

# DLGNN: A Double-layer Graph Neural Network Model Incorporating Shopping Sequence Information for Commodity Recommendation

Chuyuan Wei,<sup>1</sup> Jianwei Niu,<sup>2\*</sup> and Yanyan Guo<sup>3</sup>

<sup>1</sup>School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture,  
1 Zhanlanguan Road, Xicheng District, Beijing 100044, China

<sup>2</sup>School of Computer Science and Engineering, Beihang University,  
37 Xueyuan Road, Haidian District, Beijing 100191, China

<sup>3</sup>AI Lab, China Merchants Bank, 7088 Shennan Boulevard, Shenzhen 518040, China

(Received August 6, 2020; accepted December 19, 2020)

**Keywords:** recommendation system, graph neural network, online shopping, sequence modeling

The essence of recommendation methods for online shopping is to predict whether the user will buy specified products. The existing methods are usually modeled on a user–item–score matrix, which does not consider the transformation of users and items. The transformation is hidden in the behavior and reflected in the shopping sequence. In a smart city of the future, products can be expressed by sensors: for a given item, the changes in its audience can be observed from the sequence of users who have bought these products, and for a given user the changes in his or her need submitted as requirement can be obtained according to the sequence of purchased commodities. To better capture the shopping sequence and apply it to recommendations, we proposed an E-commerce recommendation algorithm similar to the recommendation problem in the smart city using a double-layer graph neural network (DLGNN) to model the user sequence and item sequence and predict the purchasing behavior of users. In the DLGNN, the sequence data, including the sequence of users who have bought a particular item and the sequence of purchased commodities from each user, are first modeled as graph-structure data. Then by using a graph neural network (GNN), one can express the state of each node of the graph as cells in the neural network. On this basis, the pretrained representation vectors of users and commodities are used to represent their global information. The state vectors and representation vectors can be used to model the purchasing behavior of users. Experiments based on a real dataset show that the proposed method yields better performance than existing recommendation algorithms.

## 1. Introduction

With the rapid development of mobile Internet, daily acts of people cannot be separated from their mobile devices and internet information. 5G will make connections between people and

---

\*Corresponding author: e-mail: niujianwei@buaa.edu.cn  
<https://doi.org/10.18494/SAM.2020.3056>

things, and between things and things, that is to say, objects in the home, office, and city will be connected, moving towards systems with wisdom and intelligence. However, as the content and services provided by the Internet become increasingly rich, the large amount of redundant information leads to the problem of information overload and poor user experience.

Many online services such as E-commerce, advertising, social networks, and personalized reading are becoming increasingly widespread with the rapid development of mobile Internet. The core function of recommendation systems (RSs) is to predict how likely a user will purchase an item on the basis of historical interactions such as browses, clicks, and purchases.<sup>(1)</sup> RSs rely on large datasets of historical data and find a good representation model of historical data, then design models and algorithms to predict and recommend.

The two traditional kinds of recommendation methods, content-based recommendation algorithms<sup>(2,3)</sup> and collaborative filtering (CF) recommendation algorithms,<sup>(4)</sup> have shortcomings in describing sequential data. Each item is independent of other items and the continuous preference information of each item in session cannot be modeled.<sup>(5)</sup> In recent years, the emergence of deep learning methods has brought new ideas for RSs. Relevant models and technologies based on deep learning are applied to RSs, and certain breakthroughs have been achieved. The most powerful approach to capture complex relations is to use deep learning techniques. Deep neural networks (DNNs) for RSs are also beginning to attract much attention.<sup>(6–9)</sup> User attributes and item properties can be added to the models when constructing a neural network, then the impact of these external features on the final results can be analyzed. It is easier to obtain the preference of users from user attributes than from historical user behaviors. Some attributes may even directly indicate user preferences, and the relationships between these preferences and user behaviors can be further obtained from the output of a neural network. Some existing research methods also add features extracted from relevant texts and images to the neural network. Although the accuracy of algorithms is greatly enhanced using a neural-network-based recommendation model, the effectiveness of the recommendation is not quite as good as expected.

