



Representation Learning: Recommendation With Knowledge Graph via Triple-Autoencoder

Yishuai Geng¹, Xiao Xiao^{2*}, Xiaobing Sun^{1*} and Yi Zhu¹

¹School of Information Engineering, Yangzhou University, Yangzhou, China, ²Department of Ultrasound, Affiliated Hospital of Yangzhou University, Yangzhou, China

OPEN ACCESS

Edited by:

Yucong Duan,
Hainan University, China

Reviewed by:

Peng Zhou,
Anhui University, China
Junwei Lv,
Hefei University of Technology, China
Lei Li,
Hefei University of Technology, China

*Correspondence:

Xiao Xiao
092298@yzu.edu.cn
Xiaobing Sun
xbsun@yzu.edu.cn

Specialty section:

This article was submitted to
Computational Genomics,
a section of the journal
Frontiers in Genetics

Received: 07 March 2022

Accepted: 11 April 2022

Published: 03 June 2022

Citation:

Geng Y, Xiao X, Sun X and Zhu Y
(2022) Representation Learning:
Recommendation With Knowledge
Graph via Triple-Autoencoder.
Front. Genet. 13:891265.
doi: 10.3389/fgene.2022.891265

The last decades have witnessed a vast amount of interest and research in feature representation learning from multiple disciplines, such as biology and bioinformatics. Among all the real-world application scenarios, feature extraction from knowledge graph (KG) for personalized recommendation has achieved substantial performance for addressing the problem of information overload. However, the rating matrix of recommendations is usually sparse, which may result in significant performance degradation. The crucial problem is how to extract and extend features from additional side information. To address these issues, we propose a novel feature representation learning method for the recommendation in this paper that extends item features with knowledge graph via triple-autoencoder. More specifically, the comment information between users and items is first encoded as sentiment classification. These features are then applied as the input to the autoencoder for generating the auxiliary information of items. Second, the item-based rating, the side information, and the generated comment representations are incorporated into the semi-autoencoder for reconstructed output. The low-dimensional representations of this extended information are learned with the semi-autoencoder. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder. A serial connection between the semi-autoencoder and the autoencoder is designed here to learn more abstract and higher-level feature representations for personalized recommendation. Extensive experiments conducted on several real-world datasets validate the effectiveness of the proposed method compared to several state-of-the-art models.

Keywords: personalized recommendation, autoencoder, semi-autoencoder, representation learning, collaborative filtering

1 INTRODUCTION

The success of machine learning algorithms and artificial intelligence methods heavily depends on the feature representation learning of original data (Bengio et al., 2013; Zhuang et al., 2017a). In recent decades, feature representation learning has attracted a vast amount of attention and research from multiple disciplines, such as biomedicine and bioinformatics (Wei et al., 2019; Li et al., 2021), computer vision (Kim et al., 2017), knowledge engineering (Liu et al., 2016), and personalized recommendation (Zhuang et al., 2017b; Zhu et al., 2021). In real-world applications, feature representation learning is considered to obtain the different explanatory factors of variation behind the data (Locatello et al., 2019).

For nearly three decades, effective computational methods have accelerated drug discovery and played an important role in biomedicine, such as predicting molecular properties and identifying interactions between drugs/compounds and their target proteins. In early years, quantum mechanics (Hohenberg and Kohn, 1964), such as density functional theory (DFT), was used to determine the molecular structure and calculate properties of interest for a molecule. However, the quantum computational method usually consumes tremendous computational resources and takes hours to days to calculate the molecular properties (Ramakrishnan et al., 2015), which hinders their applications to the fields of high-throughput screening. Nowadays, the powerful ability to learn representation and efficiently recommend algorithms has received significant attention. A key challenge is to learn useful molecular representation information from the huge molecular dataset.

Among all the informatics-related application scenarios, with the rapid development of the Internet, there is an urgent demand for personalized recommendation to tackle the information overload problem (Zhang et al., 2017). Notably, many successful recommendations systems share aspects of feature representation learning and have been widely applied in many online services such as electronic commerce (Ma et al., 2020) and social networks (Botangen et al., 2020). Existing methods for recommendation systems can roughly be categorized into three classes: content-based recommendation, collaborative filtering (CF), and hybrid methods (Batmaz et al., 2019). The content-based recommendation methods learn the descriptive features of items, calculate the similarity between new items and user-liked items based on these features, and generate the final recommendation (Lops et al., 2019). The collaborative filtering methods discover the inclinations of users by considering the user's historical behavior and produce recommendations (Dong et al., 2021). Hybrid recommendation methods leverage multiple approaches together and try to combine the advantages of these approaches.

Recently, collaborative filtering methods have achieved superior performance for the advantages of effectiveness and efficiency, which have far-ranging consequences in practical applications of recommendation systems (Su and Khoshgoftaar, 2009). Most of the traditional collaborative filtering methods are based on matrix factorization (MF), which combines good scalability with predictive accuracy (Luo et al., 2020). The main intuition behind these approaches is to decompose the rating matrix into user and item-based profiles, which allows the recommendation system to treat different temporal aspects separately (Yehuda et al., 2009). However, MF-based methods have inherent limitations in feature representation learning for the recommendation, which prevent further development of these approaches.

On the other hand, deep learning techniques have recently achieved great success in the computer vision and natural language processing fields. Such techniques show great potential in learning feature representations. Therefore, researchers have begun to apply deep learning methods to the field of recommendations (Salakhutdinov et al., 2007). They use a restricted Boltzmann machine instead of the traditional matrix

factorization to perform the CF, and Georgiev and Nakov,(2013) expanded the work by incorporating the correlation between users and between items. In addition, Wang et al. (2015), proposed a hierarchical Bayesian model that uses a deep learning model to obtain content features and a traditional CF model to address the rating information. These methods, based on deep learning techniques, more or less make recommendations by learning the content features of items. These methods are not applicable when we are unable to obtain the contents of items. Therefore, enhancing the effectiveness of feature learning is significant. Recent studies have shown that deep neural networks can learn more abstract and higher-level feature representations (Yi et al., 2018), which has made remarkable progress in improving recommendation performance (Chae et al., 2019). For example, He et al. (2017) proposed a general recommendation framework called Neural Network-based Collaborative Filtering, in which a deep neural network is utilized for learning the interaction between user and item features. As we can see, among all the deep neural network-based recommendation methods, many frameworks are realized on top of the autoencoder model, which is one of the most successful deep neural networks and has also been actively adopted as a CF model recently (Shuai et al., 2017; Zhuang et al., 2017c; Chae et al., 2019; Zhong et al., 2020). For example, Zhang et al. proposed a hybrid collaborative filtering framework based on an autoencoder that incorporated auxiliary information for semantic rich representations teaching (Shuai et al., 2017).

Though the autoencoder-based methods have achieved fairly good performance for personalized recommendation, there are two main problems that prevent the further development of these methods. The first is the utilization of auxiliary information from users or items, since the rating matrix in real-world applications is usually very sparse, which inevitably leads to a significant recommendation performance degradation. Most existing methods only introduce some obvious attributes, such as the age, gender, and occupation of users, or the title, release date, and genres of items. The key factors of collaborative filtering, such as the reviews of items by users, have rarely been incorporated into the autoencoder-based networks. The second problem is the optimization of neural networks. When training models to incorporate side information about items and users, the dimensions of the input and output layers are required to be equal in autoencoder-based networks, which greatly limits the scalability and flexibility of networks.

To address these problems, we propose a feature representation learning method for personalized recommendation in this paper which extends items features with knowledge graph via triple-autoencoder (KGTA for short). Specifically, the comment information between users and items is first encoded as sentiment classification. These features are then applied as the input to the autoencoder for generating the auxiliary information of items, which can be used to introduce the comment information from users to items to solve the incorporating problem of auxiliary information. Secondly, the item-based rating, the side information, and the generated comment representations are incorporated into the

semi-autoencoder for reconstructed output. It aims to address the second problem, that the dimensions of the input and output layer are required to be equal. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder for personalized recommendation. Experimental results on several datasets demonstrate the effectiveness of our proposed method compared to other state-of-the-art matrix factorization methods and deep-based methods.

