Recommendation by Users' Multimodal Preferences for Smart City Applications

Cai Xu, Ziyu Guan, *Member, IEEE*, Wei Zhao, Quanzhou Wu, Meng Yan, Long Chen, and Qiguang Miao

***Abstract*—As an essential role in smart city applications, personalized recommender systems help users to find their potentially interested items from their historically generated data. Recently, researchers have started to utilize the massive user-generated multimodal contents to improve recommendation performance. However, previous methods have at least one of the following drawbacks: 1) employing shallow models, which cannot well capture high-level conceptual information; 2) failing to capture personalized user visual preference. In this article, we present a deep users' multimodal preferences-based recommendation (UMPR) method to capture the textual and visual matching of users and items for recommendation. We extract textual matching from historical reviews. We construct users' visual preference embeddings to model users' visual preference and match them with items' visual embeddings to obtain the visual matching. We apply UMPR on two applications related to smart city: restaurant recommendation and product recommendation. Experiments show that UMPR outperforms competitive baseline methods.**

***Index Terms*—Deep neural network, multimodal, recommendation, smart city, visual preference.**

Manuscript received May 9, 2020; revised June 22, 2020; accepted July 2, 2020. Date of publication July 29, 2020; date of current version March 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61672409, under 61936006, under 61876144, and under 61876145, in part by the Major Basic Research Project of Shaanxi Province under Grant 2017ZDJC-31, in part by the Key Research and Development Program of Shaanxi under Program No. 2020ZDLGY04-07, in part by the Shaanxi Province Science Fund for Distinguished Young Scholars under Grant 2018JC-016, in part by the Natural Science Basic Research Program of Shaanxi under Program No. 2020JQ-850, and in part by the Fundamental Research Funds for the Central Universities under Grant JB190301 and Grant JB190305. Paper no. TII-20-2327. *(Corresponding author: Ziyu Guan.)*

Cai Xu, Ziyu Guan, Wei Zhao, Quanzhou Wu, and Meng Yan are with the State Key Laboratory of Integrated Services Networks, School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: cxu_3@stu.xidian.edu.cn; zyguan@xidian.edu.cn; ywzhao@mail.xidian.edu.cn; quanzhouwu@stu.xidian.edu.cn; mengyan@stu.xidian.edu.cn).

Long Chen is with the School of Communications and Information Engineering & School of Artificial Intelligence, Xi'an University of Posts & Telecommunications, Xi'an 710121, China (e-mail: chenlong@xupt.edu.cn).

Qiguang Miao is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: qgmiao@xidian.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.3008923

I. Introduction

SMART city has drawn great interest in both science and engineering fields, transforming different areas of human life, such as transportation [1], shopping [2], online social [3], [4] and so on. Recommender systems help citizens to find their potentially interested items among a variety of options rapidly. Various recommender systems, such as restaurant recommendation [2], service recommendation [5], and trip recommendation [6] are established in cities to save the time and/or promote the quality of life for citizens. For example, the catering softwares recommend potentially interesting restaurants to citizens, which could improve their happiness and increase the volume of customers for the restaurants. Recommender systems make the city smarter.

Many of the prominent recommendation methods [7]–[9] make predictions about a users interest by collecting preferences information from the historical ratings of many users. Users sharing similar preferences in the past tend to make similar choices in the future. However, in many real world applications, the number of items rated by users is negligible to the total number of items, causing the well-known sparsity problem [10].

Recent recommender systems attempt to alleviate the sparsity problem by exploiting user-generated contents such as users' textual reviews [11]. The abundant user-generated contents also play an important role in smart city [12]. For example, most shopping/restaurant-rating websites encourage users to write reviews for the purchased items. As accompanying information of ratings, reviews explain why a user assigns such a rating to an item and contain knowledge about the user's preference and the item's pros/cons. A typical restaurant recommendation scenario is shown in Fig. 1. From the review sentence, "This place has a really good vibe," we perceive the user prefers the inside environment of this restaurant. However, it is hard to understand such preference by the textual reviews only.

"A picture may paint a thousand words." Compared with textual reviews, images could convey user's preference that may not be well captured by language. Inspired by this, multimodal contents-based recommender systems [2], [13]–[15] are proposed to extract knowledge in reviews and the item's images for recommendation. However, previous methods have at least one of the following drawbacks: 1) employing shallow models, which cannot well capture high-level conceptual information in multimodal contents (especially images); 2) failing to capture personalized user visual preference. As an essential

Fig. 1. Visual description of a restaurant and the textual review of a user on this restaurant.

configuration in smart city, personalization brings the services of the city closer to citizens.

In many smart city applications such as restaurant recommendation, item's images are often grouped into visual views, which are presented to users with the corresponding item. For example, there are four visual views, "food," "drink," "inside," and "outside" in restaurant-rating websites and ten visual views, "appearance," "guest room," "lobby," "restaurant," etc., in hotel-rating websites. In such a setting, users' visual preferences can be detected by connecting related review words to visual views. Take Fig. 1 as an example. The review sentence, "This place has a really good vibe," indicates that the user likes the inside environment of this restaurant. Hence, the user visual preference on the inside environment could be extracted from this restaurant's images of the corresponding view accordingly. In this article, we aim to capture this kind of view-level user visual preference and improve recommendation performance. We choose view-level rather than image/region-level since it is difficult to obtain large-scale labeled training data for capturing image/region-level preference, while prior knowledge (aspect extraction [16]) could be exploited for accurately mining view-level preference. As will be shown by experiments, view-level preference is enough for significantly boosting recommendation performance.

In this article, we propose a users' multimodal preferences-based recommendation (UMPR) method. As shown in Fig. 2, the review network and the visual network aim to extract the multimodal matching between a user U and an item I. Specifically, the review network extracts the *textual matching*, which includes relevance matching patterns and their related sentiment evidence, between U and I from their historical reviews. The visual network extracts I's view-level visual embeddings by convolutional neural networks and matches them with U's visual preference embeddings to obtain the *visual matching*. The textual and visual matching features are then combined to predict the overall rating. To accurately capture U's positive/negative visual preference embeddings, we propose a control network to examine each training pair (U, I') and extract local visual preference from the review written by U for I' (called matching review). U's visual preference embeddings are then trained according to the extracted local visual preferences. For example, in Fig. 1 the sentence "This place has a really good vibe" indicates the user likes the "inside" view of this restaurant. Hence, we let the user's positive preference embedding for "inside" to be similar to the visual embedding of "inside" for the restaurant. We also develop proper pretraining strategies for different subnetworks to encourage correct evaluation of relevance and sentiment. For a test (U, I) pair, we only extract textual/visual matching features to predict rating and produce recommendations accordingly.