Generally speaking, a RS needs a very large number of historical user behaviors, which can be more than 10000, containing all the information of the users. The CF and matrix-factorization (MF)-based methods have been proposed to deal with user shopping data, which reflect the fact that the recommendation process has begun to depend on the decision of algorithms. Algorithms based on MF sufficiently explain the implicit relevance contained in the user–item-score matrix.<sup>(10)</sup> Thus, user behaviors are predicted by analyzing user preferences in the historical data using such methods. However, in an actual environment, user preferences are essentially dynamic and may change with time. Changes in the user preferences in historical data and the prediction of prospective trends are both ignored in these models. To obtain durable representation vectors of user preferences, constantly updated data should be used in these methods. However, this can lead to various problems in real applications. On the one hand, the constantly updated dataset is usually small, and thus the reliability of the algorithm becomes low. Conversely, changes in user preferences cannot be accurately obtained if too much historical data are used. On the other hand, the failure time of the model cannot be predicted, and the frequency for updating the model is hard to control.

Currently, some recommendation models<sup>(11)</sup> based on a recurrent neural network (RNN) use the purchasing behavior of users to model and express changes in user behaviors. However, it is known that the RNN model is more applicable to express the effect of predecessor nodes on the successor node. In fact, the effect of user historical behaviors on the subsequent behaviors is not in chronological order, but user historical behaviors construct a user behavior network that has a cross influence. Other models<sup>(12)</sup> use a graph structure to model the click sequence of users. However, these methods unduly focus on behaviors of a single user and ignore the global level, i.e., the changes in the item audience.

For a smart city in the future, the relationship between all the commodities (equivalent to the commodities in this article) can be modeled by using an algorithm similar to one for commodity recommendation. Commodities can be expressed by sensors, where people are users, and this relationship is the social relationship between people. People's choice of commodities is the same as their preference for commodities, and its essence is a kind of RS.

To solve these problems, a double-layer graph neural network (DLGNN) based on a graph neural network (GNN) is proposed as a recommendation model in this paper. The user–purchased-item sequence and purchased-item–user sequence are both used to construct the graph structure, and the GNN is applied to model these sequences to probe the transformation pattern between different nodes. The states of these transformation patterns are then used to predict user behaviors. On this basis, pretrained embedding vectors of users and commodities are used to represent their global information, and thus some additional features can be incorporated to increase the amount of prediction information. The E-commerce Recommendation Algorithm in this paper is similar to the recommendation problem in a smart city. In summary, this work provides the following main contributions:

- 1) The user–purchased-item sequence and purchased-item–user sequence are both extracted from the original purchase information and are used to construct two graph structures, which can be effectively applied to express changes in user preferences and item audiences.
- 2) We propose the DLGNN model, a new recommendation framework based on a GNN, and the nodes of the network are used to express the transition states of users and items. User behaviors are modeled and predicted by the implicit vectors of these states. The GNN can model the continuous user–purchased-item sequence and purchased-item–user sequence in a graph learning method.
- 3) Our methods are used to construct the representation vectors of users and commodities. A neural network is structured to represent the embedding of user and item information. By introducing these vectors in the above neural network, one can apply both the purchase sequence and the representation vectors of users to predict items purchased.

This paper is organized as follows. Section 2 presents the previous related work. Details of the shopping prediction model based on the GNN are introduced in Sect. 3. Experiments based on a real dataset in Sect. 4 verify the effectiveness and reliability of the proposed method. Section 5 concludes this paper.

## 2. Related Work

RSs have been widely investigated. Many approaches have been proposed to address the following issue: the traditional RS uses explicit/implicit information as the input for prediction, which has the problem of sparsity. Existing methods for RSs can be generally categorized into three classes: content-based models,<sup>(12)</sup> CF,<sup>(13)</sup> and hybrid recommendation methods.<sup>(14,15)</sup> Among the various methods, MF<sup>(16)</sup> is the most representative one, which models the user–item interaction function as the inner product of a user latent vector and an item latent vector. However, MF-based methods lack the capability to capture hidden subtle factors, because the inner product is not sufficient for modeling the complex structure of interaction data.<sup>(17,18)</sup>

In recent years, DNNs have yielded state-of-the-art performance on several tasks such as computer vision and natural language processing. Neural-network-based recommendation models have rapidly become a research hotspot, and types of network with different structures have been applied to RSs. A DNN, convolutional neural network (CNN), and RNN were respectively used in Refs. 19–21. Some neural network recommendation models have been successfully applied to the Internet. For example, the DNN model of YouTube<sup>(22)</sup> utilizes the advantages of neural networks for parameter training, and the constructed network effectively combines features of different dimensions. However, most works performed deep content feature learning and resorted to CF, which cannot effectively model the highly complex user–item interaction relationship. On the other hand, owing to the difficulty of training DNNs, existing models utilize a shallow architecture, thus limiting the expressive potential of deep learning.<sup>(18)</sup>

In fact, there are two key points to be considered: 1) embedding, which transforms users and items to vectorized representations; 2) interaction modeling, which reconstructs historical interactions based on the embeddings. GNNs<sup>(23,24)</sup> are deep-learning-based methods that operate on a graph domain. Also, graph embedding can learn to represent graph nodes, edges, or subgraphs as low-dimensional vectors.<sup>(25)</sup> Therefore, the GNN method can be used in RSs.

