8-2020

InterSentiment: Combining deep neural models on interaction and sentiment for review rating prediction

Shi FENG
Northeastern University

Kaisong SONG
Alibaba Group

Daling WANG
Northeastern University

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Yifei ZHANG
Northeastern University

Published in International journal of machine learning and cybernetics, 2020 August, 12 (2), 477-488.
https://doi.org/10.1007/s13042-020-01181-9

InterSentiment: combining deep neural models on interaction and sentiment for review rating prediction

Shi Feng[1] · **Kaisong Song**[2] · **Daling Wang**[1] · **Wei Gao**[3] · **Yifei Zhang**[1]

Abstract
Review rating prediction is commonly approached from the perspective of either Collaborative Filtering (CF) or Sentiment Classification (SC). CF-based approach usually resorts to matrix factorization based on user–item interaction, and does not fully utilize the valuable review text features. In contrast, SC-based approach is focused on mining review content, but can just incorporate some user- and product-level features, and fails to capture sufficient interactions between them represented typically in a sparse matrix as CF can do. In this paper, we propose a novel, extensible review rating prediction model called InterSentiment by bridging the user-product interaction model and the sentiment model based on deep learning. InterSentiment is a specific instance of our proposed Deep Learning based Collaborative Filtering framework. The proposed model aims to learn the high-level representations combining user-product interaction and review sentiment, and jointly project them into the rating scores. Results of experiments conducted on IMDB and two Yelp datasets demonstrate clear advantage of our proposed approach over strong baseline methods.

Keywords Review rating prediction · Deep neural networks · Matrix factorization · Sentiment analysis · User–product interaction

1 Introduction

Review rating prediction is an important sentiment analysis task which aims to detect users' sentiment intensity towards target products from vast amount of subjective reviews on online websites (e.g., 1–5 stars in Yelp, or 1–10 stars in IMDB).

Early research approaches to the task from either the angle of Sentiment Classification (SC) or that of Collaborative Filtering (CF). SC-based models primarily follow Pang and Lee [18] by concentrating text mining and regard the problem as a Single-Label Multi-Class (SLMC) classification task. Most of studies in this approach rely on handcrafted features and/or sentiment lexicons for achieving effective learning performance, which however is biased and labor intensive [5, 8]. Recently, neural network based models have achieved promising SC results. These models have strong representation learning capacity that can capture and organize discriminative features automatically extracted from data. Some of such studies have noticed the importance of user and product elements on interpreting the sentiment of reviews [27, 28], thus using user- or product-specific preference matrix/vector to adjust text semantics in their models. However, they cannot capture deep, sufficient *interactions* between users and products, which is what machine CF-based models are good at.

CF-based approach focuses on modeling users' preferences on product items according to their past *interactions*

✉ Shi Feng
fengshi@cse.neu.edu.cn

Kaisong Song
kaisong.sks@alibaba-inc.com

Daling Wang
wangdaling@cse.neu.edu.cn

Wei Gao
wei.gao@vuw.ac.nz

Yifei Zhang
zhangyifei@cse.neu.edu.cn

[1] School of Computer Science and Engineering, Northeastern University, No. 195 Chuangxin Road, Hunnan District, Shenyang, China

[2] Alibaba Group Hangzhou, Hangzhou, China

[3] Victoria University of Wellington, Wellington, New Zealand

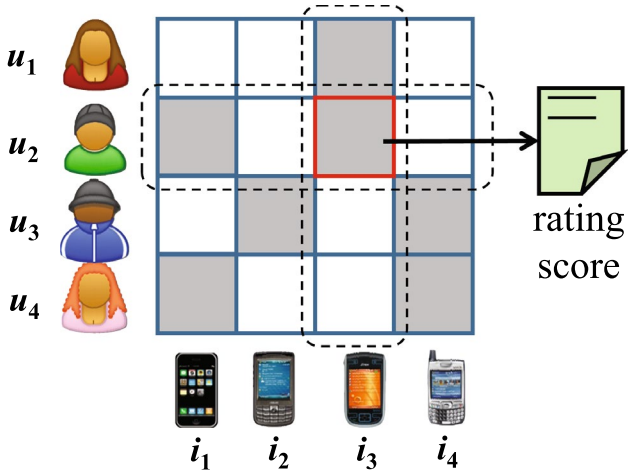

Fig. 1 User product interactions. The grey square means that the product item is rated by the corresponding user

(e.g., ratings and clicks). Figure 1 gives an illustration of user-product interaction information for an online review website. In Fig. 1, different users rate varying product items and different product items are rated by varying users. This interaction information has been successfully captured by various CF techniques, among which Matrix Factorization (MF) is the most popular one for estimating the rating score by modeling the inner product of user and product latent vectors [14, 21]. On the other hand, some recent work has advanced to take into consideration of *sentiment* in review text for improving prediction quality [13, 25, 34]. However, instead of using content features directly in the rating's modeling, they mostly treat text as auxiliary means, such as for helping interpret the models by just extracting user/product features from reviews [34], or for guiding factor estimation using shallow text representation [13, 25]. It is therefore suboptimal without leveraging deeper and valuable semantics of review text directly.

Such modeling issues discussed above motivate us to combine the review sentiment model and user–product interaction model, especially those based on the latest Deep Neural Networks (DNN), under a unified framework for review rating prediction. The intuitive idea is to use the joint model for generating and combining the high-level representations on the *interaction* and *sentiment content* more effectively. This issue has not been well addressed previously as most existing studies either do not consider the representation of interactions [27, 28] or ignore using the valuable review contents [15, 23]. In a more recent study [25], both review and interaction are jointly used. However, the review semantics is incorporated simply as Bag-of-Words, which is very shallow, since the model is designed exclusively from a CF perspective.

Some recent works have applied DNN to MF and shown promising results, however, they mostly use DNN to model review as auxiliary information [13] or to learn the user-product interaction function [12], which cannot be directly applied to our review rating prediction task. To address the issue of joint modeling, we present a unified Deep Learning based Collaborative Filtering framework (dubbed as DLCF). DLCF framework extends the Neural Collaborative Filtering (NCF) model [12] by taking into account deep text semantics of reviews, and instantiate it as a novel review rating prediction model called InterSentiment, so as to capture both high-level representations of user-product *interaction* and deep semantics of *sentiment*.

