Review Polarity-Wise Recommender

Han Liu, Yangyang Guo, Member, IEEE, Jianhua Yin, Member, IEEE, Zan Gao, Member, IEEE, and Liqiang Nie, Senior Member, IEEE

Abstract—The de facto review-involved recommender systems, using review information to enhance recommendation, have received increasing interest over the past years. Thereinto, one advanced branch is to extract salient aspects from textual reviews (i.e., the item attributes that users express) and combine them with the matrix factorization (MF) technique. However, the existing approaches all ignore the fact that semi-automatically different reviews often include opposite aspect information. In particular, positive reviews usually express aspects that users prefer, while the negative ones describe aspects that users dislike. As a result, it may mislead the recommender systems into making incorrect decisions pertaining to user preference modeling. Toward this end, in this article, we present a review polarity-wise recommender model, dubbed as RPR, to discriminately treat reviews with different polarities. To be specific, in this model, positive and negative reviews are separately gathered and used to model the user-preferred and user-rejected aspects, respectively. Besides, to overcome the imbalance of semantically different reviews, we further develop an aspect-aware importance weighting strategy to align the aspect importance for these two kinds of reviews. Extensive experiments conducted on eight benchmark datasets have demonstrated the superiority of our model when compared with several state-of-the-art review-involved baselines. Moreover, our method can provide certain explanations to real-world rating prediction scenarios.

Index Terms—Aspect-aware recommendation, review imbalance problem, review polarity, review-involved recommendation.

I. INTRODUCTION

NOWADAYS, posting reviews on e-commerce platforms has become ubiquitous among online shoppers to share their purchasing experiences. These textual reviews usually contain rich semantic information about user preferences and item attributes, thereby playing an increasingly important role in recommender systems [1]. One typical benefit is that reviews enable the machine to effectively exploit more side information and receive superior performance when compared with canonical matrix-factorization (MF)-based methods [2], as the latter methods use only the sparse rating matrix.

Previous studies on review-involved recommendation mostly adopt a standard scheme: the user and item documents are first constructed by merging the associated reviews (i.e., reviews of the user and reviews for the item), wherein each textual token is vectorized via word embedding methods [3], [4]. The two types of documents are, respectively, processed via convolutional neural networks (CNNs) to generate user and item representations, followed by a matching function (e.g., dot product and factorization machines (FM) [5]) to predict the final rating score. Based on this scheme, methods such as DeepCoNN [6], TransNets [7], D-Attn [8], and multi-pointer co-attention network (MPCN) [9] have achieved some improvements over other baselines. Distinct from these approaches, recent efforts have been dedicated to review aspect modeling [10]–[13]. Foremost, the aspect is defined as follows: Aspect—It is embodied with high-level semantics, representing the attributes of items that users comment on in their reviews [12]. For example, in the review “Excellent, pretty useful, easy to use and reliable. These Airpods work well and are comfortable to my ears,” the user mentioned the aspects easy to use, reliable, and comfortable of item Airpods (as shown in Fig. 1).

In general, these high-level aspect features are first extracted through well-developed tools, such as topic modeling [12], which are then integrated with MF backbones [2].

Despite their notable progress, one issue that hurts the performance of the existing review-involved methods is that the review polarities are not explicitly discriminated, i.e., all reviews are taken as positive feedback. In fact, users tend to convey their sentiments in reviews, i.e., higher rating scores often go with positive reviews, while negative ones meet lower scores frequently [14]–[16]. Moreover, reviews with different polarities usually contain opposite aspect information. Fig. 1 shows two opposite polarities of reviews for Bluetooth headsets. The left one implies a positive review, describing aspects of an item that the user prefers, e.g., easy to use. On the contrary, the right one is a negative review that reveals unsatisfactory aspects of an item the user dislikes, e.g., unstable. As illustrated in Fig. 2, negative reviews are quite common in the existing datasets.1 2 Simply integrating these two kinds of reviews as positive will mislead the models to recommend items similar to the ones with negative reviews in the future. It thus results in suboptimal performance and deteriorates the

Manuscript received 14 June 2021; revised 19 October 2021 and 20 January 2022; accepted 25 March 2022. Date of publication 15 April 2022; date of current version 1 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U1936203 and Grant 62172261, in part by the Key Research and Development Program of Shandong (Major Scientific and Technological Innovation Projects) under Grant 2020CSGC010111, in part by the Young Creative Team in Universities of Shandong Province under Grant 2020KJN012, and in part by the Shandong Provincial Natural Science Foundation under Grant ZR2019QF001.

Han Liu, Yangyang Guo, Jianhua Yin, and Liqiang Nie are with the School of Computer Science and Technology, Shandong University, Qingdao 266200, China (e-mail: hanliu.sdu@gmail.com; guoyang.eric@gmail.com; jhyin@sdu.edu.cn; nieliqiang@gmail.com).

Zan Gao is with the Shandong Artificial Intelligence Institute, Jinan 300384, China (e-mail: zangoashn4522@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3163789.

Digital Object Identifier 10.1109/TNNLS.2022.3163789.

1http://jmcauley.ucsd.edu/data/amazon/

2https://www.yelp.com/dataset/challenge
Fig. 1. Example of the opposite aspect information expressed in semantically different reviews.

Fig. 2. Imbalance illustration of the two-polarity reviews. The left subfigure shows the ratios of the two kinds of reviews in eight datasets. The right one shows various groups of users in Yelp. Group1 and Group2, respectively, denote the numbers of users with positive and negative review ratios over 90%, and Group3 implies the remaining users.

user experiences and faithfulness to the platform. In the light of this, toward making the recommendation more personalized and convincing, we aim to distinguish the user-preferred aspects from the user-rejected ones via explicitly performing polarity discrimination. Nevertheless, discriminatingly treating positive and negative reviews is nontrivial in recommendation, due to the following facts: 1) It is difficult to effectively exploit the semantic information of each review word for aspect modeling. 2) How to accurately model the user preferences on both user-preferred and user-rejected aspects poses another challenge for us. 3) There exists severe imbalance regarding different review polarities in current e-commerce datasets, as shown in Fig. 2. For instance, some users tend to post much more positive reviews than negative ones. It therefore exerts adverse impact on the extraction of aspect information, degrading the recommendation performance.

