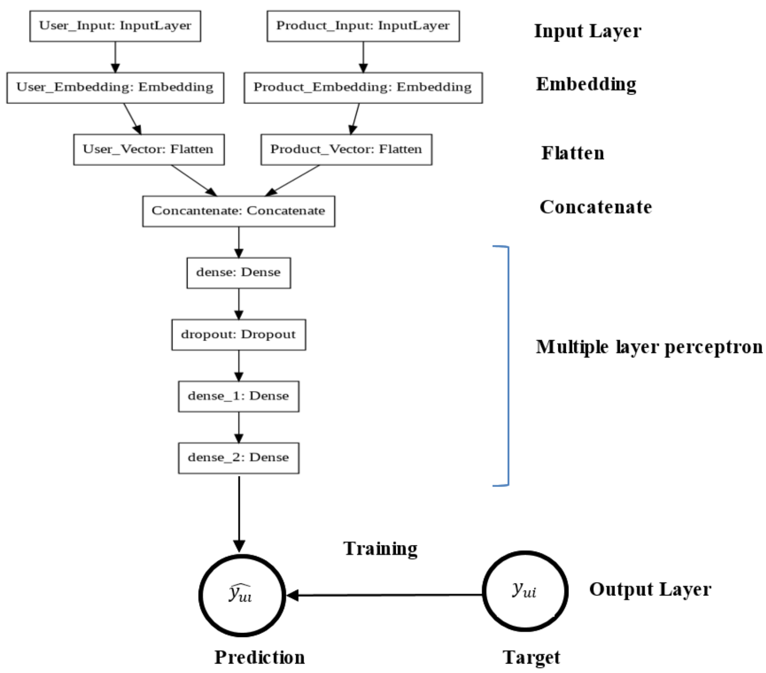

Fig. 2. Architecture of Neural Collaborative Filtering (NCF)

Cost function and evaluation metrics

Cost function

The cost function (loss function) for the entire training dataset:

$$e_{ui} = \frac{1}{s}\sum\nolimits_{u,i}\left(R_{ui} - \hat{R}_{ui}\right)^2, \tag{7}$$

where R_{ui} is observed value; $\hat{R}_{ui}$ is the predicted value; e_{ui} is the mean square error (cost function).

Gradient Descent algorithm to optimize the cost function as follows:

1. Choose an initial point $\theta = \theta_0$.
2. Update θ until we get acceptable result:

$$\theta = \theta_0 - \eta\nabla_0 J\left(\theta\right), \tag{8}$$

where $\nabla_0 J\left(\theta\right)$ is the derivation of the cost function at θ; θ is a set of variables that we need for the update; η is learning rate, it's a positive number.

In this paper, we use Adam (short for Adaptive Moment Estimation) update rule [13]:

$$m_t = \beta_1 m_{t-1} + \left(1-\beta_1\right) g_t, \tag{9}$$

$$v_t = \beta_2 v_{t-1} + \left(1-\beta_2\right) g_t^2, \tag{10}$$

$$\eta_t = \eta\frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t}, \tag{11}$$

$$\theta_t = \theta_{t-1} - \eta_t \frac{m_t}{\sqrt{v_t} + \varepsilon}, \tag{12}$$

where t indexes the current training iteration; m_t and v_t are exponential moving average (EMA) of g_t and the EMA of g_t^2 respectively; g_t is the gradient at current iteration; β_1 and β_2 are smoothing parameters, typical values are β_1 = 0:9; β_2 = 0:999 respectively; ε is a small scalar (e.g. 10^{-8}) used to prevent division by 0.

Evaluation metrics

There are several types of metrics to evaluate the effectiveness of the CF approach [14, 15]. In this paper, we use two evaluation metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the accuracy.

The MAE metric is defined as [7]:

$$MAE = \frac{1}{\left|R_{test}\right|}\sum\nolimits_{ui}\left|R_{ui} - \hat{R}_{ui}\right|, \tag{13}$$

where $\hat{R}_{ui}$ denotes prediction rating of a user u for item i and R_{test} denotes the number of ratings in the experiment.

The RMSE metric is defined as [7]:

$$RMSE = \sqrt{\frac{1}{R_{test} \vee \sum_{ui} \left(R_{ui} - \hat{R}_{ui}\right)^2}}. \quad (14)$$

From the definitions, we obviously see that a smaller MAE or RMSE value means better accuracy.

