

User Diverse Preference Modeling by Multimodal Attentive Metric Learning

Fan Liu[†], Zhiyong Cheng[§], Changchang Sun[†], Yinglong Wang[§], Liqiang Nie[†], Mohan Kankanhalli[‡]

[†]School of Computer Science and Technology, Shandong University

[§]Shandong Computer Science Center (National Supercomputer Center in Jinan),

Qilu University of Technology (Shandong Academy of Sciences)

[‡]School of Computing, National University of Singapore

{liufancs,jason.zy.cheng,nieliqiang}@gmail.com,mohan@comp.nus.edu.sg

ABSTRACT

Most existing recommender systems represent a user's preference with a feature vector, which is assumed to be fixed when predicting this user's preferences for different items. However, the same vector cannot accurately capture a user's varying preferences on all items, especially when considering the diverse characteristics of various items. To tackle this problem, in this paper, we propose a novel Multimodal Attentive Metric Learning (MAML) method to model user diverse preferences for various items. In particular, for each user-item pair, we propose an attention neural network, which exploits the item's multimodal features to estimate the user's special attention to different aspects of this item. The obtained attention is then integrated into a metric-based learning method to predict the user preference on this item. The advantage of metric learning is that it can naturally overcome the problem of dot product similarity, which is adopted by matrix factorization (MF) based recommendation models but does not satisfy the triangle inequality property. In addition, it is worth mentioning that the attention mechanism cannot only help model user's diverse preferences towards different items, but also overcome the geometrically restrictive problem caused by collaborative metric learning. Extensive experiments on large-scale real-world datasets show that our model can substantially outperform the state-of-the-art baselines, demonstrating the potential of modeling user diverse preference for recommendation.

KEYWORDS

Personalized Recommendation; Metric Learning; Attention Mechanism; Multimodal Information

ACM Reference Format:

Fan Liu[†], Zhiyong Cheng[§], Changchang Sun[†], Yinglong Wang[§], Liqiang Nie[†], Mohan Kankanhalli[‡]. 2019. User Diverse Preference Modeling by Multimodal Attentive Metric Learning. In *Proceedings of the 27th ACM*

Corresponding author: Zhiyong Cheng and Liqiang Nie.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '19, October 21–25, 2019, Nice, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6889-6/19/10...\$15.00

<https://doi.org/10.1145/3343031.3350953>

International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3343031.3350953>

1 INTRODUCTION

Recommendation has become one of the most important techniques for various online platforms, such as E-commerce (e.g., Amazon¹), streaming service (e.g., Youtube²), and social platforms (e.g., Pinterest³). Due to its importance, many recommendation techniques have been developed so far. Among them, matrix factorization (MF) has achieved great success [2, 27]. Given the user-item interaction matrix, this method maps all users and items into a latent feature space, in which each user and item is presented as a feature vector. The preference of a user (u) for an item (i) is predicted based on the dot product of the feature vectors, i.e., $\mathbf{p}_u^T \mathbf{q}_i$. Many recommendation methods have been proposed based on this idea, like WRMF [21], BPR [35], and PMF [33]. In recent years, the matrix factorization for recommendation has been greatly advanced with the development of deep neural network (DNN) techniques. A set of DNN based matrix factorization methods has been proposed, such as the neural collaborative filtering (NCF) [19] and Deep MF [49].

The success of MF is largely attributed to its simple and effective idea of exploiting the user-item interaction data to mine user preferences. However, merely relying on the interaction information also leads to some drawbacks, like 1) the incapability of modeling the fine-grained user preferences at the feature-level or aspect-level; and 2) performance degradation problem when the interaction data for items or users are insufficient [11]. To alleviate those limitations, various types of side information have been incorporated in MF, such as item images [9, 24, 47], metadata [7, 8], and user reviews [11, 31, 40]. The intuition is that side information contains item cues which adds value to user preference modeling. Among the side information, user reviews contain users' opinions on different aspects of items and thus have been widely exploited to model fine-grained user preferences [5, 10–12]. Besides, item images can help capture user preferences on the visual appearance of items, which have been extensively explored in fashion recommendation [18, 32, 37]. Different types of side information are considered to be complementary to each other. With the advancement of representation learning [3, 45], several methods have been proposed

¹<https://www.amazon.com>.

²<https://www.youtube.com>.

³<https://www.pinterest.com>.

to exploit multi-source information for recommendation, such as the Collaborative Knowledge based Embedding (CKE) [51] method using structure information, textual and visual knowledge, as well as the Joint Representation Learning (JRL) method [54] exploiting ratings, reviews and item images.

Although great progress has been achieved thus far, most existing recommender systems have not well considered to characterize user varying preferences for different items. It is common that the user preference on the various aspects of items is different. For example, a user will value the “plot” more for a suspense movie, while pays more attention to the “special effects” for a super-hero movie. From this example, we can see that users do not treat aspects equally for different items, even when the items are of the same category. Most existing recommendation methods use *the same vector to represent a user’s preferences for all items*, which cannot accurately predict the diverse preferences on various items. Recently, some researchers have noticed this problem and proposed several models to tackle it. One type of methods leverages reviews to analyze user attention on different aspects of items and then integrates the results into the matrix factorization methods for recommendation, such as ALFM [11], A³NCF [10], and ANR [12]. Another type of methods varies the target item or user vector based on the vectors of those most influential items or users with respect to the target items. These recently proposed methods are based on deep neural networks, such as deep interest networks (DIN) [58], memory attention-aware recommender system (MARS) [56], and collaborative memory network (CMN) [14].

With the consideration of users’ varying preferences towards different items, the above methods have achieved better results over their baselines. They, however, still rely on the matrix factorization method with dot product for similarity prediction. An intrinsic problem of the dot product is that it is not a metric-based distance learning, as it does not satisfy the triangle inequality⁴ [20, 50]. This problem limits the expressiveness of MF and hence causes MF-based methods incapable of capturing the fine-grained user preferences, resulting in sub-optimal performance [20, 53]. To tackle this problem, Hsieh et al. [20] proposed a metric collaborative filtering (CML) method which learns the preferences by minimizing the distance between user and item vectors (i.e., $\|p_u - q_i\|$) of positive interactions. However, directly applying of this distance metric is problematic [44]. Intuitively, this method tries to map each user and item pair with positive interaction to the same point in a low dimensional space. The problem is that each user and item has many interacted items and users, respectively. As a result, it is geometric inflexible for a large dataset since it tries to fit a user and all the interacted items into the same point. Tay et al. [41] has mathematically proved this is *geometrically restrictive* and will lead to an *ill-posed algebraic* system. They proposed to learn an adaptive relation vector r_{ui} between the interactions of a user-item pair by minimizing $\|p_u + r_{ui} - q_i\|$. Although those proposed metric-based learning approaches tackled the problem of dot product, they have not well modeled user diverse preferences for various items.

