Neurocomputing 455 (2021) 283–296

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

CARM: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms

Duantengchuan Li [a], Hai Liu [a,*], Zhaoli Zhang [a], Ke Lin [b], Shuai Fang [a], Zhifei Li [a], Neal N. Xiong [a,c]

[a] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[b] Department of Control Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[c] Department of Mathematics and Computer Science, Northeastern State University, 74464, USA

ARTICLE INFO

Article history:
Received 29 November 2020
Revised 8 February 2021
Accepted 29 March 2021
Available online 6 April 2021
Communicated by Zidong Wang

Keywords:
Recommender system
Representation learning
Matrix factorization
Interactivity
Confidence matrix
Learning resource adaptation

ABSTRACT

The recommendation systems in the online platforms often suffer from the rating data sparseness and information overload issues. Previous studies on this topic often leverage review information to construct an accurate user/item latent factor. To address this issue, we propose a novel confidence-aware recommender model via review representation learning and historical rating behavior in this article. It is motived that ratings are consistent with reviews in terms of user preferences, and reviews often contain misleading comments (e.g., fake good reviews, fake bad reviews). To this end, the interaction latent factor of user and item in the framework is constructed by exploiting review information interactivity. Then, the confidence matrix, which measures the relationship between the rating outliers and misleading reviews, is employed to further improve the model accuracy and reduce the impact of misleading reviews on the model. Furthermore, the loss function is constructed by maximum a posteriori estimation theory. Finally, the mini-batch gradient descent algorithm is introduced to optimize the loss function. Experiments conducted on four real-world datasets empirically demonstrate that our proposed method outperforms the state-of-the-art methods. The proposed method also further promotes the application in learning resource adaptation. The source Python code will be available upon request.

1. Introduction

Recommendation systems have been widely employed in many domains, such as news [1], e-commerce [2], movies [3,4], and learning resource adaptation [5] in recent years. Choosing suitable products on online platforms is increasingly becoming difficult for consumers due to information overload. Thus, recommender systems, which play an information filtering role in many online platforms, such as Amazon and Yelp, learn the hobbies and interests of users through their historical behavior data and predict user preference.

Many algorithms have been proposed for recommendation systems over the past decades [3,4,6]. Matrix factorization (MF) [7–11] can be considered a famous collaborative filtering-based methodology, which aims to model the proper item features and user preferences from the rating matrix of historical data. Although these techniques have shown impressive results, their performance is ineluctably poor with rating matrix sparsity.

Numerous studies leverage extra information to tackle the aforementioned issue and raise the recommendation performance, such as social networks [12–14], demography of users [15], images [16,17], and reviews [18–20]. Many kinds of information are introduced to construct an accurate latent factor of the user or item. In other words, additional knowledge can be utilized to constrain the latent factor of users and items in recommendation models. Thus, according to the difference of latent factor constraints, all the recommendation methods can be mainly classified into the following three categories: single latent factor (SLF), double latent factor (DLF), and user-item interactivity latent factor (ILF) constraint methods. The SLF method only utilizes extra information related to the users or items to construct an accurate user or item latent factor. In [21], Fan *et al.* restricted the construction of user latent factors through the similarities of interests between different users in social networks. Kim *et al.* [22] additionally leveraged description documents of items to improve the rating prediction accuracy by constructing an accurate latent factor of items.

* Corresponding author at: National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China.
E-mail address: hailiu0204@mail.ccnu.edu.cn (H. Liu).

https://doi.org/10.1016/j.neucom.2021.03.122
0925-2312/

The DLF methods, which can be regarded as a twin-tower structure, constrain the latent factor of users and items through user-related information and item-related information, respectively. Such as, user social information, item description information, user and item side information, and user and item review information. In addition, the review usually contains rich information regarding the preferences of an individual user and the characteristics of the items. Using reviews to extract item features and user preference is a useful approach to alleviate the data sparsity problem. Many DLF-based studies [19,23] model the latent factor of users and items respectively through user and item review texts, and further improve recommendation accuracy by deep learning technique. In fact, review text actually contains abundant semantic information of users and items. This can be regarded as an interactive behavior of the user on the item, and user review can also be regarded as an evaluation of the interactive behavior of the user on the item. This characteristic of review is first named as user-item interactivity in this paper. However, the aforementioned methods [19,21–23] ignore that the review is the interactive information between users and items. Therefore, we propose model based on ILF constraint, which utilizes the interactive information of user review to constrain the latent factors of users and items at the same time.

Moreover, reviews usually contain fake good and bad reviews, which are called misleading reviews in this article. It is reported in many studies [24,25] that recommender systems attracted many malicious users to submit misleading reviews information about items to manipulate or trap the recommendation systems. Because of financial incentives, on numerous e-commerce platforms, many businesses try to game the recommender system by posting misleading reviews to either promote or defame their targeted items [26], which will mislead the prediction results of the recommendation system. These misleading reviews can harm the fairness of e-commerce markets and ill-posed recommendations. The widely-used recommender methods based on matrix factorization suffer from varying levels of performance degradation. This is because the model is not aware of misleading reviews when it is designed, and the item/user representations are extracted from such misinformation, so it is unreliable and biased. There have been some research efforts on misleading reviews [27,28]. Several methods improve performance by considering the probabilistic distribution of users' behaviors [29] or user behavior patterns [30]. In Rayana's work [31], they combine features such as review texts, time-stamps, and user behavioral information. However, these methods ignore the fact that the misleading reviews intentionally imitate true reviews in language expression, grammar style, and other aspects, and these kinds of reviews are confusing. Although misleading reviews can easily be concealed by users, eliminating the difference between user rating behaviors corresponding to misleading reviews and historical rating behaviors is difficult. In this article, ratings, which have a large deviation from the historical ratings of users and items, are called outliers. The rating outliers can help identify the misleading reviews because the ratings are consistent with the review information, and the outliers of rating correspond to the large possibility of misleading reviews. Fig. 1 shows review-based recommendation process.

Thus, a novel constraint method based on ILF, which is called the confidence-aware recommender model (CARM) via review representation learning and historical rating behavior, is presented. First, a single review text information is extracted by convolutional neural networks (CNNs). Then, the review latent factor is regarded as a *priori* constraint of interaction between user and item latent factors. Finally, a confidence matrix is constructed through the relationship between rating outliers and misleading reviews to further improve model accuracy and reduce the influence of misleading reviews on the model construction. The major contributions of this article are summarized in three aspects.

- The interactivity of review information is revealed to model the interaction of latent factor between user and item. Then, the user-item interactivity is used to construct the recommendation model by exploiting single review information. The authors believe that interactivity features are introduced for the first time in the recommendation system field.
- A confidence matrix is explored and designed to measure the relationship between the rating outliers and misleading reviews, which helps improve the performance and robustness of the recommender system.
- Four real-world datasets are selected to validate the performance of the proposed methods. Experimental results demonstrate that the models achieve significant improvements considering prediction accuracy and training efficiency.

The remainder of this article is organized as follows. Section 2 overviews the related work about the recommendation systems. Section 3 introduces our proposed CARM and CARM-C algorithms in detail. The optimization algorithm and parameters determination are presented in Section 4. Section 5 presents and analyzes the experiments on four public datasets. Section 6 draws a conclusion of this article and presents the future work.

2. Related work

2.1. Matrix factorization

MF, one of the most common collaborative filtering (CF) approaches [32,33], is also the most extensive model-based CF approach. The MF model aims to construct latent factors of items and users by mapping item and user features to shared space. The notations in this article provide the following conventions: bold lowercase letters denote vectors, bold capital letters indicate matrices, and non-bold letters denote indexes or scalars. Given m items and n users and the observed rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, the most popular rating prediction model in MF can be formulated as follows:

$$\widehat{\mathbf{R}}_{ij} = \mathbf{u}_i \mathbf{v}_j^T + \gamma_i + \psi_j + \tau \tag{1}$$

where $\mathbf{u}_i \in \mathbb{R}^k$ and $\mathbf{v}_j \in \mathbb{R}^k$ are represented as the k-dimensional latent factors, , τ represents the overall average rating, the symbols γ_i and ψ_j denote the observed deviations of users and items, respectively. That is, the problem of the recommendation systems is converted to predict the unobserved rating $\hat{\mathbf{R}}_{ij}$ using the $\mathbf{R}_{ij}$ derived from the above MF model. Many studies attempt to enhance the expression capability of the MF model based on this conversion. Koren *et al.* [34] utilized the neighborhood information to represent the latent factors of users and items. In [35], the authors extended MF to factorization machines for the generic modeling of latent factors. Though these methods have achieved good results, the performance of the MF model will significantly degrade when the rating matrix becomes substantially sparse.

2.2. Recommendation based on additional information

To alleviate the problem of sparse data, a number of additional information is introduced to enhance recommendation performance. Social recommendation [36,37] utilizes social relations or trust relations: Ma *et al.* [36] connected the social network structure and the user-item rating matrix through the shared user latent feature space. In [37], Jamali *et al.* further restrained the user latent factor vector to approximate the weighted average of his neighbors in social network. For side information, some researches employed stacked denoising autoencoder [38,39] to estimate the prior of latent factors from the additional information introduced to the