In summary, the main contributions of our work can be distilled into the following:

- To incorporate the key information between users and items, the comments from each user for item are encoded and reconstructed as the auxiliary information
- To optimize the neural networks, a serial connection of semi-autoencoders and autoencoders are designed to learn more abstract and higher-level feature representations for personalized recommendation
- Extensive experiments on several datasets were conducted to confirm the effectiveness of the proposed method compared to other state-of-the-art matrix factorization methods and deep-based methods

2 RELATED WORK

In this section, we survey the related works of feature representation learning, personalized recommendation methods, and collaborative filtering^{1,2}.

2.1 Feature Representation Learning

Feature representation learning refers to learning data representations that make it easier to extract useful information in downstream machine learning tasks (Bengio et al., 2013). The last decades have witnessed a vast amount of research and application on feature representation learning in multiple disciplines. For example, in the field of biomedicine and bioinformatics, Wei et al. (2019) developed a bioinformatics tool for the generic prediction of therapeutic peptides. An adaptive feature representation learning method is proposed for different peptide types in the tool. Alshahrani et al. (2017) proposed a knowledge representation learning method with symbolic logic and automated reasoning, which can be applied to biological knowledge graphs for tasks such as finding candidate genes for diseases and protein-protein interactions. Li et al. (2021) proposed a triplet message mechanism to learn molecular representation based on graph neural networks, which can complete molecular property prediction and compound-protein interaction identification with few parameters and high accuracy.

Besides the fields of biomedicine and bioinformatics, feature representation learning has also been widely applied in other fields such as computer vision (Kim et al., 2017), knowledge

engineering (Liu et al., 2016) and personalized recommendation (Zhuang et al., 2017b). For example, Wang et al. proposed a high-resolution representation learning network for visual recognition problems (Wang et al., 2020), which can maintain the representation being semantically strong and spatially precise. Xu et al. (2018) proposed an aggregation method for node representation learning that can adapt neighborhood ranges to nodes. It is especially suitable for graphs that have subgraphs with diverse local structures. Niu et al. (2020) proposed a rule and path-based joint embedding method for representation learning on knowledge graphs. The Horn rules and paths are leveraged in this method to enhance the accuracy and explainability of representation learning.

2.2 Personalized Recommendation

In recent decades, with the rapid development of the Internet, personalized recommendations have provoked a vast amount of attention and research (Qian et al., 2013). The advances in personalized recommendation have far-ranging consequences in many online services applications such as electronic commerce (Ma et al., 2020) and social networks (Li et al., 2017). For example, in Facebook, Gupta et al. (2020) conducted a detailed performance analysis of recommendation models on server-scale systems present in the data center. Botangen et al. (2020) proposed a probabilistic matrix factorization-based recommendation method that considers geographic location information for designing an effective and efficient Web service recommendation.

Good feature representations of data do contribute to many machine learning tasks, such as personalized recommendation. For example, Geng et al. (2015) proposed a deep method to learn the unified feature representations for both users and images. This representation from large, sparse, and diverse social networks obviously improves the recommendation performance. Liu et al. (2019) proposed a joint representation learning method for multimodal transportation recommendations, which aims to recommend a travel plan that considers various transportation modes. Ni et al. proposed a recommendation model based on deep representation teaching (Ni et al., 2021). It contained information preprocessing and feature representation modules to generate the primitive feature vectors and the semantic feature vectors of users and items, respectively.

2.3 Collaborative Filtering

In personalized recommendations, the collaborative filtering (CF) methods aim to discover users' preferences through the interactions between users and items. Existing CF methods can be roughly categorized into two classes: matrix factorization methods and deep neural network methods.

In the matrix factorization methods, these methods have difficulty in processing sparse data and have poor generalization ability, but they have low time and space complexity and good scalability. Lee et al. proposed the classical non-negative matrix factorization (NMF) model (Lee and Seung, 2001), which can decompose the rating matrix into user and item profiles. Along this line, Sun et al. proposed a

¹<http://files.grouplens.org/datasets/movielens/ml-100k.zip>.

²<http://files.grouplens.org/datasets/movielens/ml-1m.zip>.

Probabilistic Matrix Factorization (PMF) model that scales linearly with the number of observations and performs well on very sparse and imbalanced datasets (Salakhutdinov and Mnih, 2007). In light of PMF, Salakhutdinov et al. also proposed a Bayesian Probabilistic Matrix Factorization (BPMF) model (Salakhutdinov and Mnih, 2008), which controlled model capacity automatically by placing hyper-priors over the hyper-parameters to avoid over-fitting. Koren proposed combining the factor and neighborhood models for a more accurate recommendation performance (Koren, 2008), which further extends the model to exploit both explicit and implicit feedback by the users. In recent years, to address the problem that the attributes of users are often scarce for reasons of privacy, Rashed et al. (2019) proposed a nonlinear co-embedding GraphRec model, which treats the user-item relation as a bipartite graph and constructs generic user and item attributes via the Laplacian of the user-item co-occurrence graph.

Recently, due to the powerful ability of deep learning methods, remarkable progress has been made in learning higher-level and abstract representations for personalized recommendations (Wang et al., 2015; Yu et al., 2019). These methods have nonlinear transformation and powerful representation learning ability, but poor interpretability, large data requirements, and extensive hyper-parameter tuning. For example, He et al. (2017) proposed a general recommendation framework that designs a deep neural network to learn the interaction between a user and item features. Meanwhile, to address the cold start problem and improve performance for personalized recommendations, Ni et al. (2022) proposed a two-stage embedding model to improve recommendation performance with auxiliary information. In this method, two sequential stages, graph convolutional embedding and multimodal joint fuzzy embedding, are designed to fully exploit item multimodal auxiliary information. Among all the deep learning methods for personalized recommendation, we realize many successful frameworks on top of the autoencoder, which is one of the most successful deep neural networks and has also been actively adopted as a CF model recently (Shuai et al., 2017; Zhuang et al., 2017c; Chae et al., 2019; Zhong et al., 2020). For example, Zhuang et al. (2017c) proposed a dual-autoencoder model for recommendation, which simultaneously learns the user-based and item-based features with the autoencoder model. Zhu et al. (2021) proposed a collaborative autoencoder model for personalized recommendation, which learns the hidden features of users and items with two different autoencoders for capturing different characteristics of the data.

3 PRELIMINARIES

3.1 Autoencoder

The autoencoder model aims to minimize the distance between the input and the reconstructed output. The basic autoencoder network (Bengio, 2009) generally consists of an input layer, an output layer, and one or more hidden layers. Given the input as $x \in \mathfrak{R}^{m \times n}$, when there is only one hidden layer, the encoding and decoding layer of autoencoder can be represented as follows:

$$\xi = f(Wx + b), \quad (1)$$

$$x' = g(W'\xi + b'), \quad (2)$$

where $W \in \mathfrak{R}^{k \times m}$, $W' \in \mathfrak{R}^{m \times k}$ and $b \in \mathfrak{R}^{k \times 1}$, $b' \in \mathfrak{R}^{m \times 1}$ are the weighting matrices and bias vectors, respectively. f and g are the nonlinear activation functions of the encode and decode layers, respectively. In our experiments, the sigmoid and identity functions are introduced as f and g . The objective function of the autoencoder can be shown as follows:

$$\min_{W, b, W', b'} J_r = \|x' - x\|^2. \quad (3)$$

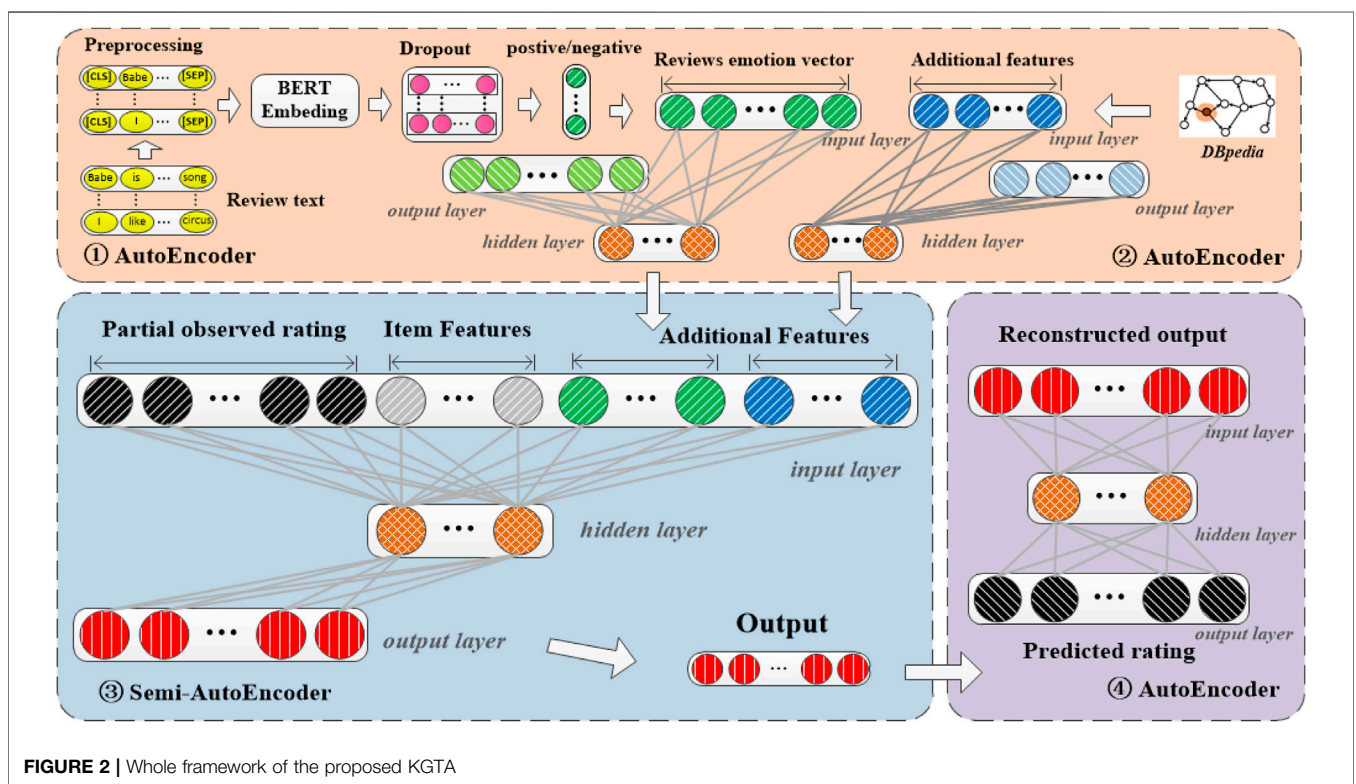
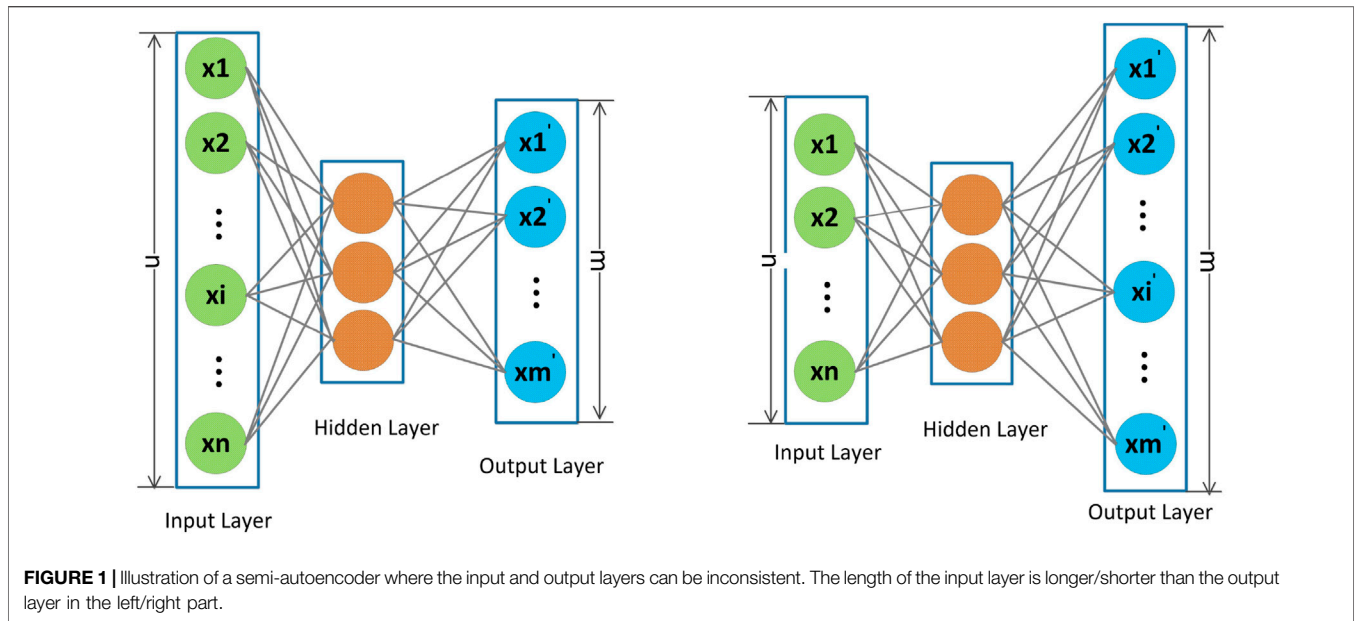
3.2 Semi-Autoencoder

In recent years, many autoencoder-based recommendation methods have achieved fairly good results with the advantages of no labeling requirement and fast convergence speed. However, the classic autoencoder model has the restriction that the dimensions of the input and the output layer must be equal, which has a great impact on introducing auxiliary information for solving the sparse problem of the rating matrix.

To address this problem, a semi-autoencoder model was proposed and generalized into a hybrid CF method for rating prediction (Shuai et al., 2017). Compared with traditional autoencoders, the input layer of semi-autoencoders is longer than the output layer, so semi-autoencoders can be utilized to capture different nonlinear feature representations and reconstructions flexibly by extracting different subsets from the inputs, and it is easy to incorporate side information into the input layer effectively to improve the item feature representation for better recommendation performance. The whole framework of the semi-autoencoder is shown in **Figure 1**, the left and right parts of **Figure 1** show the two cases in which the output layer is longer than the input layer and the output layer is shorter than the input layer, respectively. We observe that the basic framework of a semi-autoencoder is the same as that of a classical autoencoder model, which also includes an input layer, an output layer, and one or more hidden layers. Furthermore, in the right part of **Figure 1**, we can observe that the shorter output layer is the reconstruction of certain parts of the input, and the remaining part in the semi-autoencoder model is auxiliary information to learn better feature representations for addressing the sparse problem of the rating matrix.

4 METHODOLOGY

The whole framework of our proposed recommendation method with knowledge graph via triple-autoencoder (KGTA for short) is illustrated in **Figure 2**, which encompasses three main components. The first one is the representational learning of the comment information between users and items. The comments from users on each item are divided into positive and negative categories. Then the first autoencoder was introduced to reduce the dimensionality of this comment information. The second one is the learning of all the auxiliary information. A semi-autoencoder is utilized to incorporate the side information, the extended features from the knowledge graph, and the generated comment features into the item-based rating. Finally, the low-dimensional output of the semi-autoencoder is input into the



third autoencoder. Different from the semi-autoencoder model that only approximates the item-based rating; the third component tries to reconstruct all the input for the recommendation^{3,4}.

In the following, first, the commonly used notations in this paper are listed in **Table 1**, and then, the model of KGTA is described in detail.

4.1 Notations

Some important notations used in this paper and their descriptions are listed in **Table 1**.

³<http://www.librec.net/download.html>.
⁴<http://github.com/hoyeople/MeLU>.

TABLE 1 | Important notations used in this article and their descriptions.

Notations	Descriptions
R	The rating matrix
A	The attributes vectors of all items
S	The reconstructed comment vectors of all items
L	The language vectors of all items
R'	The prediction matrix $R' \in \mathbb{R}^{n \times m}$
m	The number of users
n	The number of items
r^i	The column of rating matrix
r^j	The row of rating matrix
k	The features dimension of hidden units
h	The number of hidden units
x_i	The i th instance of original input
x'_i	The reconstructed output of x_i
ξ	The hidden feature representation matrix
W, W'	The map and remap weight matrix
b, b'	The map and remap bias vectors
\bullet	The element-wise product of vectors or matrices

4.2 Comment Information Features

The personalized recommendation is to predict the interest of a user in an item based on the rating matrix information. Since the rating matrix in real-world scenarios is usually very sparse, many methods have introduced auxiliary information to address this problem. However, most existing methods only introduce some obvious attributes and ignore the key factors, such as the comments from users on each item, of collaborative filtering. To address this problem, our method learns the comment information features between users and items with the first autoencoder. The details can be seen in the upper left of **Figure 2**.