Based upon the aforementioned insights, we present a novel multimodal attentive metric learning (MAML) method by leveraging the multimodal information of items to model user diverse preferences towards different items. Our MAML model enhances the CML method by introducing a weight vector to each user-item pair. This weight vector represents the user’s attention on different aspects of the target items, and thus it is unique for each user-item pair. This is achieved by a designed attention neural network which analyzes user attention on the target item by exploiting the textual and visual features of this item. The learned attention vector is then integrated with the user and item vectors to compute the user and item distance. In this way, our model can capture a user’s varying preferences for different items in recommendation. It is also worth mentioning that our proposed MAML enjoys the following merits: 1) It satisfies the inequality property because of the adopted metric-based learning approach, and thus avoids the problem of the dot product similarity prediction method; And 2) it tackles the geometrically restrictive problem in CML. This is because the weight vector works as a transformation vector which projects the user and item into a distinct space (because the weight vector is unique for each user-item pair) to perform the proximate calculation via the Euclidean distance. Therefore, MAML allows a greater extent of geometric flexibility and modelling capability. Besides, we find that the standard attention mechanism is inferior to MAML because it leads to very small Euclidean distance between each dimension of the user and item vectors, which significantly diminish the prediction power of our model (see Sect. 3.2.2 for detail). To address this, we adjust the attention design by enlarging the summation value of the attention weights. We conduct comprehensive experiments on three benchmark datasets to evaluate the performance of our model on the top- n recommendation task. Experimental results demonstrate that our MAML outperforms the state-of-the-art MF-based and metric-based learning approaches. Beyond the improvement in accuracy, an exciting property of MAML is that it can uncover users’ varying preferences on items.

2 RELATED WORK

Diverse Preference Modeling. In recent two years, researchers have paid more attention to model the varying preferences towards different items and proposed several methods. We roughly categorize those methods into two groups. 1) Methods in the first group exploit reviews to analyze each user’s attention on different aspects of the target item, and then integrate the attention weights into the matrix factorization for recommendation [7, 10–12]. In particular, ALFM [11] applies a topic model on reviews to detect the user attention, and A³NCF [10] uses a neural attention network to learn the user attention from reviews. Following the same idea, Chin et al. [12] developed an end-to-end attentive neural network based recommendation model, which leverages reviews and ratings to learn user diverse preferences on different aspects of items. The AFM method [7] adopts a similar strategy and exploits other types of side information (e.g., item category) instead of reviews. And 2) as to the methods in the second group, they adapt the user or item vector according to the most influential users or items, which are selected based on the target item. For example, in CMN [14], the target user vector is computed based on the neighbour user

⁴It is defined as: “the distance between two points cannot be larger than the sum of their distances from a third point” [42].

vectors, where the neighbour users and their contributions to the target user vector depend on the target item. MARS [56] adapts the target user vector based on the neighbour item vectors of the target item. Both CMN and MARS fix the vector of the target item while adapt the user vector. In contrast, DIN [58] fixes the user preference vector while adapts the target item representation based on the user’s previously purchased items.

All the aforementioned methods are based on the matrix factorization framework by using dot product to estimate the similarity between the target user and item. With a different strategy in this work, we developed a novel user diverse modeling method based on the metric learning method, which automatically avoids the limitation of dot product. To the best of our knowledge, this is the first work to adopt metric learning method to capture user diverse preferences on different items.

Metric-based Learning. Several efforts have been dedicated to incorporating metric-based learning methods into collaborative filtering. Typically, the Euclidean embedding is the most popular one. For instance, Khoshneshin et al. [25] used the Euclidean embedding to model explicit feedback. Bachrach et al. [1] developed a method to transform the pre-trained dot-product space into a Euclidean space by preserving the item similarity. The most related studies are the recently proposed collaborative metric filtering (CML) [20] and latent relational metric learning (LRML) methods [41]. CML learns the user vector and item vector by minimizing the Euclidean distance of each user-item pair. Despite its simplicity, it can effectively encode the user preference, as well as user-user and item-item similarity, achieving very competitive performance on several datasets. In [41], authors pointed out that CML is geometrically restrictive and leads to an ill-posed algebraic system. This is because it tries to map each pair of user-item into the same point in the Euclidean space, and there are many neighbours for each user and item. To tackle this problem, they introduced a relational vector to connect the user and item in the embedding space. In this paper, we introduce an adaptive weight vector to transform each user-item pair into a unique space (i.e., the user-item specific attentive-aspect space) for minimization. Therefore, our method could also deal with the problem of CML. Besides, the weight vector is tailored for each pair of user-item, and thus our method can capture users’ varying preferences upon items.

Multimodal User Preference Modeling. Although many recommender systems have been designed to exploit side information to enhance the recommendation performance, most of them only exploit a single modality feature, such as textual reviews [4, 5, 10–12, 31, 39], item images [18, 24, 32, 47], and other metadata [7, 8, 15, 34]. More recently, several deep models have been proposed to model user preference by using multimodal information. TranSearch [17] exploits both textual reviews and item images to model the user preference on products for personalized search. Zhang et al. [51] proposed a knowledge base method, which extracts the multimodal knowledge as well as unstructured textual and visual knowledge to jointly learn the latent representations of items within a collaborative filtering framework. Similarly, Zhang et al. [54] first extracted user and item features from ratings, reviews, and images via deep representation learning and then concatenated the multimodal features to form the joint representations of users and items for recommendation. In this work, our method is different from the

above methods from two perspectives. Firstly, the above methods leverage the multimodal features to jointly learn fixed user and item representations for recommendation, while our method exploits the multimodal features to learn user’s varying attention to the aspects of different items. Secondly, our method is based on a metric learning strategy and adopts the Euclidean distance to predict the user preference on an item, which is greatly different from the methods relying on the matrix factorization framework and the dot product for similarity estimation.

3 OUR PROPOSED MODEL

3.1 Notations and Background

Notations. Before describing of our model, we would like to first define some basic notations. Let \mathcal{U} be the user set, \mathcal{I} be item set, and \mathbf{R} be the user-item interaction matrix. $r_{u,i} \in \mathbf{R}$ indicates the interaction between the user $u \in \mathcal{U}$ and the item $i \in \mathcal{I}$. The interaction could be implicit or explicit. For implicit feedback, $r_{u,i} = 1$ if the interaction between u and i exists; otherwise, $r_{u,i} = 0$. For explicit feedback, $r_{u,i}$ usually is the rating that u assigns to i . Let \mathcal{R} be the set of user-item (u, i) pairs whose values are non-zero and \mathcal{R}_u be the set of items that have been interacted by u . $\mathbf{p}_u \in \mathbb{R}^f$ and $\mathbf{q}_i \in \mathbb{R}^f$ denote the latent feature vector for u and i , respectively.⁵ The core of recommender systems is to learn the latent feature vectors of users and items (i.e., \mathbf{p}_u and \mathbf{q}_i). When recommending items to a user u , for each item $j \in \mathcal{R}_u$, its recommendation score $\hat{r}_{u,j}$ is computed based on \mathbf{p}_u and \mathbf{q}_j .