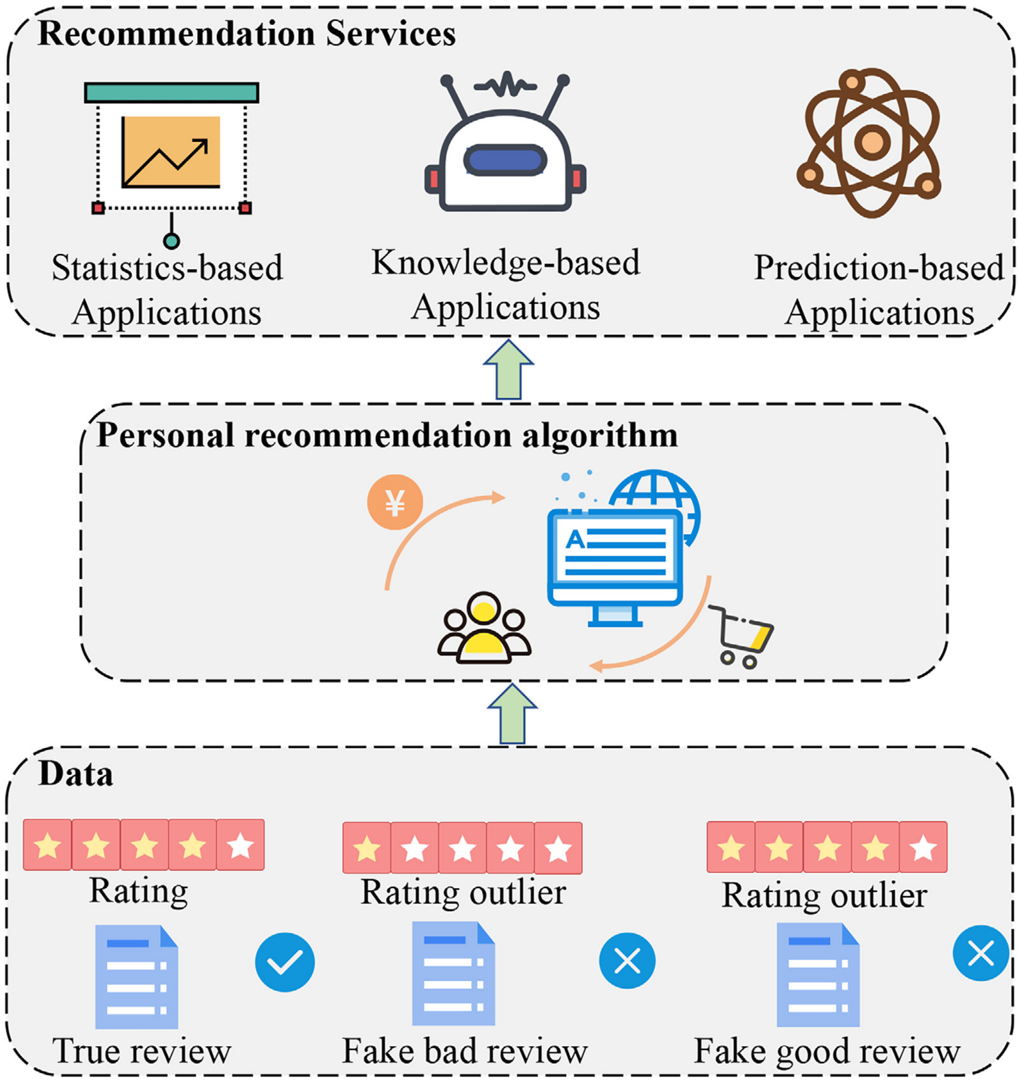

Fig. 1. Review-based recommendation process: reviews actually contain many true reviews and misleading reviews. Tick mark represents real reviews that the model needs; cross marks indicate misleading reviews that will have an impact on the accuracy of the recommended model.

recommendation system. Review-based recommendation [40–42] employs the content of items or users to address data sparse issue, such as explicit factor models (EFM) [40], rating-boosted latent topics (RBLT) [41], and hidden factors as topics (HFT) model [42]. These methods integrated topic factors of the review into the MF models to build the latent factors to improve the rating prediction performance and address the sparse issue. The experimental results also further prove that the introduction of additional information can further improve the recommendation performance.

2.3. Recommendation based on deep learning

Although the aforementioned review-based have shown significant improvements compared with conventional latent models, they have some limitations. These studies ignore local context information and word order, which result in the loss of specific information in the form of phrases and sentences. Deep learning has recently become a crucial research instrument in many domains [3–5,43]. Many models facilitate CNN architecture to capture the review contextual information to improve the capability of deep feature representation and attempt to combine CNN structures with MF frameworks to raise the recommendation performance. In [22], Kim *et al.* utilized CNNs to capture the contextual information of documents by considering local context and word order. DeepCoNN [44] uses CNNs to model the hidden latent features of users and items separately based on all their associated reviews. Chen *et al.* [23] constructed a recommendation model using neural attention mechanism network to learn useful reviews information. CARL [45] exploits mutual and local attention of CNNs to learn the relevant features from reviews via a linear fusion mechanism. However, these CNN-based models have a common trait: that is, these models use aggregated review text from users and items to model users and item, respectively. Moreover, they spend a long time in the training process. Hence, the latent factors of item and user are learned statically and independently. This article argues that the review text contains semantic information of users and items, which is also a kind of interaction information related to users and items.

3. Proposed CARM method

3.1. Outline of CARM

The proposed CARM method is summarized as three main modules, namely, confidence matrix, review latent factor representation learning and confidence-aware recommender model. The pipeline is shown in Fig. 2.

In the confidence matrix module, a confidence matrix is designed to measure the relationship between the rating outliers and misleading reviews, which improves the performance and robustness of the recommender system. In review latent factor representation learning, review text is input into CNN, which could capture the contextual information and reserve the weight of words for reviews analysis, and generate the feature vector of review text. In confidence-aware recommender model, we place zero-mean Gaussian priors on users and items latent factor vectors, respectively. At the same time, their Hadamard product, which denotes the interaction features between users and items, is subjected to a constraint of the multi-dimensional Gaussian prior distribution, whose mean is indicated by the review text feature vector. Review text information will blend in item and user factor vectors by placing these priors and confidence matrix on item and user latent factors.

3.2. Review latent factor representation model

The goal of the review latent factor representation learning is to generate latent factor of user reviews of products by use CNNs. In Fig. 3, we illustrate the review latent factor representation learning process, which contains four layers, such as the weighted word representation layer, convolution layer, pooling layer, and output layer.

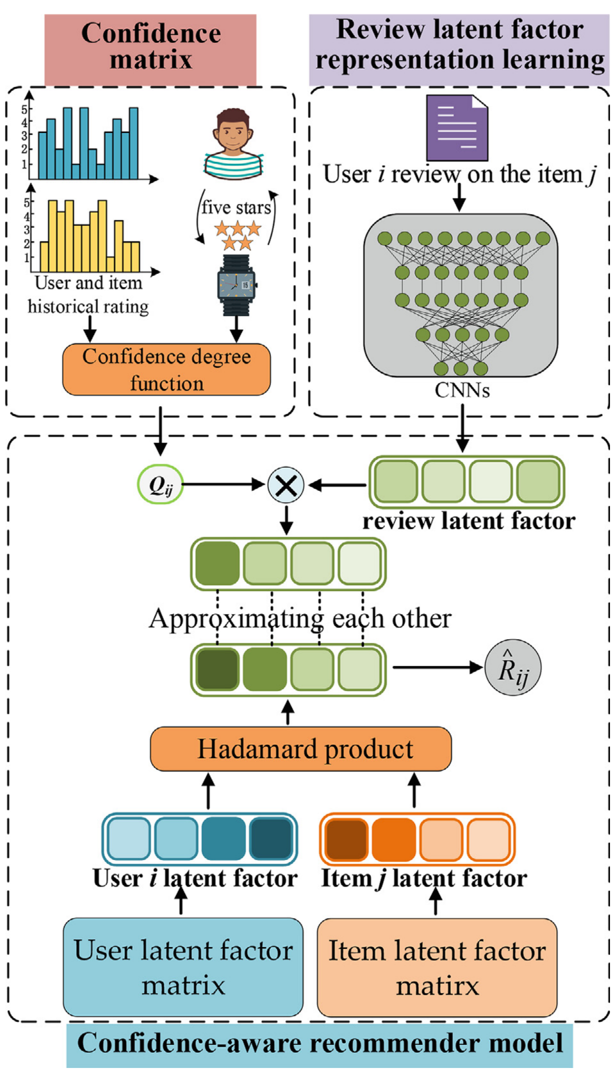

Fig. 2. Outline of the proposed CARM framework.

The first layer is the weighted word representation layer. A pre-trained word vector model [22] (word2vec) is used to convert the original text information into digital information. First, the embedding matrix converts the input text into a dense real-valued matrix, each column vector of the matrix is the vector representation of the corresponding word in the text, and the matrix is utilized as the input of the convolutional layer. Then the entire text is represented as a matrix. The review matrix $\mathbf{D} \in \mathbb{R}^{l\times p}$ can be rewritten as,

$$\mathbf{D} = \begin{bmatrix} \cdots & | & | & | & \cdots \\ & \mathbf{w}_{i-1} & \mathbf{w}_i & \mathbf{w}_{i+1} & \\ \cdots & | & | & | & \cdots \end{bmatrix}, \tag{2}$$

where l denotes the length of the input review text, and p is the dimension of the embedding vector of the word.

In fact, the word vector in the pre-training model word2vec has been able to describe the representation of the word more accurately. In order to reduce model parameters, improve model efficiency, and at the same time enable the network to have better generalization capabilities, we replace the training method of word vectors instead of training the original vectors by introducing the parameter ϕ_i with weights to train the networks. That is, ϕ_i is treated as a parameter in the network for learning, and the networks is trained by mini-batch gradient descent (MBGD) algorithm. This ensures the training efficiency of the model as well as the generalization ability of the model. Each word vector $\mathbf{w}_i$ is assigned a weight ϕ_i, then the output will be formulated as,

$$\mathbf{H} = \begin{bmatrix} \cdots & | & | & | & \cdots \\ & \phi_{i-1}\mathbf{w}_{i-1} & \phi_i\mathbf{w}_i & \phi_{i+1}\mathbf{w}_{i+1} & \\ \cdots & | & | & | & \cdots \end{bmatrix}, \tag{3}$$

Finally, the contribution of the scoring result is reflected by the weight of each word in each review. The contextual features in the review are extracted by convolution operation. The contextual feature vector $c_i^j \in \mathbb{R}$ is extracted through the j-th sharing weight $\mathbf{W}^j \in \mathbb{R}^{t\times p}$, and the window size t defines the number of surrounding words,

$$c_i^j = Relu\left(\mathbf{W}^j * \mathbf{H}_{(:,i:(i+t-1))} + b^j\right), \tag{4}$$

where i represents the current index of the filter on the review matrix, $*$ is a convolution operator, t is the length of the sliding window filter, and $b^j \in \mathbb{R}$ denotes a bias for $\mathbf{W}^j$. We employ the activation function Relu to avoid the problem of gradient disappearance. Then review contextual feature vector $c^j \in \mathbb{R}^{l-t+1}$of review with $\mathbf{W}^j$ is constructed by

$$c^j = \left[c_1^j \quad c_2^j \quad \cdots \quad c_i^j \quad \cdots \quad c_{l-t+1}^j\right]. \tag{5}$$

The third layer is the pooling layer, which is also called as the down-sampling layer. In fact, there will be many redundant features and repetitive features in the text feature vector extracted by the previous convolutional layer. Therefore, in order to reduce the excessive useless context features in c^j, max pooling will be used from the context feature vector to extract the most valuable features. The output of the pooling layer $d \in \mathbb{R}^m$can be rewritten as,

$$\mathbf{d} = \left[\max\left(c^1\right), \max\left(c^2\right), \ldots, \max\left(c^j\right), \ldots\right]. \tag{6}$$

The fourth layer is the output layer. The previous three layers are used to learn local features of the review text. In order to obtain global features, a fully connected network is used to combine these