To tackle the aforementioned issues, we present a review polarity-wise recommender model, RPR for short, as shown in Fig. 3, to perceive the review polarity toward review-involved recommendation. In particular, for user $u$ and item $i$, we first leverage their latent factor embeddings to estimate two relevance score vectors for both the user-preferred and user-rejected aspects, whereby each element expresses the preference degree on the corresponding aspect. Meanwhile, traditional topic modeling has shown certain limitations in leveraging the abundant semantic information. We thus turn to a TextCNN in parallel on the review sentences. This module is expected to extract aspect importance and provide some explicit interpretations to aspect modeling. Finally, the overall rating $r_{u,i}$ is estimated by subtracting the inner product of the score and importance vectors on user-rejected aspects from the ones on the user-preferred aspects. Based on this, positive and negative user preferences are seamlessly integrated to implement our RPR.

Besides, to overcome the imbalance between positive and negative reviews, we introduce an aspect-aware importance weighting module to capture the mapping relationship between the user-preferred and user-rejected aspect importance. The assumption is that if users focus on certain aspects they prefer, they will consistently pay roughly equal attention to the corresponding user-rejected ones, and vice versa. In view of this, according to the correspondence between user-preferred and user-rejected aspects, RPR constructs a user-rejected aspect importance offset from the user-preferred aspect importance, which is further added to the original user-rejected one. In this way, the two types of reviews mutually enhance each other for obtaining the aspect importance.

Overall, the main contributions of this article are summarized in threefold:

1) We propose a novel recommendation method to extract the semantic information of user-preferred and user-rejected aspects from positive and negative reviews, respectively. To the best of our knowledge, this work is among the first efforts to treat reviews with different polarities discriminately in review-involved recommendation.

2) We devise an aspect-aware importance weighting component to construct semantic mappings between user-preferred and user-rejected aspects. This design has been proven to be quite effective in solving the problem of data imbalance between positive and negative reviews.

3) We conduct comprehensive experiments on eight benchmark datasets to evaluate the effectiveness of the proposed model. Extensive results demonstrate the state-of-the-art performance of RPR. As a side contribution, we have released the codes, data, and parameters to facilitate researchers in this field.

The rest of this article is organized as follows. Section II briefly reviews representative literature from two directions that are highly relevant to our work. Section III outlines RPR and its architecture and describes how to optimize RPR. In Section IV and V, we experimentally evaluate RPR and analyze the evaluation results, respectively. We summarize the contributions and figure out the potential future research directions in Section VI.

---

3In the RPR model, the aspects are implicit, and the number of aspects is fixed for all the users.

4https://github.com/hanliu95/RPR
The user-preferred aspects is calculated as the positive score of user-preferred and user-rejected aspects for user \( u \) toward item \( i \) are separately learned. The inner product of the score and importance vectors on the user-preferred aspects is calculated as the positive score of user \( u \) toward item \( i \), and then the negative score is similarly computed on user-rejected aspects. RPR finally uses the subtraction of the negative score from the positive one to estimate the final rating.

II. RELATED WORK

A. Review-Involved Recommendation

Recently, exploiting reviews to enhance the recommendation performance and interpretability has been extensively studied in literature [8], [17]. Along this line, the existing methods can be broadly classified into two categories. The first category mainly focuses on the user and item modeling from their separately corresponding documents. To be more specific, user and item documents are first constructed by concatenating the user-posted and item-received reviews, respectively. The word embedding techniques are then adopted to embed the textual document into a semantic matrix. Normally, a CNN [18] is used to extract user and item representations from the matrices. Thereafter, a matching function (e.g., dot product, FMs) can be used to model the user–item interactions. Methods such as DeepCoNN [6], TransNets [7], D-Attn [8], and MPCN [9] all adopt the above scheme, obtaining superior performance compared with the prior MF ones.

The second category aims to effectively learn the aspects from reviews for recommendation, namely, the aspect-aware recommender systems [13], [19], [20]. These methods can be further summarized into the following two groups. The first group tries to extract aspects based on the existing natural language processing (NLP) tools for sentiment analysis [20]–[22]. The obtained aspect representation is then incorporated into an MF framework for more accurate recommendation. For example, explicit factor model (EFM) [20] and multitask explainable recommendation (MTER) [11] resort to a phrase-level NLP tool for aspect-level sentiment extraction. The second group devises specific internal components, e.g., topic modeling, to automatically learn explainable aspect representations of users and items from reviews [12], [13], [23]. In particular, these internal components are leveraged to achieve aspect-aware extraction of the semantic information in reviews. In a nutshell, compared with the first category, the aspect-aware methods are capable of extracting high-level semantic features from reviews, delivering improved results and interpretability.

B. Deep Learning in Recommendation

A prevailing trend in recent years has been leveraging deep learning for recommendation. It is now extensively recognized that deep learning techniques are capable of modeling complex and nonlinear user–item interactions [24], [25]. Generally speaking, the success of deep learning in recommendation mainly comes from two aspects: representation learning and matching function modeling [26]–[28]. Regarding representation learning, entity embeddings, i.e., users and items, can be greatly enhanced with advanced tools in deep learning. For example, CNNs are used to enrich item representation learning from both texts and images [6], and recurrent neural networks (RNNs) show considerable advantage in session-based recommendation [29]. For the other aspect of matching function modeling, deep learning methods often use multilevel neural networks as the interaction function to effectively aggregate low-level signals. Among these methods, multilayer perceptron (MLP) is distinguished with its strong capability to learn both the second-order and higher order feature interactions [30].

Notably, the attention mechanism [31]–[33] has been extensively integrated into recommendation, which demonstrates promising results and great potentials in the existing studies [34], [35]. For example, attentive collaborative filtering (ACF) [36] introduces a hybrid item- and component-level attention model. Meanwhile, NAIS [37] presents a neural attentive item similarity model for item-based collaborative filtering, enabling itself to identify the more important historical items in a user profile for rating prediction. In addition, attentional factorization machine (AFM) [38] learns the weights of feature interactions in FMs via neural attention.
networks. A3NCF [39] introduces a topic model-based attention method, where the attention module is used to capture attentive user preferences on each aspect of the target item.