Background. Matrix factorization (MF) [27] maps users and items into a shared latent feature space and estimates an unseen interaction by the dot product between the user and item vector, i.e., $\hat{r}_{u,i} = \mathbf{p}_u^T \mathbf{q}_i$. And \mathbf{p}_u and \mathbf{q}_i for all $u \in \mathcal{U}$ and $i \in \mathcal{I}$ are learnt by minimizing the predicted errors between the $\hat{r}_{u,i}$ and $r_{u,i}$, namely, $\sum_{u,i \in \mathcal{R}} \|\hat{r}_{u,i} - r_{u,i}\|^2$.⁶ Due to its simplicity and superior performance, matrix factorization has become the most popular collaborative filtering method over the past decade. The original MF models were designed for rating prediction [19, 27] and was extended to weighted regularized matrix factorization (WRMF) for implicit feedback prediction [21]. As the goal of recommender system is to provide the target user with a ranking list of top few items, which are most likely to be preferred by the user, recommendation is rather a ranking problem instead of rating. In light of this, MF moves to model the relative preferences between different items and the pairwise learning approach has been widely adopted to achieve the goal [35, 54]. In pairwise learning, the user and item vectors are learnt by setting $\hat{r}_{u,i} > \hat{r}_{u,k}$ for any two pairs that satisfy $(u, i) \in \mathcal{R}$ and $(u, k) \notin \mathcal{R}$. A typical example is the bayesian personalized ranking (BPR) [35]. It is a MF based method, in which $\hat{r}_{u,i} = \mathbf{p}_u^T \mathbf{q}_i$ and $\hat{r}_{u,k} = \mathbf{p}_u^T \mathbf{q}_k$ and thus $\mathbf{p}_u^T \mathbf{q}_i > \mathbf{p}_u^T \mathbf{q}_k$ (s.t. $(u, i) \in \mathcal{R}$ and $(u, k) \notin \mathcal{R}$) is used for optimization.

Despite the success of MF, it is not a metric-based learning approach as the adopted dot product similarity does not satisfy the triangle inequality property, which is critical for modeling fine-grained user preference as demonstrated in [20, 53]. From the metric learning perspective, since users and items are represented as

⁵In the paper, unless otherwise specified, notations in bold style denote matrices or vectors, and the ones in normal style denote scalars.

⁶Notice that here we ignore the regularization term for the simplicity of presentation.

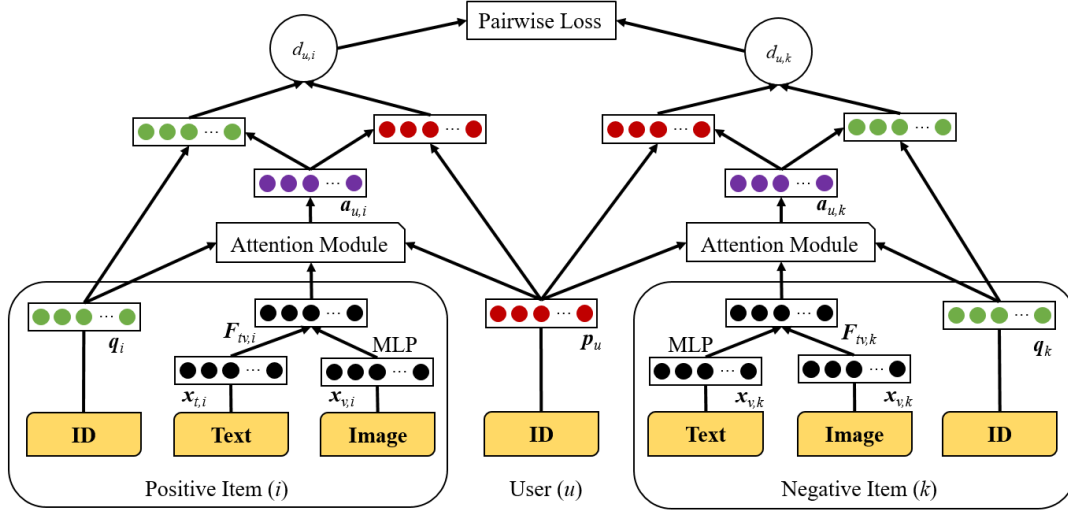


Figure 1: Overview of our MAML model.

latent vectors in a shared space, the similarity between a user and an item can be estimated based on the Euclidean distance between their vectors as:

$$d(u, i) = \|p_u - q_i\|. \quad (1)$$

Considering the pairwise learning, it is natural to derive that $d(u, i) < d(u, k)$ for $(u, i) \in \mathcal{R}$ and $(u, k) \notin \mathcal{R}$, where i denotes an item that u likes and k denotes an item that u does not like. The collaborative metric learning (CML) [20] is designed based on this simple idea. As this method is a metric-based learning approach, it naturally avoids the limitation of dot product and achieves better performance than the MF models [20]. Therefore, in this work, we also derive our user diverse preference modeling method based on this approach instead of MF.

3.2 Multimodal Attentive Metric Learning

3.2.1 Overview. In the aforementioned methods, they all utilize a fixed vector p_u to represent a user u 's preference in the feature space. In those models which map users and items into a joint latent space for similarity estimation, they all assume that each dimension in the space stands for a type of feature or an aspect of the items. Those dimensions/aspects are expected to well describe and distinguish the preferences of different users. Based on this understanding, we argue that using the same vector p_u to predict u 's preferences for all items may be not optimal, because in the real scenarios, it is common that the preference of a user on the aspects of different items is varying. For example, a user who prefers the *taste* and *price* of a restaurant may pay more attention to the *ambiance* and *service* of another restaurant, because the two restaurants serve for different purposes. When predicting the preference of a user u towards an item i , those aspects of the item i to which u pays the most attention should dominate u 's preference on the item i .

In light of this, we propose a multimodal attentive metric learning (MAML) model. For each user-item (u, i) pair, our model computes a weight vector $a_{u,i} \in \mathbb{R}^f$ to indicate the importance of i 's aspects for u . In addition, the side information of items is exploited to estimate the weight vector, as side information conveys rich features of

items, especially text reviews and item images, which are well-recognized to provide notable and complementary features of items in different aspects [9, 54]. We adopt the recent advancement of attention mechanism [6, 10] to estimate the attention vector. With the attention (weight) vector, the Euclidean distance between a user u and an item i in our model becomes:

$$d(u, i) = \|a_{u,i} \odot p_u - a_{u,i} \odot q_i\|, \quad (2)$$

where \odot denotes the element-wise product between vectors.