The contributions of this work are as follows. 1) We propose to explicitly mine users' view-level visual preference from their reviews and show that this idea can significantly improve recommendation performance; 2) we develop a novel deep learning method, UMPR, together with proper (pre)-training strategies to implement the proposed idea. UMPR can well capture textual and visual matching features between users and items for recommendation; 3) we empirically compare UMPR with state-of-the-art baselines on publicly available Yelp and Amazon product datasets (related to catering and online shopping applications, respectively). Experiment results show that UMPR outperforms baseline methods.

II. Related Work

A. Review-Based Recommendation

Review-based recommendation methods mitigate the cold-start problem. Recently, deep learning-based methods were proposed to handle this problem. Li *et al.* [17] calculate the positive and negative sentiment matching embeddings between a user and an item via a couple of capsules and use them to predict the overall rating. However, they neglect the neutral sentiment while we extract integrated sentiment information via the sentiment net (S-Net). Chen *et al.* [18] study a novel recommendation problem called question-driven recommendation. They also fuse the relevance matching patterns and related sentiment information between a user's and an item's reviews for assessing the matching between them. The key difference between our work and [18] is that we consider multimodal user preferences and develop a novel control network for capturing users' visual preference.

B. Multimodal Contents-Based Recommendation

Recently, researchers have started to pay attention to the visual information of items in recommender systems [19]. Existing multimodal contents-based recommendation methods can be divide into two categories: *Shallow methods* [2], [13], [14] and *deep learning-based methods* [15]. Shallow methods extract shallow features such as bags-of-words [13] and color name features [2] from multimodal contents, ignoring important sequential and hierarchical information in multimodal contents that may help recommendation. Our UMPR falls into the second category. Chen *et al.* [15] send a user's and an item's precomputed embeddings, as well as the item's image into a deep neural network for rating prediction. They try to learn the user's attention (i.e., preference) on image regions by matching his precomputed embedding with image region features. The attention parameters are trained by using the attentive image features to generate the user's review. However, such a scheme is hard to capture the user's real visual preference since: 1)

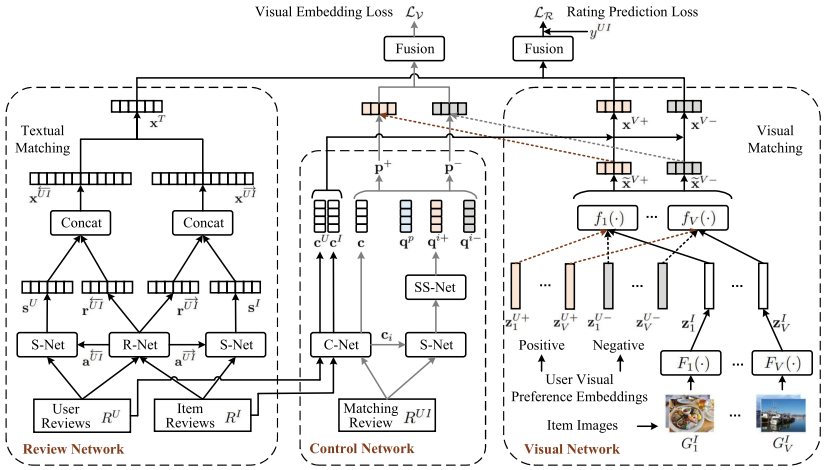


Fig. 2. Illustration of UMPR. UMPR consists of review network, visual network, and control Network. The former two capture the textual and visual matching between a user U and an item I. The control network is designed to learn the user's positive/negative visual preference embeddings based on training reviews. Finally, rating prediction is based on the textual and visual matching features.

the user's precomputed embedding is not guaranteed to capture visual preference information; 2) the unsupervised review generation training may not well capture the user's visual preference evidence in the review either. In comparison, our method try to mine a user's visual preference explicitly from his reviews by the carefully designed and pretrained control network.

III. Method

In this section, we present UMPR in detail, together with its implementation.

A. Problem Description

Given a user U and an item I, let R^{UI} be the textual review of the user U on the item I. y^{UI} is the overall rating indicating the overall satisfaction of the user U for the item I. R^U denotes U's historical reviews in the same domain as I except R^{UI}. R^I represents the reviews of the item I except R^{UI}. $G^I = G^I_1 \cup G^I_2 \cup \cdots \cup G^I_V$ is the image set of item I, where the vth subset, $G^I_v = \{g^I_{v,1}, g^I_{v,2}, \ldots, g^I_{v,K_v}\}$, contains K_v images and describes the visual information of the vth view. Given a set of training tuples $T = \{(R^{UI}, R^U, R^I, G^I, y^{UI})\}$, the goal is to develop a model that can accurately predict y^{UI} for a given (R^U, R^I, G^I).

B. Framework

The overall architecture is shown in Fig. 2. Its key components include the control network, the visual network and the review network. Based on the observations that 1) textual reviews provide comprehensive information that reveals users' textual preferences and items' textual characteristics; 2) items' images provide important complementary information in modeling users' visual preferences and items' visual characteristics, we deem that the overall satisfaction of a user U toward an item I (the overall rating y^{UI}) depends on textual matching and visual matching between U and I. The review network extracts the textual matching between R^U and R^I, while the visual network assesses visual matching by considering G^I and U's visual preference embeddings. The control network captures the cross-modal correlation in multimodal contents and guides the learning of user visual preferences. For example, with "This place has a really good vibe," the U's positive visual preference embedding for view "inside" is forced to be similar to $G^I_{v='inside'}$ by the control network. Details regarding each component will be elaborated as below.

1) Review Network: Intuitively, R^U represents U's textual preference and R^I contains I's textual pros and cons. Following [18], we use the relevance net (R-Net) and sentiment net (S-Net) to capture the relevance matching patterns and their related sentiment evidence between R^U and R^I, respectively.