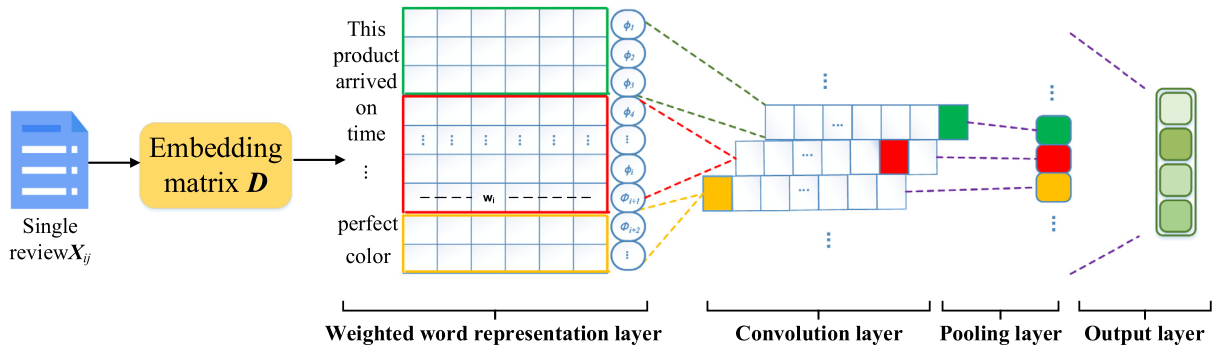

Fig. 3. Review latent factor representation model.

local features. Thus, $\mathbf{d}$ is projected into a k-dimensional space for our subsequent task, which is the same as the item and user latent vector. Finally, the traditional nonlinear projection is leveraged to obtain the review latent factor,

$$s = \tanh(\mathbf{W}_2\{\tanh(\mathbf{W}_1\mathbf{d} + b_1)\} + b_2), \tag{7}$$

where $\mathbf{W}_1 \in \mathbb{R}^{h\times m}, \mathbf{W}_2 \in \mathbb{R}^{k\times m}$ are weight matrics and $b_1 \in \mathbb{R}^h, b_2 \in \mathbb{R}^k$ denote the bias vector for $\mathbf{W}_1$ and $\mathbf{W}_2$, respectively. $s \in \mathbb{R}^k$ is the output. Eventually, the CNN architecture is regarded as a function which takes the original reviews as input, and the review feature vector that is k-dimension as output through the above process,

$$\mathbf{s}_{ij} = CNN_{\mathbf{\Theta}}(X_{ij}) \tag{8}$$

where $\mathbf{\Theta}$ denotes all variables in our network, and X_{ij} is regarded as the user i's a raw review on item j, and $\mathbf{s}_{ij}$ represents the single review feature representation which is defined as review latent factor in this paper.

3.3. Confidence-aware recommender model

In this section, we derive the objective function of the model from the view of probability (see in Fig. 4). Given m items and n users and the observed rating matrix $\mathbf{R} = [r_{ij}]_{m\times n}$, our goal is to construct appropriate item and user vectors ($\mathbf{U} \in \mathbb{R}^{k\times m}$ and $\mathbf{V} \in \mathbb{R}^{k\times n}$) that represent latent factors of items and users.

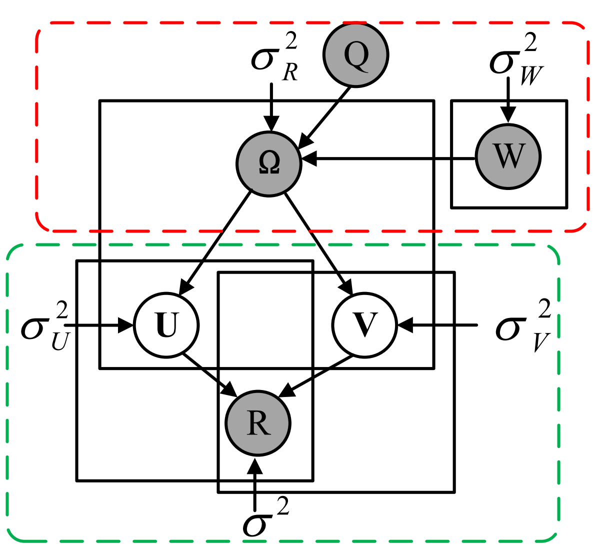

Fig. 4. Graphical model for CARM: CNNs part in up (dashed-red); latent factor part in down (dotted-green).

3.3.1. Ratings modeling

According to the rating data observation model (1), the measure errors in the rating score data are modeled as Gaussian noise. The likelihood function of observed ratings over training data can be written as,

$$p(\mathbf{R}|\mathbf{U},\mathbf{V},\gamma,\psi,\tau) = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathcal{N}(r_{ij}|\hat{r}_{ij},\sigma^2)\right]^{1_{ij}}, \tag{9}$$

where $\mathcal{N}(x|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with variance σ^2 and mean μ. The symbol 1_{ij} denotes the indicator function that equals as 1 if user i rates on item j, and equals as 0 otherwise.

3.3.2. Items and users modeling

The spherical Gaussian priors is introduced on user and item latent factors. For the item prior probability, it is defined as,

$$p(\mathbf{V}|\sigma_V) = \prod_{j=1}^{n}\mathcal{N}(\mathbf{v}_j|0,\sigma_V^2\mathbf{I}). \tag{10}$$

For the user prior probability, it is defined as,

$$p(\mathbf{U}|\sigma_U) = \prod_{i=1}^{m}\mathcal{N}(\mathbf{u}_i|0,\sigma_U^2\mathbf{I}), \tag{11}$$

where $\sigma_V^2 I$ and $\sigma_U^2 I$ are the covariance matrix of item latent matrix $\mathbf{V}$ and user latent matrix $\mathbf{U}$, respectively.

3.3.3. Single review modeling

The k dimension vector $\mathbf{u}_i \circ \mathbf{v}_j$, which denotes the Hadamard product of user latent factor $\mathbf{u}_i$ and the item latent factor $\mathbf{v}_j$, can be regarded as the interactive information representation of the user i and the object j. In addition, high scores are usually accompanied by positive reviews, and low scores are usually accompanied by negative reviews. There is consistency between the rating and the review text. That is, the difference between $\mathbf{u}_i \circ \mathbf{v}_j$ and the review latent factor representation from user i to item j is very small. This difference is considered to obey the Gaussian distribution. In this study, the zero-mean Gaussian independent and identically distributed (IID) noise is usually assumed to exist. Consequently, the Hadamard product of $\mathbf{u}_i$ and $\mathbf{v}_j$ is represented as,

$$\mathbf{u}_i \circ \mathbf{v}_j = CNN_{\mathbf{\Theta}}(X_{ij}) + \epsilon_R, \quad \epsilon_R \sim \mathcal{N}(0,\sigma_R^2 I). \tag{12}$$

Then, the conditional distribution of $\mathbf{u}_i \circ \mathbf{v}_j$ can be formulated,

$$p(\mathbf{u}_i \circ \mathbf{v}_j|X_{ij},\mathbf{\Theta},\sigma_R^2) = \mathcal{N}(\mathbf{u}_i \circ \mathbf{v}_j|CNN_{\mathbf{\Theta}}(X_{ij}),\sigma_R^2 I), \tag{13}$$

The conditional distribution of observed reviews over training is set as

$$p(\mathbf{U},\mathbf{V}|X,\boldsymbol{\Theta},\sigma_R^2) = \prod_i^m\prod_j^n \mathcal{N}\left(\mathbf{u}_i \circ \mathbf{v}_j | CNN_{\boldsymbol{\Theta}}(X_{ij}), \sigma_R^2 I\right). \tag{14}$$

Integrating the above equations, the posterior probability of our model is constructed as,

$$\begin{aligned} & p(\mathbf{U},\mathbf{V}|\mathbf{R},X,\gamma,\psi,\boldsymbol{\Theta},\sigma,\sigma_U,\sigma_V,\sigma_R) \\ & = \frac{p(\mathbf{R}|\mathbf{U},\mathbf{V},\gamma,\psi,\sigma)p(\mathbf{U}|\sigma_U)p(\mathbf{V}|\sigma_V)p(\mathbf{U},\mathbf{V}|X,\boldsymbol{\Theta},\sigma_R)}{p(\mathbf{R},X,\gamma,\psi,\boldsymbol{\Theta},\sigma,\sigma_U,\sigma_V,\sigma_R)}. \end{aligned} \tag{15}$$

The maximum a posteriori (MAP) estimation is introduced on the Eq. (15), we can achieve,

$$\begin{aligned} & max_{\mathbf{U},\mathbf{V}} p(\mathbf{U},\mathbf{V}|\mathbf{R},X,\gamma,\psi,\boldsymbol{\Theta},\sigma,\sigma_U,\sigma_V,\sigma_R) \\ & = \max_{\mathbf{U},\mathbf{V}} \underbrace{p(\mathbf{R}|\mathbf{U},\mathbf{V},\gamma,\psi,\sigma)}_{\text{likelihood probability}} \cdot \underbrace{p(\mathbf{U}|\sigma_U)p(\mathbf{V}|\sigma_V)}_{\text{prior probability}} \cdot \underbrace{p(\mathbf{U},\mathbf{V}|X,\boldsymbol{\Theta},\sigma_R)}_{\text{review prior probability}}. \end{aligned} \tag{16}$$

There are three probability density functions need to be defined, such as likelihood probability, prior probability, and review prior probability. By taking negative logarithm on Eq. (16). It can be written as,

$$\begin{aligned} L &= \log p(\mathbf{R}|\mathbf{U},\mathbf{V},\gamma,\psi,\sigma) + \log p(\mathbf{U},\mathbf{V}|X,\boldsymbol{\Theta},\sigma_\mathrm{R}) + \log p(\mathbf{U}|\sigma_U) \\ &\quad + \log p(\mathbf{V}|\sigma_V) \\ &= -\frac{1}{2\sigma^2}\sum_i^m\sum_j^n 1_{ij}\left(\mathbf{R}_{ij} - \mathbf{u_i v_j^T} - \gamma_\mathbf{i} - \psi_\mathbf{j} - \tau\right)^\mathbf{2} - \frac{1}{2\sigma_R^2}\sum_\mathbf{i}^\mathbf{m} \\ &\quad \times \sum_\mathbf{j}^\mathbf{n}\left(\mathbf{u_i}\circ\mathbf{v_j} - CNN_{\boldsymbol{\Theta}}(\mathbf{X_{ij}})\right)^\mathbf{T} I^{-\mathbf{1}}\left(\mathbf{u_i}\circ\mathbf{v_j} - CNN_{\boldsymbol{\Theta}}(\mathbf{X_{ij}})\right) \\ &\quad + \frac{1}{2\sigma_U^2}\sum_\mathbf{i}^\mathbf{m}(\mathbf{u_i})^\mathbf{T} I^{-\mathbf{1}}\mathbf{u_i} - \frac{1}{2\sigma_V^2}\sum_\mathbf{j}^\mathbf{n}(\mathbf{v_j})^\mathbf{T} I^{-\mathbf{1}}\mathbf{v_j} - C. \end{aligned} \tag{17}$$

After some manipulation, the constant C can be safely dropped. And the maximization of this posterior probability distribution is equivalent to the following regularized minimum problem,

$$\begin{aligned} E_{CARM-C} &= \sum_i^m\sum_j^n \frac{1_{ij}}{2}\left(\mathbf{R}_{ij} - \widehat{\mathbf{R}}_{ij}\right)^2 \\ &\quad + \frac{\lambda_R}{2}\sum_i^m\sum_j^n \left\|\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\boldsymbol{\Theta}}(X_{ij})\right)\right\|_F^2 + \frac{\lambda_U}{2}\sum_i^m\|\mathbf{u}_i\|_F^2 \\ &\quad + \frac{\lambda_V}{2}\sum_j^n\|\mathbf{v}_j\|_F^2, \end{aligned} \tag{18}$$

where $\lambda_U, \lambda_V, \lambda_R$ are set as $\sigma^2/\sigma_U^2, \sigma^2/\sigma_V^2, \sigma^2/\sigma_R^2$, respectively.