It is worth mentioning that, with the attention vector, our model can not only accurately capture the user's varying preferences for different items, but also tackle the geometrically restrictive problem [41] in CML. From Eq. 1, it can be found that CML tries to fit a user and all the interacted items into the same point in the latent space, however, each item in turn has many interacted users. Therefore, it is geometrically impossible to achieve the goal. In our model, as $a_{u,i}$ is unique for each user-item pair, it works as a transformation vector which transforms the target user and item into a new space for distance computation. Thus, our method can naturally avoid the geometrically restrictive problem in CML. An overview of our proposed modal MAML is illustrated in Fig. 1.

We adopt the pairwise learning for optimization and the loss function is defined as:

$$L_m(d) = \sum_{(u,i) \in \mathcal{R}} \sum_{(u,k) \notin \mathcal{R}} \omega_{ui} [m + d(u, i)^2 - d(u, k)^2]_+, \quad (3)$$

where i is an item that u likes and k is an item that u does not like; $[z]_+ = \max(z, 0)$ denotes the standard hinge loss. ω_{ui} is a ranking loss weight (described in Sect. 3.2.3) and $m > 0$ is the safety margin size. $d(u, i)$ and $d(u, k)$ are computed according to Eq. 2. In the next, we introduce how to compute the attention vector $a_{u,i}$ for each user-item pair.

3.2.2 Attention Mechanism. In this section, we introduce the attention mechanism in MAML for capturing a user u 's specific attention $a_{u,i}$ of an item i . Since text reviews and images contain rich information about user preference and item characteristic, they are used to capture u 's attention on the various aspects of i . A two-layer

neural network is used to compute the attention vector:

$$\mathbf{e}_{u,i} = \text{Tanh}(\mathbf{W}_1[\mathbf{p}_u; \mathbf{q}_i; \mathbf{F}_{tv,i}] + \mathbf{b}_1), \quad (4)$$

$$\hat{\mathbf{a}}_{u,i} = \mathbf{v}^T \text{ReLU}(\mathbf{W}_2 \mathbf{e}_{u,i} + \mathbf{b}_2), \quad (5)$$

where $\mathbf{W}_1, \mathbf{W}_2$ and $\mathbf{b}_1, \mathbf{b}_2$ are respectively the weight matrices and bias vectors of the two layers. \mathbf{v} is a vector that projects the hidden layer into an output attention weight vector. $\mathbf{F}_{tv,i}$ is the item feature vector which is a fusion of i 's textual feature and image feature (described later). $[\mathbf{p}_u; \mathbf{q}_i; \mathbf{F}_{tv,i}]$ denotes the concatenation of $\mathbf{p}_u, \mathbf{q}_i$, and $\mathbf{F}_{tv,i}$. Tanh and ReLU [29, 30, 36] are used as the activation functions for the first and second layer, respectively.⁷

Following the standard procedures of neural attention networks, there is a subsequent step to normalize $\hat{\mathbf{a}}_{u,i}$ with the softmax function, which converts the attention weights to a probabilistic distribution. Unfortunately, this standard solution does not work well in practice. This is because in our model, the attention weights are directly used to element-wise product with the Euclidean distance between \mathbf{p}_u and \mathbf{q}_i (see Eq. 2). For each dimension l , the weighted distance is $d_{u,i,l} = a_{u,i,l} \cdot (\mathbf{p}_{u,l} - \mathbf{q}_{i,l})$. After softmax normalization, the weights will be very small. For example, when the dimension f is 100, the mean value of weights is only 0.01. Notice that the distance between each dimension of \mathbf{p}_u and \mathbf{q}_i is already quite small⁸. With such a small weight $a_{u,i,l}$, the distance $d_{u,i,l}$ becomes even smaller. When the distances of all dimensions are quite small, the differences between different dimensions (aspects) become negligible. This will weaken the distinguishing power of our model, resulting in performance deterioration. To alleviate this problem, we propose to enlarge the normalized weight by a factor α . In our model, the final attention weight vector is computed as:

$$a_{u,i,l} = \alpha \cdot \frac{\exp(\hat{a}_{u,i,l})}{\sum_{l=1}^f \hat{a}_{u,i,l}}. \quad (6)$$

In experiments, we set α to be the dimension of the weight vector, i.e., $\alpha = f$, which turns out to work well (see Section 4.3). This setting is motivated by the consideration that when the weight vector is binary, only the aspects with the weight 1 take effects on the final decision, and the extremely case is that all the aspects are of the same important. In the next, we introduce how to obtain the fused item feature vector $\mathbf{F}_{tv,i}$.

Item Features. For each item, its textual and visual features are extracted from its associated reviews and images.⁹ The text feature $\mathbf{F}_{t,i}$ is extracted by the PV-DM [28] model, which learns continuous distributed vector representations for textual documents in an unsupervised way. It considers the word sequence information and tries to preserve the semantic features. PV-DM takes text documents as inputs and outputs their vector representations in a latent semantic space. The visual feature $\mathbf{F}_{v,i}$ is extracted by the Caffe reference model [23], which consists of 5 convolutional layers and 3 fully connected layers. This model was pre-trained on 1.2 million ImageNet (ILSVRC2010) images. In this work, we take the output of the second fully connected layer, resulting in a 4096-D feature vector as the visual features for each item.

⁷We have empirically studied different combinations of the activation functions in the attention neural network, and find out that Tanh and Relu can achieve relatively better performance.

⁸Due to the regularization $\|\mathbf{p}_*\|^2 \leq 1$ and $\|\mathbf{q}_*\|^2 \leq 1$, see Sect. 3.2.4

⁹In this work, each item is associated with one image.

After extracting the textual and visual features of items, we fuse them to better represent the characteristics of items [46]. Many multimodal feature fusion methods have been proposed and we adopt a widely used strategy[38, 52] by first concatenating the textual and visual features and then feeding them into a multiple-layer neural networks. Specifically, the feature fusion network is,

$$\begin{aligned} \mathbf{z}_1 &= \sigma(\mathbf{W}_1[\mathbf{F}_{t,i}; \mathbf{F}_{v,i}] + \mathbf{b}_1), \\ \mathbf{z}_2 &= \sigma(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2), \\ &\dots\dots, \\ \mathbf{z}_L &= \sigma(\mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L), \end{aligned} \quad (7)$$

where \mathbf{W}_l and \mathbf{b}_l denote the weight matrix and bias vector for the l -th layer, respectively. $\sigma(\cdot)$ is the activation function and ReLU is adopted because its biologically plausible and non-saturated property [16]. The output of last layer is the fused feature, namely, $\mathbf{F}_{tv,i} = \mathbf{z}_L$. Noticed that our focus in this paper is to exploit items' multimodal features for capturing users' varying attentions on different aspects of various items. The above extraction and fusion method is adopted for simplicity, and other advanced feature extraction and fusion methods could also be adopted in this case.