R-Net: We treat R^U and R^I as word sequences and embed each word by pretrained Glove embeddings [20] to obtain the original word embeddings, i.e., $R^U = \{\mathbf{w}^U_t\}^n_{t=1}$, $R^I = \{\mathbf{w}^I_t\}^m_{t=1}$. Then we apply the Bi-GRU [18] to capture sequential information in word sequences. The computation of the Bi-GRU for user reviews is

$$\overrightarrow{\mathbf{h}^U_t} = \overrightarrow{GRU}(\overrightarrow{\mathbf{h}^U_{t-1}}, \mathbf{w}^U_t), \quad \overrightarrow{\mathbf{h}^U_t} \in \mathbb{R}^u \tag{1}$$

$$\overleftarrow{\mathbf{h}^U_t} = \overleftarrow{GRU}(\overleftarrow{\mathbf{h}^U_{t+1}}, \mathbf{w}^U_t), \quad \overleftarrow{\mathbf{h}^U_t} \in \mathbb{R}^u \tag{2}$$

where $\overrightarrow{\mathbf{h}^U_t}$ and $\overleftarrow{\mathbf{h}^U_t}$ represent U's forward and backward hidden states of Bi-GRU, respectively. By concatenating $\overrightarrow{\mathbf{h}^U_t}$ and $\overleftarrow{\mathbf{h}^U_t}$, we obtain the complete hidden state $\mathbf{h}^U_t \in \mathbb{R}^{2u}$ as the embedding of the tth word. We use the complete hidden states of all the n words as column vectors to construct U's review matrix, i.e., $\mathbf{H}^U \in \mathbb{R}^{2u \times n}$. The item I's review matrix, $\mathbf{H}^I \in \mathbb{R}^{2u \times m}$, is obtained similarly. Then we extract the matching patterns between R^U and R^I by the coattention mechanism, which has been shown to be able to capture matching patterns between text pairs [18]. It has three steps: First, the affinity matrix is computed as follows:

$$\mathbf{A} = \tanh((\mathbf{H}^I)^\mathsf{T} \mathbf{M}^R \mathbf{H}^U), \quad \mathbf{A} \in (-1, 1)^{m \times n} \tag{3}$$

where $\mathbf{M}^R \in \mathbb{R}^{2u \times 2u}$ is a parameter matrix. The (i, j)th element of $\mathbf{A}$ reflects the similarity between the ith word of R^I and the jth word of R^U. Second, we use row-wise and column-wise maximization on $\mathbf{A}$ followed by a softmax function to generate relevance vectors $\mathbf{a}^{\overrightarrow{UI}}$ and $\mathbf{a}^{\overleftarrow{UI}}$

$$\mathbf{a}^{\overrightarrow{UI}} = \text{softmax}(\text{RowMax}(\mathbf{A})), \quad \mathbf{a}^{\overrightarrow{UI}} \in (0,1)^m$$
$$\mathbf{a}^{\overleftarrow{UI}} = \text{softmax}(\text{ColMax}(\mathbf{A})), \quad \mathbf{a}^{\overleftarrow{UI}} \in (0,1)^n. \tag{4}$$

The tth element in $\mathbf{a}^{\overrightarrow{UI}}$ ($\mathbf{a}^{\overleftarrow{UI}}$) reflects the overall relevance intensity of $\mathbf{w}_t^I$ ($\mathbf{w}_t^U$) to R^U (R^I). Last, the final relevance embeddings of R^U and R^I are calculated by performing attention aggregation on review matrices

$$\mathbf{r}^{\overleftarrow{UI}} = \mathbf{H}^U \mathbf{a}^{\overleftarrow{UI}}, \mathbf{r}^{\overrightarrow{UI}} = \mathbf{H}^I \mathbf{a}^{\overrightarrow{UI}}, \quad \mathbf{r}^{\overleftarrow{UI}}, \quad \mathbf{r}^{\overrightarrow{UI}} \in \mathbb{R}^{2u}.$$

S-Net: The sentiment evidence related to relevance matching patterns also plays an important role in textual matching. For instance, if reviews of U and I are correlated on the price aspect, we tend to recommend I to U if many reviews said I has a good price (if, according to R^U, U is harsh on price in the domain of I, we would pay more attention on the price aspect when generating recommendations). Considering the sentiment of different sentences are diverse, S-Net first extracts sentence-level sentiment. We partition U's review matrix $\mathbf{H}^U \in \mathbb{R}^{2u \times n}$ to obtain the sentence-level review matrices, $\{\mathbf{H}_1^U, \ldots, \mathbf{H}_{d^U}^U\}$, where $\mathbf{H}_i^U \in \mathbb{R}^{2u \times n_i}$ denotes the embedding of the ith sentence, d^U denotes the sentence number of R^U and n_i is the word number of the ith sentence. We feed each $\mathbf{H}_i^U$ into a self-attention module [18] to generate its sentiment attention vector $\mathbf{a}_i^U$:

$$\mathbf{a}_i^U = \text{softmax}((\mathbf{w}^s)^\mathsf{T} \tanh(\mathbf{M}^s \mathbf{H}_i^U)), \quad \mathbf{a}_i^U \in (0,1)^{n_i} \tag{5}$$

where $\mathbf{M}^s \in \mathbb{R}^{u_s \times 2u}$, $\mathbf{w}^s \in \mathbb{R}^{u_s}$ are parameters of the self-attention module with hyperparameter u_s. $\mathbf{a}_i^U$ reflects the sentiment intensity of the words in the ith sentence. Next, we perform sentence-level aggregation to obtain sentence-level sentiment embeddings: $\{\mathbf{s}_i^U = \mathbf{H}_i^U \mathbf{a}_i^U\}_{i=1}^{d^U}$. Last, we aggregate all sentence-level sentiment embeddings to obtain the review-level sentiment embedding. Since the extracted sentiment evidence should be related to relevance matching, we fuse the relevance information to calculate the final sentiment embedding of R^U

$$\mathbf{s}^U = \sum_{i=1}^{d^U} a_i^{\overleftarrow{UI}} \mathbf{s}_i^U, \quad \mathbf{s}^U \in \mathbb{R}^{2u} \tag{6}$$

where $a_i^{\overleftarrow{UI}}$ is the relevance of the ith sentence obtained by summing the elements about this sentence in the relevance vector, $\mathbf{a}^{\overleftarrow{UI}}$. In this way, $\mathbf{s}^U$ is forced to concentrate on sentences with salient relevance scores. We use the same S-Net to calculate $\mathbf{s}^I$, the sentiment embedding of R^I.

Textual Matching: We concatenate the final relevance and sentiment embeddings of U to obtain its textual embedding

$$\mathbf{x}^{\overleftarrow{UI}} = [\mathbf{r}^{\overleftarrow{UI}} \oplus \mathbf{s}^U], \quad \mathbf{x}^{\overleftarrow{UI}} \in \mathbb{R}^{4u} \tag{7}$$

where $\oplus$ represents the concatenation operation. The textual embedding of I, $\mathbf{x}^{\overrightarrow{UI}}$, is constructed similarly. $\mathbf{x}^{\overleftarrow{UI}}$ and $\mathbf{x}^{\overrightarrow{UI}}$ represent U's textual preference and the related textual pros/cons of I, respectively. We fuse them to capture the textual matching information between R^U and R^I

$$\mathbf{x}^T = \tanh(\mathbf{W}^{\overleftarrow{UI}} \mathbf{x}^{\overleftarrow{UI}} + \mathbf{W}^{\overrightarrow{UI}} \mathbf{x}^{\overrightarrow{UI}}), \quad \mathbf{x}^T \in \mathbb{R}^{2u} \tag{8}$$

where $\mathbf{W}^{\overleftarrow{UI}}$ and $\mathbf{W}^{\overrightarrow{UI}}$ are $2u \times 4u$ parameter matrices.

