

RESEARCH

Open Access



BRS_cS: a hybrid recommendation model fusing multi-source heterogeneous data

Zhenyan Ji^{1*}, Chun Yang¹, Huihui Wang², José Enrique Armendáriz-iñigo³ and Marta Arce-Urriza⁴

*Correspondence:

zhyji@bjtu.edu.cn

¹School of Software Engineering,
Beijing Jiaotong University, Beijing,
100044, China

Full list of author information is
available at the end of the article

Abstract

Recommendation systems are often used to solve the problem of information overload on the Internet. Many types of data can be used for recommendations, and fusing different types of data can make recommendations more accurate. Most existing fusion recommendation models simply combine the recommendation results from different data instead of fully fusing multi-source heterogeneous data to make recommendations. Furthermore, users' choices are usually affected by their direct and even indirect friends' preferences. This paper proposes a hybrid recommendation model BRS_cS (an acronym for BPR-Review-Score-Social). It fully fuses social data, score, and review together; uses improved BPR model to optimize the ranking; and trains them in a joint representation learning framework to get the top-*N* recommendations. User trust model is used to introduce social relationships into the rating and review data, PV-DBOW model is used to process the review data, and fully connected neural network is used to process the rating data. Experiments on Yelp public dataset show that the BRS_cS algorithm proposed outperforms other recommendation algorithms such as BRS_c, UserCF, and HRS_c. The BRS_cS model is also scalable and can fuse new types of data easily.

Keywords: Multi-source heterogeneous data, Recommendation model, Social network

1 Introduction

With the development of information technology, how to efficiently and quickly find valuable information from massive data has become a major challenge for users. In order to solve the problem of Internet information overload and enable users to quickly obtain interesting information, the recommendation system came into being. The recommendation system essentially abstracts the user's interest characteristics from a bunch of disorganized raw data and mines user's preferences to recommend different items or services [1]. Currently, the recommendation system has been successfully applied to many fields, including social networks (Facebook, Twitter), e-commerce (Amazon, Alibaba, Netflix), and information retrieval (Google, Baidu, Yahoo) [2–4].

In recent years, deep learning has been widely used in the engineering field [5]. It has achieved better results than traditional machine learning in the fields of image recognition, speech recognition, and natural language processing [6]. Compared with traditional

shallow machine learning, deep learning has an excellent ability to automatically extract abstract features, avoid complex feature engineering, and overcome the heterogeneity of multiple data. It has achieved ideal results in the representation learning of specific data such as images and text [7]. At the same time, the Internet has accumulated a wealth of information sources, such as ratings, reviews, and images, which can reflect users' preferences from different aspects. It has become a trend to fuse heterogeneous data sources for the recommendation.

The fusion of multi-source heterogeneous data recommendation models can be divided into models based on recommendation algorithm fusion and ones based on data feature fusion. The fusion models based on recommendation algorithm just select appropriate recommendation algorithms according to the data type and combine them by weighting, cascading, mixing, etc. When adding a new heterogeneous data, the entire model needs to be redesigned, so the scalability of the model is poor. Recently, due to the excellent effects of deep learning in representation learning [8], the research based on data feature fusion has received extensive attention. The fusion model based on the data feature refers to merging features of heterogeneous data by means of averaging or concatenation [9]. When adding a new heterogeneous data to the model, there is no need to redesign and train the original model so that the model is flexible and scalable.

Based on these, considering the impact of users' direct and indirect friends on users' decisions, this paper proposes a hybrid model based on social relationships that fuses multi-source heterogeneous data. The main contributions are as follows:

- This paper introduces the social network into the recommendation algorithm. Not only the direct friends' influence on users' decision is considered, but also the indirect friends' influence.
- A joint representation learning model that fuses scores, reviews, and social relations is proposed. It fuses different types of data from the data source perspective rather than combining the recommendation results from different recommendation models.
- Experiments are performed to compare the proposed model BRS_cS with other recommendation algorithms such as BS_c , UserCF, BRS_c , and HRS_c . BRS_cS performs better than other recommendation models in terms of Precision, Recall, and HT.

In this paper, Section 2 introduces the related work of recommendation algorithms. Section 3 illustrates our hybrid recommendation model based on the deep learning algorithm. The algorithms of different types of data are analyzed, and the objective function of fusing comments, scores, and social information is derived. Section 4 proves the effectiveness of adding social network information on the recommendation results and compares the proposed model with other recommendation algorithms by experiments. Section 5 concludes the paper and discusses future work.

2 Related works

2.1 Content-based recommendation

A content-based recommendation algorithm analyzes users' preferences for the items and recommends items, which have similar features to the items that the users like, to them. Based on the deep structured semantic model (DSSM), Elkahky et al. [10] proposed a multi-view deep neural network to learn the features of users and items separately. The recommended items were determined by calculating the similarity between users and

items. Zheng et al. [11] proposed a deep collaborative neural network (Deep CoNN) model that uses two parallel neural networks to learn the features of comments and then constructs an interaction layer on the two neural networks to predict the user's score. Based on DSSM, Xu et al. [12] proposed a label-based item recommendation, which inputs user information and item information related to the label into two deep neural networks respectively, and makes recommendations by calculating the similarities of abstract features between the user and the item. Seo [13] presented an attention-based convolutional neural network (CNN) model that combines reviews and ratings for product recommendations.

Deep learning can effectively alleviate the cold start problem of new projects. At the same time, it can integrate the feature extraction and recommendation process into a unified framework. However, the content-based recommendation algorithm only recommends similar items to users based on users' historical preferences. It cannot recommend new interesting items to users and implement cross-category recommendations.

2.2 Collaborative filtering algorithm

A collaborative filtering algorithm is the most widely used algorithm in the recommendation system. The main idea is to find a certain similarity between users or items and use this similarity to make recommendations for users [14]. Collaborative filtering algorithms can be divided into user-based algorithms, item-based algorithms, and model-based algorithms. The user-based collaborative filtering algorithm is the earliest recommendation algorithm. This algorithm first calculates the similarity between users and selects a similar user with the highest similarity to the target user. Then, it recommends items selected by the similar user to the target user [15]. Currently, the item-based collaborative filtering algorithm is used broadly in the industry, which recommends items similar to the items liked by users to them [16]. The model-based collaborative filtering algorithms mainly recommend items to users through machine learning and data mining models [17].

Salakhutdinov et al. [18] applied deep learning to solve the recommendation problem for the first time and proposed a collaborative filtering recommendation model based on the restricted Boltzmann machine (RBM). Sedhain et al. [19] proposed a self-encoder-based collaborative filtering method, which utilizes an encoding process and a decoding process to produce an output and optimizes the model parameters by minimizing the reconstruction error. Wu et al. [20] used the noise reduction self-encoder to solve the top- N recommendation problem and proposed a collaborative noise reduction self-encoder model. It makes recommendations by taking the user's rating vector as an input and learning the user's low-dimensional vector representation. Covington et al. [21] proposed a deep collaborative filtering model, first using the depth candidate video generation model to retrieve the candidate set and then using the depth ordering model to sort the candidate videos, which is superior to the matrix-based decomposition model.