A GNN is a type of neural network that is applied to analyze graph structures. Graph-based tasks such as representation learning are modeled by a GNN. A GNN based on an RNN has been proposed.<sup>(23)</sup> Adding gated recurrent units (GRUs) to GNNs makes the training more effective.<sup>(24)</sup> In some studies on RSs, GNNs are combined with traditional models.<sup>(26)</sup> There are also some methods that use a GNN directly to make predictions.<sup>(5)</sup>

## 3. Proposed Model

In this paper, a recommendation model based on a GNN is proposed. The construction of representation vectors of users and commodities is first introduced. Then, the method for constructing the graph data and the model for analyzing these data using a GNN are both presented. Finally, the shopping prediction model and its training mode are given.

### 3.1 Model framework

The DLGNN model predicts user ratings of a product. The framework of the DLGNN is shown in Fig. 1. The input of the model is the user's scoring matrix for the product. The output of the model is a score, which is used to predict the user purchase intention of specified commodities. The scoring matrix is first constructed as sequence data. The sequences of users and commodities are the inputs of the model, which are used to construct the graph data and GNN. In addition, to introduce global characteristic information of users and commodities into the steps of prediction, the embedding information of users and commodities is constructed from the relevant group data. Then, the GNN-analyzed state transfer matrices and representation vectors of users and commodities are used to predict the scores of users for specified commodities. Feedback in the form of real scores is used to make the model converge. For a particular user, the scores of different commodities are sorted, and those with highest scores are recommended.

### 3.2 Representation vector construction

The purpose of constructing representation vectors for users and commodities is to use global information of users and commodities as supplementary information. The transitional information between the sequences of commodities, which are restricted to the sequence nodes, can be obtained from the GNN. The generalization performance of the model can be enhanced by adding global information. The representation vectors of users and commodities can be seen as the representation vectors in the entire group. The people who bought the same commodities as a user represent the unique features of this user. The model structure used to construct the representation vector originates from the Continuous Bag-of-Words (CBOW) Model.<sup>(27)</sup> The neural network structure is shown in Fig. 2, which is applied to train the representation vectors of users.

The inputs of the neural network are one-hot vectors denoting the users that bought the same commodities as the  $i$ th user. The hidden layer cell can be obtained by the following expression:

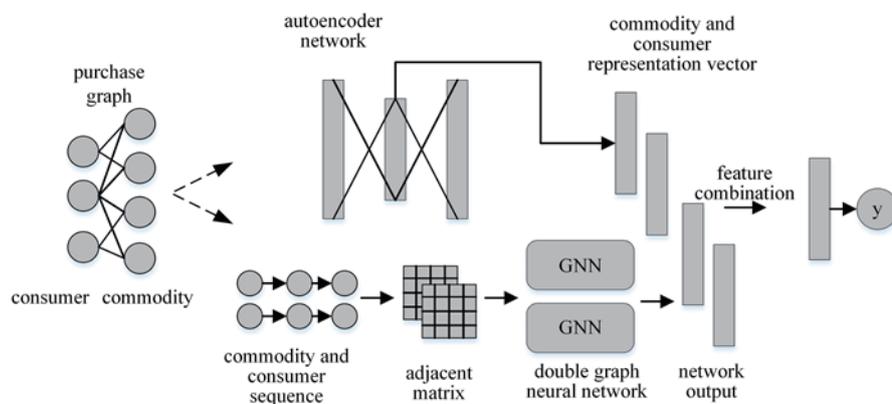


Fig. 1. Framework of DLGNN.