3.2.3 Ranking loss weight. We adopt the Weighted Approximate-Rank Pairwise (WARP) loss [48] to compute $\omega_{u,i}$. This scheme penalizes a positive item at a lower rank much more heavily than the one at the top, and produces the state-of-the-art results in previous work [20, 55]. Given a metric d , let $\text{rank}_d(u, i)$ denote the position of i in u 's recommended ranking list, $\omega_{u,i}$ is computed as,

$$\omega_{u,i} = \log(\text{rank}_d(u, i) + 1). \quad (8)$$

In WARP, $\text{rank}_d(u, i)$ is estimated through a sequential sampling procedure which repeatedly samples negative items to find impostors [48]. For each user-item pair (u, i) , let J denote the total number of items and M denote the number of impostors in N samples. The $\text{rank}_d(u, i)$ is approximated as $\lfloor (J \times M)/N \rfloor$. For more details on WARP, please refer to [48]. In our implementation, we follow the procedure described in [20] to estimate $\text{rank}_d(u, i)$.

3.2.4 Regularization. As text reviews and item images represent items' characteristics, we would like that items with similar textual and visual features to be closer in the latent feature vector. To achieve the goal, we define the following L_2 loss function,

$$L_f(q_*) = \sum_i \|\mathbf{F}_{tv,i} - \mathbf{q}_i\|. \quad (9)$$

This function penalizes an item i 's feature vector \mathbf{q}_i when \mathbf{q}_i deviates away from the extracted feature $\mathbf{F}_{tv,i}$.

To prevent the redundant of each dimension in the feature space, we then employ another regularization technique, covariance regularization [13] to reduce the correlation between the activation in a deep neural network. This technique can be also used in our model to de-correlate the dimensions in the feature space and thus to maximize the utilization of the given space. Let \mathbf{y}^n denote the latent vector of an object, which could be a user or an item; and n indexes the object in a batch of size N . The covariances between all pairs of dimensions i and j from a matrix \mathbf{C} are defined as:

$$C_{i,j} = \frac{1}{N} \sum_n (y_i^n - \mu_i)(y_j^n - \mu_j), \quad (10)$$

where $\mu_i = \frac{1}{N} \sum_n y_i^n$. We define the loss L_c to regularize the covariances:

$$L_c = \frac{1}{N} \left(\|C\|_f - \|\text{diag}(C)\|_2^2 \right), \quad (11)$$

where $\|\cdot\|_f$ is the Frobenius norm.

Finally, similar to the previous metric-based collaborative filtering methods [20, 41], we also bound all the user and item vector within a Euclidean unit sphere, i.e., $\|\mathbf{p}_*\|^2 \leq 1$ and $\|\mathbf{q}_*\|^2 \leq 1$, for regularization and preventing overfitting.

3.3 Optimization

With the consideration of all the regularization terms, the final object function of our MAML is,

$$\begin{aligned} \min_{\theta, \mathbf{p}_*, \mathbf{q}_*} \quad & L_m + \lambda_f L_f + \lambda_c L_c \\ \text{s.t.} \quad & \|\mathbf{p}_*\|^2 \leq 1 \text{ and } \|\mathbf{q}_*\|^2 \leq 1, \end{aligned} \quad (12)$$

where λ_f and λ_c are hyperparameters that control the weight of each loss term. The optimization is quite standard and the stochastic gradient descent (SGD) algorithm is adopted. In implementation, the Adam optimizer [26] is adopted to tune the learning rate.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. The public Amazon review dataset [31], which has been widely used for recommendation evaluation in previous studies [7, 9, 54], is adopted for experiments in this work. We adopted four product categories as shown in Table 1. Since we need to exploit reviews and item images to extract the item features, items without any textual reviews or images were removed. After this step, we further pre-processed the dataset to keep only the items and users which have at least 5 interactions. The basic statistics of the four datasets are shown in Table 1. As we can see, the datasets are of different sizes and sparsity. For example, the *office* dataset is relatively denser than other datasets. The diversity of datasets is useful for analyzing the performance of our method and the competitors in different situations.

In this work, we focus on the top- n recommendation task, which aims to recommend a set of n top-ranked items that will be appealing to the target user. For each dataset, we randomly selected 70% of the interactions from each user to construct the training set, and the remaining 30% for testing. Each user has at least 5 interactions, thus we had at least 3 interactions per user for training, and at least 2 interactions per user for testing.

4.1.2 Baselines and Evaluation Metrics. We compare our MAML method with the following baselines, including both shallow (BPR [35], VBPR [18]) and deep (DeepCoNN [57], NeuCF [19], JRL [54]) MF based models, as well as the metric learning based method CML [20]. Besides, they cover different information sources, i.e., ratings (BPR, CML), reviews (DeepCoNN, CML_F [20], JRL), images (CML_F, JRL). Notice that CML_F is an extension of CML. It extends the CML model by considering item features for top- n recommendation. We use CML_{text}, CML_{image}, and CML_{all} to denote the model considering the text feature, image feature, and both features, respectively.

Four widely used evaluation metrics for top- n recommendation are used in our evaluation, including *precision*, *recall*, *NDCG* [22],

Table 1: Basic statistics of the experimental datasets.

| Dataset | #user | #item | #interactions | sparsity |
|----------------|--------|--------|---------------|----------|
| Office | 4,874 | 2,406 | 53,137 | 99.55% |
| Men Clothing | 4,955 | 5,028 | 32,363 | 99.87% |
| Women Clothing | 19,244 | 14,596 | 135,326 | 99.95% |
| Toys Games | 18,748 | 11,673 | 161,653 | 99.93% |

and *hit ratio* (HR). All the metrics are computed based on the top 10 results. The reported results are the average values across all the testing users.

4.1.3 Experimental Settings. We implemented our model with TensorFlow¹⁰ and carefully tuned the key parameters. Specifically, we tuned the initial learning rate $\ell_0 \in \{0.1, 0.01, 0.001, 0.0001\}$; the margin m in the hinge loss $m \in \{0.1, 0.2, \dots, 2.0\}$; the number of negative samples (s) and the regularization parameters $\{s, \lambda_f, \lambda_c\} \in \{1, 2, \dots, 10\}$. In our experiments, the following setting works well: $\ell_0 = 0.001$, $m = 1.6$ (1.5 for ToysGames), $s = 8$ (4 for Office and Men-Clothing). $\lambda_c = 5$, and $\lambda_f = 7$ (2 for ToysGames). The optimal batch size varies across different datasets. Besides, model parameters are saved in every 10 epochs and the models are trained in maximum 1,000 epochs. In experiments, the dimension of latent vectors (i.e., \mathbf{p}_u and \mathbf{q}_i) of all methods is set to 64. Notice that all the competitors have also been carefully tuned for fair comparisons. For example, for CML, we found that $\lambda_c = 1$ (all the other hyperparameters are the same as our MAML model) works better. We released the codes and involved parameter settings to facilitate others to repeat this work¹¹.