Experiment

For the dataset, we used available Olist Ecommerce data on Kaggle [17]. We were only interested in several features such as id_customer, id_product and rating. The ratings ranged from 1 to 5 stars given by the users for the corresponding items. The dataset has more than 100k lines of data that are interactions between users and items. After preprocessing the dataset, we got the following results:

Table 1

Dataset after preprocessing

Dataset	Interactions	Items	Users	Sparsity, %
Olist Ecommerce	7064	4886	3271	99.955

We divided the dataset into 3532 lines for training and 3532 for testing. The experiment was based on the Neural Collaborative Filtering model proposed above. For the learning process in the NCF al-

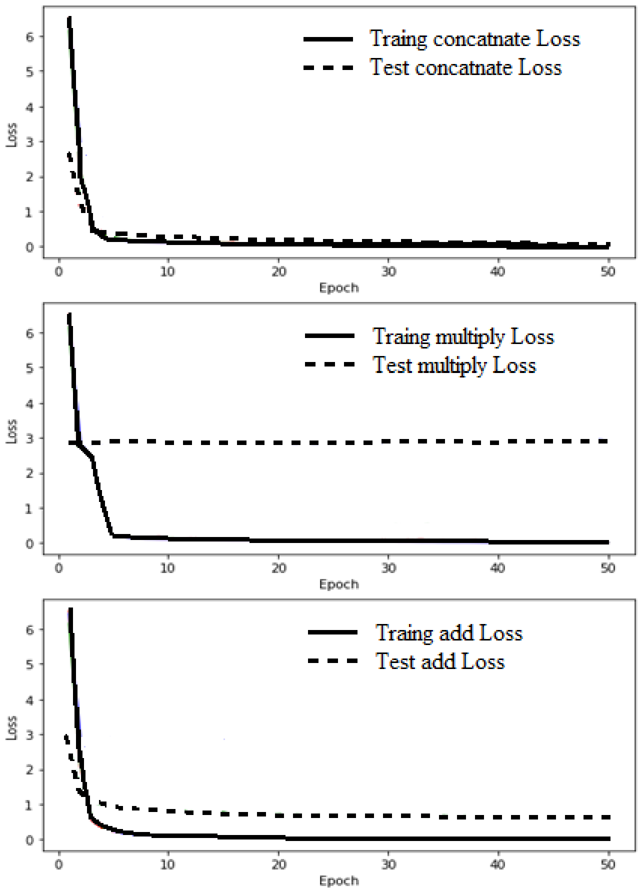

Fig. 3. Illustrating the convergence of several methods by using NCF algorithm

gorithm, beside concatenation we also used some other methods such as multiplication and addition. The RMSE output of the NCF algorithm via concatenation, multiplication, and addition is shown in Table 2.

Table 2

RMSE metric obtained by using several methods

Method	RMSE
Concatenate	**0.23**
Multiply	1.7085
Add	0.7681

Fig. 3 shows the convergence of concatenation, multiplication, and addition methods on train and test set by using the NCF algorithm.

Based on the RMSE metrics on test set shown in Table 3, the concatenation method of NCF gives the best result of 0.23 with RMSE. Besides, we used support library [16] to evaluate and compare our NCF model with the other algorithms such as MF, NMF, SVD, etc. Fig. 4 shows the RMSE metrics of several algorithms in the form of column graph.

Table 3

MAE and RMSE metrics of several algorithms

	Test MAE	Test RMSE	Algorithm
1	1.3953	1.5242	SVD
2	1.3415	1.4668	SVD++
3	1.5283	1.6858	KNN Basic
4	1.0312	1.3768	KNN with Mean
5	1.338	1.563	NMF
6	1.5413	1.68	MF
7	0.1566	0.23	NCF

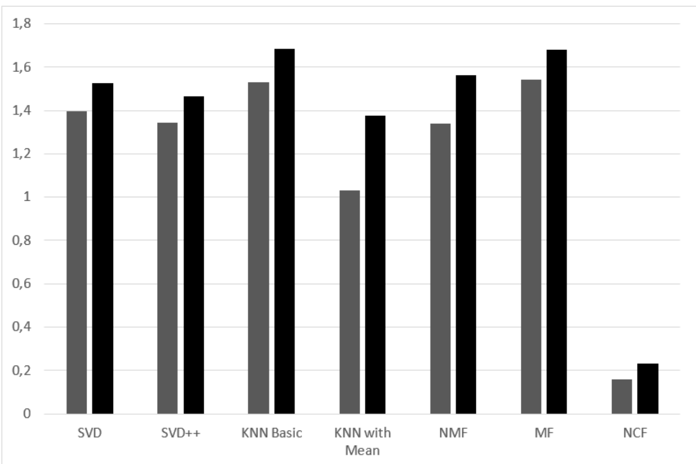

Fig. 4. MAE and RMSE metrics of several algorithms (column graph)