### III. PROPOSED METHOD

#### A. Preliminaries

In this section, we first briefly present the general framework for the aspect-aware model which exploits reviews to predict user–item interaction ratings and point out its limitation caused by ignoring the review polarity. We then recapitulate how our RPR model can overcome the issue step-by-step.

The aspect-aware recommender systems assume that a user–item rating \( r_{u,i} \) depends on user \( u \)'s score toward item \( i \) on each aspect \( a \) (i.e., aspect score \( s_{n,a,i} \) and the importance of each aspect to \( u \) (i.e., aspect importance \( \rho_{a,u} \)). It is worth noting that the aspects are implicitly defined throughout this article following [12], [23]. In general, the overall rating \( r_{u,i} \) can be predicted by

\[
\hat{r}_{u,i} = \rho_u^T s_{u,i},
\]

where the aspect importance \( \rho_u \in \mathbb{R}^{\lvert A \rvert} \) is estimated based on user reviews (\( A \) denotes the set of aspects, e.g., \{easy to use, reliable, comfortable\} for Airpods), and the aspect score \( s_{u,i} \in \mathbb{R}^{\lvert A \rvert} \) is computed through MF relying on user–item interactions. However, the expression capability of these models is largely limited, since they treat all aspects as user-preferred and ignore the fact that reviews can contain negative opinions. This problem may lead to suboptimal model performance, degrading the faithfulness of these methods.

To overcome the above issue, in this article, we aim to predict \( r_{u,i} \) via differently handling the two opposite polarities of reviews. One straightforward solution is to divide the aspects into user-preferred and user-rejected aspects from positive and negative reviews, respectively. The objective is then formulated as follows:

\[
\hat{r}_{u,i} = \rho_u^P s_{u,i}^P - \rho_u^R s_{u,i}^R,
\]

where the subtraction of the two scores considers both user-preferred and user-rejected aspects during the rating prediction of \( u \) toward \( i \). Vector \( s_{u,i}^P \) (\( s_{u,i}^R \)) consists of the estimated scores of \( u \) toward \( i \) on the user-preferred (user-rejected) aspects. \( \rho_u^P \) (\( \rho_u^R \)) denotes the importance vector of the user-preferred (user-rejected) aspects for \( u \), extracted from \( u \)'s positive (negative) reviews.

Moreover, to tackle the imbalance between positive and negative reviews, we intuitively assume that there exists a latent correlation of importance between the two kinds of aspects. We model the correlation to generate the aspect importance offsets \( \mu_u^P \) and \( \mu_u^R \) to enhance \( \rho_u^P \) and \( \rho_u^R \), respectively. Ultimately, the predictive model of RPR is given as

\[
\hat{r}_{u,i} = (\mu_u^P + \rho_u^P) s_{u,i}^P - (\mu_u^R + \rho_u^R) s_{u,i}^R.
\]

In the following, we will elaborate the proposed RPR model in threefold: aspect score estimation, aspect importance extraction, and aspect importance offset learning.

#### B. Aspect Score Estimation

Following mainstream recommender models [30], we map users and items into a latent factor space and represent user \( u \) and item \( i \) by latent factor vectors \( p_u \in \mathbb{R}^f \) and \( q_i \in \mathbb{R}^f \), respectively. According to [40], the interaction between \( u \) and \( i \) on each latent factor is characterized by \( p_u \odot q_i \), where \( \odot \) represents the element-wise product between two vectors. Similar to [12], to leverage latent factor-level interactions for aspect score prediction, we introduce two indicator matrices \( M \in \mathbb{R}^{f \times |P|} \) and \( V \in \mathbb{R}^{f \times |R|} \) (where \(|P| \) and \(|R| \) are the numbers of user-preferred and user-rejected aspects, respectively), to associate the latent factors to different user-preferred and user-rejected aspects, respectively. The weight vector \( m_i \), which is the \( x \)th column of \( M \), indicates which latent factor-level interactions are related to the score of the \( x \)th user-preferred aspect. Similarly, the weight vector \( v_i \), which is the \( y \)th column of \( V \), indicates which interactions are related to the score of the \( y \)th user-rejected aspect. Toward this end, we estimate the scores of the user-preferred and user-rejected aspects via the following formula:

\[
\begin{align*}
\hat{s}_{u,i}^P &= M^\top (p_u \odot q_i) \\
\hat{s}_{u,i}^R &= V^\top (p_u \odot q_i)
\end{align*}
\]

where \( \hat{s}_{u,i}^P \) and \( \hat{s}_{u,i}^R \) represent user’s preference scores on the user-preferred and user-rejected aspects of items, respectively.

#### C. Aspect Importance Extraction

We define that the user \( u \)'s positive document \( D_u^{pos} \) is constructed by collecting all the positive reviews posted by \( u \), and the user’s negative document \( D_u^{neg} \) is built in a similar manner. It is widely accepted that users tend to comment on aspects with opposite attitudes pertaining to different polarities of reviews. In addition, users would individually care more about certain aspects than others. For example, fashion enthusiasts often focus on the user-preferred aspect “fashionable style”
of “clothing” items, as the review words like “fashion sense” frequently appear in their reviews. In the following, we mainly detail the formulation on how to extract the user-preferred aspect importance by word-wise extraction of $D_{\text{pos}}^u$, while the user-rejected one can be obtained in a similar way.

First, we use the pretrained word embeddings to initialize word representations in the positive document, where $e_j \in \mathbb{R}^d$ is the embedding vector for the $j$th word, and $d$ denotes the embedding dimension. We then adopt a CNN model to extract the contextual information of each word \[41\], \[42\]. The convolution layer can be regarded as a tensor $[41]$, \[42\]. We then adopt a CNN model to

\begin{equation}
\begin{aligned}
    \mathbf{c}_j &= \text{ReLU}(\mathbf{W}_c (\{e_{j-\epsilon}, \ldots, e_j, \ldots, e_{j+\epsilon}\}) \ast \mathbf{K}) + \mathbf{b}_c \\
    \rho^u_y &= \text{softmax} \left( \sum_{j=1}^{l_{\text{pos}}} e_j \right)
\end{aligned}
\end{equation}

where $\mathbf{c}_j \in \mathbb{R}^N$ is the latent contextual feature vector for the $j$th word, and $\mathbf{W}_c \in \mathbb{R}^{N \times N}$ and $\mathbf{b}_c \in \mathbb{R}^N$ denote the weight matrix and bias vector, respectively.