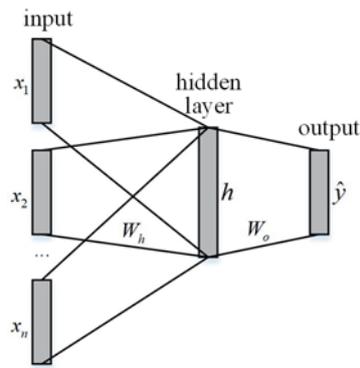


Fig. 2. Neural network structure.

$$h = \frac{1}{n} \sum_{i=1}^n x_i^T W_h + b_i, \quad (1)$$

where  $x_i$  is the one-hot vector of user  $i$ ,  $W_h$  is the weight matrix, and  $b$  is the bias. By using  $h$  and the weight matrix  $W_o$ , we can express the  $k$ -dimensional probability distribution  $y$  as

$$y = \text{softmax}(W_o h + b). \quad (2)$$

The output of the network  $y$  is a multiclass probability. The class with the largest value is the class in which the predicted user  $u_k$  is located. The loss value is calculated using the one-hot vector of user  $u_k$  and network convergence is performed by back-propagation. When the network converges, for a user  $u_k$ , the users who bought the same commodities as  $u_k$  are passed to the network to calculate the hidden layer vector  $h$  as the representation vector of user  $u_k$ . For the commodities, the items bought by the same users are used to train the commodities' representation vector.

In addition to the above methods for obtaining global representation vectors, there are many representation learning algorithms for training them. Taking the node2vec method<sup>(28)</sup> as an example, relationships between users are used to construct the user diagram  $G(v, e)$ . Sentences constructed by nodes are generated by a random walk in the network, and the word2vec model is then used to train the node-embedded expression. However, a large amount of resources is needed when dealing with a large-scale user graph and item graph.

### 3.3 Design of graph

The graph data are constructed on the basis of two types of shopping sequence, the item sequence of a single user and the user sequence of a single item; the final user-item graph is synthesized from these two sequences. The shopping sequence of a user is a list that contains all the recently purchased commodities representing the changes in user preferences; this sequence can be seen as a local graph:  $\text{userGraph} = (V_{item}, E_{item})$ .

Where the nodes of the network represent the recently purchased commodities, the directed edge  $e_{ij}$  between nodes represents the fact that the user purchased  $v_j$  after  $v_i$ . Note that the length of the sequence should be limited by using the sequence over the last  $r$  days or  $k$  purchase records to avoid further changes in historical user preferences appearing again, which introduce noise into the model.

The user sequence of single items represents the changes in the item audience. Similar to the shopping sequence of users, a node of the network represents the recent users who have purchased item  $v_k$ , and the directed edge  $e_{p,q}$  between nodes represents the fact that user  $u_q$  purchased item  $v_k$  after user  $u_p$ .

Local graphs of users and commodities are just partial considerations on changes in user preferences and item audiences; these local graphs can be spliced into a global graph. The model cannot be fully trained using only local graphs. However, arbitrary training samples can be constructed from a global graph to fully train the model.

### 3.4 Description of GNN

In the last ten years, many methods for analyzing graph data have been proposed, and neural-network-based algorithms are a research hotspot. The gated GNN (GGNN) is an improved version of the traditional GNN, and it is more suitable for modeling the transition between sequence nodes. In this paper, the GGNN is used to analyze the node transition of shopping sequences of users and commodities. Two synchronously updated GGNN networks are designed to separately handle these two shopping sequences. Here, we take shopping sequences of users with a length of four as an example, specifically,  $Seq: V_{item1} \rightarrow V_{item2} \rightarrow V_{item3} \rightarrow V_{item4}$ .

Let matrix  $A$  denote the adjacent matrix of the sequence as shown in Fig. 3. The adjacent matrix is constructed from the incoming edges and outgoing edges of nodes, and the transition relationships are represented in the form of a matrix.

The item representation vectors containing transition relationships are constructed by the following expressions:

$$V_h = [V_{item}, 0], \quad (3)$$

$$a = A[V_{h1}, \dots, V_{hk}] + b, \quad (4)$$

	Out edge				In edge			
	$v_1$	$v_2$	$v_3$	$v_4$	$v_1$	$v_2$	$v_3$	$v_4$
$v_1$	0	1	0	0	0	0	0	0
$v_2$	0	0	1	0	1	0	0	0
$v_3$	0	0	0	1	0	1	0	0
$v_4$	0	0	0	0	0	0	1	0

Fig. 3. Example of adjacent matrix of  $Seq$ .