To sum up, the major contributions of this paper are presented as follows:

- We propose a novel review rating prediction model InterSentiment based on a smooth combination of MF and Neural Networks (NN), which captures the user preferences and product characteristics on sentiment expression embedded in reviews. InterSentiment is a two-layer instantiation of our DLCF framework, and it first produces user-product combined representation and document representation with NN, and later feeds them to a MF projection layer and yields rating prediction.
- We leverage a multi-layer perceptron to endow a high level of nonlinearities for representing each user-product preference interaction, and a convolutional neutral network with multiple filters to capture local semantics of text features for the review representation.
- We perform extensive experiments on three public datasets and the results demonstrate that InterSentiment outperforms strong baseline models by a large margin.

The subsequent sections of this paper are organized as follows: Sect. 2 summarizes the work related to this article. Section 3 provides a formal definition of the review rating prediction task and a brief introduction of MF models. In Sect. 4, we introduce a unified framework DLCF that considers both user-product interaction and review text. In Sect. 5, we propose InterSentiment model as an instantiation of DLCF for rating prediction. Section 6 presents the experimental schemes and discusses the results. Finally, the concluding remarks and future work are given in Sect. 7.

2 Related work

Two types of previous literature are relevant to our work: sentiment analysis and matrix factorization.

2.1 Sentiment analysis

With the prolific rise of data from social media such as Twitter and Weibo, analyzing the sentiments and opinions embedded in these user-generated-contents has drawn great attentions from both academic researchers and commercial companies [9, 18, 33, 37]. Cai et al. [3] found that the sentiment words have diverse meanings in different context and proposed a lexicon based hybird model for domain-sensitive sentiment classification. Dridi and Recupero [7] leveraged the frame semantics and lexical resources to extract semantic features from text, and achieved better polarity detection results with the help of these features. Firdaus et al. [10] studied on the impact of the user's retweet behavior and emotion, and the experiment results validated the positive correlation between these two important factors.

Despite the success of existing methods for analyzing sentiment from subjective text, recent attention is increasingly shifting towards considering the influences of users and products on sentiments. Tang et al. [28] proposed a review rating prediction model based on NN, but they only consider the effects of users on sentiment expression at word level. Later, Tang et al. [27] further presented a strong sentiment classification model called User Product Neural Network model (UPNN) based on convolutional neural network, to which our work is more closely related. Dou [6] employed a deep memory network to capture the user and product information for better classification results. Amplayo et al. [1] utilized shared vectors from similar users/products to alleviate the sparseness problem in the insufficient training data.

These neural-based models still suffer the following problems: (1) The introductions of preference matrix for each user/product in their methods are insufficient and difficult to be well trained with limited reviews, which might significantly drag down the prediction quality; (2) The user and product effects on ratings should be reflected on the document level rather than word level since documents always reflect high-level semantics and interact with users/products directly. Different from their methods, our approach does not rely on user/product preference matrix. In addition, we aim to explicitly model the user-product interaction preferences on the semantics of documents.

2.2 Matrix factorization

Popularized by the Netflix Prize, MF has been widely used in online recommendation systems [2, 22, 34]. However, most of them focus on predicting ratings from user-product interactions. Latent factor model (LFM) is a popular MF method for rating prediction in the recommendation field [32]. Much research effort has been devoted to enhancing LFM, such as integrating it with neighbor-based models [14], combining it with topic models of item content [29], and extending it to factorization machines [19] for a generic modeling of features.

Recently, some literature have employed text information. Li et al. [15] proposed a user-product-word tensor factorization model for review rating prediction. Mukherjee et al. incorporated author preferences to capture the facet level ratings [17]. Later, Song et al. [23, 24] proposed personalized microblog sentiment classification problem, which was also explored by Wu and Huang [30] in a multi-task learning framework. Zhang et al. [36] integrated the rich attributes of items and social links of users into MF models for alleviating the rating sparsity effect. Compared with neural network methods, these MF based methods mostly utilize the Bag-of-Words model and can not fully use text features.

Although DNN and MF have achieved promising results in review rating prediction, the exploration of DNN on recommender systems has received relatively less scrutiny. Recently, He et al. [12] proposed a NCF framework by adopting a multi-layer representation to model a user-item interaction, which generalized the basic MF under a neural network architecture. However, their model is unfit for our task for ignoring the text information. In this work, we incorporate users, products and reviews into the factorization process of MF by combining with NN, and finally propose a novel review rating prediction model.

3 Preliminaries

For a typical online review website such as Yelp[1] or IMDB[2], we would have a set of users $\mathcal{U}$ writing reviews $\mathcal{R}$ on a set of products $\mathcal{I}$. We use y_{ui} and $r_{ui} \in \mathcal{R}$ to respectively denote a rating level and the text review that user $u \in \mathcal{U}$ gives on a product $i \in \mathcal{I}$. Let $p_u \in \mathbb{R}^d$ and $q_i \in \mathbb{R}^d$ be the user-factors vector and product-factors vector. The basic MF model [14, 21] estimates the rating score $\hat{y}_{ui}$ by modeling the inner product of p_u and q_i as below:

$$\hat{y}_{ui} = p_u^{\mathrm{T}} q_i \tag{1}$$

The MF cannot capture the complex structure of interaction data sufficiently, because the inner product models the two-way interaction of user and item latent factors, assuming each dimension of the latent space is independent of each other and linearly combining them with the same weight. Therefore, MF can be regarded as a linear model of latent factors. Moreover, the existing variants of MF model could not make full use of text features, because they usually

[1] https://www.yelp.com/

[2] https://www.imdb.com/.

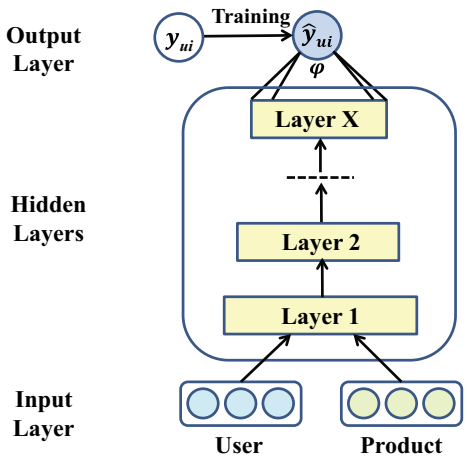

Fig. 2 The overall framework of Neural Collaborative Filtering [12]

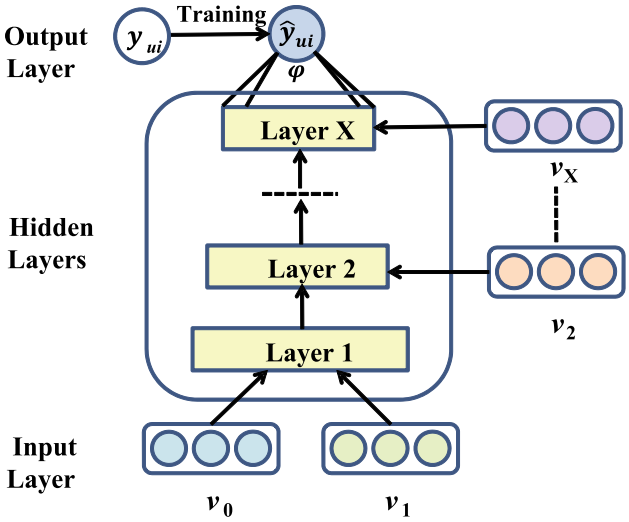

Fig. 3 The overall framework of DLCF

employ Bag-of-Words models and thus ignore the word orders and context information [12, 15, 23, 25].