However, reviews contain misleading reviews, which are harmful to model construction. Thus, a confidence matrix is introduced to the review data since not all of the data play the equal role to model the latent factors. The value in the confidence matrix measures the degree of trust in each review, and the reviews corresponding to the outliers in the rating matrix should be given low confidence, and vice versa. Thus, the corresponding energy functional is rewritten as,

$$\begin{aligned} E_{CARM} &= \sum_i^m\sum_j^n \frac{1_{ij}}{2}\left(\mathbf{R}_{ij} - \widehat{\mathbf{R}}_{ij}\right)^2 + \sum_i^m\sum_j^n \mathbf{Q}_{ij} \\ &\quad \times \frac{\lambda_R}{2}\left\|\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\boldsymbol{\Theta}}(X_{ij})\right)\right\|_F^2 + \frac{\lambda_U}{2}\sum_i^m\|\mathbf{u}_i\|_F^2 \\ &\quad + \frac{\lambda_V}{2}\sum_j^n\|\mathbf{v}_j\|_F^2, \end{aligned} \tag{19}$$

where $\mathbf{Q}_{ij}$ represents an element in confidence matrix $\mathbf{Q}$. The value of each element in matrix $\mathbf{Q}$ will be determined in detail in the next section.

3.4. Confidence matrix

The confidence matrix, which is calculated by confidence degree function $\mathscr{F}(\cdot)$, represents the levels of contribution of different reviews to the model. The elements of this matrix represent the reciprocals of the probability of the outlier rating. The matrix $\mathbf{Q}$ can be regarded as the relative adjustment of regularization at each review. For convenience, the element values are scaled to the range [0, 1].

On the one hand, large element values should be selected for true reviews to learn the review interaction information from the review latent factor representation learning. On the other hand, small element values of the confidence matrix $\mathbf{Q}$ can be selected to construct the latent factor from the rating data. Thus, the following confidence degree function $\mathscr{F}(\cdot)$ determines the element values in matrix $\mathbf{Q}$,

$$\mathbf{Q}_{ij} = \mathscr{F}(\mathbf{R}_{ij},\beta) = \begin{cases} e^{-Relu\left(|\overline{\sum_{p\neq j,\mathbf{R}_{ip}\leqslant 3}\mathbf{R}_{ip}} - \mathbf{R}_{ij}| - \beta\right) - Relu\left(|\overline{\sum_{q\neq i,\mathbf{R}_{qj}\leqslant 3}\mathbf{R}_{qj}} - \mathbf{R}_{ij}| - \beta\right)}, & \mathbf{R}_{ij}\leqslant 3 \\ e^{-Relu\left(|\overline{\sum_{p\neq j,\mathbf{R}_{ip}>3}\mathbf{R}_{ip}} - \mathbf{R}_{ij}| - \beta\right) - Relu\left(|\overline{\sum_{q\neq i,\mathbf{R}_{qj}>3}\mathbf{R}_{qj}} - \mathbf{R}_{ij}| - \beta\right)}, & \mathbf{R}_{ij} > 3 \end{cases} \tag{20}$$

where β represents a threshold of deviation, which is set as 0.8 in this article. Misleading reviews are mainly caused by fake good and bad reviews. When a user gives a high score to an item, this score substantially deviates from the historical high score behavior average of the user and the historical score average of the item. Then, abnormal points are highly possible. Therefore, the corresponding reviews are also given a low weight. In Fig. 5, we illustrate the process of building confidence values.

4. Optimization and parameters determination

4.1. Optimization

Currently, a variety of optimization methods [46,47] are developed to optimize the objective function. In order to optimize our loss function, the mini-batch gradient descent (MBGD) algorithm is utilized to optimize the model (19). The update rule for all variables in CARM is provided as follows.

Update the U: The **U** estimation subproblem corresponds to minimize,

$$\begin{aligned} E_{CARM}^{\mathbf{U}} &= \sum_i^m\sum_j^n \frac{1_{ij}}{2}\left(\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}\right)^2 + \frac{\lambda_U}{2}\sum_i^m\|\mathbf{u}_i\|_\mathbf{F}^\mathbf{2} + \frac{\lambda_R}{2}\sum_i^m \\ &\quad \times \sum_j^n \mathbf{Q}_{ij}\left\|\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\boldsymbol{\Theta}}(\mathbf{X}_{ij})\right)\right\|_\mathbf{F}^\mathbf{2}. \end{aligned} \tag{21}$$

Then, the updated formulation can be defined by,

$$\begin{aligned} \mathbf{u}_i &\leftarrow \mathbf{u}_i - \alpha\frac{\partial E_{CARM}^{\mathbf{U}}}{\partial\mathbf{u}_i} \\ &= \mathbf{u}_i - \alpha\left\{\sum_j^n 1_{ij}\left(\mathbf{R}_{ij} - \widehat{\mathbf{R}}_{ij}\right)\cdot\mathbf{v}_j + \lambda_U\mathbf{u}_i + \sum_j^n 1_{ij}\mathbf{Q}_{ij}\lambda_R\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\boldsymbol{\Theta}}(X_{ij})\circ\mathbf{v}_j\right)\right\}. \end{aligned} \tag{22}$$

Update the V: Since **V** and **U** are equivalent, the objective function with respect to **V** can be written as,

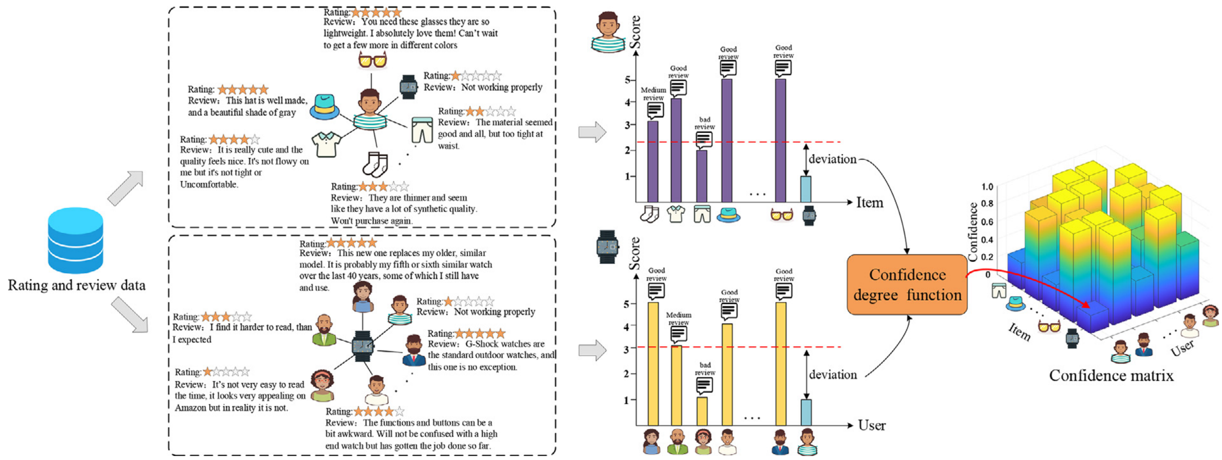

Fig. 5. The detailed process of confidence values determination in the proposed CARM model.

$$E^{\mathbf{V}}_{CARM} = \sum_{i}^{m}\sum_{j}^{n}\frac{1_{ij}}{2}\left(\mathbf{R}_{ij}-\widehat{\mathbf{R}}_{ij}\right)^2 + \frac{\lambda_V}{2}\sum_{j}^{n}||\mathbf{v}_i||^2_{\mathbf{F}} + \frac{\lambda_R}{2}\sum_{i}^{m} \times \sum_{j}^{n}\mathbf{Q}_{ij}||(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\Theta}(\mathbf{X}_{ij}))||^2_{\mathbf{F}}, \tag{23}$$

and,

$$\mathbf{v}_j \leftarrow \mathbf{v}_j - \alpha\frac{\partial E^{\mathbf{V}}_{CARM}}{\partial \mathbf{v}_j} = \mathbf{v}_j - \alpha\left\{\sum_{i}^{m}1_{ij}\left(\mathbf{R}_{ij}-\widehat{\mathbf{R}}_{ij}\right)\cdot\mathbf{u}_i + \lambda_V\mathbf{v}_j + \sum_{i}^{m}1_{ij}\mathbf{Q}_{ij}\lambda_R\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\Theta}(X_{ij})\circ\mathbf{u}_i\right)\right\}. \tag{24}$$

Update the γ, ψ: Similarly, the updated formulation of γ, ψ can be written as

$$\gamma_i \leftarrow \gamma_i - \alpha\frac{\partial E^{\gamma}_{CARM}}{\partial \gamma_i} = \gamma_i - \alpha\sum_{i}^{m}1_{ij}\left(\mathbf{R}_{ij}-\widehat{\mathbf{R}}_{ij}\right) \tag{25}$$

and,

$$\psi_j \leftarrow \psi_i - \alpha\frac{\partial E^{\psi}_{CARM}}{\partial \psi_j} = \psi_j - \alpha\sum_{j}^{n}1_{ij}\left(\mathbf{R}_{ij}-\widehat{\mathbf{R}}_{ij}\right) \tag{26}$$

Update the Θ: Since the gradients of the parameters in the internal structure of CNN are too complicated, the gradient of Θ is replaced by the partial derivative operator of $CNN_{\Theta}(X_{ij})$ function,

$$E^{\Theta}_{CARM} = \frac{\lambda_R}{2}\sum_{i}^{m}\sum_{j}^{n}\mathbf{Q}_{ij}||(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\Theta}(\mathbf{X}_{ij}))||^2_{\mathbf{F}}. \tag{27}$$

Then, the updated formulation can be denoted as,

$$\Theta \leftarrow \Theta - \alpha\frac{\partial E^{\Theta}_{CARM}}{\partial \Theta} = \Theta - \alpha\left\{\lambda_R\sum_{i}^{m}\sum_{j}^{n}\mathbf{Q}_{ij}\left(\mathbf{u}_i\circ\mathbf{v}_j - CNN_{\Theta}(X_{ij})\cdot\frac{\partial CNN_{\Theta}(X_{ij})}{\partial\Theta}\right\} \tag{28}$$

where α denotes the learning rate in CARM. After parameters of $\mathbf{U}, \mathbf{V}, \gamma$, and ψ are optimized, the unknown ratings can be predicted by Eq. (1). The parameters updating rule and optimazation process of CARM-C model are similar with CARM.