where  $a$  is the sequence of item representation vectors. The item representation vectors use a  $d$ -dimensional vector to represent a specified item, and this vector contains the transition information between this item and other items. When using an RNN to process  $\text{seq}.a$ , the reset gate and update gate of the GRU are used to select necessary information in  $a$  and activate the neural cells. The calculation process is shown in the following expressions.

$$z^t = \sigma(W_z a^t + U_z v^{t-1}) \quad (5)$$

$$r^t = \sigma(W_r a^t + U_r v^{t-1}) \quad (6)$$

$$\tilde{v}^t = \tanh(W_o a^t + U_o (r^t \odot v^{t-1})) \quad (7)$$

$$v^t = (1 - z^t) \odot v^{t-1} + z^t \odot \tilde{v}^t \quad (8)$$

Here,  $z^t$  and  $r^t$  are the reset and update gate vectors, respectively,  $\sigma$  is the sigmoid function, and  $\odot$  is the elementwise multiplication operator.  $v^t$  indicates the latent vector of the sequence node  $V_{itemt}$ . Then, the hidden state of the previous point in time and the present transition state are used to update the present hidden state. The output vector  $v^k$  of the final sequence node  $V_{itemk}$  represents the hidden state of all user changes for the final item after buying a series of items. Similarly, for the user sequences after GGNN processing, the output vector denotes the state vector of users after all changes in the item audience. In the actual modeling, two GNNs with the same structure are used to process sequences of users and items simultaneously.

### 3.5 Prediction step

After the processing of the two GNNs, the output vectors  $S_{user_i}$  and  $S_{item_j}$  of the user sequence and item sequence can be obtained, where  $S_{item_j}$  represents the cumulative state of the audience transition of  $item_j$  and  $S_{user_i}$  denotes the cumulative state of the purchase preference transition of  $user_i$ . The pretrained representation vector  $S_{item_j}$  for  $item_j$  denotes the representation vector of  $item_j$  over all items. Similarly, the representation vector  $E_{user_i}$  of  $user_i$  denotes the representation vector of  $user_i$  over all users. The prediction of scores of  $user_i$  for  $item_j$  is based on only these four vectors. To solve the problem that the dimensions of the state transition vector and the representation vector are not the same, a feature-fusion computation is used, and the feature vector representing users and commodities can be obtained as follows:

$$f_u = \sigma(W_{uemb} E_{user} + W_{utran} S_{user} + b_u), \quad (9)$$

$$f_i = \sigma(W_{iemb} E_{item} + W_{itran} S_{item} + b_i), \quad (10)$$

where  $W_{iemb}$  and  $W_{itran}$  control the weights of these two vectors. Activating and normalizing

feature nerve cells are the two functions of the sigmoid function. Values of the user feature vector and item feature vector are limited within  $[0, 1]$ . Then, the features of users and commodities are spliced into context features:

$$C = [f_u, f_i]. \quad (11)$$

A fully connected neural network layer is used to process the context features and for regression prediction. The equation for predicting scores can be expressed as

$$\hat{y} = W_b C + b. \quad (12)$$

The model is trained by continuous optimization to make it approach real scores. The loss function is defined by the real scores  $y$  and predicted scores  $\hat{y}$  as

$$LOSS = x(\hat{y} - y)^2, \hat{y} \in R. \quad (13)$$

The stochastic gradient descent algorithm and back-propagation are used to minimize the loss function until the model converges.

## 4. Experiments and Analysis

To evaluate the performance of the model, the real dataset used and some existing recommendation methods are first introduced in this paper. The practical effects of the proposed algorithm are then compared with those of the baseline model by considering different evaluation indicators. The effects of embedded vectors of users and commodities on the proposed model are also considered. Finally, the training process and its characteristics are discussed.

### 4.1 Dataset

A real dataset from the well-known Amazon.com is used to evaluate the performance of the proposed algorithm. As a global e-commerce website, Amazon has large and full-scale datasets,<sup>(29)</sup> the statistical properties of which are favorable such as a homogeneous distribution and lack of bias. The dataset Movies and TV is used in this paper, the data of which contain the scores of users for specified commodities and the purchase timestamps. The time span of the data is from 1996 to 2014. The scale of the dataset is listed in Table 1.

The average numbers of purchases in the dataset Movies and TV are given in this dataset, and each item is purchased approximately 75.3 times on average. From this, it can be inferred that the lengths of the user sequence and item sequence are sufficient to be applied in the proposed method, i.e., changes in user preferences and the item audience are contained. This dataset is also favorable for use with baseline methods.

Table 1  
Description of experimental dataset.