4.2 Performance Comparison

The results of our model and all the competitors over the four datasets are reported in Table 2. It can be found that our method outperforms all the competitors consistently across all the test datasets in terms of different metrics. Besides, by grouping all the methods into four categories, we gained some interesting observations.

Firstly, we focused the performance of the methods in the first block, which only use the user-item interaction information. Specifically, NeuCF adopts neural networks which can better model user-item interactions and achieve better results than BPR in general. The *Office* dataset has a smaller size but is relatively denser, which might be the reason that BPR can obtain good performance on this dataset.¹² CML outperforms both BPR and NeuMF by a large margin, because CML can capture fine-grained user preferences by using the metric-based learning approach. It can encode not only user preference but also the user-user and item-item similarity.

The second and third blocks are the methods exploiting one type of side information, i.e., text and image, respectively. MAML_{text} and MAML_{image} are variants of our model MAML, which exploits only text and image, respectively. As we can see, the recommendation performance can be greatly improved with the exploitation

¹⁰<https://www.tensorflow.org>.

¹¹<https://github.com/liufancs/MAML>

¹²Remind that MF based method suffers from the cold-start problem (i.e., very limited interactions for users or items) and NeuMF needs relatively more data for better learning.

Table 2: Performance of our MAML model and the competitors over four datasets. Noticed that the values are reported by percentage with '%' omitted.

| Datasets | Office | | | | Men Clothing | | | | Women Clothing | | | | Toys Games | | | |
|-----------------------|---------------|---------------|----------------|---------------|---------------|---------------|----------------|---------------|----------------|---------------|----------------|---------------|---------------|---------------|----------------|---------------|
| Metrics | NCDG | RECALL | HR | PR | NCDG | RECALL | HR | PR | NCDG | RECALL | HR | PR | NCDG | RECALL | HR | PR |
| BPR | 4.557 | 7.780 | 21.410 | 2.562 | 1.229 | 2.375 | 4.611 | 0.478 | 0.572 | 1.086 | 2.062 | 0.244 | 0.807 | 1.733 | 3.235 | 0.385 |
| NeuCF | 4.145 | 6.609 | 19.504 | 2.516 | 1.416 | 2.599 | 6.017 | 0.625 | 1.136 | 2.115 | 4.377 | 0.491 | 2.374 | 4.558 | 9.801 | 1.374 |
| CML | 6.680 | 8.986 | 24.025 | 3.141 | 2.520 | 4.040 | 7.815 | 0.811 | 3.026 | 4.542 | 9.157 | 1.034 | 6.207 | 8.638 | 17.454 | 2.204 |
| DeepCoNN | 2.418 | 3.842 | 11.273 | 2.160 | 0.813 | 1.510 | 3.643 | 0.491 | 0.648 | 1.214 | 2.474 | 0.394 | 1.496 | 2.581 | 5.665 | 1.145 |
| CML _{text} | 6.889 | 9.171 | 24.778 | 3.223 | 3.212 | 4.952 | 9.659 | 1.005 | 3.507 | 5.139 | 9.972 | 1.144 | 6.345 | 8.787 | 17.957 | 2.260 |
| MAML _{text} | 7.094 | 9.346 | 25.165 | 3.222 | 3.282 | 4.961 | 9.843 | 1.019 | 3.590 | 5.246 | 10.329 | 1.175 | 6.410 | 8.954 | 18.093 | 2.289 |
| VBPR | 3.072 | 4.942 | 15.381 | 1.817 | 1.061 | 1.914 | 3.873 | 0.392 | 0.921 | 1.644 | 3.357 | 0.377 | 0.846 | 1.595 | 3.738 | 0.418 |
| CML _{image} | 6.987 | 9.232 | 24.560 | 3.214 | 3.542 | 5.259 | 10.419 | 1.086 | 3.704 | 5.499 | 10.804 | 1.237 | 6.362 | 8.923 | 17.951 | 2.301 |
| MAML _{image} | 7.077 | 9.286 | 25.155 | 3.261 | 3.585 | 5.401 | 10.927 | 1.148 | 3.733 | 5.554 | 10.956 | 1.246 | 6.486 | 9.087 | 18.312 | 2.314 |
| JRL | 3.391 | 3.421 | 12.345 | 1.525 | 2.574 | 3.920 | 7.247 | 0.657 | 1.729 | 2.906 | 4.965 | 0.457 | 2.302 | 4.599 | 8.232 | 1.382 |
| CML _{all} | 7.032 | 9.349 | 24.911 | 3.236 | 3.718 | 5.408 | 10.627 | 1.127 | 3.798 | 5.562 | 10.985 | 1.241 | 6.409 | 8.928 | 18.135 | 2.308 |
| MAML _{all} | 7.139* | 9.386* | 25.177* | 3.265* | 3.733* | 5.574* | 10.719* | 1.152* | 3.809* | 5.636* | 11.144* | 1.260* | 6.547* | 9.111* | 18.341* | 2.327* |

The symbol * denotes that the improvement is significant with p -value < 0.05 based on a two-tailed paired t-test.

of additional item features, which has been demonstrated in many previous studies. DeepCoNN utilizes text information but performs unsatisfactorily. This is because the DeepCoNN uses text reviews not only in training but also in the test stage. In real scenario, the user review for an item is unavailable before the purchasing behavior happened. Therefore, in our experiments, we did not use review information in the testing stage, resulting in the performance degradation of DeepCoNN (which has also been demonstrated in [4]). It is interesting that the performance based on images outperforms that based on texts (by comparing CML_{image} and MAML_{image} to CML_{text} and MAML_{text}). This might because the image features are more important for the selected four types of products (e.g., *Clothing*). VBPR achieves satisfactory performance in *Women Clothing* but it is inferior to NeuCF and BPR on other datasets.

Finally, in the last block, the methods exploit both text and image features. The best performance of CML_{all} and MAML_{all} demonstrates the effectiveness of fusing multimodal features, which can indeed enhance the recommendation accuracy by considering multi-source item information. JRL achieves the best results over all the other baselines except the CML based ones (and BRP on the *Office* dataset). The CML based methods outperform all the other baselines, verifying the potential of metric-based learning approach. The only difference between MAML and CML is that the former models user diverse preferences by using the attention mechanism. The consistent and significant improvement of MAML over CML validates the importance of considering user diverse preferences on different aspects of items, and also demonstrates the effectiveness of our proposed model.