Second, we focus on how to extract the user-preferred aspect importance based on the word contextual feature vectors. In fact, some salient words are supposed to contribute more to aspect importance modeling of users. For example, the word “cost-effective” in reviews implies that the user probably puts more emphasis on aspects like “good price” and “effectiveness.” In addition, it is natural that if an aspect-specific word is frequently mentioned in $D_{\text{pos}}^u$, the user $u$ will attach more importance to aspects related to this word. For example, “delicious” and “yummy” would be repeatedly written by a user who pays much attention to the “good taste” aspect. Inspired by this, we develop a fine-grained semantic extraction network for the aspects. Specifically, we resort to the fully connected layers to automatically discriminate the importance contribution of each review word:

\begin{equation}
\begin{aligned}
    \mathbf{e}_j &= \text{ReLU}(\mathbf{W}_r \mathbf{e}_j + \mathbf{b}_r) \\
    \rho^u_y &= \text{softmax} \left( \sum_{j=1}^{l_{\text{pos}}} \mathbf{e}_j \right)
\end{aligned}
\end{equation}

where the semantic embedding $\mathbf{e}_j \in \mathbb{R}^{[P]}$ is projected from the $j$th word contextual features through a matrix $\mathbf{W}_r \in \mathbb{R}^{[P] \times N}$, bias $\mathbf{b}_r \in \mathbb{R}^{[P]}$, and the rectified linear unit (ReLU) activation function. $l_{\text{pos}}$ represents the number of words in the user’s positive document, and $\rho^u_y \in \mathbb{R}^{[P]}$ is the user-preferred aspect importance, reflecting the emphasis degree on each user-preferred aspect attached by user $u$.

From the above formulation, it can be seen that each element in the word semantic embedding contributes distinctively to aspect importance modeling. In the light of this, we adopt a loose hypothesis that each aspect associates closely with certain review words. We hence assume that one word can express one aspect (i.e., one element in the aspect vector) if the corresponding element from the word semantic embedding is the largest, since the dimensions of word semantic embedding and aspects are the same.\[5\] For example, as shown in Fig. 4, the $j$th word, whose first element $e_{j,1}$ is the largest in semantic embedding $e_j$, can be associated with the first aspect.

Based on the same procedure, we can also extract the user’s importance vector for user-rejected aspects $\rho^u_y^\text{neg}$ with a similar module from her/his negative document $D_{\text{neg}}^u$.\[6\]

D. Aspect Importance Offset Learning

Though we have obtained the expected scores and the corresponding importance distribution for the user-preferred and user-rejected aspects, it is still insufficient to correctly predict the overall rating. The reason is that in a practical scenario, it is common that the objective user $u$ provides much more reviews from one polarity than the other, referring to Fig. 2. For a better understanding, we make an extreme assumption that if user $u$ only posts positive reviews, the learned user-rejected aspect importance vector $\rho^u_y^\text{neg}$ would approximately approach zero in RPR. Furthermore, if we leverage the user-rejected aspect importance vector to predict the ratings toward the items in aspects the user distastes, the predicted rating would be unfavorable.

To solve the imbalance in reviews with different polarities, we propose to construct the relationships between the user-preferred and user-rejected aspect importance. In particular, we intuitively assume that if a user attaches more importance to a user-preferred aspect, she/he will correspondingly pose roughly equal importance to the related user-rejected aspects. For example, a user pays attention to the user-preferred aspect of “elegant environment” and has chosen a restaurant, she/he would similarly care about user-rejected aspects like “obsolete decoration” and “poor sanitary situation.” Based on this assumption, we leverage an aspect-aware importance weighting module to construct the mapping relationship between the user-preferred and user-rejected aspect importance.

To construct the relationships between two opposite polarities of aspects, we use the common $\text{user/item latent space}$ as a bridge. Given that the aspect is associated with specific latent factors as shown in (4), the related aspects from the other polarity are expected to be more similar in the latent space. As weight vectors $\mathbf{m}_x$ and $\mathbf{v}_y$ are responsible for latent aspect modeling (as mentioned in Section III-B), RPR takes these indicator vectors as inputs to the aspect-aware importance weighting module for measuring the similarity among aspects. Specifically, the following function is used to obtain the attention weight of the user-rejected aspect $y$ to each user-preferred aspect

\begin{equation}
\begin{aligned}
    \phi^y_{x,t} &= \text{h}_r \text{ReLU}(\mathbf{W}_a (\mathbf{v}_y \odot \mathbf{m}_x) + \mathbf{b}_a) \\
    \phi_{y,t} &= \frac{\exp(\phi^y_{x,t})}{\sum_{y' \in \mathcal{P}} \exp(\phi^y_{x, t'})}
\end{aligned}
\end{equation}

where $\phi^y_{x,t}$ denotes the attention weight of the user-rejected aspect $y$ to the $x$th user-preferred aspect, and $\mathbf{h}_r$, $\mathbf{W}_a$, and $\mathbf{b}_a$ are the parameters of the aspect attention network. For simplicity, we concatenate these attention weights into a matrix $\Phi \in \mathbb{R}^{[P] \times [R]}$. The $y$th column $\phi_y$ of matrix $\Phi$ denotes the attention weight vector of the user-rejected aspect $y$, whose $x$th element is $\phi^y_{x,t}$.
We take the attention weights as the mapping relationships between the user-preferred and user-rejected aspect importance. As for the aforementioned imbalance problem, we can use an offset vector as an addition to the user-rejected aspect importance vector using the matrix $\Phi$. Thus, the enhanced user-rejected aspect importance vector of user $u$ equals

$$
\begin{align*}
\mu_u' &= \Phi^T \rho_u^p \\
\rho_u^r &= \rho_u^p + \mu_u^r
\end{align*}
$$

where $\mu_u'$ denotes the offset for user-rejected aspect importance, and $\rho_u^p$ is the user-preferred aspect importance vector that has already been extracted in (6). The $y$th element of the offset vector $\mu_u'$ equals the inner product $\phi_y^u \cdot \rho^r_y$, and the attention weights of the $y$th user-rejected aspect on each user-preferred aspect model a linear relationship, which maps the extracted user-preferred aspect importance to the objective user-rejected aspect importance space. In this way, we can achieve a more intuitive user-rejected aspect importance vector of a user from her/his positive reviews indirectly, even if negative reviews are inadequate. Fig. 5 illustrates the generation process of importance offset for the $y$th user-rejected aspect.