Dataset	Number of users	Number of items	Rating scale	Number of data
Movies and TV	14219	14299	[1, 5]	4607047

## 4.2 Baseline

The following algorithms are used to evaluate the effectiveness of the proposed DLGNN.

- 1) Bayesian probabilistic matrix factorization (BPMF):<sup>(30)</sup> The BPMF is based on the factorization of the probability matrix, which uses the Markov chain Monte Carlo (MCMC) method for approximate inference.
- 2) SVD++:<sup>(31)</sup> SVD++ is based on MF and implicit feedback. During the factorization of the score matrix of users, the implicit feedback of scores is also considered in this model.
- 3) TimeSVD:<sup>(32)</sup> On the basis of SVD++, the impact of time on the purchase is introduced.
- 4) Factorization machines (FMs):<sup>(33)</sup> The FMs are a type of machine learning algorithm based on MF, which can be used for regression prediction of user scores. The crossing relations of different characteristics are expressed by implicit vectors.

## 4.3 Evaluation metrics

The indicators used to evaluate the DLGNN and baseline methods are defined as follows:

- 1) RMSE: The root mean square error (RMSE) indicates the bias between predicted values and real values; this is a commonly used evaluation indicator.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (14)$$

- 2) MAE: The mean absolute error (MAE) is expressed as

$$MAE = \sqrt{\frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|}. \quad (15)$$

The actual situation of the predicted error can be better indicated using the MAE.

## 4.4 Performance evaluation

The following parameter setup is used in the experiments conducted. The dimension of the trained vectors is 128 when training the representation vectors for users and commodities. The lengths of purchase sequences are set to 15. The length of the GNN output vector after processing the sequence data is set to 128. A Gaussian distribution is used to initiate the parameters of the neural network; the Adam optimizer with a learning rate of 0.001 is used

Table 2  
Performances of DLGNN and baseline methods.

Algorithm	RMSE	MAE
BPMF	1.2996	0.925
SVD++	0.992	0.731
TimesSVD	1.072	0.772
FM	1.142	0.903
DLGNN	0.915	0.691

to train these parameters. A batch of data has a size of 256. 80% of the data are trained as training samples and the other 20% of the data are used as test samples for further analysis. The performances of the DLGNN and baseline methods on the dataset are shown in Table 2.

From Table 2, the prediction result of the DLGNN yields the smallest RMSE and MAE. The performance of BPMF is the lowest owing to the fact that no latent information is utilized. FM, SVD++, and TimeSVD introduce latent vectors into the model, which increase the amount of information in the model and improve the accuracy of the results by training implicit information. This comparison illustrates that the more information used in the model, the better the result achieved. Although TimeSVD utilizes information of the purchase time, this algorithm does not work well because it does not consider changes in user preferences. A user's old shopping record interferes with the prediction of new items owing to changes in user preferences. This is a drawback of methods based on the user-item-score MF method. These methods rely too much on the logic in the matrix and cannot capture the transfer of user interest. Although FM introduces more latent information, the choice of features of the data has a major impact on the results. The DLGNN constructs purchase sequences from the user-item-score matrix to analyze changes in user preferences and changes in product audiences. This allows it to capture external changes in users and commodities, resulting in higher accuracy of the model.

#### 4.5 Analysis of representation module

In the proposed DLGNN method in this paper, the pretrained representation vectors of users and commodities are used as global information; this information and the output vectors of the GNN are combined to generate the prediction for user scores. The specified impact of representation vectors on the final results is analyzed. The full DLGNN model with global representation vectors and the fragmented DLGNN model without global representation vectors are separately constructed for the following discussion. The fragmented DLGNN model directly splices two GNN outputs, and linear regression is used to obtain predictions. Changes in the loss of these two algorithms are shown in Fig. 4.

#### 4.6 Analysis of sequence length

The length of the purchase sequence is expected to affect the convergence and performance of the training process. Purchase sequences with three different lengths are used here to train the proposed model, and the detailed information is listed in Table 3.