4 Deep learning based collaborative filtering

Limited literature have been published on combining the MF models with DNN for rating prediction. He et al. discussed the defects of MF models [12] and proposed a generic framework NCF [12] to learn complex user-product interactions based on DNN, as shown in Fig. 2.

NCF is essentially an extensible neural network framework for recommendation tasks, which has multi-layer perceptrons for learning better prediction model. Compared with the classical inner product operation of MF models, NCF is endowed with high level nonlinear modelling ability to capture user-product interactions, which is formulated as follows.

$$\hat{y}_{ui} = \phi_{out}\big(\phi_X(\dots \phi_2(\phi_1(p_u, q_i))\dots)\big) \tag{2}$$

where ϕ_{out} and ϕ_x $(x = 1, 2, \dots, X)$ respectively denote the mapping function for the output layer and the x-th NCF layer, and there are X layers in total. Each layer of the multi-layer representations can be customized to discover certain latent structures of interactions. Meanwhile, MF can be interpreted as a special case of NCF framework. Specifically, the mapping function of the first NCF layer as:

$$\phi_1(p_u, q_i) = p_u \odot q_i \tag{3}$$

where $\odot$ denotes the element-wise product of two vector p_u and q_i, and then the results are fed into the output layer.

$$\hat{y}_{ui} = a_{out}(h^{\mathrm{T}}(p_u \odot q_i)) \tag{4}$$

where a_{out} and h represent the activation function and edge weights of the output layer. If a_{out} is set to be an identity function (i.e. $a_{out}(x) = x$) and h to be a uniform vector of 1, the Formula 4 is completely equivalent to Formula 1. If more neural layers are added, NCF has the ability to model any interaction structure, and thus capture complex user-product interaction information. Although NCF has laid a theoretical foundation and shown positive effects of exploiting DNN on MF, the framework only considers the input information of user and product, and focuses on the recommendation task regardless of review text content. However, the text is the most important carrier of sentiments and the most intuitive reflection of user's rating score. Ignoring the valuable text features further limits the usability of NCF on rating prediction task. In summary, NCF can not directly incorporate other important and helpful features, which will lead to the limitations of the framework in practical use.

In this paper, we propose a novel and extensible framework called Deep Learning based Collaborative Filtering (DLCF) as shown in Fig. 3, which is a natural extension of NCF by allowing to add more helpful elements (e.g., reviews). DLCF can be considered as a further generalization of NCF framework. We now formulate the DLCF's predictive model as:

$$\hat{y}_{ui} = f(\overline{v}_0, \dots, \overline{v}_X) = \phi_{out}\big(\phi_X(\overline{v}_X, \dots \phi_2(\overline{v}_2, \phi_1(\overline{v}_1, \overline{v}_0))\dots)\big) \tag{5}$$

where $\overline{v}_X$ denotes the vector representation of any interactive element. such as p_u, q_i and moreover review representation $\overline{v}(r_{ui})$. As an extensible framework, more features could be incorporated into DLCF according to different application

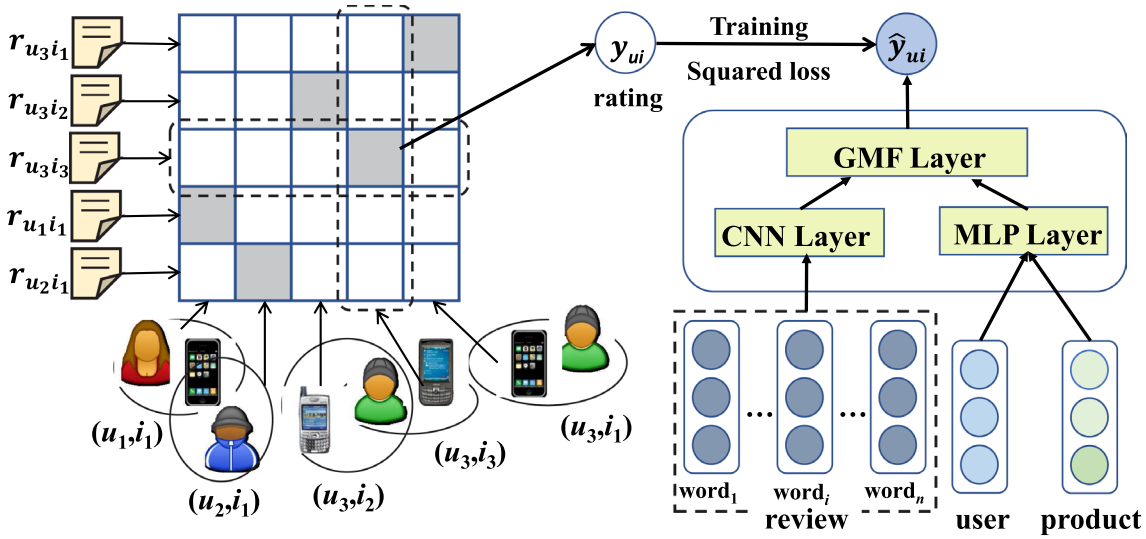

Fig. 4 The overview of InterSentiment model, which is a two-layer instantiation of DLCF framework

scenarios for better performance. This paper focuses on the task of personalized review rating prediction. Thus, we instantiate DLCF as a deep MF model InterSentiment that considers the most relevant information such as user, product and review content simultaneously. Note that NCF and existing non-neural MF variants could not fully capture content information in review r_{ui}, as most of them assume the Bag-of-Words model (if r_{ui} is considered) that ignores context such as surrounding words and word orders [12, 15, 23, 25].