Similarly, we can obtain the enhanced user-preferred aspect importance $\rho_u'^+$, where the process is omitted due to space limitation. Extensive experiments have demonstrated that the offset components can effectively resolve the imbalance problem between positive and negative reviews, which will be elaborated in Section V-B.

E. Overall Objective

Up to now, we have obtained the scores and importance of the user-preferred and user-rejected aspects, respectively. The expected rating $\hat{r}_{u,i}$ that user $u$ would give to item $i$ is computed as follows:

$$
\hat{r}_{u,i} = \rho_u'^+ s_{u,i} - \rho_u'^+ r_{u,i}. \tag{9}
$$

In this way, both the user positive and negative preferences can be simultaneously considered and discriminated. The estimation of model parameters is to minimize the rating prediction error on the training dataset. In this way, the objective function is

$$
\min_{p_u, q_i, M, V, \theta} \frac{1}{2} \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \beta_1 (\|M\|_1 + \|V\|_1) + \frac{\beta_2}{2} \left( \|p_u\|_2^2 + \|q_i\|_2^2 + \sum_{\theta \in \Theta} \|\theta\|_2^2 \right). \tag{10}
$$

With the minimization of this objective function, all the model parameters can be effectively updated through the gradient decent strategy. The involved parameters include user and item latent vectors $p_u$ and $q_i$, indicator matrices $M$ and $V$ from the aspect score estimation module, and the remaining parameters $\Theta$ from the aspect importance extraction and the aspect importance offset learning modules. $\| \cdot \|_1$ and $\| \cdot \|_2$, respectively, denote the $l_1$ and $l_2$ regularization norms, with the corresponding hyperparameters $\beta_1$ and $\beta_2$. Considering the fact that $l_1$ regularization yields sparse solution of the weights, we thus use $l_1$ norm to obtain approximately binary matrices $M$ and $V$ for better indication ability. The $l_2$ regularization of $p_u$, $q_i$, and $\theta$ prevents these parameters from uncontrollable values and over-fitting.

**Optimization:** We leverage the Adam optimization algorithm [43] to learn all the parameters by minimizing the objective function in (10). Note that for training stability, the parameter is optimized in the following sequence: the user matrix, the item matrix, the weight matrices, and the remaining parameters.

IV. Experimental Setup

In this section, we first presented the evaluation datasets, and then introduced our experimental settings. Finally, we listed several baseline methods for comparison.

A. Datasets

We conducted experiments on two publicly available datasets: Amazon$^1$ and Yelp$^2$. The Amazon dataset provides rich review information with rating scores. In our experiments, we applied its seven subdatasets: Musical Instruments, Office Products, Digital Music, Tools Improvement, Automotive, Toys and Games, and Video Games. Yelp is a famous online review platform for business, such as restaurants, bars, and spas. We selected the dataset from the latest version and used a 20-core setting to provide a denser dataset. Each record in our datasets consists of user Identity (ID), item ID, rating, and the corresponding review text. For all the datasets, we filtered out the empty-review records. The target rating score used in these datasets ranges from 1 to 5. Similar to [44], reviews with rating scores higher than or equal to 3 are split into positive documents, while the ones lower than 3 are regarded negative on all the datasets. Table I summarizes the detailed statistics of the evaluated datasets, where “pos/neg ratio” denotes relative proportions between positive and negative reviews of each dataset.

For each dataset, we randomly split its records into two parts: 80% for training and the rest 20% for testing. Moreover, 10% records in the training set are randomly selected as the validation set for hyperparameter tuning. Note that we slightly adjusted the training and testing sets to ensure that at
layer in the review-based models, the filter size is set to 3, and we tested various numbers of filters among \{20, 50, 100, 200\} to select the optimal one. Dropout is appended after all fully connected and convolution layers with a dropout rate of 0.2. For TransNet, we used two transform layers, following the model setting adopted in the original article [7]. For MPCN, the number of pointers is tuned among \{1, 3, 5, 8, 10\}. For ALFM, we tuned the numbers of aspects and latent factors following the original article [12]. For CARP, we took the suggestion in the original article [23], and set the capsule number and the predefined threshold value to 5 and 3, respectively.

V. EXPERIMENTAL RESULTS

To validate the effectiveness of our proposed method, we conducted extensive quantitative and qualitative experiments to answer the following questions.

1) Q1: Can our proposed method outperform both the state-of-the-art review-involved and traditional recommendation baselines?
2) Q2: How do different components (e.g., the aspect-aware importance weighting component for learning importance offsets) contribute to the overall performance of our proposed model?
3) Q3: How do the key hyperparameters (e.g., the number of latent factors) affect our model performance?
4) Q4: Can our model provide explicit interpretations?

A. Q1: Performance Comparison

The results of our method and other baselines over the experimented datasets are presented in Table II. The key observations can be summarized as follows.

1) The recommendation methods from the first category obtain the worst performance on all the datasets. The deep learning-based methods (i.e., NeuMF and MLP) can achieve superior performance compared with the factorization-based ones (i.e., FM and MF), since they can model more complex interactions than the factorization ones.

2) The review-involved methods consistently surpass the interaction-based ones, demonstrating that reviews contain valuable side information for accurate recommendation. Moreover, among the review-involved baselines, it is obvious that MPCN largely outperforms both DeepCoNN (D-CON) and TransNet (T-NET), which mainly benefits from the fact that MPCN adopts the pointer-based scheme to filter important reviews for recommendation. Nevertheless, TransNet outperforms DeepCoNN, since it takes the review of target user–item pair as the approximation object in the training process.