The biggest advantage of a collaborative filtering algorithm based on deep learning is to introduce nonlinear feature transformation into the process of learning the implicit representations of user and item [22]. Compared with the traditional collaborative filtering method, it has better performance. However, new items cannot be recommended to users because they have not been rated. The algorithm cannot solve the data-sparse problem and the cold start problem.

2.3 Hybrid model

A hybrid model combines different recommendation models to take advantage of different models' merit and avoid their disadvantages [23]. Commonly used hybrid recommendation algorithms include weighted hybrid recommendation algorithm, cross-harmonic recommendation algorithm, and meta-model mixed recommendation algorithm [24]. For example, Lee et al. [25] learned semantic representation from the context of user conversations by combining recurrent neural networks and convolutional neural networks. Dai et al. [26] proposed a dynamic recommendation algorithm that combines the convolutional neural network and multivariate point process by learning the co-evolutionary model of user-commodity implied features.

To sum up, although the recommendation of single or dual source data based on deep learning has achieved good results, the recommendation accuracy is still poor [27–30]. The reason is that most hybrid recommendation models utilize limited kinds of heterogeneous data. With the development of the Internet, more and more data can be obtained. Using deep learning to fuse multiple heterogeneous data in the data source layer to improve the accuracy of the recommendation results is still worth studying [31–34].

3 Methods

3.1 Overview

This paper proposes a recommendation model based on deep learning, which can process multi-source heterogeneous data: score, review, and social information.

For the score, the traditional matrix decomposition method suffers problems of sparse data and low accuracy. This paper adopts the neural network to transform scores into the user/item representations. For the reviews, the traditional topic model cannot accurately represent the characteristics of the text. This paper utilizes the Distributed Bag of Words version of Paragraph Vector (PV-DBOW) algorithm to learn the feature representations of reviews. PV-DBOW assumes that the words in the document are independent and unordered, and uses document vector representation to predict the words with higher accuracy. For social network data, this paper takes into account the impact of users' friends on users' selection, introduces the user trust model, and integrates the social relationship information into the pairwise learning method, which improves the accuracy of the recommendation results.

3.2 Recommendation process

Due to the heterogeneity of different data, the traditional hybrid recommendation model usually fuses data at the algorithm level [30], i.e., makes the final recommendation by combining recommendation results from algorithms based on different data. With the development of deep learning, multi-source heterogeneous data such as scores and reviews can be accurately represented through deep networks, which makes it possible to fuse multi-source heterogeneous data fully at the data source level [30]. The multi-source heterogeneous data recommendation model proposed in this paper combines ratings, reviews, and social network information to make a more accurate recommendation. It has the advantages of high accuracy and strong scalability.

The recommendation process is shown in Fig. 1. The score is a user's overall evaluation for an item, which reflects the user's satisfaction with the item. The multi-layer fully connected neural network is used to directly learn the feature vector representations of the