5 InterSentiment

In this section, we propose a novel deep MF model named InterSentiment, as shown in Fig. 4. InterSentiment actually is a two-layer instantiation of DLCF framework, which takes the user vector, product vector and word vector as inputs for predicting the rating score. Comparing InterSentiment with the DLCF framework, v_0 and v_1 in Fig. 3 denote user and product vectors, and v_2 represents the review vector in InterSentiment, i.e., the output of a convolutional neural network Layer. We formulate InterSentiment model as:

$$\hat{y}_{ui} = GMF\big(CNN(r_{ri}), MLP(u,i)\big) \tag{6}$$

where the three major components of InterSentiment includes: a deep user-product interaction model based on Multi-Layer Perceptron (MLP), a deep sentiment model based on Convolutional Neural Network (CNN) and a Generalized Matrix Factorization component (GMF). The role of each component is briefly described as follows.

User–product interaction model This component utilizes MLP to model the complex interactions between users and products, and finally captures the high-level representation of the interaction structure.

Neural sentiment model This component leverages hundreds of convolutional filters to extract text features, so as to make full use of word order and context information for better review representation.

Generalized matrix factorization GMF component is an extension of MF model, which could project the user-product interaction vectors and review vectors into final rating score.

In the following subsections, we will describe the above three components in detail.

5.1 User–product interaction model

We represent the user-product interaction (u, i) as a vector $\bar{v}_{(u,i)}$ using a simple mapping $\bar{v}_{(u,i)} = \phi(p_u, q_i)$, where ϕ can be a vector concatenation operation $\bar{v}_{(u,i)} = [p_u; q_i]$. However, such a simple concatenation ignores the complex interactions between user and product factors. In order to capture sufficient interactions between users and products, we add multiple hidden layers on the concatenated vector, which is formulated into a MLP as below:

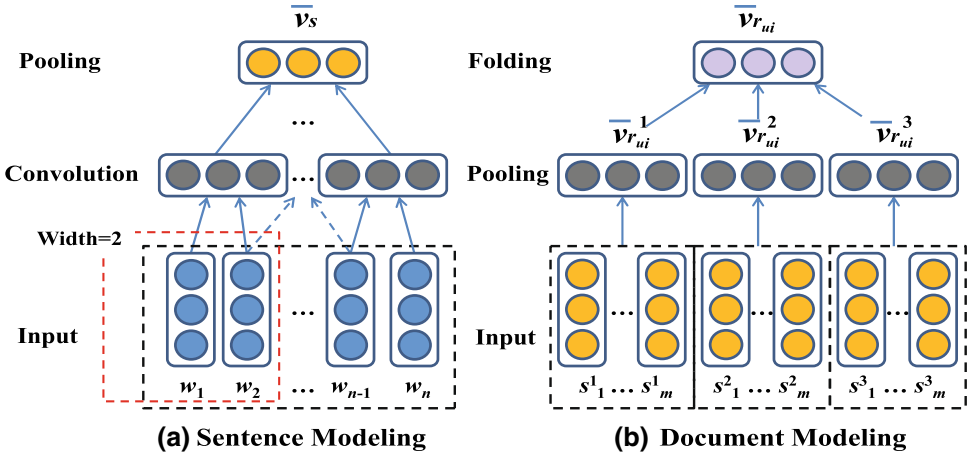

Fig. 5 CNN architecture for sentence and document modeling: **a** Sentence representation based on convolutional filter with Width 2; **b** document representation based on m sentences derived from three filters with Width 1, 2 and 3

$$\begin{aligned}
&\text{MLP Layer } 1: \ \bar{v}^1_{(u,i)} = tanh(W^1 \bar{v}_{(u,i)} + b^1) \\
&\dots \\
&\text{MLP Layer } l: \ \bar{v}^l_{(u,i)} = tanh(W^l \bar{v}^{l-1}_{(u,i)} + b^l) \\
&\dots \\
&\text{MLP Layer } L: \ \bar{v}^L_{(u,i)} = tanh(W^L \bar{v}^{L-1}_{(u,i)} + b^L)
\end{aligned} \tag{7}$$

where $\bar{v}^l_{(u,i)}$, W^l and b^l denote the interaction presentation, weight matrix and bias vector of the l-th layer, and hyperbolic tangent function $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is used as the activation function. To design the network structure, existing approaches generally adopt a tower pattern, where the bottom layer is the widest and each successive layer has a smaller number of neurons. In this work, we empirically halve the layer size for each successive higher layer, that is,

$$2 * size\left(\bar{v}^l_{(u,i)}\right) = size\left(\bar{v}^{l-1}_{(u,i)}\right) \tag{8}$$

where $1 \le l \le L$ and function $size(\cdot)$ returns the vector length. For example, if the output length of the last layer $size\left(\bar{v}^3_{(u,i)}\right) = K$, the architecture is $2^2 * K \to 2^1 * K \to K$. Therefore, the perceptron can endow a large extent of flexibility and non-linearity to learn the latent user-product interactions.

5.2 Neural sentiment model

CNN has been proven powerful for sentiment classification [20, 27] since it can capture local semantics of n-grams of various granularities. Since a review usually consists of a sequence of sentences and each sentence consists of a sequence of words, we produce the representation for each review by two stages:

1. We produce the vector for each sentence in review using word vectors;
2. We compose sentence vectors into a review vector.

The architecture for the CNN we used is displayed in Fig. 5, where three kinds of filters are utilized with 1, 2, 3 window size (Width) respectively. Note that Width = 1, 2, 3 is consistent with the unigram, bigram and trigram feature settings of traditional machine learning models. Figure 5a depicts the sentence representation generation based on convolutional filter with Width 2; Fig. 5b describes the document representation generation based on m sentences derived from three filters with Width 1, 2 and 3. The review text modeling process based on CNN is detailed as follows.

Stage 1 Let us denote a sentence s consisting of n words as $\{w_1, \dots, w_n\}$. For each word w_i, we use a look-up matrix $E \in \mathbb{R}^{d \times V}$ to obtain its word vector $e_{w_i} \in \mathbb{R}^d$, where d and V is the size of the word vector and the vocabulary, respectively. E is typically initialized with pre-trained word embeddings. For any convolutional filter γ with the window Width of ω, we follow the *Lookup-Convolution-Pooling* steps as shown in Fig. 5a to produce the sentence representation $\bar{v}_s$ for sentence s:

$$\begin{aligned}
&\text{Lookup Layer: } e^{w_{i+\omega-1}}_{w_i} = [e_{w_i}; e_{w_{i+1}}; \dots; e_{w_{i+\omega-1}}] \\
&\text{Convolution Layer: } c^{w_{i+\omega-1}}_{w_i} = tanh(W^\gamma e^{w_{i+\omega-1}}_{w_i} + b^\gamma) \\
&\text{Pooling Layer: } \bar{v}_s = [avg(\mathcal{C}_1), \dots, avg(\mathcal{C}_k), \dots, avg(\mathcal{C}_K)]
\end{aligned} \tag{9}$$

where $e \in \mathbb{R}^{d \times \omega}$ is the concatenation of ω successive word representations in the window, $c \in \mathbb{R}^K$ is the convolved feature derived from the filter with weight matrix $W^\gamma \in \mathbb{R}^{K \times (d \times \omega)}$

and bias vector $b^{\gamma} \in \mathbb{R}^{d\times\omega}$; $\mathcal{C}_k \in \mathbb{R}^{n-\omega+1}$ is the k-th row vector of K-by-$(n-\omega+1)$ feature matrix $\mathcal{C} = [c_{w_1}^{w_\omega}; \ldots; c_{w_{n-\omega+1}}^{w_n}]$ which is the concatenation of all the convolved features in the sentence s and will yield $\bar{v}_s$ by applying average-pooling operation $avg(\cdot)$ across all the elements in each row; K is the output length of convolved feature being consistent with the size of user-product interaction representation.