2) Visual Network: As aforementioned, items' images could convey user preference that may not be well captured by language. Therefore, we propose the visual network to extract an item's visual embeddings and match them with a user's visual preference embeddings to obtain the visual matching.

We extract an item's visual embeddings from its image set, G^I. For each view v, we calculate the high-level representation of each image $g_{v,k}^I \in G_v^I$ as follows:

$$\mathbf{z}_{v,k}^I = F_v(g_{v,k}^I), \mathbf{z}_{v,k}^I \in \mathbb{R}^{u_c} \tag{9}$$

where $F_v(\cdot)$ is the CNN network for the vth view and initialized by the VGG-16 network [21]. Then we conduct within-view average pooling to obtain I's visual embeddings $\{\mathbf{z}_v^I\}_{v=1}^V$, where

$$\mathbf{z}_v^I = \frac{1}{K_v} \sum_{k=1}^{K_v} \mathbf{z}_{v,k}^I, \mathbf{z}_v^I \in \mathbb{R}^{u_c}. \tag{10}$$

We define $\{\mathbf{z}_v^{U+}/\mathbf{z}_v^{U-}\}_{v=1}^V$ as U's positive/negative visual preference embeddings on all the V views, representing U's view-level visual preference. Intuitively, U's visual preference could be collected from images of the items commented by him. In the next subsection, we will present the control network to guide the learning of $\{\mathbf{z}_v^{U+}/\mathbf{z}_v^{U-}\}_{v=1}^V$.

By the following calculation, we can obtain the preliminary visual matching vectors $\widetilde{\mathbf{x}}^{V+}$ and $\widetilde{\mathbf{x}}^{V-}$:

$$\widetilde{x}_v^{V+} = \tanh(\left\| f_v(\mathbf{z}_v^{U+}) - f_v(\mathbf{z}_v^I) \right\|_2)$$
$$\widetilde{x}_v^{V-} = \tanh(\left\| f_v(\mathbf{z}_v^{U-}) - f_v(\mathbf{z}_v^I) \right\|_2) \tag{11}$$

where $\tanh(\cdot)$ transforms the preliminary visual matching into the [0,1) interval. $f_v(\cdot)$ is a fully connected neural network to encode the visual embeddings of the vth view into a matching space. $\widetilde{x}_v^{V+}/\widetilde{x}_v^{V-}$ is the vth element of the positive/negative preliminary visual matching vector $\widetilde{\mathbf{x}}^{V+}/\widetilde{\mathbf{x}}^{V-} \in [0,1)^V$.

However, the visual (preference) embeddings described above ignore the diverse importance of visual views w.r.t users and items. For example, a user may frequently comment on the "inside" environment of restaurants, indicating the user is more concerned with the "inside" view; a restaurant could also receive more comments on some views, reflecting its popular pros/cons which users care more about. In view of these observations, we define *concern intensity* to measure the importance of a visual view w.r.t a user/an item. Let $\mathbf{c}^U, \mathbf{c}^I$, respectively, be the concern intensity vectors for user U and item I. Then the final visual matching between U and I is computed as

$$\mathbf{x}^{V+} = \mathbf{c}^U \odot \mathbf{c}^I \odot (\mathbf{1} - \widetilde{\mathbf{x}}^{V+})$$
$$\mathbf{x}^{V-} = \mathbf{c}^U \odot \mathbf{c}^I \odot (\mathbf{1} - \widetilde{\mathbf{x}}^{V-}) \tag{12}$$

where $\mathbf{1}$ denotes all-1 vector, $\odot$ represents element-wise multiplication. The vth element of $\mathbf{x}^{V+}$ ($\mathbf{x}^{V-}$) encodes the weighted positive (negative) matching degree of U on I's vth view. $\mathbf{c}^U$ and

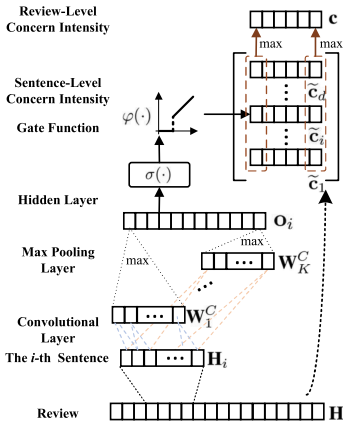


Fig. 3. Illustration of the Concern Net (C-Net).

$\mathbf{c}^I$ are obtained by the control network which will be presented in the next subsection.

3) Control Network: Based on the training (U, I) pairs, we could extract users' visual preference information from each matching review R^{UI}. The intuition is that, if U shows positive attitude in R^{UI} toward I's visual view v, we should encourage $\mathbf{z}_v^{U+}$ to be near to $\mathbf{z}_v^I$. The same idea holds for $\mathbf{z}_v^{U-}$ when the attitude is negative. The control network takes each R^{UI} as input and extract *local visual preference* of U on each I's visual view. The local visual preference consists of three measures: *Local concern intensity* (i.e., how intensely does U discuss about I's view v), *sentiment polarity,* and *sentiment intensity*. The former measure is obtained by the concern net (C-Net) and the latter two by the sentiment scoring net (SS-Net). In this subsection, we just focus on U's local preference in R^{UI} and hence omit the superscript (UI) temporally for clarity.