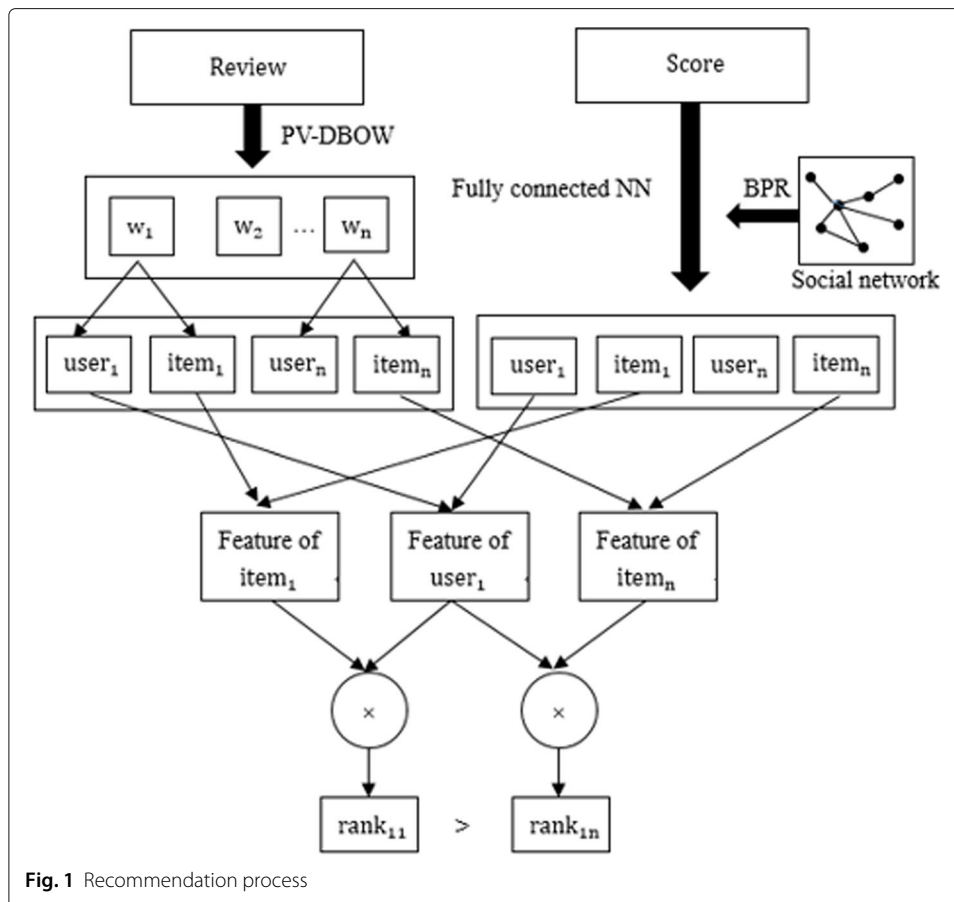


Fig. 1 Recommendation process

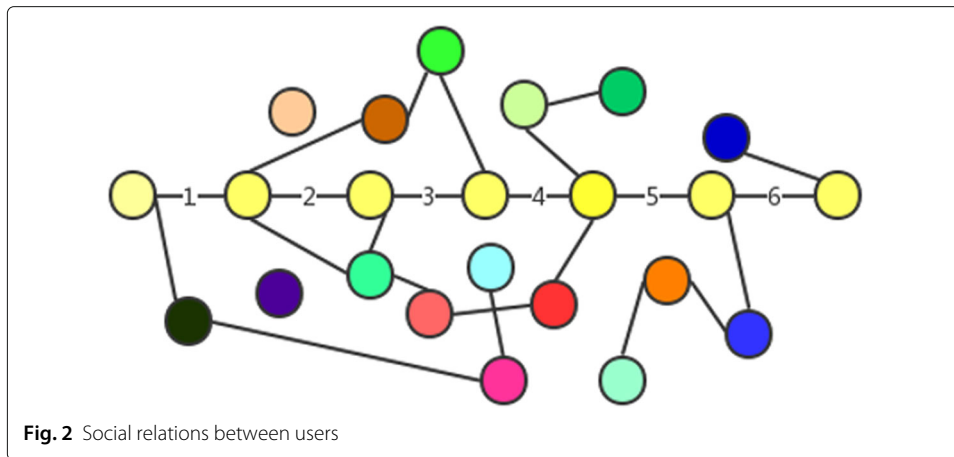
user and the item. Reviews can reflect users’ evaluations for items in detail and contain rich information about users and items. PV-DBOW algorithm is used to learn the feature representation of the paragraph and thus obtains the feature vector representations of the user and the item. The social network reflects friendships between users. The preferences of users’ friends will indirectly affect the users’ choices. The social network can be used to improve the prediction accuracy of a user’s potential purchase behavior. Bayesian Personalized Ranking (BPR) model is used to rank the nonlinear characteristics of users and items, and further improves the accuracy of the recommendation results.

3.3 Recommendation model

The recommendation model of multi-source heterogeneous data consists of four steps. Firstly, construct the user and item triplet optimization model. Secondly, extract social relations from the social network, and fuse social relation data, review data, and scores together. Thirdly, obtain the feature representations of users and items through deep learning. Finally, a top- N recommendation list is acquired from the feature representations of users and items. The model is described in detail as below.

3.3.1 User trust model

Social networks can reflect the friendship between users. In real life, users are more likely to choose items that their friends buy or like. Thus, a user’s behavior and preferences can be more precisely predicted based on the user’s direct and indirect friend relationship.



The trust-based recommendation model assumes that users have similar preferences to their trusted users. In general, direct and indirect friends can affect a user’s decisions on different levels, and indirect friends have less impact on the user’s decisions than direct friends. According to Kevin Bacon’s 6 degrees of separation concept [35, 36], the similarity between users can be defined in (1):

$$s(a, b) = \begin{cases} 0.2 \times (6 - l_{ab}) & \text{if } l_{ab} < 6 \\ 0.1 & \text{otherwise} \end{cases} \tag{1}$$

Among them, a and b represent any two users. l_{ab} represents the distance between user a and user b , the distance of direct friend is 1, the distance of indirect friend is 2, 3, 4, \dots , and $s(a, b)$ represents the similarity between two users. Figure 2 shows the distance values between users.

The similarity between users can be calculated based on the distances between users. We name the direct friend as the first-degree friend, the indirect friends with distance 2 as the second-degree friend, and so forth. We consider an indirect friend with distance 6 at most so that we name the model as 6 Degree Model. Algorithm 1 gives the model’s pseudo-codes and shows how to calculate similarities between users.

Algorithm 1 6 Degree Model

Input: dataset $User$, social relation data $Relation$;

Output: similarity set S ;

repeat:

1: Construct a graph by using all users and relations in $User$ and $Relation$;

2: Calculate l_{ab} using igraph;

3: If $l_{ab} < 6$

4: $s(a, b) = 0.2 \times (6 - l_{ab})$;

5: else

6: $s(a, b) = 0.1$;

7: **until** calculate all users’ relations;

8: **return** S .

After obtaining the similarity between users from social networks, the influence of different friends on the user's selection can be get; then, it can be input to a unified joint representation learning framework together with the other types of data.

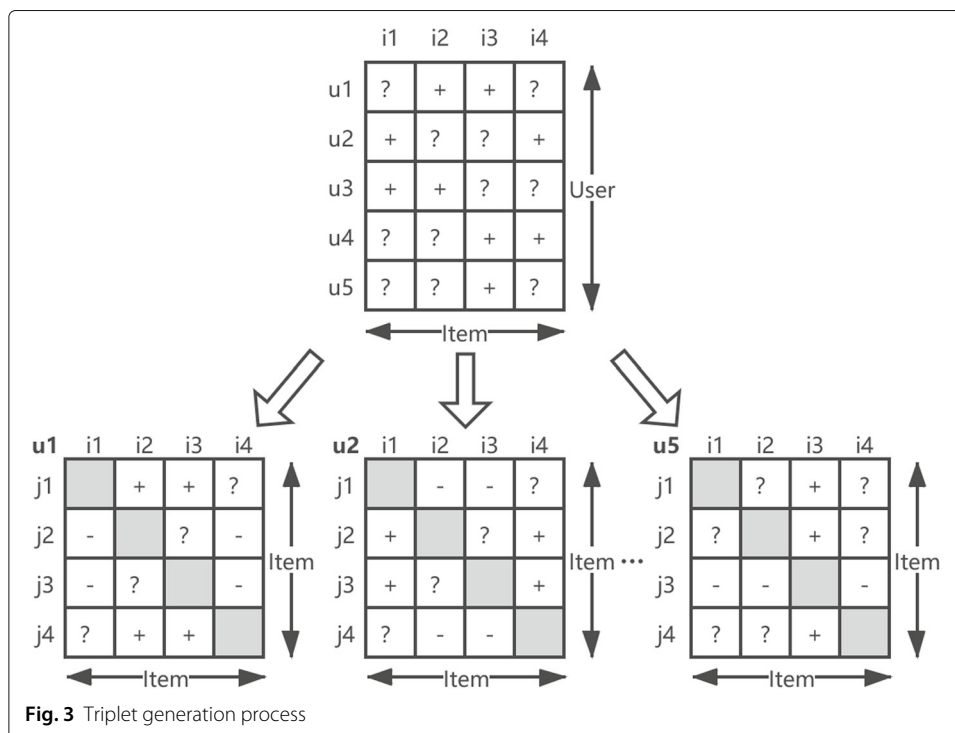
3.3.2 Improved BPR model

BPR is a pairwise learning model [37]. A triplet (u, i, j) is constructed based on the user's preferences. A triplet can represent three cases:

- User u purchases item i but does not purchase item j . It means user u has a preference on item i than item j .
- User u purchases neither item i nor item j . It means the user's preferences cannot be determined.
- User u purchases both item i and item j . It means the user's preferences cannot be distinguished.

Comparing with pointwise learning, the BPR model has two advantages. The first is considering both the items purchased by the user and the items not purchased during learning, and the items not purchased are the ones to be sorted in the future. The second advantage is that this model can reach good results when a small amount of data is selected for recommendation.

BPR is a ranking algorithm based on matrix decomposition. Comparing with algorithms such as funkSVD, it is not a global scoring optimization but a ranking optimization for each user's own commodity preferences. Its result is more accurate. Figure 3 shows the triplet generation process. Plus (+) means user u prefers item i over item j . Minus (-) means user u prefers item j over item i . Question mark (?) means that the user's preference cannot be determined.