4.3 Effects of Our Attention Mechanism

Remind that in our MAML model, we adapt the standard attention mechanism by enlarging the sum of all the attention weights by a factor α . In our implementation, the value of α is set to be equal to the dimension of the user/item feature vectors, i.e., $\alpha = f$

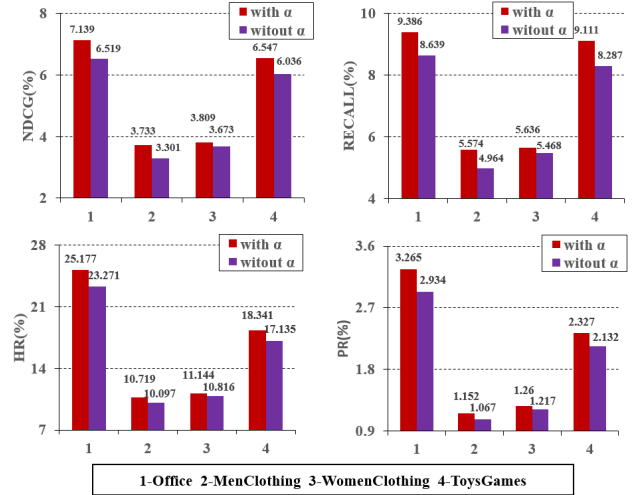


Figure 2: Our adapted attention mechanism (with α) vs. the standard attention mechanism (whiteout α).

(see Sect. 3.2.2 for details). In this section, we examine the effectiveness of this tailored attention mechanism in our model. In particular, we compare the performance of our model with the adapted attention mechanism (with α) and the standard one (without α). We tuned MAML with the standard attention mechanism carefully (e.g., $m \in \{0.001, 0.005, 0.01, \dots, 2.0\}$, and $\{\lambda_f, \lambda_c\} \in \{0.0001, 0.001, \dots, 10\}$). The best results are used for comparisons. As shown in Fig. 2, with the adapted attention mechanism, our model can achieve consistent improvement by a large margin across all the test datasets, indicating the effectiveness of our attention mechanism in MAML.

4.4 Visualization

In the proposed method, we model user diverse preferences by capturing user attention to different aspects of items. Our intuition

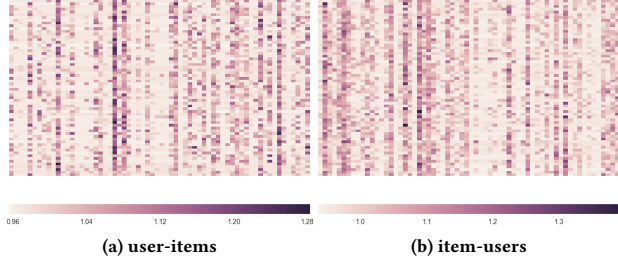


Figure 3: Visualization of Attention Vector.

is that the preference of a user on different aspects is varying when facing different items. Therefore, for each user-item pair, our model computes a unique attention weight for each aspect of the item (i.e., different dimensions of the feature vector). In this section, we would like to: 1) visualize the learned attention weights to illustrate user varying preferences on different aspects of various items; and 2) visualize user diverse preferences based on the purchased items.

Attention Visualization. We use heatmap to visualize the attention weights as shown in Fig. 3. The color scale represents the intensities of attention weights, where a deeper color indicates a higher value and a lighter color indicates a lower value. Fig. 3a visualizes the attention weights of a user for 64 items that were purchased by the user (in the *Office* dataset). The x -axis represents the dimensions of attention vector (i.e., 64), and y -axis represents the 64 items. At the beginning, the user attention on all aspects (or dimensions) is set to be the same (i.e., the all the values in the attention vector is 1). During the training process, the attention weight of each dimension will gradually adjust within the range of $[0, 64]$, and finally converge to a constant value. Fig. 3a visualizes the converged values, which demonstrates that the user preference on different aspects is varying across different items. Fig. 3b illustrates the attention weights of 64 users on the aspects of a randomly selected item (also from the *Office* dataset). The x -axis also represents the dimensions of attention vector, and y -axis represents different users. We can see that, different users pay attention to different aspects of the same item.

User diverse preference visualization. We argued that a user preference is diverse, because a user 1) likes items with diverse features; and 2) has different preferences on the aspects of various items. In Fig. 4, we visualize the user preference by clustering the purchased items using 1) item embedding and 2) user attention vectors with respect to items. Fig. 4 shows the clustering results of all the items that a user purchased from *Office* and *WomenClothing*, respectively. The left figures visualize the clustering results based on item embedding (i.e., q_i); and the right ones illustrate the clustering results based on user attention vectors. Every dot in the figures denotes an item. Besides, in both Fig. 4a and Fig. 4b, the dots with the same color (e.g., blue) in the two figures denote the same set of items. We use t-SNE [43] for clustering and visualization. Notice that the item vector (i.e., q_i) is learned based on its interactions with all the users. Therefore, this vector should well characterize all the aspects of the item. In contrast, an attention vector $a_{u,i}$ characterizes a user’s attention to the different aspects of an item. Therefore, the clustering results based on the item embedding (left figures) can demonstrate the user diverse preferences on items

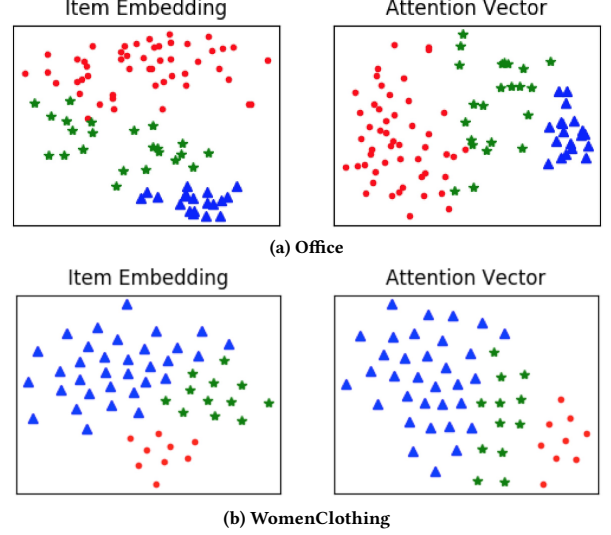


Figure 4: User preference visualization based on item embedding and attention vectors.

with different features; the clustering results based on the attention vector (right figures) validate user varying attention on different aspects of items. Besides, we can see that the distribution of the same items (e.g., red dots in Fig. 4a) is different in the two types of figures. It also demonstrates that a user pays different attention to the various aspects of different items.