Concern Net (*C-Net*): The architecture of C-Net is shown in Fig. 3. We use the same low-level structure (with separate parameters) as in R-Net to obtain the review matrix $\mathbf{H} \in \mathbb{R}^{2u \times l}$ of R^{UI}. Since different sentences usually describe different views, the local concern should be first estimated at the sentence-level. Hence we partition $\mathbf{H}$ to obtain the sentence-level review matrices $\{\mathbf{H}_1, \ldots, \mathbf{H}_d\}$, where d denotes the sentence number in R^{UI}. We employ the CNN network to capture the regional features related to visual views. As shown in Fig. 3, the convolutional layer applies K filters on sentences. Each filter $\mathbf{W}_k^C \in \mathbb{R}^{2u \times h}$ is applied to a window of h words to produce a regional feature value

$$o_{i,k}[t] = \phi(\mathbf{H}_{i,t:t+h-1} \otimes \mathbf{W}_k^C + b_k) \tag{13}$$

where $\mathbf{H}_{i,t:t+h-1}$ represents $[\mathbf{h}_{i,t} \quad \mathbf{h}_{i,t+1} \quad \cdots \quad \mathbf{h}_{i,t+h-1}]$, $\mathbf{h}_{i,t}$ is the embedding of the tth word in sentence $\mathbf{H}_i$, $\otimes$ denotes convolution, b_k is the bias of the current filter, $\phi(\cdot)$ denotes the rectified linear unit (ReLU) activation function, $o_{i,k}[t]$ is the computed regional feature at position t. Computing $o_{i,k}[t]$ at all possible positions in sentence $\mathbf{H}_i$ yields a feature map $[o_{i,k}[1] \quad o_{i,k}[2] \quad \cdots \quad o_{i,k}[l_i - h + 1]]^\mathsf{T}$, where l_i is the word number of the ith sentence. Then we perform the max pooling to generate a fixed-length embedding of sentence i: $\mathbf{o}_i = [o_{i,1} o_{i,2} \cdots o_{i,K}]^\mathsf{T}$, where $o_{i,k} = \max\{o_{i,k}[1], \ldots, o_{i,k}[l_i - h + 1]\}$. Since the convolutional filters try to detect regional features related to visual views, $\mathbf{o}_i$ can be regarded as extracting the most salient features from the sentence. Next, $\mathbf{o}_i$ is projected by a nonlinear layer to generate the preliminary local concern intensity

$$\widetilde{\mathbf{c}}_i^s = \sigma(\mathbf{W}^o \mathbf{o}_i + \mathbf{b}^o), \widetilde{\mathbf{c}}_i^s \in (0,1)^V \tag{14}$$

where $\mathbf{W}^o \in \mathbb{R}^{V \times K}$ and $\mathbf{b}^o \in \mathbb{R}^V$ are parameters, and $\sigma(\cdot)$ denotes the sigmoid function. Each element of $\widetilde{\mathbf{c}}_i^s$ represents the probability that the sentence $\mathbf{H}_i$ concerns the corresponding view. It is possible that a sentence concerns multiple views.

However, review sentences may not all be related to visual views, since visual views usually cover only a few item aspects. Therefore, we further filter $\widetilde{\mathbf{c}}_i^s$ with a predefined threshold, ω, using the following gate function:

$$\varphi(x) = \begin{cases} x \text{ if } x \geq \omega \\ 0 \text{ otherwise.} \end{cases} \tag{15}$$

The final sentence-level local concern intensities are obtained via putting $\widetilde{\mathbf{c}}_i^s$ into the gate function: $\mathbf{c}_i^s = \varphi(\widetilde{\mathbf{c}}_i^s)$. The v-th element in $\mathbf{c}_i^s$ (i.e., $c_{i,v}^s$) denotes U's local concern intensity for the vth visual view according to the ith sentence.

Note that when considering the whole review R^{UI}, we take more care of the frequently concerned visual views. Hence, the sum of square operation is performed to fuse the sentence-level local concern intensities and put more emphasis on sentences with high confidence

$$c_v = \sum_{i=1}^{d} {c_{i,v}^s}^2 \tag{16}$$

where c_v is the vth element of the review-level local concern intensity $\mathbf{c}$.

Recall that in Section III-B2, we need to obtain the global concern intensities of U and I w.r.t visual views [$\mathbf{c}^U$ and $\mathbf{c}^I$ in (12)]. They can be obtained by simply feeding R^U (R^I) into C-Net.

To encourage C-Net to capture regional features related to visual views, we propose a pretraining method based on view-specific keywords. We will elaborate it in Section III-D.

SS-Net: SS-Net is intrinsically a fully connected neural network and takes $\{\mathbf{s}_i\}_{i=1}^d$, the sentence-level sentiment embeddings of R^{UI} generated by S-Net, as input. We feed each $\mathbf{s}_i$ through SS-Net to capture its sentiment score

$$\widetilde{q}_i^s = \sigma\left(f^{ss}(\mathbf{s}_i)\right) \tag{17}$$

where $f^{ss}(\cdot)$ represents the transformation of SS-Net. By applying the sigmoid function, the sentiment score $\widetilde{q}_i^s$ is forced to be in (0,1), where 0 and 1 represent the most intense values for negative and positive sentiment, respectively. To facilitate accurately capturing the sentiment polarity and intensity, we pretrain SS-Net by sentences with the most intense positive/negative sentiment, which will be elaborated in Section III-D.

After obtaining sentence-level sentiment scores for R^{UI}, the next question is how to assess review-level sentiment polarity and intensity for each visual view, so that they can be used cooperatively with the local concern intensity for user visual

preference training. The idea is to give high weights to highly concerned sentences when averaging sentence-level sentiment scores for a view

$$q_v = \left(\sum_{i=1}^{d} \widetilde{q}_i^s {c_{i,v}^s}^2\right) \Big/ \left(\sum_{i=1}^{d} {c_{i,v}^s}^2\right) \tag{18}$$

where q_v denotes U's overall sentiment score on I's vth view.[1] We could acquire a high (low) q_v when most highly concerned sentences confirm U enjoys (disrelishes) I's view v. On the contrary, if U expresses opposite opinions in the concerned sentences of R^{UI}, the overall sentiment score would be near 0.5 which indicates a neutral sentiment. Therefore we use 0.5 as the threshold for polarity judgement. Regarding sentiment intensity, we further adjust q_v to suppress ambiguous scores

$$\begin{cases} \text{if } q_v > 0.5, \text{then}, q_v^p = 1, q_v^{i+} = 4(q_v - 0.5)^2, q_v^{i-} = 0 \\ \text{if } q_v < 0.5, \text{then}, q_v^p = 0, q_v^{i+} = 0, q_v^{i-} = 4(0.5 - q_v)^2. \end{cases}$$

By grouping the results for all the V views, we get the sentiment polarity vector $\mathbf{q}^p \in \{0,1\}^V$, the positive/negative sentiment intensity vectors $\mathbf{q}^{i+}/\mathbf{q}^{i-} \in (0,1)^V$ of user U for item I. Finally we calculate the local visual preference of U by synthesizing $\mathbf{c}$, $\mathbf{q}^p$, $\mathbf{q}^{i+}$, and $\mathbf{q}^{i-}$

$$\mathbf{p}^+ = \mathbf{c} \odot \mathbf{q}^p \odot \mathbf{q}^{i+}, \mathbf{p}^- = \mathbf{c} \odot (\mathbf{1} - \mathbf{q}^p) \odot \mathbf{q}^{i-} \tag{19}$$

where higher concern and sentiment intensities lead to a higher preference weight.