However, the triplets constructed based on the standard BPR model are randomly sampled [38], and the effect of social relationships on the sampling process is not considered. In real life, users prefer the items that their friends have selected. So the similarity between users and friends can be applied to the sampling of the BPR model. By considering friends' influences on the user and adding social relation constraints to the sampling process, the triplet can more precisely reflect the user's preferences, and thereby, the recommendation accuracy can be improved.

According to the user's purchase records and the friendships reflected by the social network, for each user u , the item purchased by the user is defined as i , the item that the user has not purchased is defined as j , and the item purchased by the user's direct or indirect friend is defined as p . All the items set in the system are defined as D . The set of items purchased by the user u is defined as D_u . The set of items purchased by the user's direct and indirect friends is defined as D_p . The item set representing the user's strong preference is firstly D_u and secondly $D_p \setminus D_u$. The reason is that the user is likely to purchase the item $D_p \setminus D_u$ purchased by the direct or indirect friends but not by the user according to the influences of the friends on the user's preference. Finally, the item that the user is least likely to purchase is $D \setminus (D_u \cup D_p)$. Constructing a triplet of users and items as a training set based on social network information, the train set T can be expressed as follows. Where user-item triplet (u, i, j) represents that the user u has a greater preference for the item i than the item j . Item i is purchased by the user or by the direct or indirect friends of the user. Item j means the item not purchased by the user or his/her friends. In this way, a user-item triplet based on social relations is constructed.

$$T := \{(u, i, j) | i \in (D_u \cup D_p), j \in D \setminus (D_u \cup D_p)\} \tag{2}$$

According to the Bayesian formula, it is necessary to maximize the following posterior probabilities for finding a list of items recommended. In (3), (u, i, j) represents a constructed triplet with the user's preference, and θ represents the parameters of the model. To make the triplet (u, i, j) has the highest probability of occurrence, adjusting the model parameters.

$$p(\theta | (u, i, j)) \propto p((u, i, j) | \theta) p(\theta) \tag{3}$$

To simplify the aforementioned formula, we assume that the item pairs (i, j) are independent of each other. Then, we have:

$$\prod p((u, i, j) | \theta) = \prod_{u \in (u, i, j)} p(i > j | \theta) \cdot \prod_{u \notin (u, i, j)} (1 - p(i > j | \theta)) \tag{4}$$

According to the integrity and anti-symmetry of pairwise learning, the above formula can be further simplified to:

$$\prod p((u, i, j) | \theta) = \prod_{u \in (u, i, j)} p(i > j | \theta) \tag{5}$$

To obtain the final sorting, a model needs to be constructed to calculate the probability of recommendation for each item. The sigmoid function is used to construct the model in which the probability of the user's purchasing item i is greater than the one of purchasing item j . Where $x_{uij}(\theta)$ is an arbitrary parametric model that describes the potential relationship between the user and the item. In other words, any model that describes the

relationship between the user and the item can be used.

$$p((u, i, j)|\theta) := \sigma(x_{uij}(\theta)) \tag{6}$$

Improved BPR model is used to directly optimize the recommendation result based on the item recommendation ranking.

3.3.3 PV-DBOW model

PV-DBOW model is used to learn review data to obtain feature representations of corresponding users and items. As Fig. 4 shows, the model samples a text window, then samples a random word from the text window and forms a classification task given the Paragraph Vector [39]. PV-DBOW assumes that the words in a sentence are independent of each other and requires only a small amount of data to be stored.

In our model, paragraph vectors are used to predict words. Each review will be mapped into a semantic space and then trained to predict words. The probability that the word w appears in sentence d can be calculated by calculating the softmax function.

$$P(w|d_{um}) = \frac{e^{w^T d_{um}}}{\sum_{w' \in V} e^{w'^T d_{um}}} \tag{7}$$

where d_{um} denotes the review given by user u to item m , w denotes the word, and V denotes the vocabulary. In order to reduce the cost of computation and improve the calculation efficiency, the negative sampling example strategy is adopted in this model. Therefore, the following objective function can be constructed:

$$L_1 = \sum_{w \in V} \sum_{(u,m) \in R} f_{w,d_{um}} \log \sigma(w^T d_{um}) + \sum_{w \in V} \sum_{(u,m) \in R} f_{w,d_{um}} (t \cdot E_{w_N \sim P_V} \log \sigma(-w_N^T d_{um})) \tag{8}$$

where $f_{w,d_{um}}$ represents the frequency of word and review pairs. $E_{w_N \sim P_V}$ represents the expected value on the noise distribution P_V , and t represents the negative sample

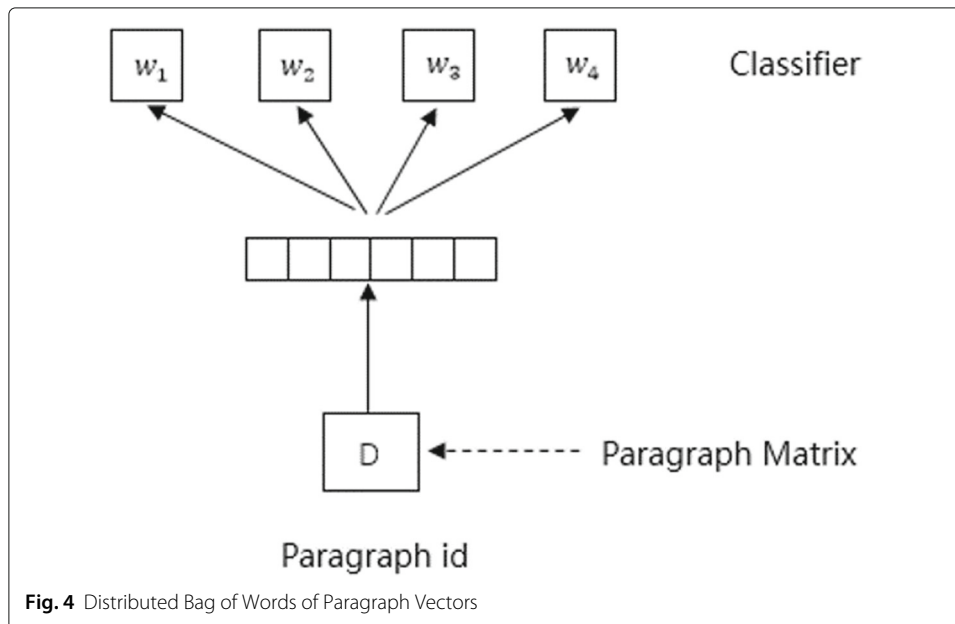


Fig. 4 Distributed Bag of Words of Paragraph Vectors

numbers. According to (8), the comment representation d_{um} can be obtained and d_{um} corresponds to the user u and the item m . The feature representations of u and m can be obtained from the comment data.