3) The aspect-aware methods surpass the plain review-involved ones in most cases. This indicates that capturing aspect-aware information from reviews is effective. Among these baselines, the recently proposed CARP model yields a better result than ALFM. It is because CARP introduces the capsule network, which is proposed to model the complex relationships among

---

**TABLE I**

| Datasets           | Users | Items | Ratings | pos/neg ratio |
|--------------------|-------|-------|---------|---------------|
| Musical Instruments| 1,429 | 900   | 10,262  | 10.11         |
| Office Products    | 4,905 | 2,420 | 53,258  | 9.10          |
| Digital Music      | 5,341 | 3,568 | 64,706  | 5.67          |
| Tools Improvement  | 16,638| 10,217| 134,345 | 6.14          |
| Automotive         | 2,928 | 1,835 | 20,473  | 9.20          |
| Toys and Games     | 19,412| 11,924| 167,597 | 8.09          |
| Video Games        | 24,303| 10,672| 231,377 | 5.25          |
| Yelp               | 40,500| 58,755| 2,024,283| 2.70          |

---

least one record for each user/item would be included in the training set. The target reviews in the validation and testing sets are excluded since they are unavailable in the practical scenario.

B. Experimental Settings

1) Evaluation Metrics: To thoroughly evaluate our model and the baselines, we adopted the mean squared error (MSE) and the mean absolute error (MAE) as the evaluation metrics to measure the rating prediction performance.

2) Implementation Details: We implemented our model via the development tool Tensorflow.\(^6\) The embedding matrix of the document is initialized via word vectors which have been pretrained in GloVe\(^7\) (used 50-d vectors for its efficiency). For the convolutional layer in the proposed model, the number and size of filters are set to 50 and 3, respectively. We used the popular approach of Xavier [45] to initialize the weights in our model. We adopted grid search to tune the hyperparameters based on the results from the validation set. Moreover, we varied the number of both the user-preferred and user-rejected aspects within the set \{1, 2, 3, 4, 5\}, the dimension of user and item latent factor vectors among \{4, 8, 16, 32, 64\}, the learning rate among \{1E-05, 1E-04, 1E-03, 1E-02\}, and the size of training mini-batch among \{100, 200, 500, 1,000\}.

C. Baseline Comparison

We compared the performance of our proposed method with a series of state-of-the-art recommendation methods. To summarize, we divided the baselines into three categories. The first category is interaction-based, including MF, FM [46], MLP [30], and NeuMF [30]. The second category is plain review-involved, including DeepCoNN [6], TransNet [7], and MPCN [9]. The last category is aspect-aware, including ALFM [7], FM [46], and CARP [23].

We adopted the publicly available implementations of FM, NeuMF, DeepCoNN, TransNet, MPCN, ALFM, and CARP in our experiments. When training all these baseline models, we set the maximum training epoch to 50 for a fair comparison. For the interaction-based models, we varied the embedding size within \{8, 16, 32, 64\}. All word embeddings in review-based baselines are initialized using pretrained word vectors in GloVe [4] or word2vec [3].\(^8\) For the convolutional

\(^6\)http://www.tensorflow.org
\(^7\)https://nlp.stanford.edu/projects/glove/
\(^8\)https://code.google.com/archive/p/word2vec/
TABLE II

Performance Comparison on Eight Datasets. The Best Performance Is Highlighted in Boldface. ΔCA Denotes the Relative Improvement (%) of RPR Over the Best Baseline CARP. P-Value Reflects the t-Test Result of RPR Compared With CARP.

| Datasets       | Musical Instruments | Office Products | Digital Music | Tools Improvement |
|----------------|---------------------|-----------------|---------------|-------------------|
|                | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| MF             | 1.970 | 1.404 | 1.144 | 1.070 | 1.956 | 1.399 | 1.560 | 1.249 |
| FM             | 1.167 | 1.080 | 1.095 | 1.048 | 1.395 | 1.181 | 1.320 | 1.149 |
| MLP            | 1.082 | 1.040 | 1.120 | 1.058 | 1.356 | 1.164 | 1.331 | 1.162 |
| NeuMF          | 1.187 | 1.089 | 1.078 | 1.038 | 1.334 | 1.155 | 1.314 | 1.146 |
| Di-CON         | 1.286 | 1.135 | 0.975 | 0.823 | 1.330 | 1.153 | 1.248 | 1.063 |
| T-NET          | 1.130 | 0.872 | 0.951 | 0.789 | 1.325 | 1.052 | 1.124 | 0.923 |
| MPCN           | 0.923 | 0.860 | 0.879 | 0.738 | 1.291 | 0.936 | 1.097 | 0.912 |
| ALFM           | 0.891 | 0.735 | 0.870 | 0.758 | 1.280 | 0.976 | 1.065 | 0.870 |
| CARP           | 0.879 | 0.714 | 0.827 | 0.686 | 1.236 | 0.945 | 1.069 | 0.847 |
| RPR            | 0.795 | 0.652 | 0.814 | 0.641 | 1.141 | 0.836 | 0.987 | 0.784 |
| ΔCA            | 9.6  | 5.7  | 1.6  | 6.6  | 7.7  | 11.5 | 7.7  | 7.4  |
| p-value        | 9.8E-4 | 5.6E-4 | 1.5E-3 | 4.8E-4 | 1.5E-3 | 1.5E-3 | 9.7E-4 | 4.5E-4 |

TABLE III

Performance Comparison of the Model Variants on the Automotive and Video Games Datasets.

| Setup                        | Automotive | Video Games |
|------------------------------|------------|-------------|
| Base                         | 0.896      | 1.142       |
| Coarse-grained Model         | 0.952      | 1.319       |
| W/o Review Polarity          | 0.949      | 1.325       |
| Uniform Aspect Importance    | 0.964      | 1.330       |
| W/o Importance Offset        | 0.947      | 1.384       |

Finally, we can observe that RPR substantially outperforms all the baselines on the eight datasets. When comparing with the review-involved baselines regarding the MSE metric, its relative improvement is satisfying with gains up to 38.2% (DeepCoNN), 29.6% (TransNet), and 13.9% (MPCN), respectively. Moreover, it is obvious that RPR consistently exceeds the aspect-aware baselines by a considerable margin. Jointly analyzing Fig. 2 and Table II, we observed that the improvements over the best baseline are more significant in datasets Digital Music, Tools Improvement, Video Games, and Yelp, where the negative reviews are more sufficient. This observation indicates that discriminately treating reviews with different polarities would promote the recommendation performance.