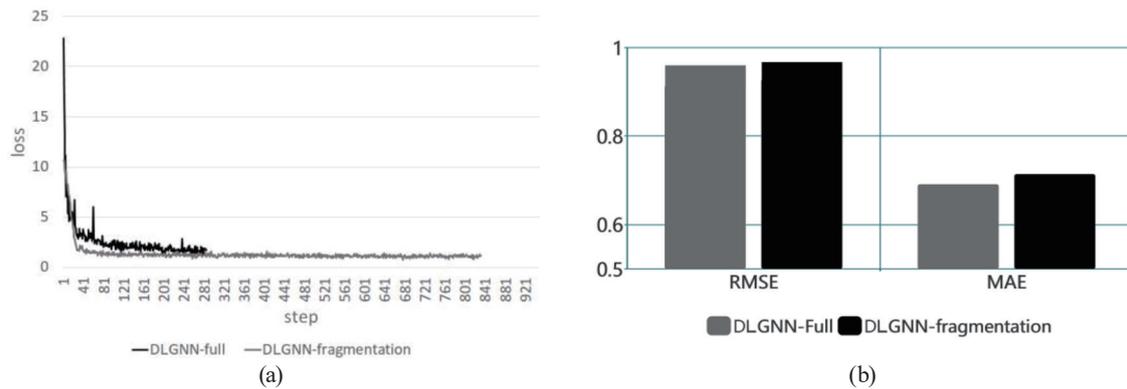


Fig. 4. (a) Loss and (b) performance of two algorithms.

Table 3  
Differences among variants of DLGNN.

Variant of DLGNN	User sequence length	Item sequence length	RMSE	MAE
DLGNN-L55	5	5	0.923	0.694
DLGNN-L1515	15	15	0.915	0.691
DLGNN-L2525	25	25	0.994	0.703

It can be seen from the results that too long and too short sequences have a negative impact on the model. When the sequence is too short, the user's purchase preference information and the audience information of the product contained in the sequence are insufficient. The transfer of users and items occurs more than once during a long sequence, which shifts the output of the final GNN. In our follow-up work, we will additionally focus on the processing of some long purchase sequence dataset. The suitable sequence length is not fixed and should be adjusted according to actual conditions. In the case of a single product type, the sequence length should be set shorter. If the variety of goods is large, the sequence length should be set longer.

## 5. Conclusion

A recommendation model, DLGNN, based on a GNN is proposed in this paper. The user–purchased-item sequence and purchased-item–user sequence are both used to construct the graph structure, and the GNN is applied to model these sequences to probe the transformation pattern between different nodes. On this basis, the pretrained embedding vectors of users and commodities are used to represent their global information. Experiments based on a real dataset show that the proposed method yields better performance than existing recommendation algorithms.