3.3.4 Fully connected neural network

For the reason that neural networks have the ability to quickly find optimal solutions, the fully connected neural network is used to process the scoring data [30]. The representations of user and item can be obtained from the score data. In this experiment, two fully connected layers are used to fit the nonlinear correlation:

$$\hat{r}_{um} = \phi(U_2 \phi(U_1(r_u \odot r_m) + c_1) + c_2) \tag{9}$$

where $\phi(\cdot)$ is the ELU activation function and U_1, U_2, c_1, c_2 are the parameters to be learned. Then, the following objective function can be get:

$$L_2 = \sum_{(u,m) \in R} (\hat{r}_{um} - r_{um})^2 \tag{10}$$

The goal of the scoring model is to make the difference between the predicted score and the true score as small as possible.

3.3.5 BRS_cS model

To fuse multi-source heterogeneous data to make a recommendation, we propose a model named BRS_cS (an acronym for BPR-Review-Score-Social). In the model, improved BPR model is used to optimize the ranking, user trust model is used to introduce social relationships into the rating and review data, PV-DBOW model is used to process the review data, and fully connected neural network is used to process the rating data. Finally, an integrated objective function is given to optimize.

The unified objective function for model optimization is given as (11). u represents the fusion feature representation of the user, and i and j represent the fusion feature representation of the items. According to the previous definition, it is known that the user u has a greater preference for the item i than the item j . $g(\cdot)$ is a loss function that combines user and item features; this paper defines $g(\cdot)$ as a sigmoid function to calculate the user's different preferences for different items. Here, $g(u, i, j) = \sigma(u^T i - u^T j)$. L_1 is the objective function of the review data, and L_2 is the objective function of the score data. When adding a new data source to the recommendation system, we only need to add the corresponding objective function in (11) instead of redesigning the model. The model proposed has good scalability.

$$\begin{aligned} \max_{W, \theta} L &= \sum_{u,i,j} g(u, i, j) + \lambda_1 L_1 - \lambda_2 L_2 \\ &= \sum_{u,i,j} \left\{ \sigma(u^T i - u^T j) \right. \\ &\quad + \lambda_1 \left(\sum_{w \in V} \sum_{(u,m) \in R} f_{w,d_{um}} \log \sigma(w^T d_{um}) \right) \\ &\quad + \sum_{w \in V} \sum_{(u,m) \in R} f_{w,d_{um}} (t \cdot E_{w_N \sim P_V} \sigma(-w_N^T d_{um})) - \\ &\quad \left. \lambda_2 (\phi(U_2 \cdot \phi(U_1(r_u \odot r_m) + c_1) + c_2) - r_{um})^2 \right\} \end{aligned} \tag{11}$$

$W = \{W_1, W_2\}$ denotes the weight parameters of each model. In the review representation learning model, the weight parameter W_1 is different for different user's different

review and need to be learned. In the score representation learning model, the features of users and items are directly obtained, that is, the weight parameter W_2 can be set to 1. It is unnecessary to update W_2 by the optimization objective function. θ represents other parameters to be learned, $\theta = \{\theta_1, \theta_2\} = \{w, d_{um}\}, \{U_1, U_2, c_1, c_2, r_u, r_m\}$. λ is the penalty parameter for each model, and its value is in the interval $[0, 1]$. The objective function L_2 of the score model is preceded by a negative sign because the objective function of the score model should be minimized, while the objective function of the overall model should be maximized. The stochastic gradient descent (SGD) method [40] can be used to optimize (11).

In the end, a recommendation list can be obtained by multiplying the user feature representations and the item feature representations:

$$s = u^T m \tag{12}$$

The larger the s , the higher possibility for the user to select the item. A user's top- N recommendation list is obtained from (12) in descending order.

Algorithm 2 BRS_cS model

Input: score dataset *Score*, social relation dataset *Relation*, review dataset *Review*, vocabulary V ;

Output: user representation U , item representation M , recommendation list L ;

- 1: Initialize θ , embedding size= 300, batch size= 64, negative sample = 5;
 - 2: **for** epoch= 1, 2, . . . , n **do**
 - 3: split the dataset *Score*, *Relation* and *Review* into training datasets (70%) and testing datasets (30%);
 - 4: construct positive and negative sample triplets $g(u, i, j)$ based on BPR;
 - 5: learn the frequency of word-review pair $f_{w,d_{um}}$ and the expected value $E_{w_N \sim P_V}$;
 - 6: get review representation d_{um} ;
 - 7: learn U_1, U_2, c_1, c_2 from score data;
 - 8: get score representation r_u, r_m ;
 - 9: get distance l_{ab} between users;
 - 10: calculate $\sum_{u,i,j} g(u, i, j) + \lambda_1 L_1 - \lambda_2 L_2$;
 - 11: update $\theta = \{\theta_1, \theta_2\}$ with back propagation;
 - 12: get corresponding user and item representations U, M according to (11);
 - 13: **end for**
 - 14: compute s according to (12);
 - 15: **return** recommendation list L .
-

4 Results and discussion

Two groups of experiments are performed on a single-core GPU GeForce GTX 1080 Ti with Ubuntu 16.04 operating system. The programming environment consists of python3.6, igraph, tensorflow1.4, and IntelliJ IDEA.

4.1 Dataset

Yelp is a business directory service and crowd-sourced review forum in America. It covers businesses in restaurants, shopping centers, hotels, and tourism. Users can rate, submit comments, and exchange experiences on the Yelp website. This paper uses Yelp public dataset which can be obtained from the Yelp official website for experiments. Yelp dataset is in JSON format and contains details of users and businesses. Data contains the IDs of users and businesses, users' comments and ratings on businesses, and the friendship between the users. The social relationship between users is transformed into user-friend relationship pairs. Comments and ratings are used to analyze users' preferences.

Since the dataset is too sparse, it is necessary to filter out some users with few comments to verify the validity of the proposed model. Data with more than 20 comments are extracted from the Yelp dataset, and the new dataset is named as New-Yelp. Table 1 shows the detailed statistics of the New-Yelp dataset.

4.2 Comparable experiment

Four indicators are used to measure the experimental results:

Recall: the ratio of items purchased by users on the recommendation list to all items purchased by users.

Precision: the ratio of the number of recommended items purchased by users to the total recommended items.

NDCG: normalized discounted cumulative gain. It is used to calculate the ranking quality of recommended items.

HT: the hit rate refers to whether the user has purchased the recommended item. If the user has purchased the recommended item, it means hits. Otherwise, it means misses.

4.2.1 Experiment I

To select a model to deal with texts and prove the positive effect of fusing social data on recommendation, experiment I compares six models as below. HDC and SEL are the most commonly used models to process text information in the recommendation system. HRS model uses the HDC to process reviews, and SRS model uses the SEL to process reviews. BR, BRS, BRS_c , and BRS_cS are the models using the BPR framework and PV-DBOW algorithm to process reviews:

- BR (BPR+Review) model is based on the BPR framework and uses the reviews for the recommendation.