Stage 2 Let r_{ui} be a review consisting of m sentences $\{s_1, \ldots, s_m\}$, we concatenate their sentence representations and produce the input $D \in \mathbb{R}^{K\times m}$ which will be later input into a pooling layer and yield the review representation $\bar{v}_{r_{ui}}^{\gamma}$ with the filter γ (see Fig. 5b). Then we can have the following:

$$\begin{aligned} &\text{Lookup Layer: } D = [\bar{v}_{s_1}; \bar{v}_{s_2}; \ldots; \bar{v}_{s_m}] \\ &\text{Pooling Layer: } \bar{v}_{r_{ui}}^{\gamma} = \left[avg(D_1), \ldots, avg(D_k), \ldots, avg(D_K)\right] \end{aligned} \tag{10}$$

We consider multiple filters with Γ different window widths, then concatenate all review vectors $[\bar{v}_{r_{ui}}^{1}; \ldots \bar{v}_{r_{ui}}^{\gamma} \ldots; \bar{v}_{r_{ui}}^{\Gamma}]$ and construct matrix $\mathcal{D}\simeq \in \mathbb{R}^{K\times\Gamma}$. Afterwards, $\mathcal{D}\simeq$ is input to a folding layer and generates final representation $\bar{v}_{r_{ui}}$:

$$\text{Folding Layer: } \bar{v}_{r_{ui}} = \left[avg(\mathcal{D}\simeq_1), \ldots, avg(\mathcal{D}\simeq_k), \ldots, avg(\mathcal{D}\simeq_K)\right] \tag{11}$$

where the function $avg(\cdot)$ returns the average value of the vector in $\mathcal{D}\simeq$ and the final review representation $\bar{v}_{r_{ui}} \in \mathbb{R}^K$.

5.3 Generalized matrix factorization

To combine the two deep models above, we construct a sparse (user, product)-review rating matrix $\mathbf{Y} : |\mathcal{U}\times\mathcal{I}|\times|\mathcal{R}|$ (see Fig. 4). Each observed entry will be approximated by our InterSentiment model which bridges the user-product interaction model and neural sentiment model based on an instantiation of DLCF. DLCF is instantiated by taking three original inputs u, i and r_{ui} and treating the above two deep models as core interactive components. For each observed rating y_{uir} in $\mathbf{Y}$, it can be approximated by $\hat{y}_{uir}$ as below:

$$\hat{y}_{uir} = GMF\left(CNN(r_{ui}), MLP(u,i)\right) \tag{12}$$

where $CNN(\cdot)$ and $MLP(\cdot)$ are the models introduced in previous sections, and the generalized matrix factorization GMF is specifically instantiated as follows:

$$\hat{y}_{uir} = \varphi\left(h^{\mathrm{T}}\left(\phi_{1:X}\left(\bar{v}_{r_{ui}} \odot \bar{v}_{(u,i)}\right)\right)\right) \tag{13}$$

where $\odot$ is the elementwise product of two vectors, $\bar{v}_{(u,i)} \in \mathbb{R}^K$ and $\bar{v}_{r_{ui}} \in \mathbb{R}^K$ denote the representation of interaction (u, i) learned from MLP and that of review content r_{ui} learned from CNN, respectively; $\phi_{1:X}$ denotes a $X-$layer perceptron to learn high-level representation, weight vector $h \in \mathbb{R}^{\frac{K}{2^X}}$ assigns varying importance to different dimensions which is to be learned by the model. Moreover, we define the activation function φ as:

$$\varphi(x) = 1 + \frac{\zeta - 1}{1 + e^{-x}} \tag{14}$$

$\varphi(x)$ generates model outputs in the range of valid rating values $[1, \zeta]$, where ζ is the highest rating level. The formulation ensures that, if $x \to +\infty$ (or $x \to -\infty$), $\varphi(x) = \zeta$ (or 1).

5.4 Model training

We define our objective function G that minimizes the sum of squared errors over training set $\mathcal{T}$ with a regularization term as below:

$$G = \frac{1}{2}\sum_{y_{ui}\in\mathcal{T}}(y_{uir} - \hat{y}_{uir})^2 + \frac{\lambda}{2}\| \Theta \|^2 \tag{15}$$

where Θ is the set of parameters to be estimated, which includes $\{p_u\}, \{q_i\}, \{W^l, b^l | 1 \le l \le L\}, \{W^{\gamma}, b^{\gamma} | 1 \le \gamma \le \Gamma\}$, $\{W^x, b^x | 1 \le x \le X\}$, E and h. $\{p_u\}$ and $\{q_i\}$ represent the vector set of products and users. λ is a coefficient that controls the weight of regularization.

We take the derivative of G through back-prorogation with respect to Θ, and update parameters with stochastic gradient descent. According to DLCF framework Formula 5, if the parameters of layer x are represented by Θ^x, then we can infer the partial derivative by chain rules as:

$$\frac{\partial G}{\partial \Theta^x} = \frac{\partial G}{\partial \phi^x}\frac{\partial \phi^x}{\partial \Theta^x} \tag{16}$$

$$\frac{\partial G}{\partial \phi^{x-1}} = \frac{\partial G}{\partial \phi^x}\frac{\partial \phi^x}{\partial \phi^{x-1}} \tag{17}$$

The optimization of the objective function is started by calculating the partial derivative $\frac{\partial G}{\partial \phi^x}$ of the last layer, and thus $\frac{\partial \phi^x}{\partial \Theta^x}\cdot\frac{\partial G}{\partial \phi^x}$ is learned by Formula 16 and the partial derivatives are calculated in a layer-wise way until the input layer. The procedure of parameter learning is shown in Algorithm 1.