In our method, we take natural language text as the input for sentiment classification and output emotion score $\in \{1, -1\}$. -1 represents negative emotion and 1 represents positive emotion. Our method has two stages from input sentence to output score, which are described below.

In the first stage, we perform the following preprocessing steps on the comment text before we feed it into the model. First, we remove all the digits, punctuation symbols, and accent marks, and convert everything to lowercase. Secondly, we then tokenize the text using the WordPiece tokenizer (Schuster and Nakajima, 2012). It breaks the words down into their prefix, root, and suffix to better handle unseen words. Finally, we add the [CLS] and [SEP] tokens at the appropriate positions.

In the second stage, we build a simple architecture with just a dropout regularization (Srivastava et al., 2014) and a softmax classifier layer on top of the pretrained BERT layer. The upper left corner of **Figure 2** shows the overall architecture of our sentiment classification model. There are four main stages. The first is the processing step, as described earlier. Then we compute the sequence embedding from BERT. We then apply a dropout with a probability factor of 0.1 to regularize and prevent over-fitting. Finally, the softmax classification layer will output the probabilities of the input text belonging to each of the class labels such that the sum of the probabilities is

1. The softmax layer is just a fully connected neural network layer with the softmax activation function. The output node with the highest probability is then chosen as the predicted label for the input.

Given the rating matrix $R \in \mathbb{R}^{m \times n}$, where m and n denote the number of users and items respectively. For each item, the comments from each user are classified by sentiment using BERT (Devlin et al., 2018) first, and then we obtain the comment feature vector c_i for each item. Since the comment information from users to items is usually sparse, just like the rating matrix, the first autoencoder was introduced for feature dimensionality reduction and representation learning. The process of the first autoencoder can be shown as follows:

$$\xi_s = f(W_s C + b_s), \quad (4)$$

$$s = g(W'_s \xi_s + b'_s), \quad (5)$$

where $W_s \in \mathbb{R}^{k_1 \times n}$ and $W'_s \in \mathbb{R}^{n \times k_1}$ are the weighting matrices, $b_s \in \mathbb{R}^{k_1 \times 1}$ and $b'_s \in \mathbb{R}^{n \times 1}$ are the bias vectors, f and g are the functions of nonlinear activation, and k_1 is the feature dimension of hidden units. The hidden features of the first autoencoder, i.e., the low-dimensional representations of s , are denoted as S^I , which are incorporated into the second semi-autoencoder for capturing different representations and reconstructions by sampling different subsets from all the inputs.

4.3 Co-Embeddings With the Semi-Autoencoder

After obtaining the reconstructed comment features, a semi-autoencoder is introduced to incorporate the item rating vector r_i and other auxiliary information such as attributes vector a_i , reconstructed comment features s_i , and the KG-extended features l_i . The input of the semi-autoencoder can be defined as $con(r_i, a_i, s_i, l_i)$

$$con(r_i, a_i, s_i, l_i) = \text{connection of } r_i, a_i, s_i, \text{ and } l_i. \quad (6)$$

The $con(R^I, A^I, S^I, L^I) \in \mathbb{R}^{n \times (m+y+k_1+k_2)}$ refers to the connection of R^I , A^I , S^I and L^I , where $R^I \in \mathbb{R}^{n \times m}$ represents the item-based rating vectors, $A^I \in \mathbb{R}^{n \times y}$ represents the attribute vectors of all items, which are the obvious attributes such as the title, release date, and genres in movie recommendation datasets, $S^I \in \mathbb{R}^{n \times k_1}$ represents the reconstructed comment features for all n items, $L^I \in \mathbb{R}^{n \times k_2}$ represents the language vectors collected from the knowledge graph and autoencoder. Considering that the experiments are conducted on MovieLens datasets, the languages of the movies are obtained from open KGs such as DBpedia, and the languages are encoded with the multi-hot method and input into the autoencoder model for learning the hidden representations L^I . The process of L^I learning is consistent with that of S^I , the details can be seen in the upper right of **Figure 2**.

Then the $con(R^I, A^I, S^I, L^I)$ is input into the second autoencoder, i.e. a semi-autoencoder, to learn the compressed

reconstructed output, the encode stage of the semi-autoencoder can be defined as (7)

$$\xi = f(W \text{con}(R^l, A^l, S^l, L^l) + b), \quad (7)$$

where $W \in \mathfrak{R}^{(m+y+k_1+k_2) \times k}$ and $b_1^l \in \mathfrak{R}^k$ are the weight matrix and bias item, respectively, k is the feature dimension of the hidden layer, and f is the sigmoid function for nonlinear activation. Then, the decode stage can be shown as follows:

$$R'_{semi} = g(W' \xi + b'). \quad (8)$$

Similarly, where $W' \in \mathfrak{R}^{k \times m}$ and $b'_2 \in \mathfrak{R}^m$ are the weight matrix and bias item of decoding layer respectively, g is the identity function for the activation function. Notably, the SGD (stochastic gradient descent) method is utilized in the semi-autoencoder for model optimization. The details can be seen in the bottom left of **Figure 2**.

4.4 Triple-Autoencoder for Recommendation

From **Eqs. 7, 8**, we can obviously observe that the output of a semi-autoencoder model is the reconstruction of a certain part of the inputs. When computing the loss function, the result of the semi-autoencoder is a reconstruction of the rating matrix R^l instead of the whole input $\text{con}(R^l, A^l, S^l, L^l)$, which may result in a performance degradation for recommendation. To this end, we design the third autoencoder model to learn the reconstruction of the whole input, that is triple-autoencoder for the recommendation. The encode and decode stage of the triple-autoencoder can be shown as follows:

$$R' = g(W'_t f(W_t R'_{semi} + b_t) + b'_t). \quad (9)$$

To avoid over-fitting, the ℓ_2 norm regularization of the weight matrix W_t and W'_t is added to the objective function, which can be shown as follows:

$$J_r = (\|W_t\|_2^2 + \|W'_t\|_2^2). \quad (10)$$

Thus, the objective function of the triple-autoencoder can be shown as follows:

$$J_{item} = \|(R' - R'_{semi})\|^2 + \alpha J_r, \quad (11)$$

where α is the trade-off parameter that controls the balance of regularization terms. To minimize the distance between the input R'_{semi} and the output R' , the deviations are minimized to obtain representations for the recommendation. When the model converges, the output layer of the triple-autoencoder is the prediction matrix R' for the recommendation, the details can be shown in the bottom right of **Figure 2**. Details of the proposed KGTA are summarized in **Algorithm 1**.

Algorithm 1. Recommendation with knowledge graph via triple-autoencoder (KGTA)

Input: The rating matrix $R \in \mathfrak{R}^{u \times o}$, trade-off parameter α , the number of latent feature dimensions k , the number of hidden units h , and the dimension of item attribute vector, comment vector and languages auxiliary vector is y , k_1 and k_2 .

Output: The predicated rating matrix R' .

- 1: Get the additional information vector a_i for each item;
- 2: Input the comment between users and each item into the BERT model, and get the raw comment vector c_i from the sentiment classification model;
- 3: Get the reconstructed comment features s_i by Eq. (5);
- 4: Get the language information for each item from open KG and encode with multi-hot method;
- 5: Get the reconstructed language vectors l_i ;
- 6: Input the concatenation vectors $\text{con}(R^l, A^l, S^l, L^l)$ of the items rating vector R^l , attribute vector A^l , comment features S^l and languages features L^l into semi-autoencoder;
- 7: Initialize W , b and W' , b' randomly respectively;
- 8: Minimize $\|R'_{semi} - R^l\|_2^2$ with SGD method until convergence;
- 9: Input the reconstructed vectors R'_{semi} into autoencoder;
- 10: Initialize W_t , b_t and W'_t , b'_t randomly respectively;
- 11: Minimize Eq.(11) with SGD method until convergence;
- 12: **return** The predicated rating matrix R' ;

5 EXPERIMENTS

In this section, experiments are conducted on two datasets, MovieLens 100K and MovieLens 1M, to evaluate the effectiveness of our proposed KGTA. In the following, we first introduce the details of two experimental datasets. Secondly, the compared methods, including the MF-based and deep neural network-based methods, are given. In addition, the evaluation metrics such as MAE and RMSE are also presented. Then, the comparative experimental results and their observations are presented in detail. Finally, the main properties such as parameter sensitivity are analyzed for certain datasets.