Table 1 Detailed statistics of New-Yelp

Dataset	New-Yelp
User	16,371
Business	22,581
User-friend relation	6,111,137
Review	356,990
Sparsity	99.90%

Table 2 Experiment I of top-5 recommendation

Method/100%	Top-5 recommendation			
	Recall@5	Pre@5	NDCG@5	HT@5
BR	3.422	1.119	2.37	6.334
BRS	4.175	1.308	2.967	7.425
HRS	1.12	0.326	0.751	1.93
SRS	1.853	0.608	1.321	3.492
BRS _c	3.65	1.208	2.541	6.794
BRS _c S	4.363	1.352	3.084	7.71

- BRS (BPR+Review+Social) model is based on the BPR framework and uses the reviews and social network information for the recommendation. The social relation here contains both direct friend relation and indirect friend relation.
- HRS (HDC+Review+Social) model is a regularized embedding-based language model that combines reviews and social network information for the recommendation.
- SRS (SEL+Review+Social) model is based on a simplified embedding-based language model. It uses reviews and social network information for the recommendations.
- BRS_c (BPR+Review+Score) model is based on the BPR framework. It uses reviews and scores for the recommendation.
- BRS_cS (BPR+Review + Score+Social) model is based on the BPR framework. It combines reviews, scores, and social network information for the recommendation.

In the experiment, 70% of the data is used for training and 30% of the data for testing. The batch size is 64, the max train epoch is 40, the negative sample is 5, and the feature dimension is 300.

BRS_cS is the hybrid model proposed by us. The top-5 recommendation experimental results are shown in Table 2 and Fig. 5. The top-10 recommendation experimental results are shown in Table 3 and Fig. 6. The best results are shown in bold.

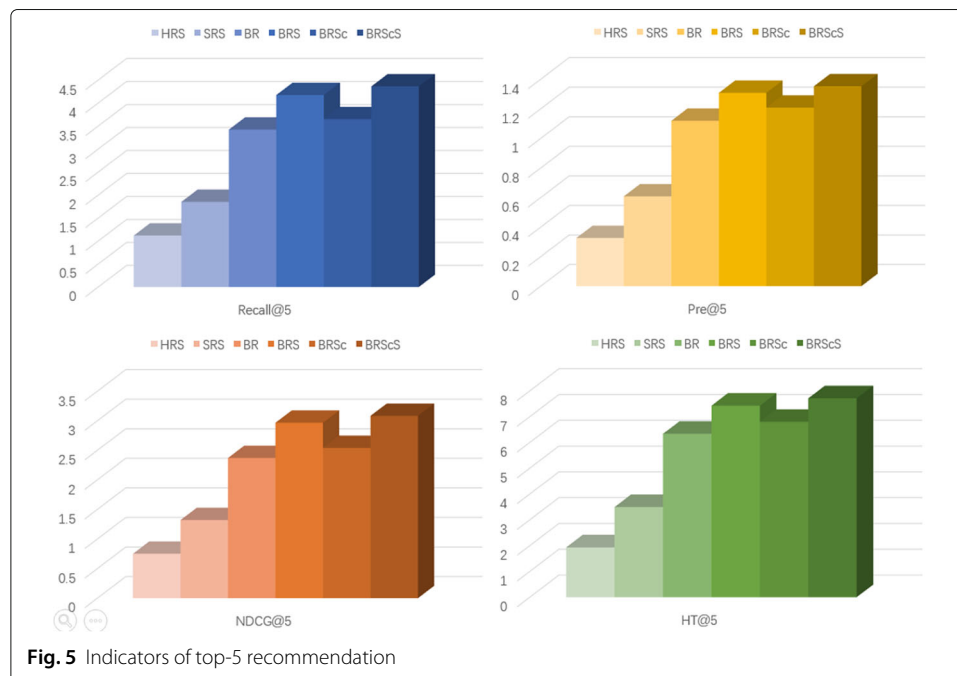


Table 3 Experiment I of top-10 recommendation

Method/100%	Top-10 recommendation			
	Recall@10	Pre@10	NDCG@10	HT@10
BR	5.56	1.004	3.103	10.049
BRS	6.617	1.125	3.784	11.483
HRS	1.855	0.304	1.003	3.256
SRS	2.843	0.517	1.651	5.268
BRS _c	5.787	1.057	3.264	10.48
BRS _c S	6.85	1.168	3.915	11.83

In the experiment, we compare BRS, HRS, and SRS, which are 3 different recommendation models based on reviews and social information. The results present that BRS model outstrips HRS model and SRS model significantly and proves that utilizing BPR framework in our model is a wise decision. Furthermore, experiments show that BRS model surpasses BR model and BRS_cS model surpasses BRS_c model, which proves that adding social networks can improve recommendation accuracy. Comparing all the models, we can see that the BRS_cS model proposed by us performs best in terms of Precision, Recall, NDCG, and HT.

4.2.2 Experiment II

Experiment II compares our model with broadly used classic recommendation models.

BS_c model is based on the BPR framework and uses scores to make a recommendation. UserCF (user-based collaborative filtering algorithm) is one of the most popular recommendation algorithms and uses scores as input. BRS_c is a model combining reviews and scores for the recommendation. BRS_cS is the model proposed by us. The experimental results are shown in Table 4 and Fig. 7. The best indicator results are shown in bold.