Table 1 Statistics of experimental datasets we used

Dataset	Scale	#users	#items	#reviews	Length (avg)	#reviews/#users
IMDB	1~10	1310	1635	84,919	394.6 (word)	64.82
Yelp 2014	1~5	4818	4194	231,163	196.9 (word)	47.97
Yelp 2013	1~5	1631	1633	78,966	189.3 (word)	48.42

Algorithm 1 InterSentiment Parameter Learning Algorithm.

Require:
Training dataset T; Learning rate η; Iteration number N;
Ensure:
Model parameter set $\Theta = \{\{p_u\}, \{q_i\}, \{W^l, b^l | 1 \leq l \leq L\}, \{W^\gamma, b^\gamma | 1 \leq \gamma \leq \Gamma\}, \{W^x, b^x | 1 \leq x \leq X\}, E, h\}$;

for each $i \in [1, N]$ **do**
  shuffle the training set T;
  for each review $r_{ui} \in T$ **do**
    Compute the gradients for related parameters by backpropagation
    for each word $w \in W(r_{ui})$ **do**
      $e_w \leftarrow e_w - \eta(\partial G / \partial e_w)$ // update word vector
    end for
    $h \leftarrow h - \eta(\partial G / \partial h)$ // update parameter h
    for each layer $l \in [1, L]$ **do**
      $W^l \leftarrow W^l - \eta(\partial G / \partial W^l)$ // update parameter matrix for MLP layer
      $b^l \leftarrow b^l - \eta(\partial G / \partial b^l)$ // update bias for each MLP layer
    end for
    for each $\gamma \in [1, \Gamma]$ **do**
      $W^\gamma \leftarrow W^\gamma - \eta(\partial G / \partial W^\gamma)$ // update parameter matrix for filters
      $b^\gamma \leftarrow b^\gamma - \eta(\partial G / \partial b^\gamma)$ // update bias for filters
    end for
    $p_u \leftarrow p_u - \eta(\partial G / \partial p_u)$ // update user feature vector
    $q_i \leftarrow q_i - \eta(\partial G / \partial q_i)$ // update product feature vector
  end for
  $\eta \leftarrow \eta \times 0.9$
end for

The word vectors are initialized by Sentiment-Specific Word Embeddings (SSWE) [26] pre-trained from our training sets. We follow Glorot and Bengio [11] to initialize other parameters with uniform distribution, namely sampling from $(-\sqrt{6/(r+c)}, \sqrt{6/(r+c)})$, where r and c are the numbers of rows and columns of the matrices or vectors.

6 Experiment

In this section, we evaluate the proposed review rating prediction methods by three publicly available review rating datasets. The effectiveness of the proposed methods is validated by comparing with the strong baselines of Sentiment Analysis models (SA) and Recommendation Systems (RS). The details of comparison methods are shown in Sect. 6.2.1. Moreover, we will analyze the impact of the sparsity of rating matrix on sentiment prediction problem and discuss the preliminary experiment results for the cold-start users and products.

6.1 Experimental settings

Datasets We conduct experiments on three public datasets[3]: IMDB, Yelp 2013 and Yelp 2014, which are built by Tang et al. [27]. The datasets are tokenized and splitted into training, development and test sets with a 80/10/10 split. The statistics of the datasets are shown in Table 1.

Table 2 Settings of MLP layer L on our datasets

Dataset	MLP-0	MLP-1	MLP-2	MLP-3
IMDB	1.436	1.428	**1.425**	1.435
Yelp 2014	0.712	0.684	0.682	**0.680**
Yelp 2013	0.706	0.695	0.694	**0.685**

Values in bold indicate the best performance in the corresponding category

[3] http://ir.hit.edu.cn/~dytang/paper/acl2015/dataset.7z.

Table 3 Comparison of RMSE scores among different SC-based methods

Dataset	Ngram°	AvgVec°	SSWE°	ParVec°	UPNN°	Ours
IMDB	1.783	1.985	1.973	1.814	1.602	**1.418**
Yelp 2014	0.804	0.893	0.851	0.802	0.764	**0.660**
Yelp 2013	0.814	0.898	0.849	0.832	0.784	**0.673**

Values in bold indicate the best performance in the corresponding category

Table 4 Comparison of RMSE scores among different CF-based methods

Dataset	MF	JMARS°	TFM*	PSC	TLFM*	Ours
IMDB	1.995	1.773	1.598	1.502	1.495	**1.418**
Yelp 2014	1.020	0.999	0.835	0.813	0.712	**0.660**
Yelp 2013	0.987	0.985	0.836	0.808	0.716	**0.673**

Values in bold indicate the best performance in the corresponding category

Evaluation metric We evaluate the quality of prediction results by Root Mean Squared Error:

$$RMSE = \sqrt{\sum_{y_{uir} \in \mathcal{T}'} (y_{uir} - \hat{y}_{uir})^2 / |\mathcal{T}'|} \tag{18}$$

where $\mathcal{T}'$ is the test set.

Parameter settings We use three convolutional filters with different window sizes $\omega = \{1, 2, 3\}$ to encode the semantics of unigrams, bigrams and trigrams. This window size setting also follows UPNN model's setting [27], which is one of our strong baselines. We set the learning rate as 0.05, λ as 0.001 and d as 200 according to our observation on the best RMSE scores on the development set. We optimize the output length K on the development set by searching on all values of $\{4, 8, 16, 32, 64\}$. It is found that the performance becomes stable when $K \geq 16$, so we fix $K = 16$.

We optimize the MLP layer number L and X by searching all values of $\{0, 1, 2, 3\}$, and the results are shown in Table 2. It can be seen that better performance is achieved with deeper MLP layers. This is because InterSentiment model can capture the impact of complex user-product interactions when MLP component has deeper structure. In the following experiments, we fix $L = 3$ because of faster convergence and better performance. Our models converge in the first 10 iterations with the drop of RMSE on the development set.

Experiment environment The experiments are conducted on a commercial PC with Intel Core i7-6700 CPU and 16G RAM. Our method is implemented using the Java programming language and has been made publicly available.[4]

6.2 Performance comparison

In this subsection, we first compare the proposed model with strong baseline methods and further conduct the ablation experiments. The impact of data sparsity and the problems of cold-start users and products are also discussed.

6.2.1 Comparison of different approaches

We compare our InterSentiment model with some traditional and advanced baselines, and display the results in Tables 3 and 4. The results with superscript ∘ and ∗ are reported in [27] and [25], respectively.