5.1 Datasets

The details of two real-world datasets used in the experiments are listed in **Table 2**, including rating density, the number of users, items, and ratings.

MovieLens 100K1: it is a well-known and most widely applied dataset for evaluating recommendation performance. There are 943 users and 1,682 movies with 100,000 ratings on a scale of 1–5, and each user rated at least 20 movies. In MovieLens 100K, item attributes such as the title, release date, and genres of movies are also provided for improving recommendation performance.

MovieLens 1M2: It is an enlarged version of the MovieLens 100K dataset, which has also been widely applied in the recommendation. It has 6,040 users and 3,706 movies with 1,000,209 ratings. Similar to MovieLens 100K, the ratings are scaled from 1 to 5, and auxiliary information such as movie title, release date, and category are also provided.

5.2 Compared Methods and Evaluation Metrics

5.2.1 Compared Methods

To evaluate the effectiveness of the proposed KGTA, the following matrix factorization methods, meta-learning methods, and deep neural network methods were conducted:

- Non-negative matrix factorization (NMF) (Lee and Seung, 2001). It is the basic matrix factorization method for the

TABLE 2 | Details of the three datasets used in our experiments.

Dataset	Number of users	Number of items	Number of ratings	Rating density (%)
MovieLens 100K	943	1,682	100,000	6.3
MovieLens 1M	6,039	3,883	1,000,209	4.27

TABLE 3 | The performance of RMSE on MovieLens 100K and MovieLens 1M datasets.

Datasets	Methods	Proportion of training data			
		50%	60%	70%	80%
MovieLens 100K	-	50%	60%	70%	80%
	NMF	0.991	0.976	0.965	0.960
	SVD++	0.943	0.927	0.915	0.909
	MetaHIN	1.062	1.046	1.041	1.032
	MeLU	1.154	1.144	1.132	1.121
	AutoRec	1.023	1.003	0.981	0.964
	HCRSA	0.948	0.937	0.923	0.919
	PRKG	0.928	0.917	0.907	0.899
	KGTA	0.859	0.847	0.840	0.832
MovieLens 1M	NMF	0.928	0.925	0.921	0.918
	MetaHIN	1.024	0.993	0.965	0.959
	MeLU	1.082	1.038	1.008	0.973
	NCF	0.914	0.911	0.909	0.907
	AutoRec	0.914	0.905	0.896	0.888
	HCRSA	0.903	0.892	0.884	0.874
	PRKG	0.885	0.875	0.868	0.861
	KGTA	0.823	0.814	0.807	0.8

The bold values provided in Table 3 represent the experimental results of our proposed method (KGTA) and are the best results among all the comparison methods.

recommendation. In our experiments, we use the generalized Kullback–Leibler divergence as the update rules in NMF.

- Singular value decomposition plus (SVD++) (Koren, 2008). It exploits explicit and implicit feedback from users to combine the latent factor model and the neighborhood model into a unified model for the recommendation.
- Meta-learned user preference estimator (MeLU) (Lee et al., 2019). It estimates user preferences based on a small number of items to alleviate the cold start problem for the recommendation.
- Meta-learning method for cold start recommendation on Heterogeneous Information Networks (MetaHIN) (Lu et al., 2020). It creates a semantic-enhanced task constructor for exploring rich semantics, and a co-adaptation meta-learner with semantic- and task-wise adaptations within each task.
- Neural collaborative filtering (NCF) (He et al., 2017). It is a general recommendation framework that uses designs a deep neural network to learn the interaction between a user and item features.
- Item-based recommendation via autoencoder (AutoRec) (Sedhain et al., 2015). It is the first autoencoder framework in the recommendation, which learns the effective feature representations of items for collaborative filtering.
- Hybrid Collaborative Recommendation via Semi-Autoencoder (HCRSA) (Shuai et al., 2017). It is a hybrid

collaborative filtering framework based on the semi-autoencoder, which incorporates auxiliary information for semantic rich representation learning.

- Personalized recommendation with knowledge graph via dual-autoencoder (PRKG) (Yang et al., 2021). The side information of items is extracted from DBpedia and encoded into low-dimensional representations in this method, and a semi-autoencoder is introduced to incorporate this auxiliary information for the recommendation.

5.2.2 Implementation Details and Parameter Settings

The PREA toolkit (Lee et al., 2014) is adopted for the implementation of MF-based methods such as NMF and SVD++. For the methods of MeLU, MetaHIN, and HCRSA, we re-compile the source code as 4, 5, and 6. The default parameters of these three methods remain unchanged as reported in the original paper in the MovieLens dataset. For the method AutoRec, we select an item-based autoencoder that can achieve better performance than the user-based autoencoder. For fairness, the parameters of AutoRec and PRKG are consistent with ours in all two datasets. In our experiments, we set $\alpha = 0.1$ after some preliminary tests for all datasets. The maximum number of iterations in gradient descent is set at 300. The number of hidden units is set at 300 for all datasets^{5,6}.

5.2.3 Evaluation Metrics

In the experiments, we introduced root mean square error (RMSE) to measure the performance of our proposed KGTA and all compared methods in the recommendation, which can be shown as (12). It is worth mentioning that the smaller value of RMSE indicates better results.

$$\text{RMSE} = \sqrt{\frac{\sum_{r_{u,i} \in \text{TestSet}} (r_{u,i} - r'_{u,i})^2}{|\text{TestSet}|}}, \quad (12)$$

where $r_{u,i}$ and $r'_{u,i}$ represent the original rating matrix and the predication matrix, respectively.

5.3 Experimental Results

For each data set, the percentages of 50%, 60%, 70%, and 80% are sampled into training data, respectively, and the rest are used for test data. The experimental results of RMSE on the MovieLens 100K and MovieLens 1M datasets are recorded in **Table 3** and **Figures 3, 4** respectively. Notably, all the results are obtained by

⁵<https://github.com/rootlu/MetaHIN>.

⁶<https://github.com/cheungdaven/semi-ae-recsys>.

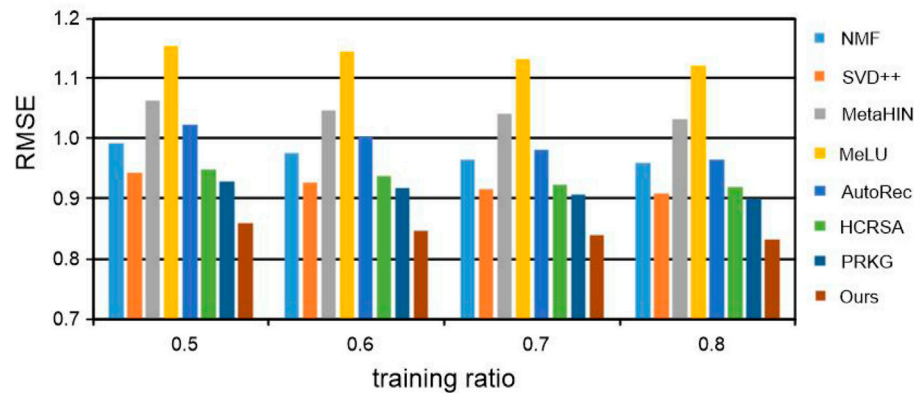


FIGURE 3 | RMSE of our KGTA and compared methods on the MovieLens 100K dataset.