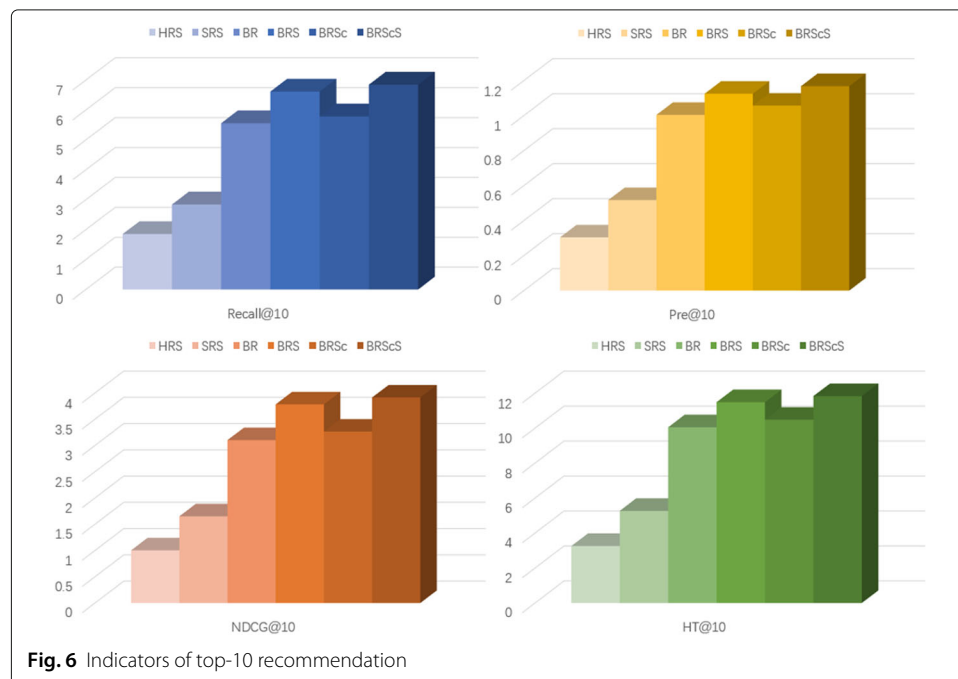


Table 4 Experiment II of recommendation

Measure/100%	BSc	UserCF	BRS _c	BRS _c S
Recall@5	1.532	1.93	3.65	4.363
Recall@10	2.26	2.78	5.787	6.85
Precision@5	0.671	0.25	1.208	1.352
Precision@10	0.227	0.18	1.057	1.168
HT@5	5.784	6.38	6.794	7.71
HT@10	8.628	9.15	10.48	11.83

The experimental results show that BRS_c surpasses BSc and UserCF in terms of three indicators (Recall, Precision, and HT). That proves the positive effect of fusing reviews together with scores to make a recommendation. The model proposed by us, BRS_cS, performs best in terms of all the indicators. It proves that the fusion model based on deep learning outperforms the traditional collaborative filtering algorithm due to the merits of feature representation by deep learning. Our model can fully fuse multi-source heterogeneous data such as scores, reviews, and social network information from data source level by deep learning so as to make a more accurate recommendation. Introducing social network information can also solve the cold start and data-sparse problems because direct and indirect friends' data can be used for making a recommendation.

5 Conclusions

To utilize heterogeneous data to improve recommendation accuracy and solve cold start and data-sparse problems, we propose a hybrid recommendation model that can fuse multi-source heterogeneous data such as scores, reviews, and social network information. The model is named as BRS_cS based on deep learning. Experiments are performed to compare our model with other recommendation models. The results show that our model can outperform the other models in Recall, Precision, NDCG, and HT. Introducing the

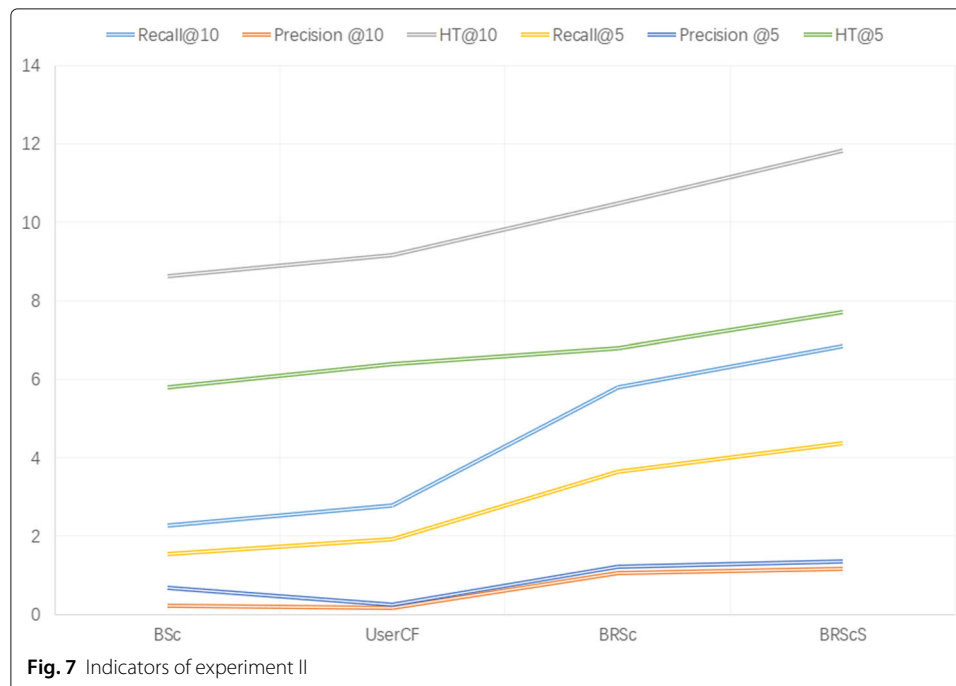


Fig. 7 Indicators of experiment II

user trust model to fuse social data together with scores and reviews can solve the cold start problem and data-sparse problem effectively because friends' data can be used to make a recommendation. The model is scalable and can fuse more types of heterogeneous data easily. However, the model proposed is based on a neural network, and it is difficult to explain the recommendation results, i.e., the interpretability of this model is weak.

In the future, we plan to introduce image data into our recommendation model because images contain rich semantic information.

Abbreviations

BPR: Bayesian Personalized Ranking; DSSM: Deep structured semantic model; CNN: Convolutional neural network; CF: Collaborative filtering; RBM: Restricted Boltzmann machine; PV-DBOW: Paragraph Vector-Distributed Bag of Words; Deep CoNN: Deep collaborative neural network

Acknowledgements

This work was supported by the Research Center of Software Engineering in the School of Software Engineering, Beijing Jiaotong University.

Authors' contributions

All authors contributed to this work. The authors read and approved the final manuscript.

Funding

This work is supported by the National Key Research and Development Program of China under grant no. 2018YFC0831903 and Major Project of National Natural Science Foundation of China under grant no. 51935002.

Availability of data and materials

The dataset used in this paper is available from the Yelp website [<http://www.yelp.com/dataset>]; more details are available from the corresponding author.

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Software Engineering, Beijing Jiaotong University, Beijing, 100044, China. ²Department of Engineering, Jacksonville University, Jacksonville, FL, 32211, USA. ³Department of Statistics, Computer Science and Mathematics, Public University of Navarre, 31006, Pamplona, Spain. ⁴Department of Business Management, Public University of Navarre, 31006, Pamplona, Spain.