We first compare to SC-based methods:

- *Ngram* is a support vector machine (SVM) classifier which is trained on unigram, bigram and trigram features [8];
- *AvgVec* averages word embeddings learned from training and development sets with *word2vec* [16] as document representation, and then trains a SVM classifier;
- *SSWE* is similar to AvgVec but applies sentiment-specific word embeddings [26];
- *ParVec* is a SVM classifier trained on paragraph representations of documents.
- *UPNN* is a neural network model which modifies word embeddings in the input layer with user/product preference matrix, and then concatenates user/product vector with generated review representation via softmax layer [27].

We also compared to some CF-based methods:

- *MF* is the basic matrix factorization [21];
- *JMARS* is a probabilistic model based on collaborative filtering and topic modeling, which considers user and aspects of a review [4];
- *TFM* is a linear model by combining the entries estimated based on a user-product-word tensor factorization model [15];
- *PSC* extends MF by constructing a user-text matrix, which considers sentiment and topic units in subjective text [23];

[4] https://github.com/source-code-doc/InterSentiment.

Table 5 Comparison of RMSE scores among different configurations

	Method	Full	$-\mathcal{U}$	$-\mathcal{I}$	$-\mathcal{UI}$	$-\mathcal{R}$
IMDB	Ours	**1.418**	**1.533**	**1.438**	**1.548**	**1.751**
	UPNN°	1.602	1.743	1.712	1.629	N/A
	TLFM*	1.495	1.613	1.521	N/A	1.959
Yelp 2014	Ours	**0.660**	**0.702**	**0.692**	**0.711**	**0.959**
	UPNN°	0.764	0.778	0.776	0.808	N/A
	TLFM*	0.712	0.745	0.740	N/A	0.998
Yelp 2013	Ours	**0.673**	**0.710**	**0.700**	**0.719**	**0.950**
	UPNN°	0.784	0.828	0.802	0.812	N/A
	TLFM*	0.716	0.781	0.762	N/A	0.981

Values in bold indicate the best performance in the corresponding category
'N/A' indicates the model is not applicable in the case

Table 6 Comparison among different matrix densities

Dataset	$n = 25$		$n = 50$		$n = 75$		$n = 100$	
	$\rho\%$	RMSE	$\rho\%$	RMSE	$\rho\%$	RMSE	$\rho\%$	RMSE
IMDB	14.3	**1.324**	7.31	1.345	4.55	1.405	3.14	1.425
Yelp 2014	3.95	**0.655**	2.02	0.664	1.28	0.674	0.90	0.680
Yelp 2013	9.66	**0.648**	5.08	0.666	3.28	0.676	2.34	0.685

Values in bold indicate the best performance in the corresponding category

- *TLFM* is a variant of latent factor model that captures the review text based on Bag-of-Words [25].

As can be seen from Table 3, Ngram is shown in most cases more powerful than SSWE, AvgVec and ParVec that are trained on word embeddings, which indicates that handcrafted features are more effective. However, all these methods still cannot compete with UPNN which additionally considers user and product information. InterSentiment outperforms UPNN by 11.4%, 13.6% and 14.1% on IMDB, Yelp 2014 and Yelp 2013 datasets, respectively.[5] This indicates the effectiveness of InterSentiment on learning the representations of both user-product interactions and review text, and jointly modeling based on DNN.

As shown in Table 4, MF that largely considers text features as unnecessary or auxiliary information cannot beat TFM, PSC and TLFM that focus on detecting sentiment from subjective text. Our InterSentiment outperforms TLFM by 5.2%, 7.3% and 6% on IMDB, Yelp 2014 and Yelp 2013 datasets, respectively,[6] which verifies that InterSentiment can not only better utilize the user-product interaction representations based on MLP, but also better learn text features via deep learning based on CNN.

[5] The improvement is calculated by $|RMSE(\text{Ours}) - RMSE(\text{UPNN})| / RMSE(\text{UPNN})$.

[6] The improvement is calculated by $|RMSE(\text{Ours}) - RMSE(\text{TLFM})| / RMSE(\text{TLFM})$.

Table 7 Prediction of AvgUI and UnkUI on masked test sets

Dataset	Full–$\mathcal{UI}$	AvgUI	UnkUI
IMDB	1.560	**1.454**	1.467
Yelp 2014	0.734	0.705	**0.700**
Yelp 2013	0.755	**0.704**	0.707

Values in bold indicate the best performance in the corresponding category

6.2.2 Ablation experiments

We compare the performance of five different configurations among UPNN, TLFM and InterSentiment. Each configuration considers only part of useful features, which includes:

- *Full* is the fully configured model;
- $-\mathcal{U}$ represents the models which ignore user information;
- $-\mathcal{I}$ ignores product information;
- $-\mathcal{UI}$ only considers review text;
- $-\mathcal{R}$ only models user-product interactions.

The ablation experiment results with different configurations are shown in Table 5.

It is clear that any partial configuration cannot compete with the full ones indicating that user, product and text information should be considered together. $-\mathcal{UI}$ is better than $-\mathcal{R}$, which suggests that the content of reviews can reflect sentiments more accurately. Besides, $-\mathcal{I}$ performs better than $-\mathcal{U}$ indicating that user information is more helpful.

InterSentiment in full configuration outperforms both UPNN and TLFM indicating the effectiveness of our model on capturing user-product interactions as well as text semantics.

6.2.3 Discussion about matrix density

Because the density of user-product rating matrix may have an impact on the overall RMSE on sentiment analysis task, we further study the influence of matrix density, which is defined as:

$$\rho = \frac{|\mathcal{R}|}{(|\mathcal{U}| \times |\mathcal{I}|)} \tag{19}$$

where $|\mathcal{R}|, |\mathcal{U}|, |\mathcal{I}|$ denote the number of reviews, users and products respectively. The popularity of a user or a product is usually determined by the frequency of the user's or product's presence in the training set. We select the top $n\%$ users and products that have the most training instances as a group, and calculate RMSE with respect to ρ, where the grouped results are displayed in Table 6.

In Table 6, n = 25 means that we select the top 25% users and products with the most training instances for the experiments. We can find that the group with higher ρ will have lower RMSE since user/product factors are estimated more accurately. This is because the active users or products usually have more training instances so as to generate better feature representations. Therefore, we can infer that considering personalized information for active users or products will usually achieve better prediction performance.

6.2.4 Discussion about cold-start problem

Despite of InterSentiment's promising results, the rating prediction methods that consider personalized information depend on adequate training data. However, in practical applications, there are unobserved users or rarely rated products in the test set. The lack of ratings from these users\products may weaken the parameter estimation. Technically, this is referred to cold-start problem [31, 35].

For dealing with unseen users or unseen products, inspired by [27] we adopt two solutions called AvgUI and UnkUI [25]. The AvgUI averages over all the observed p_u or q_i in the training set as the representations of new users\products. The UnkUI method learns a shared "unknown" representation for new users\products by randomly drawing 200 reviews as their alternative training instances.