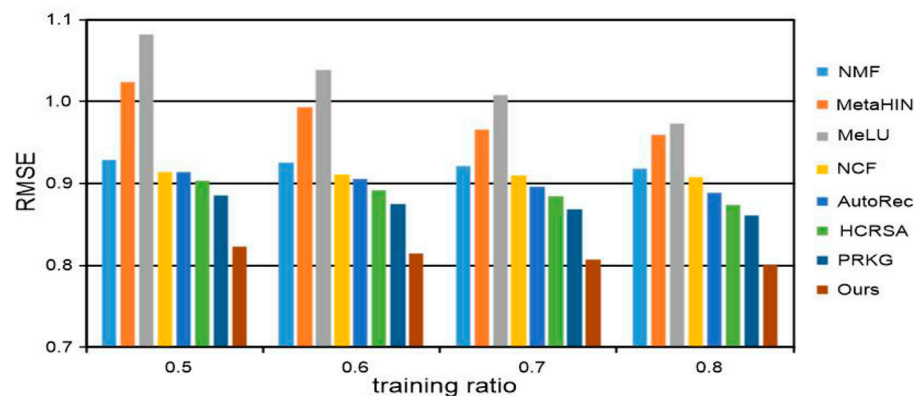
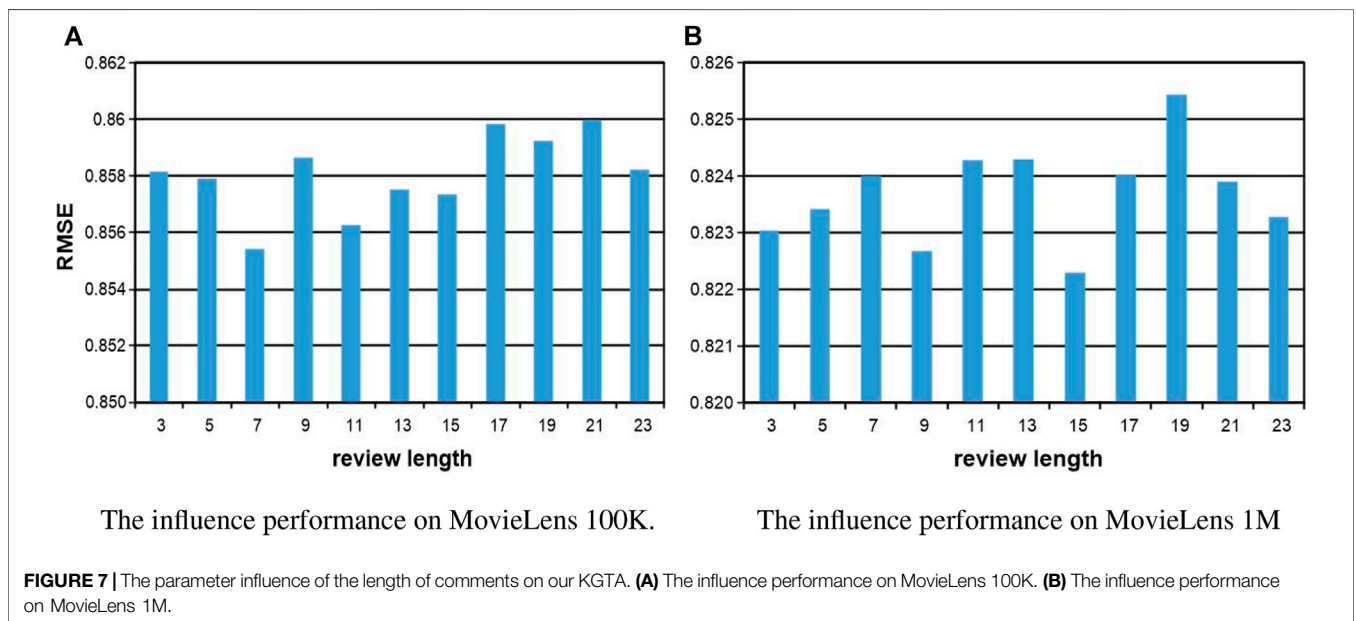
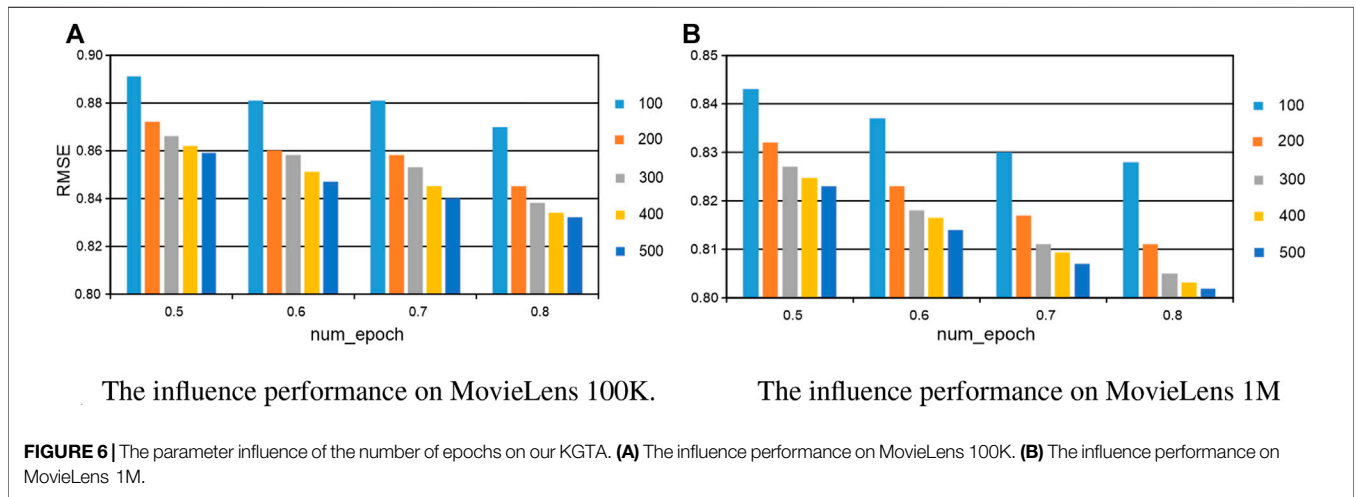
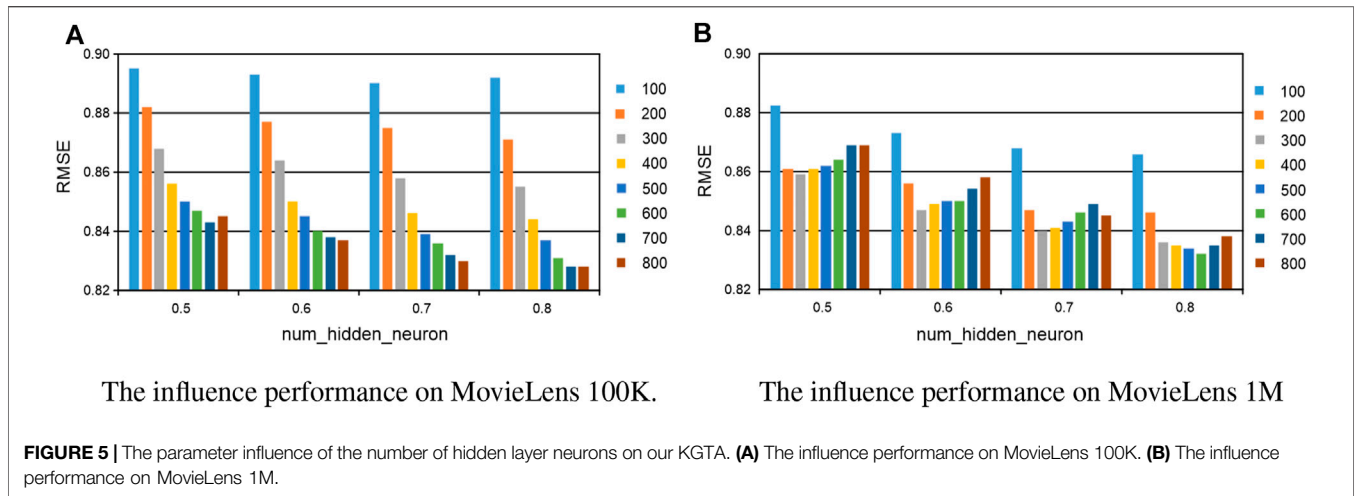


FIGURE 4 | RMSE of our KGTA and compared methods on the MovieLens 1M dataset.

repeating the experiments 5 times and taking the average value. From all the results, we have the following insightful observations:

- The performance of all recommended methods is improved with the increase of training data. It is worth mentioning that meta-learning methods such as MetaHIN and MeLU have not changed much, which may be due to the meta-learning methods being designed to alleviate the cold start problem for the recommendation.
- Generally, among the three types of methods, meta-learning methods perform the worst, probably because they are primarily designed to address the cold start problem. The methods for deep neural networks can achieve more desirable performance in most cases than both meta-learning and matrix factorization methods, which reveals the powerful ability of deep neural networks in learning the feature representations for personalized recommendation.
- Among all the deep neural network methods for recommendation, our KGTA is significantly better than NCF and AutoRec, which shows the superiority of introducing auxiliary information for addressing the problem of data sparsity and improving the performance of personalized recommendations.
- In the method of HCRSA, attributes such as the title, release date, and genre of a movie are introduced to the semi-autoencoder model for prediction. From the results listed in **Table 3** and **Figures 3, 4**, we can observe that our KGTA consistently outperforms HCRSA, which demonstrates the superiority of incorporating the key factors of collaborative filtering, such as the comments from users to items, to improve the performance of personalized recommendation.
- Although both the methods introduce auxiliary information, our KGTA outperforms PRKG by up to 7 RMSE points on two well-known datasets, which shows the advantage of designing a serial connection of semi-autoencoder and autoencoder for learning more abstract and higher-level feature representations in the recommendation.
- Overall, the proposed KGTA performs best in all groups, which validates the effectiveness of incorporating the key information between users and items and designing a serial connection of semi-autoencoder and autoencoder for the



recommendation. It should be noted that KGTA can achieve stable performance in both MovieLens 100K and MovieLens 1M. These results demonstrate that our KGTA can perform well even if the dataset is sparse.