4.2. Parameters determination

In the CARM model, there are four parameters that need to be discussed. In review latent factor representation learning, k denotes the dimensionality of the review text and user/item latent factor vectors, which can control the representation ability of the proposed models. Meanwhile, the hyperparameter φ that affects the generalization ability refers to the parameter in dropout. The function of parameters λ_U, λ_V, and λ_R is to balance the constraint strength of regularization items in optimizing user and item representations.

To select the best values for the parameters, some approaches have been proposed to determine this parameter automatically such as the *L*-curve method [48], discrepancy principle [49], and generalized cross-validation (GCV) [50]. In this paper, the GCV algorithm is used to validate the parameter values in a large range and determine the best ones automatically. In other words, the parameters are determined heuristically. We suggest that promising performance can be achieved with the parameters φ=0.5, k=10, b = 256, $\lambda_U = \lambda_V \in$[0.0001, 0.001, 0.01, 0.1, 1] and $\lambda_R \in$[0.01, 0.1, 1, 10, 100]. More parameters details will be discussed in next section. The proposed algorithm is presented as Algorithm 1.

Algorithm 1. Confidence-aware recommender model via review representation learning and historical rating behavior

Input: $\mathbf{R}$: User-item rating matrix, X: Review text of items set
Set: Dimensionality k, learning rate α, batch size b;
1: Initialize $\mathbf{U}, \mathbf{V}, \gamma, \psi$ randomly;
2: While not E_{CARM} is converged do:
  i) Sample a mini batch from $\mathbf{R}$ and corresponding review text X_{ij} in size b;
  ii) Calculate the output result of the review representation learning model for the review samples;
  iii) Calculate the confidence value $\mathbf{Q}_{ij}$ of the current review based on the historical rating behavior;
  Update $\mathbf{U}, \mathbf{V}, \gamma, \psi, \Theta$ via (22), (24), (25), (26), (28) with mini batch.
End while
Output: $\mathbf{U}, \mathbf{V}, \gamma, \psi, \Theta$

5. Experiments and discussion

5.1. Experiment preparation

5.1.1. Evaluation metrics

The well-known root mean square error (RMSE) and mean absolute error (MAE) are adopted for recommendation performance evaluation,

Table 1
Comparison of characteristics by the traditional approaches and CNN-based technology.

Characteristics	Offset	PMF	HFT	DeepCoNN [44]	NARRE [23]	CARL [45]	CARM-C	CARM
Ratings	✓	✓	✓	✓	✓	✓	✓	✓
Textual reviews	–	–	✓	✓	✓	✓	✓	✓
Deep learning	–	–	–	✓	✓	✓	✓	✓
Pre-train word vector	–	–	–	✓	✓	✓	✓	✓
Review confidence degree	–	–	–	–	–	–	–	✓

Table 2
Statistical details of the four public datasets for recommendation.

Datasets	Users	Items	Ratings & Reviews	Density
Automotive	2,928	1,835	20,473	0.381%
Movies_and_TV	14,169	17,795	673,342	0.267%
Video_Games	24,303	10,672	231,780	0.090%
Yelp_2018	36,989	48,813	1,578,463	0.087%

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathbf{R}_{test}|}\sum_{\mathbf{R}_{ij}} \in \mathbf{R}_{test}\left(\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij}\right)^2}, \tag{29}$$

and,

$$\mathrm{MAE} = \frac{1}{|\mathbf{R}_{test}|}\sum_{\mathbf{R}_{ij}\in\mathbf{R}_{test}} |\mathbf{R}_{ij} - \widehat{\mathbf{R}_{ij}}|_{abs}, \tag{30}$$

where $|\mathbf{R}_{test}|$ represents the cardinality of the testset, and $|\bullet|_{abs}$ means the absolute value operation.

5.1.2. Tested methods

To compare with the proposed model, several the state-of-the-art recommendation methods are selected as follows. Their conceptual differences are illustrated in Table 1.

- *Offset*: The offset method takes the average of all the training set ratings as predictions in the test set.
- *PMF* [51]: It constructs the latent factors of items and users by adopting Gaussian distribution. This method corresponds to the model represented by Eq. (1).
- *HFT*[42]: It is a famous topic-based method, which utilizes topic model to learn latent factors from review.
- *DeepCoNN* [44]: This method uses two parallel CNNs to extract the latent factors from item and user review documents.
- *NARRE* [23]: It adopts neural attention mechanism to build latent feature vectors of items and users by selecting highly-useful review.
- *CARL*[45]: A novel approach is proposed in the CARL model, which introduces a dynamic linear fusion mechanism and CNNs to construct the feature vectors.
- *CARM-C*: This method is proposed in this article, we propose a novel constraint method based on user-item interactivity latent factor by single review feature representation learning model, which does not introduce the confidence matrix.
- *CARM*: CARM is a variation of CARM-C, which reduces the impact of misleading reviews on the model by introducing confidence matrix **Q**.

5.1.3. Datasets

The proposed models are evaluated on four public datasets. The first datasets subsets are selected from Yelp Dataset Challenge.[1] The other three datasets (Automotive, Video_Games, and Movies_and_TV) are selected from Amazon dataset.[2] Note that each record in the datasets includes the review text and corresponding rating score (from 1 to 5). Because the Yelp_2018 and Movies_And_TV data sets are too large for the experimental environment, they are preprocessed to ensure that all users and items have at least 20 ratings data. The detailed statistics of the four public datasets are shown in Table 2. In our experiments, each rating dataset is divided into 80% training and 20% testing, which are executed by using five-fold cross-validation technique.

5.2. Experimental implementation

The parameters of the comparison methods are adjusted as the suitable ones according to their papers. In CARM-C, word latent vectors are initialized by the GoogleNews word vectors. In review, the feature representation learning, various window sizes with shared weights are adopted to grasp the surrounding information of various length words. Subsequently, the dropout rate φ is set at 0.5, the number of latent factor k is equal to 10, the number of batch size b is set as 256, and the learning rate α is set as 0.001. CARM has the same parameter settings as CARM-C. The experiment was performed on a PC server equipped with an Intel(R) Core(TM) i7-7700K CPU@4.2 GHz, NVIDIA GeForce GTX 1080 Ti GPU, and 32 GB RAM and are implemented by utilizing the software library TensorFlow [52].

5.3. Results and discussion

5.3.1. Accuracy analysis

All the recommendation methods are carried out on the four public datasets. The comparisons of rating prediction results are summarized in Table 3. The best results are highlighted in bold font. According to the prediction results, the comparative analysis can be obtained as three aspects.

First, the metric values of HFT, DeepCoNN, NARRE, and CARL methods are less than those of PMF and Off-set methods in Table 3. The former four methods consider the reviews for the rating prediction, whereas the two latter methods only consider the rating information. The latent factor representation ability is improved greatly due to the rich side information in the review text.

Second, comparing with the traditional method using review and rating data, the deep learning-based methods (e.g., DeepCoNN, NARRE, and CARL) achieve prodigious improvement in RMSE and MAE on four datasets. The reason is that the contextual features of review can be extracted by the deep learn-ing techniques non-linearly. Moreover, the dropout operation in deep learning-based methods can avoid the overfitting issue and raise recommendation performance potentially.

[1] https://www.yelp.com/dataset_challenge.
[2] http://jmcauley.ucsd.edu/data/amazon.

Table 3
Comparison of rating prediction results by using the proposed method and six comparison methods. The best values are marked by bold font. Δ% denotes the improvement of CARM over the best baseline performer.

	Automotive		Movies_And_TV		Video_Games		Yelp_2018	
Methods	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Offset	0.9691	0.7339	1.1853	0.9268	1.1992	0.9436	1.1997	0.9757
PMF [51]	1.0768	0.8564	1.0428	0.7878	1.3965	1.0981	1.1938	0.9191
HFT [42]	1.0222	0.7277	1.0267	0.7579	1.1115	0.8435	1.1252	0.8702
DeepCoNN [44]	0.9305	0.6925	1.0096	0.7323	1.0706	0.8050	1.1218	0.8633
NARRE [23]	0.9187	0.6446	0.9947	0.7162	1.0607	0.7938	1.1106	0.8454
CARL [45]	0.9078	0.6207	0.9831	0.7048	1.0637	0.7942	1.1204	0.8494
CARM-C	0.9072	0.6152	0.9787	0.6830	1.0581	0.7770	1.0997	0.8374
CARM	**0.8960**	**0.5965**	**0.9742**	**0.6797**	**1.0531**	**0.7691**	**1.0936**	**0.8261**
Δ%	1.30%	3.90%	0.90%	3.56%	1.00%	3.11%	1.53%	2.28%

Thirdly, the proposed methods obtain the lowest values at the MAE and RMSE metrics in Table 3. It shows that CARM-C has a certain improvement in performance. This also shows that the review latent factor learned from a single review is useful for the construction of user and item true unbiased preference vector. Furthermore, the performance of CARM, which considers the impact of user rating behavior on review confidence, is improved compared to CARM-C model. This also illustrates that assigning a lower weight to misleading reviews will help to achieve a more accurate model. Specifically, comparing with the PMF model, CARM achieves 6.6%–24.6% improvement in RMSE and 10.1%–30.3% in MAE. CARM demonstrates bet-ter prediction performance than CARM-C at the aspects of MAE and RMSE because it reduces influence of misleading review on the model construction. Thus, the proposed models outperform all the baseline methods consistently.

5.3.2. Training efficiency analysis of CARM

Afterward, DeepCoNN, NARRE, and CARL use aggregated review text to model user/item feature vectors. This comparison will cause the input text to be extremely large in each training iteration.

Thus, the scale of model parameters is extremely large in the aforementioned model. NARRE lags far behind DeepCoNN as the attention mechanism over feature extraction incurs extra burden. Similarly, CARL adopts factorization machines to constrain the representations of item and user vector, which increases the training time cost for each epoch. Unlike these methods, CARM-C and CARM train the model with only one review text at a time, thereby enhancing model training efficiently and reducing the scale of model parameters prominently. Experimental results prove that CARM achieves better performance on training efficiency when compares with DeepCoNN, NARRE, and CARL methods. In Fig. 6, we plot the training process of these methods with epoch number increasing. CARM method (red line in Fig. 6) achieves high converge speed and stability. Moreover, the lowest RMSE values illustrate that the proposed method can extract the review features efficiently by network in Fig. 3.

5.3.3. Sensitivity analysis of the parameters

From the objective function Eq. (18) of the CARM-C model and the objective function Eq. (19) of the CARM, it can be seen that the values of the hyperparameters λ_U, λ_V and λ_R will directly affect the performance of the model, so it is necessary to explore the influence of different values on the model. Fig. 7 analyze the impact of λ_U and λ_V on RMSE values by four datasets. Fig. 8. shows the impact of λ_U, λ_V and λ_R different values on the mean square error of the model rating prediction on all datasets.