B. Q2: Ablation Study

We conducted detailed ablation studies to validate how each component contributes to the overall performance of our model. In particular, we compared our proposed model with the following variants.

1) **Base**: The base refers to our complete model with the optimal setting.

2) **Coarse-Grained Model**: Instead of extracting the word-wise aspect importance distribution in (5), this variant introduces the max-pooling layer to the convolutional layer in the base.

3) **W/o Review Polarity**: We removed the components of distinguishing the positive and negative reviews of the user and followed the previous review-involved methods to collect all the reviews as the user document.

4) **Uniform Aspect Importance**: We replaced $\rho_u^p$ and $\rho_u^r$ in (9) with the uniform importance distributions, i.e., all user-preferred and user-rejected aspects are assumed to be equally important.

5) **W/o Importance Offset**: We removed the aspect-aware importance weighting component from the complete model. It is worth noting that the imbalance problem is not specifically tackled in this variant.

Table III shows the MSE results of the above variants on the Automotive and Video Games datasets. First, the word-wise extraction of aspect importance outperforms the document-wise one, which verifies that the fine-grained semantic extraction boosts the performance of review-involved recommendation. Second, we can observe that ignoring the polarities of reviews will deteriorate the performance, and it is thus necessary to distinguish positive and negative reviews for review-involved recommendation. In addition, it is reasonable for a user to attach different importance to different aspects of an item. Thus, uniforming the aspect importance would limit
the modeling capacity of user preferences. Finally, the result of the last variant reflects that the aspect-aware importance weighting module in our model can effectively reduce the impact of review imbalance.

C. Q3: Effectiveness of Key Hyperparameters

In this section, we analyzed the effectiveness of the key hyperparameters in our method for the overall performance. We primarily focused on the number of aspects and latent factors (i.e., the dimension of embeddings $p_u$ and $q_i$). Fig. 6 shows the performance variations with changing the number of aspects and latent factors on three datasets.

1) Number of Aspects: We set the number of latent factors to 32 for better studying the effect of aspects. With the number of aspects changing from 1 to 5 as illustrated in Fig. 6, we observed that the optimal number of aspects varies with different datasets, which is probably because users comment different aspects in reviews for different categories of items. In addition, promising performance can be obtained when the number of user-preferred/user-rejected aspects is from 2 to 4.

2) Number of Latent Factors: To study the effect of latent factors, we fixed the number of user-preferred/user-rejected aspects to 3. From Fig. 6, it can be seen that the MSE decreases with increasing the latent factors, since the rating prediction is still based on MF in our model. Therefore, increasing latent factors can better represent the user/item, contributing to a more accurate rating prediction.

To visualize the joint effects of aspects and factors, we presented 3-D figures by varying the number of user-preferred/user-rejected aspects in $\{1, 2, 3, 4, 5\}$ and the number of factors in $\{4, 8, 16, 32, 64\}$ and illustrated the results of the Office Products and Yelp datasets. From Fig. 7, it can be recognized that the optimal numbers of aspects and factors are different across datasets. In general, more latent factors usually lead to better performance, while the optimal number of user-preferred/user-rejected aspects might depend on the review details of different datasets.

D. Q4: Interpretability

In our RPR model, a user’s preference on an item depends on the scores and the importance of both user-preferred and user-rejected aspects to the user. The importance on user-preferred/user-rejected aspects is computed by the aspect-specific semantic embeddings in the user’s positive/negative reviews. Since we assume one element corresponds to one aspect, accordingly, review words which hold the largest corresponding elements in their semantic embeddings are adopted as the semantic explanation for this aspect. Table IV records the positive and negative reviews of a randomly selected user from the Musical Instruments dataset.

In our experiments, we set the number of user-preferred and user-rejected aspects to 2 and filtered their corresponding review words. Table V shows the top 10 aspect words (we removed stop words for better illustration). Based on this experiment, the two user-preferred aspects can be semantically interpreted as “Performance” and “Fine Strings,” and the two user-rejected aspects can be interpreted as “Poor Quality” and “Crack Sensitive.”

Next, we aimed to study the interpretability of our proposed model on high and low ratings [11], [47]. From the same dataset, we chose “item 1” and “item 2,” which are rated with 5 and 1 by the selected user, respectively. We first obtained the selected user’s aspect importance on the user-preferred and user-rejected aspects (i.e., $\rho_p^{u,i}$ and $\rho_r^{u,i}$), and then computed the aspect scores on the user-preferred and user-rejected aspects (i.e., $s_p^{u,i}$ and $s_r^{u,i}$) of “item 1” and “item 2,” respectively. As shown in Table VI, we could observe that the

Fig. 6. Effect of the number of aspects and factors in our model.

Fig. 7. Confounding effect of the number of aspects and factors.
TABLE IV
POSITIVE AND NEGATIVE REVIEWS OF A RANDOMLY SELECTED USER FROM \textit{Musical Instruments}

| Positive Reviews                                                                 | Negative Reviews                                                                 |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| ...My guitar player had a different lock system, and his $1000 Les Paul         | It’s a decent unit, but stopped working completely after about 6 months.          |
| fell to the stage, completely knocking it out of tune. Mine stayed locked         | Tried every thing I could, but it’s gone. I would not recommend this pedal.       |
| perfectly...                                                                     | The plastic piece that screws the wire into the end was made of very thin plastic,|
|                                                                                | and cracked in two within a week of light use.                                    |
| These strings are nice and easy to fly over. No buildup, or residue on your     | Nothing sounds as bright as these strings. I’ve tried many, and these are the    |
| fingers either. I really like this stuff.                                        | best by far. I know, cause I play guitar really god-like.                          |
|                                                                                | I expected more from this manufacturer, but I guess the quality is not the same   |
|                                                                                | as it used to be.                                                                  |
| I often use this to record my band’s gigs. Now if they would make one that      |                                                                                |
| will hold a PAR 38 so I can turn my extra microphone stands into single can     |
| lighting racks!                                                                   |                                                                                |