C. Loss Function

We train UMPR to predict the overall rating y^{UI} and also explicitly learn users' visual preference embeddings. We now describe different components of our loss function.

1) Visual Embedding Loss: We construct the visual embedding loss to guide the learning of users' visual preference embeddings

$$\mathcal{L}_{\mathcal{V}} = \sum \left((\mathbf{p}^+)^{\mathsf{T}} \widetilde{\mathbf{x}}^{V+} + (\mathbf{p}^-)^{\mathsf{T}} \widetilde{\mathbf{x}}^{V-}\right) \tag{20}$$

where the summation is over all the training (U, I) pairs. This loss forces user visual preference to be similar with item visual embeddings if the corresponding estimated local visual preference $\mathbf{p}^+/\mathbf{p}^-$ is high.

2) Rating Prediction Loss: We fuse the textual matching and the visual matching to predict the overall rating indicating the overall satisfaction of U to I

$$\hat{y}^{UI} = f^p(\mathbf{x}^T, \mathbf{x}^{V+}, \mathbf{x}^{V-}). \tag{21}$$

Then we use the mean squared error (MSE) to construct the rating prediction loss:

$$\mathcal{L}_{\mathcal{R}} = \sum_{U,I} (y^{UI} - \hat{y}^{UI})^2. \tag{22}$$

3) Joint Loss: By synthesizing the visual embedding loss and rating prediction loss, the overall optimization problem of UMPR is formulated as

$$\min_{\Theta, \{\mathbf{z}_v^{U+}, \mathbf{z}_v^{U-}\}_{\forall v, U}} \mathcal{L} = \mathcal{L}_{\mathcal{R}} + \epsilon \mathcal{L}_{\mathcal{V}} \tag{23}$$

where Θ denotes all the model parameters, ϵ is a hyperparameter.

D. Training Strategies

Since the performance of neural networks can be rather sensitive to how the parameters are initialized [22], we develop a transfer training strategy: Pretraining important subnets and then fine-tuning the whole network by the joint loss. We follow the strategies in [18] to pretrain R-Net and S-Net. Next we describe the pretraining of C-Net, SS-Net, and fine-tuning.

Pretraining for C-Net: We employ view keywords as prior knowledge to pretrain C-Net. First, we perform the state-of-the-art aspect extraction toolkit [16] to construct a keyword lexicon for each view. Then, the target local concern intensity on each view ($c_{i,v}^s$) is obtained by the following criterion: Review sentences with/without overlapping view keywords are deemed to be concerned/unconcerned instances with $c_{i,v}^s = 1/0$. Finally, we use an MSE loss on the target intensities to train C-Net.

Pretraining for SS-Net: We treat review sentences with rating 5/1 as intense positive/negative instances. The weak label of $\widetilde{q}_i^s$ is set to 1/0 accordingly. We treat the pretraining of SS-Net as a binary classification task.

Fine-tuning: We initialize the parameters of UMPR by the pretrained R-Net, S-Net, C-Net, and SS-Net. The other parameters are initialized randomly. Then we minimize the joint loss ($\mathcal{L}$) to fine-tune the parameters. In the test stage, for a test pair (U, I), there is no matching review (R^{UI}) between them. Therefore, we only retain C-Net in the control network to assess U's and I's global concern intensities.

IV. Experiments

We evaluate the proposed UMPR against several state-of-the-art baseline methods on publicly available Yelp and Amazon datasets.

A. Dataset

Yelp[2] is a restaurant recommendation dataset. It contains rating and review, as well as 2 00 000 restaurant images labeled to four categories (views): 'Food," "drink," "inside," and "outside". We first select restaurants with at least one image in each view and then adopt 5-cores setting [14] to remove the restaurants and users with less than 5 reviews.

Amazon-5-cores [23] contains the user-generated contents (review, rating, etc.) and the item's metadata (price, image, etc.) from Amazon. We utilize the review, rating, and image in our experiments. Amazon-5-cores dataset contains 24 product categories. We evaluate our model on the *Clothing, Shoes and Jewelry* category in which visual information is significant [15], [19]. Each item is accompanied with an image in this dataset.

[1] If $c_v = 0$, we do not need to compute q_v and set $q_v = 0.5$ for completeness.

[2] [Online]. Available: https://www.yelp.com/dataset

TABLE I
DATASET SUMMARY

Datasets	# users	# items	# ratings	# images per item
Yelp	23141	1762	274437	30.06
Amazon-5-cores	39389	23033	278677	1

We can fairly compare UMPR with the methods that can only be applied to single item image case [15], [19]. We degenerate UMPR by: 1) Extracting an item's visual embedding from its single image and define a user's single-view positive/negative visual preference embeddings; 2) only retaining the S-Net and SS-Net in the control network to calculate the sentiment polarity in the matching review. The degenerate UMPR still learns multimodal matching for personalized recommendation.

Important statistics are summarized in Table I.

B. Evaluation Methodology

The baseline methods contain matrix factorization (MF) [7] which is the most popular shallow recommendation method. It factorizes rating matrices to obtain users' and items' latent embeddings and matches them for recommendation. Neural matrix factorization (NeuMF) [8] employs a neural network to capture hierarchical interactions between users' and items' latent embeddings. Dual attention mutual leaning (DAML) [11] is the state-of-the-art review-based recommendation method. DAML extracts the latent embeddings of users and items by CNNs and highlights their matching features by coattention. The final embeddings are used for rating prediction. Visual Bayesian personalized ranking (VBPR) [19] extracts an item's visual embedding and matches it with a user's visual preference embedding to calculate the visual matching. Then VBPR infuses visual matching into a collaborative filtering framework. Multiview visual Bayesian personalized ranking (MVBPR) [24] is a multiview extension of VBPR. It extracts an items's multiview visual embeddings by CNNs and matches it with a user's visual preference embeddings for rating prediction. Visually explainable collaborative filtering (VECF) [15] is the state-of-the-art multimodal contents-based deep recommendation method. Since VECF and VBPR can be only applied to single item image case, for Yelp dataset, we randomly select one image for each item. Multimodal aspect-aware latent factor model (MMALFM) [14] is a multimodal contents-based shallow recommendation method. The reviews and images are used to estimate the importance of the aspects in users' and items' latent embeddings. For easy understanding, we summarize the similarities and differences of all the methods in Table II.