Looking at Fig. 4 above, with an RMSE metric being 0.23, our NCF method has intuitively outperformed the other algorithms. The RMSE metrics of the remaining algorithms are much higher meaning that the accuracy of the recommendation is lower.

Conclusion

Neural collaborative filtering combined with deep learning model has an advantage over other methods. We used the Olist data for our experiment to create a system of recommendations based on joint filtering with a large and sparse dataset. We have obtained better results than those produced by other methods.

The Neural collaborative filtering method gives a noticeable advantage in processing speed in both linear and quadratic metrics. This method gives the value of a quadratic metric of 0.23 and 0.1566 in the case of a linear metric. This value is several times less than the other methods considered.

REFERENCES

1. **Pazzani M.J., Billsus D.** Content-based recommendation systems. *LNCS*, 2007, Vol. 4321.

2. **Aggawal C.C.** *Content-based recommender systems*. Recommender Systems Textbook. Switzerland, Springer International Publishing, 2016, Pp. 139–166.

3. **Schafer J.B., Frankowski D., J. Herlocker, J. Shilad Sen.** Collaborative filtering recommender systems. *LNCS*, 2007, Vol. 4321.

4. **Herlocker J.L., Konstan J.A., Terveen L.G., Riedl J.** Evaluating collaborative filtering recommender systems. *ACM Transaction on Information Systems*, 2004, Vol. 22 (1).

5. **Aggawal C.C.** *Neighborhood-based collaborative filtering*. Recommender Systems Textbook. Switzerland, Springer International Publishing, 2016, Pp. 29–70.

6. **Aggawal C.C.** *Model-Based Collaborative Filtering*. Recommender Systems Textbook. Switzerland, Springer International Publishing, 2016, Pp. 71–138.

7. **Ricci F., Rokach L., Shapira B.** *Recommender Systems Handbook*. Springer, 2011.

8. **Aghdam M.H., Analoui M., Kabiri P.** A novel non-negative matrix factorization method for recommender systems. *Natural Science Publishing Journal, Applied Mathematics& Information Sciences*, 2015, Vol. 9.

9. **He X., Liao L., Zhang H., Nie L., Hu X., Chua T.** Neural collaborative filtering. *Creative Commons*, 2017.

10. **Li Fei-Fei, Johnson J., Yeung S.** Convolutional neural networks for visual recognition, 2021. Available: *http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf* (Accessed: 2021).

11. **Srivastava N., Salakhutdinov R.** Multimodal learning with deep Boltzmann machines. *Advances in Neural Information Processing Systems*, 2012, no. 25, Pp. 2222–2230.

12. **Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, Tat-Seng Chua.** Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, Pp. 187–196.

13. **Juntang Zhuang, Tommy Tang, Sekhar Tatikonda, Nicha Dvornek.** AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients, 2020. Available: *https://arxiv.org/abs/2010.07468* (Accessed: 2021).

14. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web*, 2001.

15. **Olmo F.N., Gaudioso E.** Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 2008, Vol. 3, Pp. 790–804.

16. **Hug N.** Surprise, 2019. Available: *http://surpriselib.com* (Accessed: 2021).

17. Olist Dataset. Available: *https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce* (Accessed: 2021).

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Vy Van
Ван Ви
E-mail: vanv@hcmue.edu.vn

Alexander S. Gruzdev
Груздев Александр Станиславович
E-mail: gruzdev_spb@mail.ru

Quang Tan Nguyen
Нгуен Куан Тан
E-mail: tannq@hcmue.edu.vn

Ngoc Tan Nguyen
Нгуен Нгок Тан
E-mail: ngoctan1610@yahoo.com

Submitted: 11.01.2022; Approved: 23.05.2022; Accepted: 30.05.2022.
Поступила: 11.01.2022; Одобрена: 23.05.2022; Принята: 30.05.2022.