5 CONCLUSIONS

In this paper, we presented a novel Multimodal Attentive Metric Learning (MAML) model for top- n recommendation. In particular, this model is designed to model user diverse preferences on different aspects of items. This is achieved by a proposed attention neural network, which exploits the multimodal features of the target item to estimate a user’s specific attention on each aspect of this item. Moreover, our model is developed based on a metric-based learning approach, which avoids the inherent limitation of matrix factorization based methods and thus can capture fine-grained user preference. Extensive experiments have been conducted on four real-world datasets from Amazon. The results show that our method significantly outperforms a variety of competitors, demonstrating the effectiveness of our method and the potential of modeling user diverse preference for recommendation. Besides, we also studied the effectiveness of the designed attention mechanism and visualized the user diverse preference based on both the learned item vectors and the attention vectors.

6 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China, No.:61772310, No.:61702300, No.:61702302, No.:61802231, No.:U1836216; the Project of Thousand Youth Talents 2016; the Shandong Provincial Natural Science and Foundation, No.:ZR2019JQ23, No.:ZR2019QF001; the Future Talents Research Funds of Shandong University, No.:2018WLJH63.

REFERENCES

- [1] Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nave, and Ulrich Paquet. 2014. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *RecSys*. ACM, 257–264.
- [2] Robert M Bell and Yehuda Koren. 2007. Lessons from the Netflix prize challenge. *SIGKDD Explorations Newsletter* 9, 2 (2007), 75–79.
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *TPAMI* 35, 8 (2013), 1798–1828.
- [4] Rose Catherine and William Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. ACM.
- [5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. IW3C2, 1583–1592.
- [6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM, 335–344.
- [7] Jingwu Chen, Fuzhen Zhuang, Xin Hong, Xiang Ao, Xing Xie, and Qing He. 2018. Attention-driven Factor Model for Explainable Personalized Recommendation. In *SIGIR*. ACM, 909–912.
- [8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishikesh Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Isir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.
- [9] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable recommendation by leveraging reviews and images. *TOIS* 37, 2 (2019), 16.
- [10] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. 2018. A³NCF: An Adaptive Aspect Attention Model for Rating Prediction. In *IJCAI*. AAAI Press, 3748–3754.
- [11] Zhiyong Cheng, Ying Ding, Lei Zhu, and Kankanhalli Mohan. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *WWW*. IW3C2, 639–648.
- [12] Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. ANR: Aspect-based Neural Recommender. In *CIKM*. ACM, 147–156.
- [13] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. 2015. Reducing overfitting in deep networks by decorrelating representations. *arXiv e-prints* (2015).
- [14] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *SIGIR*. ACM, 515–524.
- [15] Li Gao, Hong Yang, Jia Wu, Chuan Zhou, Weixue Lu, and Yue Hu. 2018. Recommendation with multi-source heterogeneous information. In *IJCAI*. AAAI Press, 3378–3384.
- [16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *AISTATS*. JMLR, 315–323.
- [17] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. 2018. Multi-modal preference modeling for product search. In *MM*. ACM, 1865–1873.
- [18] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*. AAAI Press, 144–150.
- [19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. IW3C2, 173–182.
- [20] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *WWW*. IW3C2, 193–201.
- [21] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*. IEEE, 263–272.
- [22] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *TOIS* 20, 4 (2002), 422–446.
- [23] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *MM*. ACM, 675–678.
- [24] Yannis Kalantidis, Lyndon Kennedy, and Li-Jia Li. 2013. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *ICMR*. ACM, 105–112.
- [25] Mohammad Khoshneshin and W Nick Street. 2010. Collaborative filtering via euclidean embedding. In *RecSys*. ACM, 87–94.
- [26] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- [27] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 42–49.
- [28] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. JMLR, 1188–1196.
- [29] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. 2018. Attentive Moment Retrieval in Videos. In *SIGIR*. ACM, 15–24.
- [30] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. 2018. Cross-modal Moment Localization in Videos. In *MM*. ACM, 843–851.
- [31] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. ACM, 165–172.
- [32] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. ACM, 43–52.
- [33] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*. MIT Press, 1257–1264.
- [34] Liqiang Nie, Xuemeng Song, and Tat Seng Chua. 2016. Learning from Multiple Social Networks. *Synthesis Lectures on Information Concepts Retrieval Services* 8 (2016), 1–118.
- [35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI, 452–461.
- [36] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural compatibility modeling with attentive knowledge distillation. In *SIGIR*. ACM, 5–14.
- [37] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. Neurostylist: Neural compatibility modeling for clothing matching. In *MM*. ACM, 753–761.
- [38] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *NIPS*. MIT press, 2222–2230.
- [39] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*. AAAI Press.
- [40] Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. 2015. User modeling with neural network for review rating prediction. In *IJCAI*. AAAI Press, 1340–1346.
- [41] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *WWW*. IW3C2, 729–739.
- [42] Amos Tversky and Itamar Gati. 1982. Similarity, separability, and the triangle inequality. *Psychological review* 89, 2 (1982), 123.
- [43] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *JMLR* 9 (2008), 2579–2605.
- [44] Meng Wang, Yue Gao, Ke Lu, and Yong Rui. 2013. View-Based Discriminative Probabilistic Modeling for 3D Object Retrieval and Recognition. *TOIS* 22 (2013), 1395–1407.
- [45] Meng Wang, Hao Li, Dacheng Tao, Ke Lu, and Xindong Wu. 2012. Multimodal Graph-Based Reranking for Web Image Search. *TOIS* 21 (2012), 4649–4661.
- [46] Mingyu Wang, Changzhi Luo, Bingbing Ni, Jun Yuan, Jianfeng Wang, and Shuicheng Yan. 2018. First-Person Daily Activity Recognition With Manipulated Object Proposals and Non-Linear Feature Fusion. *TCSVT* 28 (2018), 2946–2955.
- [47] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *WWW*. IW3C2, 391–400.
- [48] Jason Weston, Samy Bengio, and Nicolas Usunier. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning* 81, 1 (2010), 21–35.
- [49] Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *IJCAI*. AAAI Press, 3203–3209.
- [50] Xun Yang, Mingyu Wang, and Dacheng Tao. 2018. Person Re-Identification With Metric Learning Using Privileged Information. *TOIS* 27 (2018), 791–805.
- [51] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*. ACM, 353–362.
- [52] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from Scratch: Towards Automatically Identifying, Modeling, and Naming Visual Attributes. In *MM*. ACM, 187–196.
- [53] Shuai Zhang, Lina Yao, Yi Tay, Xiwei Xu, Xiang Zhang, and Liming Zhu. 2019. Metric Factorization: Recommendation beyond Matrix Factorization. In *ICDE*. IEEE.
- [54] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *WSDM*. ACM, 1449–1458.
- [55] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *CIKM*. ACM, 821–830.
- [56] Lei Zheng, Chun-Ta Lu, Lifang He, Sihong Xie, Vahid Noroozi, He Huang, and Philip S Yu. 2018. MARS: Memory attention-aware recommender system. *arXiv e-prints* (2018).
- [57] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. ACM, 425–434.
- [58] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. ACM, 1059–1068.