Regarding the changes of the best performing values of λ_U and λ_V from Fig. 7(a)–(d), we insert different number of batch size $\lambda_U, \lambda_V \in [0.0001, 0.001, 0.01, 0.1, 1.0]$ to observe the impact of different λ_U and λ_V on RMSE values by four datasets. We found that when λ_U is equal to λ_V, model can produce the best results. This result is also in line with the assumptions of the previous model. That is, we place zero-mean spherical Gaussian priors on users and items latent factor vectors respectively. Since the prior assumptions of both user latent factors and item latent factors are consistent with Gaussian distribution, it is reasonable to set their regularization parameters λ_U and λ_V consistently. Experimental results in Fig. 7 demonstrate that the reasonableness of setting λ_U and λ_V to be the same.

By observing the best parameters in the Fig. 8 experimental results, when the rating matrix becomes sparse, the best values of λ_U and λ_V will become smaller and the best value of λ_R becomes larger. To be precise, the best values of λ_U, λ_V and λ_R for CARM model on four datasets are (0.1, 0.1, 1), (0.1, 0.1, 1), (0.01, 0.01, 10), and (0.01, 0.01, 10), respectively. This result is reasonable because relatively large value of λ_R will increase the contribution of review information to latent factors of users and items when rating data is insufficient to construct accurate latent factors. Appropriate λ_U and λ_V can raise the generalization ability of model representation, but too large will make the model difficult to fit the rating data. That is, when λ_R is increasing, the latent factor representation of users and items will be more dependent on review information rather than rating data. In other words, λ_R is leveraged in the proposed model to tradeoff the contribution of the rating information and the review information. Thus, the suitable parameters λ_U, λ_V and λ_R can balance the importance of ratings and review texts and obtain the optimal result.

Moreover, the parameter k is robust on different datasets. Fig. 9 depicts the performance of CARM by various the latent factor dimension size $k \in [5, 10, 15, 20]$. The model achieves the lowest RMSE and MAE values when k is equal to 10 on all four datasets. In addition, it is necessary to analyze the number of batch size. The different number of batch size $b \in [32, 64, 128, 256, 512]$ are inserted to observe the impact of different batch sizes on RMSE and MAE values by four datasets, and the learning rate is set at 0.001. The experimental results, which is presented in Fig. 10, demonstrate that the number of batch size has little effect on CARM model performance. Furthermore, when the number of batch size b are set as 256 and 512, the model performance is almost the same and achieves good results on RMSE and MAE values. Taking into account the use of GPU memory and the efficiency of the model, the number of batch size b is suitable to set as 256 for our proposed models.

The dropout ratio φ plays an important role in enhancing the generalization ability of the proposed CARM. The experiments are carried out on four different datasets with the changing dropout ratio φ. In Fig. 11, we plot the RMSE and MAE values by using CARM method. The experimental results demonstrate that the optimal dropout ratio on the four datasets is 0.5. Dropout can

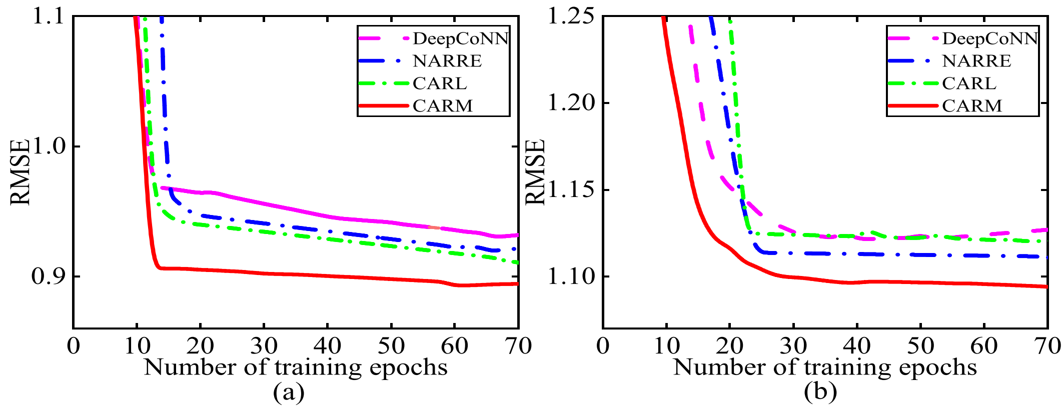

Fig. 6. Comparison of RMSE values by four different methods with the number of training epochs increasing. (a) Automotive, (b) Yelp_2018.

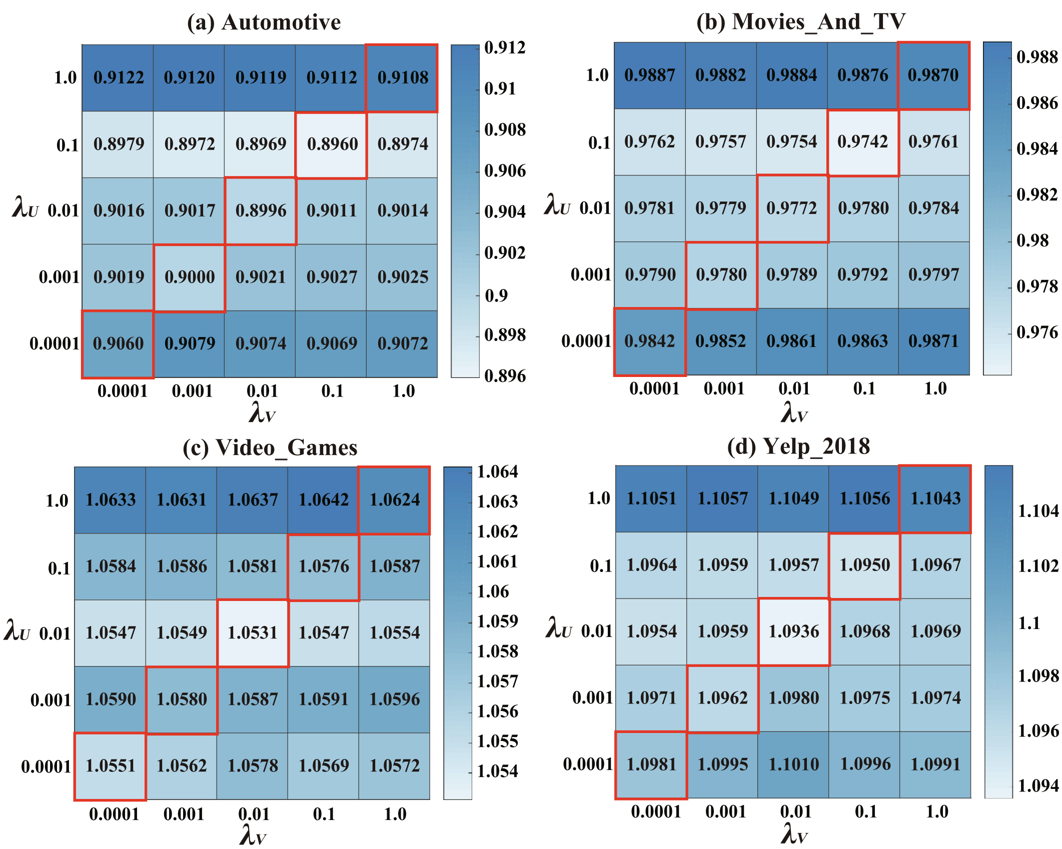

Fig. 7. Parameter analysis of λ_U and λ_V in CARM on RMSE values by four datasets. (a) Automotive, (b) Movies_and_TV, (c) Video_Games, (d) Yelp_2018.

improve the predication performance effectively. Thus, the dropout ratio φ is suggested as 0.5 in real scenarios.

5.3.4. Word weight analysis

The weight values of the word vector in the review latent factor representation learning can reflects the contribution of the word to the model construction. Fig. 12 shows that some reviews are selected to perform weighted visualization experiment results. In order to extract the features between contexts, which can also be considered as phrase information (e.g., bigrams or trigrams), multiple convolution kernels (e.g., $2 \times 2, 3 \times 3, 4 \times 4$) of different sizes are introduced to extract contextual features. Thus, our model can finally show the contribution of the word and phrases (e.g., bigrams or trigrams) to the model construction. One review corresponding to a five-point rating from the Automotive dataset of Amazon and another review with a one-point rating are randomly selected. In Fig. 12, the highlighted words are valuable to the final rating. The most significant words are highlighted in red, the moderately important words are highlighted in light orange, and the insignificant words are not highlighted. Fig. 12 reveals that stop words, prepositions, and some nouns are given low weights. This finding illustrates that these words unimportant to the results. Meanwhile, the weights of adverbs, adjectives, and interjections are highlighted in red. These words play important roles in our model due to their strong subjective color. This experiment intuitively shows the influence of different words in the review to model construction, which strengthens the interpretability of the proposed model.

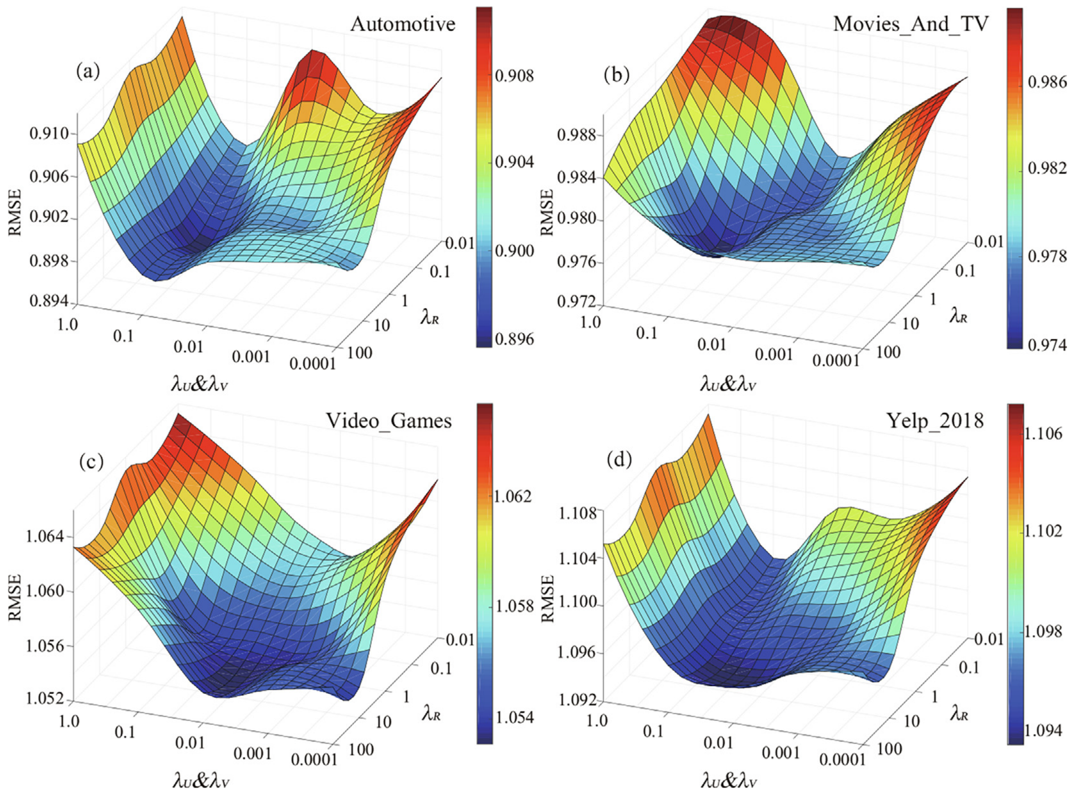

Fig. 8. Regularization parameter discussion in the proposed CARM method. The robustness of the parameters can be verified on different datasets. (a) Automotive, (b) Movies_and_TV, (c) Video_Games, (d) Yelp_2018.