5.4 Parameter Sensitivity

In this section, we investigate the influence of parameters in our proposed method, including the number of hidden layer neurons, the number of epochs, and the length of comments in the training. When one parameter is changed, the others are fixed in the experiments. The number of hidden layer neurons is varied from 100 to 800, the number of epochs is altered from 100 to 500, and the length of comments is sampled from the set {3, 5, 7, 9, 11, 13, 15, 17, 19, 21, and 23}. In the experiments, the validation was conducted on MovieLens 100K and MovieLens 1M, respectively. For the number of hidden layer neurons and the number of epochs, the experiments are conducted with 50%–80% of the training data. All the results are reported in **Figures 5, 6**, and we set *the number of epoch = 500* for both datasets, *the number of hidden layer neurons = 300* and *thenumberofhiddenlayerneurons = 400* for MovieLens 100K and MovieLens 1M, respectively. For the length of comments, experiments are conducted on 50% of the training data with the best and most stable parameters configuration of the number of hidden layer neurons and epoch, all the results are reported in **Figure 7**, and we set *the length of comments = 5* for both the datasets.

6 CONCLUSION

In this paper, we propose a feature representation learning method with a knowledge graph via triple-autoencoder for personalized recommendation called KGTA. We propose a serial connection between the semi-autoencoder and autoencoder methods. In our method, we were able to incorporate side information distilled from DBpedia for more useful item feature representations, and the key factors of collaborative filtering, such as comment information between users and items, are incorporated into the autoencoder as

auxiliary information. Moreover, the item-based rating and all the external information are incorporated into the semi-autoencoder to obtain low-dimensional information representation. Finally, the reconstructed output generated by the semi-autoencoder is input into a third autoencoder to learn better feature representations for personalized recommendation. Extensive experiments demonstrate the proposed method outperforms other state-of-the-art methods in effectiveness. In future work, we will try to achieve superior performance by incorporating less information and utilizing an attention network to strengthen the feature integration or without auxiliary information from the open knowledge base.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

YG: methodology, software, formal analysis, and writing. XX: conceptualization, supervision, and project administration. YZ: data curation, visualization, and writing. XS: visualization and validation.

FUNDING

This research was partially supported by the National Natural Science Foundation of China (61906060 and 62076217), Yangzhou University Interdisciplinary Research Foundation for Animal Husbandry Discipline of Targeted Support (yzuxk202015), the Opening Foundation of the Key Laboratory of Huizhou Architecture in Anhui Province under grant HPJZ-2020-02, and the Open Project Program of the Joint International Research Laboratory of Agriculture and Agri-Product Safety (JILAR-KF202104).

REFERENCES

- Alshahrani, M., Khan, M. A., Maddouri, O., Kinjo, A. R., Queralt-Rosinach, N., and Hoehndorf, R. (2017). Neuro-Symbolic Representation Learning on Biological Knowledge Graphs. *Bioinformatics* 33 (17), 2723–2730. doi:10.1093/bioinformatics/btx275
- Batmaz, Z., Yurekli, A., Bilge, A., and Kaleli, C. (2019). A Review on Deep Learning for Recommender Systems: Challenges and Remedies. *Artif. Intell. Rev.* 52 (1), 1–37. doi:10.1007/s10462-018-9654-y
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (8), 1798–1828. doi:10.1109/tpami.2013.50
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Found. Trends[®] Mach. Learn.* 2 (1), 1–127. doi:10.1561/2200000006
- Botangan, K. A., Yu, J., Sheng, Q. Z., Han, Y., and Yongchareon, S. (2020). Geographic-Aware Collaborative Filtering for Web Service Recommendation. *Expert Syst. Appl.* 151, 113347. doi:10.1016/j.eswa.2020.113347
- Chae, D.-K., Kim, S.-W., and Lee, J.-T. (2019). Autoencoder-Based Personalized Ranking Framework Unifying Explicit and Implicit Feedback for Accurate Top-N Recommendation. *Knowl. Based Syst.* 176, 110–121. doi:10.1016/j.knsys.2019.03.026
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.
- Dong, B., Zhu, Y., Li, L., and Wu, X. (2021). Hybrid Collaborative Recommendation of Co-Embedded Item Attributes and Graph Features. *Neurocomputing* 442, 307–316. doi:10.1016/j.neucom.2021.01.129
- Geng, X., Zhang, H., Bian, J., and Chua, T.-S. (2015). “Learning Image and User Features for Recommendation in Social Networks,” in Proceedings of the IEEE International Conference on Computer Vision, ICCV 2015 took place at the CentroParque Convention Center in Santiago, Chile, December 7–13, 2015. 4274–4282. doi:10.1109/iccv.2015.486
- Georgiev, K., and Nakov, P. (2013). “A Non-iid Framework for Collaborative Filtering with Restricted Boltzmann Machines,” in International Conference on Machine Learning, Atlanta, GA, USA, June 16–21, 2013. (PMLR), 1148–1156.