## References

- 1 Y. Cao, X. Wang, X. He, Z. Hu, and T. Chua: The World Wide Web Conf. (2019) 151. <https://doi.org/10.1145/3308558.3313705>
- 2 M. Pazzani and D. Billsus: The Adaptive Web (Springer-Verlag, 2007) p. 325. [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
- 3 P. Lops, M. D. Gemmis, and G. Semeraro: Recommender Systems Handbook (Springer US, 2011) p. 73. <https://doi.org/10.1007/978-0-387-85820-3-3>
- 4 B. Sarwar, G. Karypis, J. Konstan, and J. Riedl: Proc. 10th Int. Conf. World Wide Web (2001) 285. <https://doi.org/10.1145/371920.372071>
- 5 B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk: Browse State-of-the-Art (2015). <http://arxiv.org/abs/1511.0693>.
- 6 M. Fu, H. Qu, Z. Yi, L. Lu and Y. Liu: IEEE Trans. Cybern. **49** (2019) 1084. <https://doi.org/10.1109/TCYB.2018.2795041>
- 7 A. V. d. Oord, S. Dieleman, and B. Schrauwen: Proc. 26th Int. Conf. Neural Information Processing Systems **2** (2013) 2643. <https://dl.acm.org/doi/10.5555/2999792.2999907>
- 8 S. Li, J. Kawale, and Y. Fu: Proc. 24th ACM Int. Conf. Information and Knowledge Management (2015) 811. <https://doi.org/10.1145/2806416.2806527>
- 9 H. Zhang, T. W. S. Chow, and Q. M. J. Wu: IEEE Trans. Neural Networks Learning Syst. **27** (2016) 2537. <https://doi.org/10.1109/TNNLS.2015.2496281>
- 10 H. Ma, H. Yang, M. R. Lyu, and I. King: Proc. 17th ACM Conf. Information and Knowledge Management (2008) 931. <https://doi.org/10.1145/1458082.1458205>
- 11 S. Wu, Y. Tang, Y. Zhu, L. Wang, and T. Tan: Proc. AAAI Conf. Artificial Intelligence (2018). <https://doi.org/10.1609/aaai.v33i01.3301346>
- 12 G. Eason, B. Noble, and I. N. Sneddon: Philos. Trans. R. Soc. London, Ser. A **247** (1955) 529. <https://doi.org/10.1098/rsta.1955.0005>
- 13 G. Adomavicius and A. Tuzhilin: IEEE Trans. Knowl. Data Eng. **17** (2005) 734. <https://doi.org/10.1109/TKDE.2005.99>
- 14 M. Claypool, A. Gokhale, T. Mir, P. Murnikov, and M. Sartin: Proc. 6th Int. Workshop on Information Retrieval with Asian Languages **11** (2003) 17. <https://doi.org/10.3115/1118935.1118938>
- 15 M. J. Pazzani: Artif. Intell. Rev. **13** (1999) 393. <https://doi.org/10.1023/A:1006544522159>.
- 16 C. Zhang, L. Yu, X. Zhang and N. Chawla: IEEE Int. Conf. Big Data (2017) 857. <https://doi.org/10.1109/BigData.2017.8258001>
- 17 X. He, L. Liao, L. Nie, X. Hu, and T. S. Chua: Proc. 26th Int. Conf. World Wide Web (WWW, 2017) 173. <https://doi.org/10.1145/3038912.3052569>
- 18 Q. Li, X. Zheng, and X. Wu: Neural Collaborative Autoencoder (2017). <https://arxiv.org/abs/1712.09043>
- 19 W. Huang, Z. Wu, C. Liang, P. Mitra, and C. L. Giles: 29th AAAI Conf. Artificial Intelligence (2015) 2404. <https://dl.acm.org/doi/10.5555/2886521.2886655>
- 20 Y. Gong and Q. Zhang: Proc. 25th Int. Joint Conf. Artificial Intelligence (2016) 2782. <https://dl.acm.org/doi/10.5555/3060832.3061010>
- 21 T. Bansal, D. Belanger, and A. McCallum: ACM (2016) 107. <https://doi.org/10.1145/2959100.2959180>
- 22 P. Covington, J. Adams, and E. Sargin: Proc. 10th ACM Conf. Recommender Systems (2016) 191. <https://doi.org/10.1145/2959100.2959190>
- 23 F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini: IEEE Trans. Neural Networks **20** (2008) 61. <https://doi.org/10.1109/TNN.2008.2005605>
- 24 Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel: Gated Graph Sequence Neural Networks (2015). <https://arxiv.org/abs/1511.05493>
- 25 J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun: arXiv Preprint (2018). <https://arxiv.org/abs/1812.08434>.
- 26 R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec: Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2018) 974. <https://arxiv.org/abs/1806.01973>
- 27 A. Grover and J. Leskovec: Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2016) 855. <https://doi.org/10.1145/2939672.2939754>
- 28 T. Kenter, A. Borisov, and M. Rijke: Proc. 54th Annu. Meeting of the Association for Computational Linguistics **1** (2016) 941. <https://doi.org/10.18653/v1/P16-1089>
- 29 R. He and J. McAuley: International World Wide Web Conf. Steering Committee (2016) 507. <https://doi.org/10.1145/2872427.2883037>
- 30 R. Salakhutdinov and A. Mnih: Proc. 25th Int. Conf. Machine Learning (2008) 880. <https://doi.org/10.1145/1390156.1390267>

- 31 Y. Koren: Proc. 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2008) 426. <https://doi.org/10.1145/1401890.1401944>
- 32 Y. Koren: Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2009) 447. <https://doi.org/10.1145/1557019.1557072>
- 33 S. Rendle: 2010 IEEE Int. Conf. Data Mining (2010) 995. <https://doi.org/10.1109/ICDM.2010.127>

### About the Authors



**Chuyuan Wei** received his Ph.D. degree from Beijing Institute of Technology, China, in 2016. From 2016, he has been an assistant professor at Beijing University of Civil Engineering and Architecture. His research interests are in big data and smart cities.



**Jianwei Niu** received his Ph.D. degree from Beihang University, China, in 2002. From 2004, he was an assistant professor at Beihang University. Since 2007, he has been a professor at Beihang University. His research interests are in mobile computing and wireless sensors.



**Yanyan Guo** received her M.S. degree from Beihang University, China, in 2019. Her research interests are in data mining and recommendation systems. ([guoyanyan@cmbchina.com](mailto:guoyanyan@cmbchina.com))