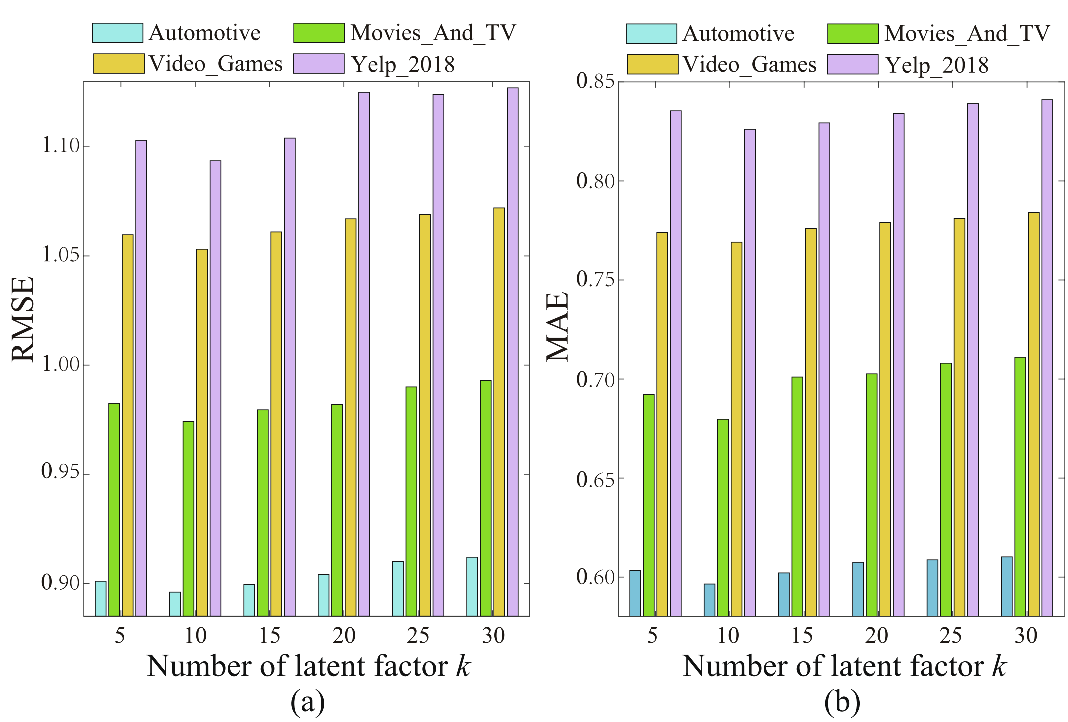

Fig. 9. Discussion of dimension number k in CARM on four different datasets. (a) RMSE. (b) MAE.

According to the previous quantitative analysis, the Table 4 provides a qualitative evaluation of all the methods in terms of speed and performance, using the PMF method as a benchmark. The properties include the method type, latent factor constraints type, input data type, computing speed, and performance. The proposed method achieves the highlight performance.

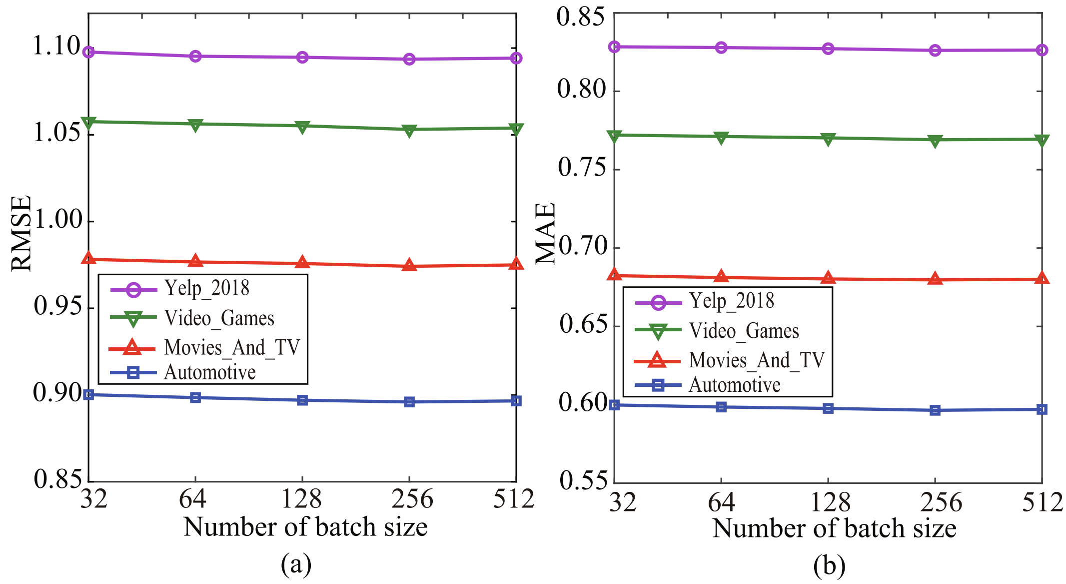

Fig. 10. Discussion of the number of batch size in CARM on four different datasets. (a) RMSE. (b) MAE.

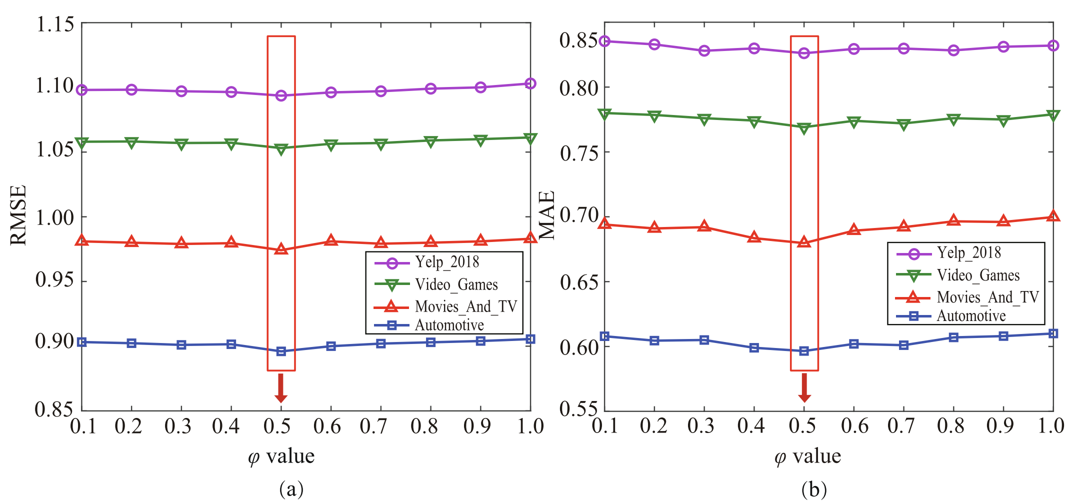

Fig. 11. Different datasets affect the dropout φ setting, which increases from 0.1 to 1. φ the in proposed CARM is robust (φ=0.5) to the changing datasets. (a) RMSE. (b) MAE.

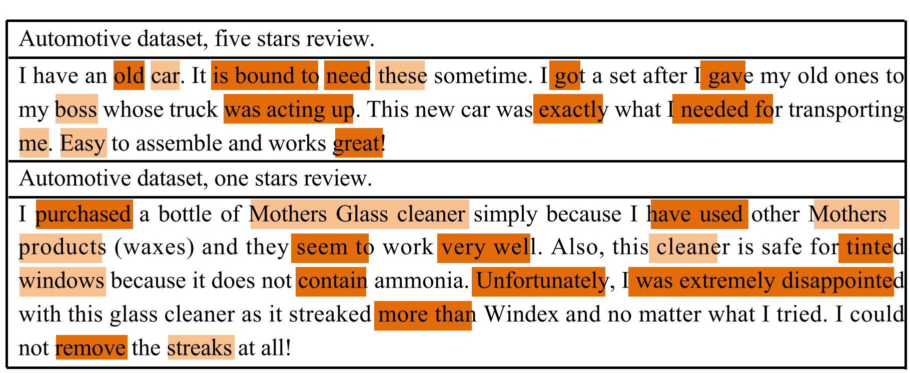

Fig. 12. Word weight analysis in user review, the darker the color, the greater the weight.

Table 4
Comparison the performance between the proposed method and the state-of-the-art methods. Speed indicates the model training time in seconds, and RMSE is utilized to measure the performance of all models.

Methods	Latent factor constraints type	Input data type	Speed (s)	Performance (RMSE)
PMF [51]	SLF	Ratings	★★★★★	★★
HFT [42]	SLF	Ratings&Reviews	★★★★	★★★
DeepCoNN [44]	DLF	Ratings&Reviews	★★★☆	★★★☆
NARRE [23]	DLF	Ratings&Reviews	★★	★★★★
CARL [45]	DLF	Ratings&Reviews	★★★☆	★★★★☆
CARM	ILF	Ratings&Single review	★★★★☆	★★★★★

6. Conclusion

In this study, we propose a novel confidence-aware recommendation model via review representation learning and historical rating behavior. To reduce the rating data sparseness issue, we introduce a novel constraint method based on user-item interactivity by exploiting single review information to construct interaction latent factors of items and users. Meanwhile, the confidence matrix is introduced considering the impact of misleading reviews on the model to further improve the accuracy of the recommendation. The loss function is also constructed by maximum a posteriori estimation theory. Finally, the mini-batch gradient descent algorithm is introduced to optimize the loss function. Experiments demonstrate the effectiveness of the proposed review representation learning scheme and user-item interactivity as well as the significant and consistent improvements over other state-of-the-art recommendation methods. The proposed CARM model will promote the application of recommendation system in the field of learning resource adaptation. In the future, user-side and rating time series cues will be introduced to build accurate recommendation models.

CRediT authorship contribution statement

Duantengchuan Li: Writing - original draft. **Hai Liu:** Writing - review & editing. **Zhaoli Zhang:** Data curation. **Ke Lin:** Data curation. **Shuai Fang:** Data curation. **Zhifei Li:** Data curation. **Neal N. Xiong:** Conceptualization, Methodology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. **Acknowledgements**: The authors would like to thank the reviewers for their valuable and helpful suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant 62011530436, Grant 62077020, Grant 62005092, and Grant 61875068, and the Fundamental Research Funds for the Central Universities under Grant CCNU20ZT017, Grant CCNU2020ZN008, Grant 2020YBZZ009.