To evaluate the proposed method, we first randomly select 10% users and products from test set, replace their names with unseen names so as to obtain a new cold-start test set. The final experiment results are shown in Table 7.

In Table 7, AvgUI and UnkUI obviously outperform Full–$\mathcal{UI}$ that only considers text features. These two methods utilize simple and straightforward solutions to the cold-start problem. However, compared with the results of full datasets in Table 3, AvgUI and UnkUI are slightly worse than InterSentiment, but dramatically outperform UPNN. These experiment results validate the effectiveness of AvgUI and UnkUI in alleviating the cold-start problem.

7 Conclusions and future work

In this work, we present a novel review rating prediction model called InterSentiment by bridging a user–product interaction and a sentiment model based on deep neural networks. The experimental results on IMDB and two Yelp datasets show that InterSentiment outperforms strong baselines with clear margin. In the future, we will further study how to combine the interactions and semantics more closely in deep neural models and apply our method to different recommendation tasks.

Acknowledgements The work was supported by the National Key R&D Program of China under Grant 2018YFB1004700, the National Natural Science Foundation of China (61872074, 61772122), and the Fundamental Research Funds for the Central Universities (N180716010).

References

1. Amplayo RK, Kim J, Sung S, Hwang S (2018) Cold-start aware user and product attention for sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, vol 1: Long Papers, pp 2535–2544
2. Bell RM, Koren Y (2007) Lessons from the netflix prize challenge. SIGKDD Explorr 9(2):75–79
3. Cai Y, Yang K, Huang D, Zhou Z, Lei X, Xie H, Wong T (2019) A hybrid model for opinion mining based on domain sentiment dictionary. Int. J. Mach. Learn. Cybernet. 10(8):2131–2142
4. Diao Q, Qiu M, Wu C, Smola AJ, Jiang J, Wang C (2014) Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: SIGKDD, pp 193–202
5. Ding X, Liu B, Yu PS (2008) A holistic lexicon-based approach to opinion mining. In: WSDM, pp 231–240
6. Dou Z (2017) Capturing user and product information for document level sentiment analysis with deep memory network. In: Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017, pp 521–526
7. Dridi A, Recupero DR (2019) Leveraging semantics for sentiment polarity detection in social media. Int. J. Mach. Learn. Cybernet. 10(8):2045–2055
8. Fan R, Chang K, Hsieh C, Wang X, Lin C (2008) LIBLINEAR: a library for large linear classification. JMLR 9:1871–1874
9. Feng S, Wang D, Yu G, Gao W, Wong K (2011) Extracting common emotions from blogs based on fine-grained sentiment clustering. Knowl Inf Syst 27(2):281–302
10. Firdaus SN, Ding C, Sadeghian A (2019) Topic specific emotion detection for retweet prediction. Int. J. Mach. Learn. Cybernet. 10(8):2071–2083

11. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: AISTATS, pp 249–256
12. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: WWW
13. Kim DH, Park C, Oh J, Lee S, Yu H (2016) Convolutional matrix factorization for document context-aware recommendation. In: RecSys, pp 233–240
14. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: ICDM, pp 426–434
15. Li F, Liu NN, Jin H, Zhao K, Yang Q, Zhu X (2011) Incorporating reviewer and product information for review rating prediction. In: IJCAI, pp 1820–1825
16. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS, pp 3111–3119
17. Mukherjee S, Basu G, Joshi S (2013) Incorporating author preference in sentiment rating prediction of reviews. In: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13–17, 2013, Companion Volume, pp 47–48
18. Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL, pp 115–124
19. Rendle S (2010) Factorization machines. In: ICDM 2010, The 10th IEEE international conference on data mining, 2010, pp 995–1000
20. Ren Y, Zhang Y, Zhang M, Ji D (2016) Context-sensitive twitter sentiment classification using neural network. In: AAAI, pp 215–221
21. Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization. In: NIPS, pp 1257–1264
22. Shen R, Zhang H, Yu H, Min F (2019) Sentiment based matrix factorization with reliability for recommendation. Expert Syst Appl 135:249–258
23. Song K, Feng S, Gao W, Wang D, Yu G, Wong K (2015) Personalized sentiment classification based on latent individuality of microblog users. In: IJCAI, pp 2277–2283
24. Song K, Chen L, Gao W, Feng S, Wang D, Zhang C (2016) Persentiment: A personalized sentiment classification system for microblog users. In: Proceedings of the 25th international conference on World Wide Web, WWW 2016, pp 255–258
25. Song K, Gao W, Feng S, Wang D, Wong K, Zhang C (2017) Recommendation vs sentiment analysis: a text-driven latent factor model for rating prediction with cold-start awareness. In: IJCAI, pp 2744–2750
26. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: ACL, pp 1555–1565
27. Tang D, Qin B, Liu T (2015) Learning semantic representations of users and products for document level sentiment classification. In: ACL, pp 1014–1023
28. Tang D, Qin B, Liu T, Yang Y (2015) User modeling with neural network for review rating prediction. In: IJCAI, pp 1340–1346
29. Wang H, Wang N, Yeung D (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, 2015, pp 1235–1244
30. Wu F, Huang Y (2016) Personalized microblog sentiment classification via multi-task learning. In: Proceedings of the thirtieth AAAI conference on artificial intelligence, pp 3059–3065
31. Xu J, Yao Y, Tong H, Tao X, Lu J (2015) Ice-breaking: Mitigating cold-start recommendation problem by rating comparison. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI 2015, pp 3981–3987
32. Yuan Y, Luo X, Shang M (2018) Effects of preprocessing and training biases in latent factor models for recommender systems. Neurocomputing 275:2019–2030
33. Yue L, Chen W, Li X, Zuo W, Yin M (2019) A survey of sentiment analysis in social media. Knowl Inf Syst 60(2):617–663
34. Zhang Y, Lai G, Zhang M, Zhang Y, Liu Y, Ma S (2014) Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: SIGIR, pp 83–92
35. Zhang M, Tang J, Zhang X, Xue X (2014) Addressing cold start in recommender systems: a semi-supervised co-training algorithm. In: Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval, SIGIR '14, pp 73–82
36. Zhang J, Chow C, Xu J (2017) Enabling kernel-based attribute-aware matrix factorization for rating prediction. IEEE Trans Knowl Data Eng 29(4):798–812
37. Zhao J, Dong L, Wu J, Xu K (2012) Moodlens: an emoticon-based sentiment analysis system for Chinese tweets. In: The 18th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '12, pp 1528–1531