TABLE V
TOP 10 WORDS FOR EACH USER-PREFERRED AND USER-REJECTED ASPECTS OF THE SELECTED USER FROM \textit{Musical Instruments}. THE “ASPECT LABELS” ARE ATTACHED BASED ON THE INTERPRETATION OF THE ASPECT

| User-preferred Aspects | User-rejected Aspects |
|------------------------|-----------------------|
| Performance            | Fine Strings          |
| record                 | tried                 |
| system                 | build-up              |
| gigs                   | fingers               |
| stayed                 | extra                 |
| tune                   | perfectly             |
| band                   | sounds                |
| guitar                 | strings               |
| stuff                  | easy                  |
| stage                  | bright                |
| knocking               | guitar                |
| microphone             | lock                  |
|                       | but                   |
|                       | cracked               |
|                       | working               |
|                       | lasted                |
|                       | stopped               |
|                       | unit                  |
|                       | use                   |
|                       | months                |
|                       | quality               |
|                       | tape                  |
|                       | frustrating           |
|                       | piece                 |
|                       | wire                  |
|                       | gigs                 |
|                       | end                   |
|                       | pedal                |
|                       | manufacturer         |
|                       | work                  |
|                       | pay                   |
|                       | guess                |
|                       | decent               |
|                       | screw                |

VI. CONCLUSION AND FUTURE WORK
In this article, we present a RPR which treats reviews with different polarities discriminately. Specifically, RPR simultaneously learns the scores and importance of the user-preferred and user-rejected aspects to a user. The final rating is then estimated via the mathematical difference in the positive and negative scores, which are the weighted sum of the relevance scores with the corresponding importance. To overcome the problem of imbalanced review polarity, RPR implements an aspect-aware importance weighting module to effectively learn the mapping relationships for one aspect importance based on the other. In addition to its remarkable performance over eight datasets, RPR is also capable of providing explicit explanation for the recommendation results.

In future, we will apply pairwise learners to strengthen RPR and validate its effectiveness via extensive experiments. Moreover, we are particularly interested in discriminating the user-preferred and user-rejected aspects of multimedia items [48], which contain abundant aspect information to reflect users’ preferences. Another interesting direction is to extend RPR to solve the long-tail recommendation problem by extracting semantic information from attribute descriptions of less frequent items, which requires urgent solution for e-commerce platforms.

REFERENCES
[1] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinakal, and M. S. Kankanhalli, “MMALFM: Explainable recommendation by leveraging reviews and images,” \textit{ACM Trans. Inf. Syst.}, vol. 37, no. 2, pp. 1–28, 2019.
[2] K. Liu, X. Li, Z. Zhu, L. Brand, and H. Wang, “Factor-bounded nonnegative matrix factorization,” \textit{ACM Trans. Knowl. Discovery Data}, vol. 15, no. 6, pp. 1–18, May 2021.
[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in \textit{Proc. Adv. Neural Inf. Process. Syst.}, 2013, pp. 3111–3119.
[4] J. Pennington, R. Socher, and C. Manning, “GLOVE: Global vectors for word representation,” in \textit{Proc. EMNLP}, 2014, pp. 1532–1543.
[5] Y. Guo, Z. Cheng, J. Jing, Y. Lin, L. Nie, and M. Wang, “Enhancing factorization machines with generalized metric learning,” \textit{IEEE Trans. Knowl. Data Eng.}, early access, Oct. 29, 2020, doi: 10.1109/TKDE.2020.3034613.
[6] L. Zheng, V. Noroozi, and P. S. Yu, “Joint deep modeling of users and items using reviews for recommendation,” in \textit{Proc. WSDM}, 2017, pp. 425–434.
[7] R. Catherine and W. Cohen, “Transnets: Learning to transform for recommendation,” in \textit{Proc. RecSys}, 2017, pp. 288–296.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Jianhua Yin (Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, Shandong University, Jinan, China. He has published several articles in top venues, such as Association for Computing Machinery Transactions on Information Systems (ACM TOIS), IEEE Transactions on Knowledge and Data Engineering (TKDE), ACM International Conference on Multimedia (MM), ACM Knowledge Discovery and Data Mining (SIGKDD), ACM International Conference on Research on Development in Information Retrieval (SIGIR), and IEEE International Conference on Data Engineering (ICDE). His research interests mainly include data mining and machine learning applications.

Zan Gao (Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011. From September 2009 to September 2010, he was with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. From July 2016 to January 2017, he was also with the School of Computing, National University of Singapore, Singapore. From 2011 to 2018, he worked with the School of Computer Science and Engineering, Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, Tianjin, China. He is currently a Full Professor with the Shandong Artificial Intelligence Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. He has authored more than 100 scientific articles in international conferences and journals including IEEE Transactions on Image Processing (TIP), IEEE Transactions on Multimedia (TMM), IEEE Transactions on Cybernetics (TCYBE), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), ACM International Conference on Multimedia (MM), International Conference on Research on Development in Information Retrieval (SIGIR), International World Wide Web Conferences (WWW), and AAAI Conference on Artificial Intelligence (AAAI). His research interests include artificial intelligence, multimedia analysis and retrieval, computer vision, and machine learning.

Liqiang Nie (Senior Member, IEEE) received the B.Eng. degree from Xi’an Jiaotong University, Xi’an, China, in 2009, and the Ph.D. degree from the National University of Singapore (NUS), Singapore, in 2013. After Ph.D., he continued his research in NUS as a Research Fellow for more than three years. He is currently a Professor with the School of Computer Science and Technology, Shandong University, Qingdao, China. Meanwhile, he is the Adjunct Dean with the Shandong Artificial Intelligence (AI) Institute Jinan, China. His research interests lie primarily in multimedia computing and information retrieval. He has published around 100 articles in the top conferences or journals, with more than 10000 Google Scholar citations. He was granted several awards like SIGIR and the ACM International Conference on Multimedia (MM) Best Paper Nomination Award, the ACM Special Interest Group in Multimedia (SIGMM) Rising Star, the Alibaba DAMO Academy (DAMO) Young Fellow, the AI 2000 the Most Influential Scholar in artificial intelligence, and the Massachusetts Institute of Technology (MIT) TR35 China. He is an Associate Editor (AE) of IEEE TKDE, TMM, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), and Information Science. He serves as a Program Committee (PC) Chair of International Conference on Internet Multimedia Computing and Service (ICIMCS) 2017, Pacific-Rim Conference on Multimedia (PCM) 2018, and ChinaMM 2020, and an Area Chair of ACM MM 2018-2021.