- Gupta, U., Wu, C.-J., Wang, X., Naumov, M., Reagen, B., Brooks, D., et al. (2020). "The Architectural Implications of Facebook's Dnn-Based Personalized Recommendation," in 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), San Diego, CA, USA, February 22–26, 2020. 488–501. doi:10.1109/hpca47549.2020.00047
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, April 3–7, 2017. 173–182. doi:10.1145/3038912.3052569
- Hohenberg, P., and Kohn, W. (1964). Inhomogeneous Electron Gas. *Phys. Rev.* 136 (3B), B864–B871. doi:10.1103/physrev.136.b864
- Kim, D. H., Baddar, W. J., Jang, J., and Ro, Y. M. (2017). Multi-Objective Based Spatio-Temporal Feature Representation Learning Robust to Expression Intensity Variations for Facial Expression Recognition. *IEEE Trans. Affective Comput.* 10 (2), 223–236. doi:10.1109/TAFFC.2017.2695999
- Koren, Y. (2008). "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008. 426–434.
- Lee, D. D., and Seung, H. S. (2001). "Algorithms for Non-Negative Matrix Factorization," in International Conference on Neural Information Processing Systems, Shanghai, China, November 11–15, 2001. 556–562.
- Lee, H., Im, J., Jang, S., Cho, H., and Chung, S. (2019). "Melu: Meta-Learned User Preference Estimator for Cold-Start Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, August 4–8, 2019. 1073–1082.
- Lee, J., Sun, M., Lebanon, G., and Sonnenburg, S. (2014). PREA: Personalized Recommendation Algorithms Toolkit. *J. Machine Learn. Res.* 13 (3), 2699–2703. doi:10.3166/EJC.18.485-495
- Li, P., Li, Y., Hsieh, C. Y., Zhang, S., Liu, X., Liu, H., et al. (2021). Trimnet: Learning Molecular Representation from Triplet Messages for Biomedicine. *Brief Bioinform* 22 (4), bbaa266. doi:10.1093/bib/bbaa266
- Li, Z., Fang, X., and Sheng, O. R. L. (2017). A Survey of Link Recommendation for Social Networks: Methods, Theoretical Foundations, and Future Research Directions. *ACM Trans. Manage. Inf. Syst. (TMIS)* 9 (1), 1–26. doi:10.1145/3131782
- Liu, H., Li, T., Hu, R., Fu, Y., Gu, J., and Xiong, H. (2019). Joint Representation Learning for Multi-Modal Transportation Recommendation. *Proc. AAAI Conf. Artif. Intelligence* 33, 1036–1043. doi:10.1609/aaai.v33i01.33011036
- Liu, Z., Sun, M., Lin, Y., and Xie, R. (2016). Knowledge Representation Learning: A Review. *J. Comput. Res. Develop.* 53 (2), 247. doi:10.7544/issn1000-1239.2016.20160020
- Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., et al. (2019). "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," in The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA 2019), Cairo, Egypt, March 28–30, 2019. 4114–4124.
- Lops, P., Jannach, D., Musto, C., Bogers, T., and Koolen, M. (2019). Trends in Content-Based Recommendation. *User Model. User-Adap Inter.* 29 (2), 239–249. doi:10.1007/s11257-019-09231-w
- Lu, Y., Fang, Y., and Shi, C. (2020). "Meta-Learning on Heterogeneous Information Networks for Cold-Start Recommendation," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1563–1573. doi:10.1145/3394486.3403207
- Luo, X., Yuan, Y., Chen, S., Zeng, N., and Wang, Z. (2020). Position-Transitional Particle Swarm Optimization-Incorporated Latent Factor Analysis. *IEEE Trans. Knowl. Data Eng.*, 1. doi:10.1109/tkde.2020.3033324
- Ma, Y., Narayanaswamy, B., Lin, H., and Ding, H. (2020). "Temporal-Contextual Recommendation in Real-Time," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020. 2291–2299. doi:10.1145/3394486.3403278
- Ni, J., Huang, Z., Cheng, J., and Gao, S. (2021). An Effective Recommendation Model Based on Deep Representation Learning. *Inf. Sci.* 542, 324–342. doi:10.1016/j.ins.2020.07.038
- Ni, J., Huang, Z., Hu, Y., and Lin, C. (2022). A Two-Stage Embedding Model for Recommendation with Multimodal Auxiliary Information. *Inf. Sci.* 582, 22–37. doi:10.1016/j.ins.2021.09.006
- Niu, G., Zhang, Y., Li, B., Cui, P., Liu, S., Li, J., et al. (2020). Rule-Guided Compositional Representation Learning on Knowledge Graphs. *Proc. AAAI Conf. Artif. Intell.* 34, 2950–2958. doi:10.1609/aaai.v34i03.5687
- Qian, X., Feng, H., Zhao, G., and Mei, T. (2013). Personalized Recommendation Combining User Interest and Social Circle. *IEEE Trans. Knowl. Data Eng.* 26 (7), 1763–1777. doi:10.1109/TKDE.2013.168
- Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. (2015). Big Data Meets Quantum Chemistry Approximations: The δ -Machine Learning Approach. *J. Chem. Theor. Comput.* 11 (5), 2087–2096. doi:10.1021/acs.jctc.5b00099
- Rashed, A., Grabocka, J., and Schmidt-Thieme, L. (2019). "Attribute-Aware Non-Linear Co-Embeddings of Graph Features," in Proceedings of the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 16–20, 2019. 314–321. doi:10.1145/3298689.3346999
- Salakhutdinov, R., and Mnih, A. (2008). "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo," in Proceedings of the 25th international conference on Machine learning (ICML 2008), Helsinki, Finland, July 5–9, 2008. 880–887. doi:10.1145/1390156.1390267
- Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). "Restricted Boltzmann Machines for Collaborative Filtering," in Neural Information Processing Systems (NIPS 2007), Vancouver, British Columbia, Canada, December 4–7, 2006. 791–798. doi:10.1145/1273496.1273596
- Salakhutdinov, R., and Mnih, A. (2007). "Probabilistic Matrix Factorization," in Proceedings of the 24th international conference on Machine learning (ICML 2007), ICML 2007 was held in conjunction with the 2007 International Conference on Inductive Logic Programming at Oregon State University in Corvallis, Oregon, June 20–24, 2007. 1257–1264.
- Schuster, M., and Nakajima, K. (2012). "Japanese and Korean Voice Search," in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012), Kyoto, Japan, March 25–30, 2012. (IEEE), 5149–5152. doi:10.1109/icassp.2012.6289079
- Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). "Autorec: Autoencoders Meet Collaborative Filtering," in Proceedings of the 24th International Conference on World Wide Web (WWW 2015), Florence, Italy, May 18–22, 2015. 111–112.
- Shuai, Z., Yao, L., Xu, X., Wang, S., and Zhu, L. (2017). "Hybrid Collaborative Recommendation via Semi-Autoencoder," in International Conference on Neural Information Processing, 185–193.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Machine Learn. Res.* 15 (1), 1929–1958. doi:10.5555/2627435.2670313
- Su, X., and Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* 2009, 1–19. doi:10.1155/2009/421425
- Wang, H., Wang, N., and Yeung, D.-Y. (2015). "Collaborative Deep Learning for Recommender Systems," in Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, NSW, Australia, August 10–15, 2015. 1235–1244. doi:10.1145/2783258.2783273
- Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020). Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (10), 3349–3364. doi:10.1109/TPAMI.2020.2983686
- Wei, L., Zhou, C., Su, R., and Zou, Q. (2019). Pepred-Suite: Improved and Robust Prediction of Therapeutic Peptides Using Adaptive Feature Representation Learning. *Bioinformatics* 35 (21), 4272–4280. doi:10.1093/bioinformatics/btz246
- Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-I., and Jegelka, S. (2018). "Representation Learning on Graphs with Jumping Knowledge Networks," in the 35th International Conference on Machine Learning (ICML 2018), Stockholm, Sweden, July 10–15, 2018. 5453–5462.
- Yang, Y., Zhu, Y., and Li, Y. (2021). Personalized Recommendation with Knowledge Graph via Dual-Autoencoder. *Appl. Intell.* 52, 1–12. doi:10.1007/s10489-021-02647-1
- Yehuda, K., R, B., and C, V. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer* 42 (8), 30–37. doi:10.1109/MC.2009.263
- Yi, Z., Hu, X., Zhang, Y., and Li, P. (2018). Transfer Learning with Stacked Reconstruction Independent Component Analysis. *Knowl. Based Syst.* 152, 100–106. doi:10.1016/j.knsys.2018.04.010

- Yu, Z., Lian, J., Mahmoody, A., Liu, G., and Xie, X. (2019). "Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation," in the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China, August 10–16, 2019. 4213–4219. doi:10.24963/ijcai.2019/585
- Zhang, Y., Ai, Q., Chen, X., and Croft, W. B. (2017). "Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources," in the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017), Singapore, November 6–10, 2017. 1449–1458. doi:10.1145/3132847.3132892
- Zhong, S.-T., Huang, L., Wang, C.-D., Lai, J.-H., and Yu, P. S. (2020). An Autoencoder Framework with Attention Mechanism for Cross-Domain Recommendation. *IEEE Trans. Cybern.* (11), 1–13. doi:10.1109/tcyb.2020.3029002
- Zhu, Y., Wu, X., Qiang, J., Yuan, Y., and Li, Y. (2021). Representation Learning with Collaborative Autoencoder for Personalized Recommendation. *Expert Syst. Appl.* 186, 115825. doi:10.1016/j.eswa.2021.115825
- Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q. (2017a). Supervised Representation Learning with Double Encoding-Layer Autoencoder for Transfer Learning. *ACM Trans. Intell. Syst. Technol. (TIST)* 9 (2), 1–17. doi:10.1145/3108257
- Zhuang, F., Luo, D., Yuan, N. J., Xie, X., and He, Q. (2017b). "Representation Learning with Pair-Wise Constraints for Collaborative Ranking," in the 10th ACM International Conference on Web Search and Data Mining (WSDM 2017), Cambridge, UK, February 6–10, 2017. 567–575. doi:10.1145/3018661.3018720
- Zhuang, F., Zhang, Z., Qian, M., Shi, C., Xie, X., and He, Q. (2017c). Representation Learning via Dual-Autoencoder for Recommendation. *Neural Networks* 90, 83–89. doi:10.1016/j.neunet.2017.03.009

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor YD declared a past co-authorship with the author XS.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Geng, Xiao, Sun and Zhu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.