Each dataset is randomly divided into training set (90%), validation set (5%), and test set (5%). By the reason that our UMPR needs to train user's visual preference embeddings, we ensure the training set contains all users. All the hyperparameters of UMPR and baselines are selected based on the performance on the validation set. Following related works [11], [14], we use the widely adopted MSE as evaluation metric. The averaged performance is reported by running each test case five times.

TABLE II
SUMMARY OF THE METHODS. ✓ DENOTES THE CORRESPONDING INFORMATION IS USED

Method	Reference	Rating	Review	Image	Depth
MF	[7]	✓	✗	✗	Shallow method
NeuMF	[8]	✓	✗	✗	Deep method
DAML	[11]	✓	✓	✗	Deep method
VBPR	[19]	✓	✗	✓	Deep method
MVBPR	[24]	✓	✗	✓	Deep method
VECF	[15]	✓	✓	✓	Deep method
MMALFM	[14]	✓	✓	✓	Shallow method
UMPR	-	✓	✓	✓	Deep method

C. Implementation Details

We use stochastic gradient decent (SGD) and apply the Adam [25] for training. The learning rate is set as 1e-6. The first and second momentum coefficients are set to 0.9 and 0.999, respectively. The minibatch size for SGD is set to 32. The length of the hidden state vectors of GRUs (u) and the dimension of the self-attention space (u_s) are set to 64. The CNN in the visual network is initialized by VGG-16 [21]. In our experiments, we take the output of the second fully connected layer (i.e., FC-7), to obtain the $u_c = 4096$ dimensional visual embeddings. $\{f_v(\cdot)\}_{v=1}^{V}$ are two-layer fully connected neural networks activated by ReLU function to encode the visual embeddings into the matching space. Layer sizes are set as 4096-4096-1000. The CNN in the C-Net has $K = 120$ filters with 40 filters in each window size, i.e., $h = 1, 2, 3$. $f^{ss}(\cdot)$ is also a two-layer fully connected neural network with layer size 128-64-1 and activated by ReLU function. It calculates the sentiment scores from the sentence-level sentiment embeddings. The hyperparameters ϵ and ω are set as 10^{-1} and 0.35, respectively. Our model is implemented by TensorFlow 1.15 and runs on Ubuntu Linux 16.04.

D. Performance Comparison

Table III shows the experiment results. We can obtain the following points from the experiment results: 1) The methods based on only the rating information (MF, NeuMF) perform the worst, which verifies the effectiveness of users' reviews and items' images in recommendation. 2) DAML outperforms all the other baselines, including multimodal content-based methods on the Yelp dataset. The reasons are: a) DAML captures the textual matching while VECF only uses review data in model training; b) DAML employs CNNs to better extract local contextual information. MMALFM treats reviews as combination of words and ignores the correlation in those words. 3) VECF outperforms DAML on the Amazon-5-cores dataset. The reason for this could be that the items' images are crucial in selecting clothing, shoes, and jewelry. 4) We test the significance of performance difference between UMPR and baselines by t-test with significance level 0.05. Results show that UMPR significantly outperforms all the baselines on all datasets. Compared with DAML, UMPR captures a user's visual preference which provides significant complementary information to his textual preference. In Section IV-E, we perform ablation studies to evaluate the effects of the visual network and the review network.

TABLE III
PERFORMANCE COMPARISON ON YELP AND AMAZON-5-CORES DATASETS FOR ALL METHODS. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED BY **BOLDFACE** AND UNDERLINED, RESPECTIVELY. △% DENOTES THE PERFORMANCE IMPROVEMENT OF UMPR OVER THE BEST BASELINE

Datasets	MF	NeuMF	DAML	VBPR	MVBPR	VECF	MMALFM	UMPR	△%
Yelp	1.866	1.735	1.502	1.729	1.652	1.694	1.547	**1.431**	4.72
Amazon-5-cores	1.524	1.437	1.335	1.368	1.362	1.316	1.357	**1.266**	3.80
Average on all datasets	1.695	1.586	1.419	1.549	1.507	1.520	1.452	**1.349**	4.93

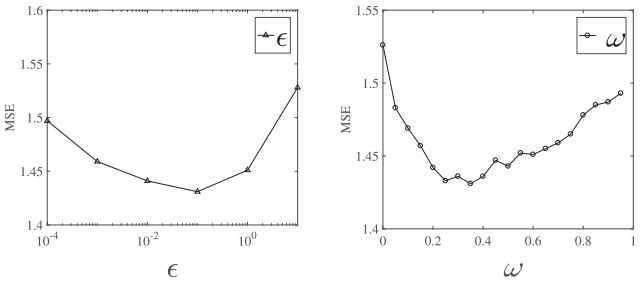


Fig. 4. Parameter analysis on the Yelp dataset. (a) Varying ϵ when $\omega = 0.35$. (b) Varying ω when $\epsilon = 0.1$.

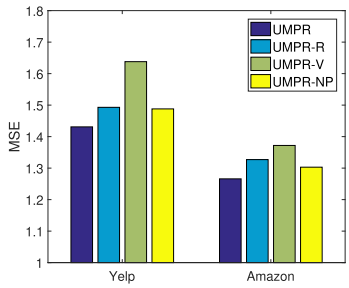


Fig. 5. Ablation studies on Yelp and Amazon-5-cores datasets.

E. Analysis

Parameter Analysis: UMPR contains two important hyperparameters, ϵ and ω. ϵ controls the importance of visual embedding loss. ω is the threshold of the gate function in C-Net. It determines whether a sentence concerns a visual view. Here we explore their impact on the Yelp dataset. We vary one hyperparameter and fix another one. We turn hyperparameters by looking at the performance on the validation set. Fig. 4 shows the results. We find a general pattern: The MSE curves first go down and then go up when increasing the hyperparameters, which proves the validities of the visual embedding loss and the gate function in C-Net. We finally set ϵ and ω as 10^{-1} and 0.35, respectively.

Ablation Studies: To prove the importance of the textual and visual matching in recommendation, we propose two variations of UMPR, UMPR-R, and UMPR-V which only utilize review and image, respectively. UMPR-R only performs the review network to obtain the textual matching for recommendation. Similar to UMPR, UMPR-V learns visual matching and users' visual preference embeddings via the visual network. Besides, to verify the effectiveness of pretraining, we propose UMPR-NP which removes the pretraining stage in UMPR. Fig. 5 shows the results of ablation studies. The performance of UMPR-V drops dramatically, which indicates reviews contain more integrated aspects than images.