References

[1] J. Xuan, X. Luo, J. Lu, G. Zhang, Explicitly and implicitly exploiting the hierarchical structure for mining website interests on news events, Inf. Sci. 420 (2017) 263–277.

[2] W. Pan, A survey of transfer learning for collaborative recommendation with auxiliary data, Neurocomputing 177 (2016) 447–453.

[3] J. Niu, X. Zhao, L. Zhu, H. Li, Affivir: An affect-based internet video recommendation system, Neurocomputing 120 (2013) 422–433.

[4] H. Li, J. Cui, B. Shen, J. Ma, An intelligent movie recommendation system through group-level sentiment analysis in microblogs, Neurocomputing 210 (2016) 164–173.

[5] J. Shu, X. Shen, X. Zhou, B. Yi, Z. Zhang, A content-based recommendation algorithm for learning resources, Multimed. Syst. 24 (2018) 163–173.

[6] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A.C. Ammari, A. Alabdulwahab, Generating highly accurate predictions for missing qos data via aggregating nonnegative latent factor models, IEEE Trans. Neural Netw. Learn. Syst. (2016) 524–537.

[7] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[8] X. Luo, M. Zhou, S. Li, M. Shang, An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications, IEEE Trans. Ind. Inform. 14 (2018) 2011–2022.

[9] X. Luo, M. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, H. Leung, Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing qos data, IEEE T. Cybern. 48 (2018) 1216–1228.

[10] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, N. Xiong, Deep matrix factorization with implicit feedback embedding for recommendation system, IEEE Trans. Ind. Inform. 15 (2019) 4591–4601.

[11] X. Shen, B. Yi, H. Liu, W. Zhang, Z. Zhang, S. Liu, N. Xiong, Deep variational matrix factorization with knowledge embedding for recommendation system, IEEE Trans. Knowl. Data Eng. 33 (2021) 1906–1918.

[12] T. Guo, J. Luo, K. Dong, M. Yang, Differentially private graph-link analysis based social recommendation, Inf. Sci. 463–464 (2018) 214–226.

[13] Z. Guo, H. Wang, A deep graph neural network-based mechanism for social recommendations, IEEE Trans. Ind. Inform. 17 (2020) 2776–2783.

[14] C. Hsu, M. Yeh, S. Lin, A general framework for implicit and explicit social recommendation, IEEE Trans. Knowl. Data Eng. 30 (2018) 2228–2241.

[15] S. Li, J. Kawale, Y. Fu, Deep collaborative filtering via marginalized denoising auto-encoder, in: Proc. 24nd Int. Conf. Inf. Knowledge Manage., 2015, pp. 811–820..

[16] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, IEEE Trans. Knowl. Data Eng. 32 (2020) 1854–1867.

[17] D.C.G. Pedronette, R. da S. Torres, Exploiting pairwise recommendation and clustering strategies for image re-ranking, Inf. Sci. 207 (2012) 19–34.

[18] D. Rafailidis, F. Crestani, Adversarial training for review-based recommendations, in: Proc. 42nd Int. ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1057–1060.

[19] X. He, T. Chen, M.-Y. Kan, X. Chen, Trirank, Review-aware explainable recommendation by modeling aspects, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1661–1670.

[20] D. Hyun, C. Park, J. Cho, H. Yu, Learning to utilize auxiliary reviews for recommendation, Inf. Sci. 545 (2021) 595–607.

[21] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: Proc. World Wide Web Conf., 2019, pp. 417–426.

[22] D. Kim, C. Park, J. Oh, H. Yu, Deep hybrid recommender systems via exploiting document context and statistics of items, Inf. Sci. 417 (2017) 72–87.

[23] C. Chen, M. Zhang, Y. Liu, S. Ma, Neural attentional rating regression with review-level explanations, in: Proc. World Wide Web Conf., 2018, pp. 1583–1592..

[24] C. Aggarwal, Attack-resistant recommender systems, Recommend. Syst. (2016) 385–410.

[25] M. Jiang, P. Cui, C. Faloutsos, Suspicious behavior detection: Current trends and future directions, IEEE Intell. Syst. 31 (2016) 31–39.

[26] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, What yelp fake review filter might be doing?, in: ICWSM, 2013, pp 409–418.

[27] V. Sandulescu, M. Ester, Detecting singleton review spammers using semantic similarity, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 971–976.

[28] A. Mukherjee, B. Liu, N. Glance, Spotting fake reviewer groups in consumer reviews, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 191–200.

[29] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, J. Shao, Bimodal distribution and co-bursting in review spam detection, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1063–1072.

[30] Z. Wu, Y. Wang, Y. Wang, J. Wu, J. Cao, L. Zhang, Spammers detection from product reviews: A hybrid model, in: 2015 IEEE International Conference on Data Mining, 2015, pp. 1039–1044.

[31] S. Rayana, L. Akoglu, Collective opinion spam detection: Bridging review networks and metadata, in: Proc. 21rd ACM SIGKDD Int. Conf. Knowl. Discov. Data min., 2015, pp. 985–994..

[32] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proc. World Wide Web Conf., 2001, pp. 285–295..

[33] C. Leng, H. Zhang, G. Cai, I. Cheng, A. Basu, Graph regularized lp smooth non-negative matrix factorization for data representation, IEEE/CAA J. Automat. Sin. (2019) 584–595.

[34] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2008, pp. 426–434..

[35] S. Rendle, Factorization machines, in: Proc. IEEE Int. Conf. Data Min., 2010, pp. 995–1000..

[36] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec, Social recommendation using probabilistic matrix factorization, in: Proc. CIKM., 2008, pp. 931–940.

[37] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proc. ACM Conf. Recomm. Syst., 2010, pp. 135–142..

[38] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data min., 2017, pp. 305–314..
[39] H. Wang, D. Yeung, Towards bayesian deep learning: A framework and some existing methods, IEEE Trans. Knowl. Data Eng. 28 (2016) 3395–3408.
[40] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2014, pp. 83–92..
[41] Y. Tan, M. Zhang, Y. Liu, S. Ma, Rating-boosted latent topics: Understanding users and items with ratings and reviews., in: Int. Joint Conf. Artif. Intell., vol. 16, 2016, pp. 2640–2646..
[42] J. McAuley, J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in: Proc. ACM Conf. Recomm. Syst., 2013, pp. 165–172..
[43] K. Lin, L. Gong, Y. Huang, C. Liu, J. Pan, Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network, Front. Plant Sci. 10 (2019) 155.
[44] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, 2017, pp. 425–434.
[45] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, X. Luo, A context-aware user-item representation learning for item recommendation, ACM Trans. Inf. Syst. 37 (2019) 1–29.
[46] X. Luo, W. Qin, A. Dong, K. Sedraoui, M. Zhou, Efficient and high-quality recommendations via momentum-incorporated parallel stochastic gradient descent-based learning, IEEE/CAA J. Automat. Sin. 8 (2021) 402–411.
[47] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE T. Cybern. 51 (2021) 1085–1093.
[48] P.C. Hansen, Analysis of discrete ill-posed problems by means of the l-curve, SIAM Rev. 34 (1992) 561–580.
[49] H.W. Engl, Discrepancy principles for tikhonov regularization of ill-posed problems leading to optimal convergence rates, J. Optim. Theory Appl. 52 (1987) 209–215.
[50] G. Golub, MichaelHeath, GraceWahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics 21 (1979) 215–223.
[51] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: Adv. Neural Inf. Process. Syst., 2007, pp. 1257–1264..
[52] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, Tensorflow: a system for large-scale machine learning, in: Proc. USENIX Symp. Oper. Syst. Des. Implement., vol. 16, 2016, pp. 265–283..

Duantengchuan Li is currently working toward the M. S. degree in computer science with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China. His research interests include deep learning, representation learning, natural language processing and their applications in recommendation system and big data analysis.

Hai Liu (M'14-SM'21) received the M.S. degree in applied mathematics from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010, and the Ph. D. degree in pattern recognition and artificial intelligence from the same university, in 2014. Since June 2017, he has been an Assistant Professor with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan. From 2017 to 2019, he was a postdoctoral fellow in the Department of Mechanical Engineering, City University of Hong Kong, Kowloon, Hong Kong, where he was hosted by the Professor You-Fu Li. In 2020, he was selected as "China-European Commission Talent Programme" under National Natural Science Foundation of China (NSFC). He is a senior researcher with UCL Interaction Centre, University College London, London, United Kingdom, where he will host by the Professor Sriram Subramanian; he will hold the position one year till February 2022. He has authored more than 70 peer-reviewed articles in international journals from multiple domains. More than six articles are selected as the ESI highly cited articles. His current research interests include deep learning, artificial intelligence, recommendation system, head pose estimation, gaze estimation, educational technology and pattern recognition. Dr. Liu has been frequently serving as a reviewer for more than six international journals including the IEEE Translations on Industrial Informatics, IEEE Translations on Cybernetics, IEEE/ASME Transactions on Mechatronics, and IEEE Translations on Instrumentation and Measurement. He is also a Communication Evaluation Expert for the NSFC from 2016 to present. He won the first prize of Science and Technology Progress Award by the Hubei Province of China in 2020.

Zhaoli Zhang (M'18) received the M.S. degree in Computer Science from Central China Normal University, Wuhan, China, in 2004, and the Ph.D. degree in Computer Science from Huazhong University of Science and Technology in 2008. He is currently a professor in the National Engineering Research Center for ELearning, Central ChinaNormalUniversity. His research interests include human-computer interaction, deep learning, signal processing, knowledge services and software engineering. He is a member of IEEE and CCF (China Computer Federation).

Ke Lin is currently working toward the Ph.D. degree in Control Science and Engineering at Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include deep reinforcement learning, machine learning, and their applications in recommendation system.

Shuai Fang received the B.S. degrees from Sichuan University, Chengdu, China, in 2019. He is currently working in Central China Normal University, Wuhan, China. His research interests include pattern recognition, human-machine interface and recommendation system.

Zhifei Li received the M.S. degree in Collaborative Innovative Center for Educational Technology from Central China Normal University in 2018. He is currently working toward the Ph.D. degree in the National Engineering Research Center for E-Learning, Central China Normal University. His current research interests include knowledge graph, graph neural network and recommendation system. He is a student member of CCF (China Computer Federation).

Neal N. Xiong (M'08-SM'12) received the Ph.D. degrees in sensor system engineering and in dependable sensor networks from Wuhan University and the Japan Advanced Institute of Science and Technology, respectively. He is currently a Professor with the Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK, USA. His research interests include cloud computing, machine learning, representation learning, and optimization theory. He is serving as the Editorin-Chief, an Associate Editor, or an Editor member for over ten international journals, including an Associate Editor for the IEEE Transactions on Systems, Man, And Cybernetics: Systems and Information Sciences.