Received: 11 February 2020 Accepted: 23 April 2020

Published online: 16 June 2020

References

1. Z. Ji, H. Pi, W. Wei, B. Xiong, M. Wozniak, R. Damasevicius, Recommendation based on review texts and social communities: a hybrid model. *IEEE Access*. **7**, 40416–40427 (2019)
2. J. B. Schafer, J. A. Konstan, J. Riedl, E-commerce recommendation applications. *Data Min. Knowl. Discov.* **5**, 115–153 (2001)
3. J. Wang, Y. Zhang, in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Opportunity model for e-commerce recommendation, (2013), pp. 303–312. <https://doi.org/10.1145/2484028.2484067>
4. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**, 734–749 (2005)
5. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
6. G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets. *Neural Comput.* **18**(7), 1527–1554 (2006)
7. G. E. Hinton, S. Osindero, Y.-W. Teh, Bengio yoshua. *Foundations and Trends® in Machine Learning*. **2**(1), 1–127 (2009)
8. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015)
9. M. Golz, D. Sommer, M. Chen, D. Mandic, U. Trutschel, Feature fusion for the detection of microsleep events. *J. VISI Signal Process. Syst. Signal Image Video Technol.* **49**, 329–342 (2007)
10. A. M. Elkahky, S. Yang, X. He, in *The 24th International Conference*, A multi-view deep learning approach for cross domain user modeling in recommendation systems, (2015). <https://doi.org/10.1145/2736277.2741667>
11. L. Zheng, V. Noroozi, P. S. Yu, *Joint deep modeling of users and items using reviews for recommendation*, (2017), pp. 425–434. <https://doi.org/10.1145/3018661.3018665>
12. Z. Xu, C. Chen, T. Lukasiewicz, *Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling*, (2016), pp. 1921–1924. <https://doi.org/10.1145/2983323.2983874>
13. S. Seo, *Representation learning of users and items for review rating prediction using attention-based convolutional neural network*, (2017)

14. B. S. Greg Linden, J. York, Amazon. com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput.* **7**, 76–80 (2003)
15. Z. D. Zhao, M. S. Shang, in *Third International Conference on Knowledge Discovery and Data Mining, WKDD 2010, Phuket, Thailand, 9–10 January 2010*, User-based collaborative-filtering recommendation algorithms on hadoop, (2010), pp. 478–481. <https://doi.org/10.1109/wkdd.2010.54>
16. M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* **22**, 143–177. <https://doi.org/10.1145/963770.963776>
17. Z. Ji, W. Yao, W. Wei, H. Song, H. Pi, Deep multi-level semantic hashing for cross-modal retrieval. *IEEE Access.* **7**, 23667–23674. <https://doi.org/10.1109/access.2019.2899536>
18. R. Salakhutdinov, A. Mnih, G. Hinton, in *Proceedings of the 24th International Conference on Machine Learning*, Restricted boltzmann machines for collaborative filtering, (2007), pp. 791–798. <https://doi.org/10.1145/1273496.1273596>
19. S. Sedhain, A. K. Menon, S. Sanner, L. Xie, Autorec: autoencoders meet collaborative filtering, 111–112. <https://doi.org/10.1145/2740908.2742726>
20. W. Yao, C. Dubois, A. X. Zheng, M. Ester, in *The Ninth ACM International Conference*, Collaborative denoising auto-encoders for top-n recommender systems, pp. 153–162. <https://doi.org/10.1145/2835776.2835837>
21. P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, 191–198 (2016). <https://doi.org/10.1145/2959100.2959190>
22. I. Arel, D. C. Rose, T. P. Karnowski, Deep machine learning - a new frontier in artificial intelligence research [research frontier]. *Comput. Intell. Mag. IEEE.* **5**, 13–18 (2010)
23. X. Wang, W. Ye, in *the ACM International Conference*, Improving content-based and hybrid music recommendation using deep learning, (2014), pp. 627–636. <https://doi.org/10.1145/2647868.2654940>
24. O. J. Bostandjiev Svetlin, H. Höllner Tobias, in *the ACM International Conference*, Tasteweights: a visual interactive hybrid recommender system, (2012), pp. 35–42. <https://doi.org/10.1145/2365952.2365964>
25. H. Lee, Y. Ahn, H. Lee, S. Ha, S. G. Lee, in *the 39th International ACM SIGIR Conference*, Quote recommendation in dialogue using deep neural network, (2016), pp. 957–960. <https://doi.org/10.1145/2911451.2914734>
26. H. Dai, Y. Wang, R. Trivedi, in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, Recurrent coevolutionary latent feature processes for continuous-time recommendation, (2016), pp. 29–34. <https://doi.org/10.1145/2988450.2988451>
27. P. Hicks, D. J. Cooper, S. Webb, J. Myburgh, I. Seppelt, S. Peake, C. Joyce, D. Stephens, A. Turner, C. French, G. Hart, I. Jenkins, A. Burrell, in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, vol. 36, The surviving sepsis campaign: international guidelines for management of severe sepsis and septic shock: 2008. an assessment by the australian and new zealand intensive care society, (2008), pp. 149–151. <https://doi.org/10.1177/0310057x0803600202>
28. M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation. *Commun. ACM.* **40**(3), 66–72 (1997)
29. Z. Ji, W. Yao, H. Pi, L. Wei, H. Jing, H. Wang, in *National Conference of Theoretical Computer Science*, A survey of personalised image retrieval and recommendation, (2017), pp. 233–247. https://doi.org/10.1007/978-981-10-6893-5_18
30. Z. Yongfeng, et al., in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Joint representation learning for top-n recommendation with heterogeneous information sources, (2017), pp. 1449–1458. <https://doi.org/10.1145/3132847.3132892>
31. W. Zhang, Z. Zhang, S. Zeadally, H.-C. Chao, V. Leung, Masm: a multiple-algorithm service model for energy-delay optimization in edge artificial intelligence. *IEEE Trans. Ind. Inform.* (2019). <https://doi.org/10.1109/tii.2019.2897001>
32. S. Houbing, S. Ravi, S. Tamim, S. Jeschke, *Smart cities foundations, principles, and applications*. (Wiley, 2017), pp. 1–906
33. Z. Zhang, W. Zhang, F.-H. Tseng, Satellite mobile edge computing: improving qos of high-speed satellite-terrestrial networks using edge computing techniques. *IEEE Netw.* **33**(1), 70–76. <https://doi.org/10.1109/mnet.2018.1800172>
34. J. Yang, H. Wang, Z. Lv, W. Wei, H. Song, M. Erol-Kantarci, B. Kantarci, S. He, Multimedia recommendation and transmission system based on cloud platform. *Futur. Gener. Comput. Syst.* **70**, 94–103. <https://doi.org/10.1016/j.future.2016.06.015>
35. L. Zhang, W. Tu, in *Proceeding of the Web Science*, Six degrees of separation in online society, pp. 1–5
36. H. Zhang, I. Ganchev, N. S. Nikolov, M. O'Droma, A trust-enriched approach for item-based collaborative filtering recommendations (2016). <https://doi.org/10.1109/iccp.2016.7737124>
37. S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012)
38. M. Al-Mashari, M. Zairi, Revisiting bpr: a holistic review of practice and development. *Bus. Process Manag. J.* **6**(1), 10–42 (2000)
39. Q. V. Le, T. Mikolov, Distributed representations of sentences and documents. *Int. Conf. Mach. Learn.*, 1188–1196 (2014)
40. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.