V. CONCLUSION

In this article, we explored a novel recommendation problem that is extracting users' multimodal preferences for personalized recommendation. We proposed an UMPR method for this problem. UMPR captured the textual and visual matching between users and items via the review network and the visual network, respectively. The multimodal matching features were combined to predict the overall ratings. In addition, we designed the control network to guide learning users' visual preference embeddings. We also proposed pretraining strategies to better initialize UMPR. Experimental results on Yelp and Amazon datasets confirmed the effectiveness of UMPR.

REFERENCES

[1] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-tte: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2659–2666, Apr. 2020.
[2] W.-T. Chu and Y.-L. Tsai, "A hybrid recommendation system considering visual information for predicting favorite restaurants," *World Wide Web*, vol. 20, no. 6, pp. 1313–1331, 2017.
[3] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Inf. Syst.*, vol. 92, 2020, Art. no. 101522.
[4] T. Cai, J. Li, A. S. Mian, R. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Trans. Knowl. Data Eng.*, p. 1, 2020.
[5] N. Gutowski, T. Amghar, O. Camp, and S. Hammoudi, "A framework for context-aware service recommendation for mobile users: A focus on mobility in smart cities," *From Data Decision*, 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01566223
[6] R. Logesh, V. Subramaniyaswamy, V. Vijayakumar, X.-Z. Gao, and V. Indragandhi, "A hybrid quantum-induced swarm intelligence clustering for the urban trip recommendation in smart city," *Future Generation Comput. Syst.*, vol. 83, pp. 653–673, 2018.
[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.
[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Int. World Wide Web Conf. Committee*, 2017, pp. 173–182.
[9] B. Wu, X. He, Z. S. Sun, L. Chen, and Y. Ye, "A: An attentive translation model for next-item recommendation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1448–1459, Mar. 2020.
[10] W. Yue, Z. Wang, B. Tian, M. Pook, and X. Liu, "A hybrid model-and memory-based collaborative filtering algorithm for baseline data prediction of Friedreich's ataxia patients," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2020.2984540.
[11] D. Liu, J. Li, B. Du, J. Chang, and R. Gao, "Daml: Dual attention mutual learning between ratings and reviews for item recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery & Data Mining*, 2019, pp. 344–352.
[12] B. Tang *et al.*"Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2140–2150, Oct. 2017.

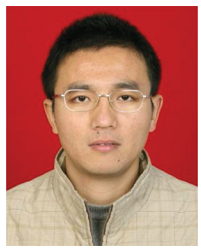
[13] S. Qian, T. Zhang, and C. Xu, "Multi-modal multi-view topic-opinion mining for social event analysis," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 2–11.
[14] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "MMALFM: Explainable recommendation by leveraging reviews and images," *ACM Trans. Inform. Syst.*, vol. 37, no. 2, 2019, Art. no. 16.
[15] X. Chen, H. Chen, H. Xu, Y. Zhang, Y. Cao, Z. Qin, and H. Zha, "Personalized fashion recommendation with visual explanations based on multimodal attention network," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 765–774.
[16] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 388–397.
[17] C. Li, C. Quan, L. Peng, Y. Qi, Y. Deng, and L. Wu, "A capsule network for recommendation and explaining what you like and dislike," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 275–284.
[18] L. Chen, Z. Guan, Q. Xu, Q. Zhang, H. Sun, G. Lu, and D. Cai, "Question-driven purchasing propensity analysis for recommendation," in *Proc. Assoc. Advancement Artif. Intell.*, 2020, pp. 35–42.
[19] R. He and J. McAuley, "Vbpr: Visual Bayesian personalized ranking from implicit feedback," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 144–150.
[20] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[22] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010.
[23] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst. Conf.*, 2013, pp. 165–172.
[24] H. Luo, X. Zhang, B. Chen, and G. Guo, "Multi-view visual Bayesian personalized ranking from implicit feedback," in *Proc. 26th Conf. User Model., Adaptation Personalization*, 2018, pp. 361–362.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Cai Xu received the B.S. and M.S. degrees in communication and information system from Xi'an University of Posts & Telecommunications, Xi'an, China, in 2014 and 2017, respectively. He is currently working toward the Ph.D. degree in computer science with the School of Computer Science and Technology, Xidian University, Xi'an.

His current research interests include machine learning, multiview learning and recommender systems.

Ziyu Guan (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2004 and 2010, respectively.

He has worked as a Research Scientist with the University of California at Santa Barbara, CA, USA, from 2010 to 2012. He is currently a Professor with the School of Information and Technology at Northwest University, Xi'an, China, and the School of Computer Science and Technology at Xidian University, Xi'an, China. His current research interests include attributed graph mining and search, machine learning, expertise modeling and retrieval, and recommender systems.



Wei Zhao received the B.S. degree in detection technology and instrumentation, the M.S. degree in signal and information processing, and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2002, 2005 and 2015, respectively.

He is currently a Professor with the School of Computer Science and Technology at Xidian University. His current research interests include pattern recognition and intelligent systems, with specific interests in attributed graph mining and search, machine learning, signal processing, and precision guiding technology.

Quanzhou Wu received the B.S. degree in software engineering from the Software Institute at Jinlin University, Changchun, China, in 2017. He is currently working toward the postgraduate degree in computer technology with the School of Computer Science and Technology, Xidian University, Xi'an, China.

His current research interests include deep learning, machine learning, and knowledge graph.

Meng Yan received the B.S degree in information security from Guangxi University, Nanning, China, in 2019. She is currently working toward the postgraduate degree in software engineering with the School of Computer Science and Technology, Xidian University, Xi'an, China.

Her current research interests include recommender systems and deep learning.

Long Chen received the B.S. degree in electronic information engineering from Xi'an Jiaotong University City College, Xi'an, China, in 2012, and the M.S. degree in electronics and communications engineering and the Ph.D. degree in computer application technology from Northwest University, Xi'an, in 2015 and 2019, respectively.

He is currently a Lecturer with the School of Communications and Information Engineering and School of Artificial Intelligence at Xi'an University of Posts & Telecommunications, Xi'an. His current research interests include deep learning, sentiment analysis, question answering, and natural language processing.

Qiguang Miao received the doctoral degree in computer application technology from Xidian University, Xi'an, China, in 2005.

He is a Professor and Ph.D. Student Supervisor with School of Computer Science and Technology in Xidian University, Xi'an. In recent years, he has authored or coauthored over 100 papers in the significant domestic and international journals or conferences. His current research interests include machine learning, intelligent image processing and malware behavior analysis and understanding.