This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

Multimodal Representation Learning for Recommendation in Internet of Things

Zhenhua Huang, Xin Xu, Juan Ni, Honghao Zhu, and Cheng Wang, *Senior Member, IEEE*

Abstract—Recommender system has recently drawn a lot of attention to the communities of information services and mobile applications. Many deep learning-based recommendation models have been proposed to learn the feature representations from items. However, in Internet of Things (IoT), items' description information are typically heterogeneous and multimodal, posing a challenge to items' representation learning of recommendation models. To address this challenge and to improve the recommendation effectiveness in IoT, a novel multimodal representation learning-based model (MRLM) was proposed. In MRLM, two closely related modules were trained simultaneously; they are global feature representation learning and multimodal feature representation learning. The former was designed to learn to accurately represent the global features of items and users through simultaneous training on three tasks: triplet metric learning, softmax classification, and microscopic verification. The latter was proposed to refine items' global features and to generate the final multimodal features by using items' multimodal description information. After MRLM converged, items' multimodal features and users' global features could be used to calculate users' preferences on items via cosine similarity. Through extensive experiments on two real-world datasets, MRLM remarkably improved the recommendation effectiveness in IoT.

Index Terms—Internet of Things, Multimodal representation, Recommender system, Deep learning, Multi-task optimization

1 Introduction

In recent years, the rapid growth of big data and information technology has led to the problem of information overload. A recommender system is an effective way to solve this problem [1]. It can filter information from overwhelming data based on users' historical preferences, find out the contents that users are really interested in, and help users to efficiently obtain the information.

With great successes in the areas of computer vision (CV) and natural language processing (NLP), deep learning begins to attract great interests in the area of recommender system and bring more opportunities for improving the recommendation effectiveness. In order to produce high-quality recommendation results, it is very important for deep learning-based recommendation models to accurately capture the features of items and users [2, [3]. Generally, extraction of user features is relatively simple. The existing models utilize users' structured attribute values to learn their feature representations [4-6]. Specifically, for each input user, existing models first convert his every attribute value into a one-hot vector, and then feed the concatenation of all obtained one-hot vectors into a deep neural network to learn his feature vector. While for the extraction of items' features, most existing models utilize items' structured attribute values [7-11] or text description [12-18] to learn their feature representations. The feature extraction of items' structured attribute values is similar to that of users. For feature extraction of items' text description, the existing models usually use convolutional neural network (CNN), long short-term memory (LSTM), gated recurrent unit (GRU), or attention mechanism to learn item feature representations [19].

Recently, Internet of Things (IoT) has attracted a lot attention in both industry and academia. It has been widely used in many applications, such as smart city, intelligent monitoring, smart shopping, and intelligent transportation. Data generated in IoT are usually heterogeneous, and in specific, recommended items in IoT have the multimodal characteristics [20]. For example, for each mobile phone in a smart shopping system, it may have the following multimodal description information: user reviews, user ratings, brief introduction, images, promotional videos, audio demos, and semantic information in knowledge base (or knowledge graph).

To cater for the applications of recommendation in IoT, the aforementioned models [7-18] have been extended in a straightforward way by introducing more modal description information like image [21-23], audio [24, 25], and knowledge base [23]. Firstly, for each item, the features from its different modal description information were separately extracted, and all the extracted features

Zhenhua Huang and Xin Xu are with the School of Computer Science, South China Normal University, No. 55 Zhongshan Avenue West, Guangzhou, 510631, China (E-mail: jukiehuang@163.com; 2095522151@qq.com).

Juan Ni is with the School of Politics & Administration, South China Normal University, No. 55 Zhongshan Avenue West, Guangzhou, 510631, China (E-mail: nijuanshkj@126.com).

Honghao Zhu is the Department of Computer Science and Engineering, Bengbu University, No. 1866 Caoshan Road, Bengbu 233000, China (E-mail: bbxyzhh@163.com).

Cheng Wang is with the Department of Computer Science and Engineering, Tongji University, No. 4800 Caoan Highway, Shanghai 201804, China (E-mail: cwang@tongji.edu.cn).

Corresponding authors: Zhenhua Huang; Juan Ni.

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

were concatenated as its final feature representation. Then, these feature representations were integrated into the existing models [7-18] to realize recommendation.

However, the extended models may lead to a poor recommendation performance due to three main reasons: (*a*) they ignore the mutuality and the complementarity among description information of items' different modalities; (*b*) they are unable to efficiently capture the potential influences of description information of items' different modalities on user preferences; and (*c*) when the number of modalities is increased, they are easy to lead to the problems of feature redundancy and dimension disaster for a recommender system.

For ensuring the effectiveness of recommendation in IoT, new models and methods are needed for a recommender system. In this work, a novel multimodal representation learning-based model (MRLM) was proposed to realize accurate recommendation in IoT. In MRLM, two closely related modules were simultaneously trained; they are global feature representation learning (GFRL) and multimodal feature representation learning (MFRL), and the joint loss function of these two modules was minimized.

Before training MRLM, data preprocessing was carried out, which mainly included two aspects: (*a*) deep learning-based recommendation models were utilized to preliminarily extract features for all users and items based on their structured attribute values, which were treated as initial feature representations [8-10, 16, 17]; and (*b*) for each modality of item, the following processes were implemented: first, state-of-the-art feature extraction methods were used to get the auxiliary features of each item, and then the obtained auxiliary features of all items were clustered by using *k*-means algorithms [26, 27]. When the processes were finished, each item has a unique cluster ID on its modality.

GFRL took a triplet of initial features (user, positive item, negative item) as a training sample and employed a CNN-based triplet network to extract global and low-level features for the triplet, respectively. Global features were extracted from the last layer of the triplet network and utilized to perform two tasks: triple metric learning and softmax classification. The former was given to minimize the preference distance between user and positive item, and to maximize the preference distance between user and negative item. The latter was given for ID recognition of user and items. Low-level features were extracted from the first layer of the triplet network and utilized to perform the task of microscopic verification. This task was designed to identify the user's preference difference between positive and negative items from the micro-level.

MFRL aimed to refine items' global features by employing their multimodal description information. It took the global features of positive and negative items from GFRL as input and employed a CNN-based Siamese network to extract their multimodal features. For either a positive or a negative item, the multimodal feature was extracted from the last layer of the Siamese network and employed to jointly optimize multiple classifiers. Each classifier corresponded to a modality of an item and performed a multi-classification task. The number of classes in each classifier equals the number of clusters on its corresponding modality. The class ID of an item in each classifier equals the cluster ID of an item on its corresponding modality.

After MRLM was converged, items' multimodal features and users' global features were used to calculate users' preferences on items through cosine similarity. In order to capture each user's dynamic preference efficiently, the global feature was combined with the multimodal features of the latest interacted items to obtain the final feature representation of a user.

Furthermore, the effectiveness of our model MRLM was studied through extensive experiments on two real-world datasets. The results showed that MRLM markedly outperformed the state-of-the-art models in terms of various evaluation metrics in IOT.

The rest of the paper was organized as follows: Section 2 introduced related works to the proposed model. Section 3 presented the details of the proposed MRLM model. Experimental results were presented in Section 4. Finally, Section 5 concluded the proposed work.

2 Related Works

In this section, the works related to the proposed model were introduced, which focused on extracting item features through deep learning-based methods applied in recommendation models. They mainly consisted of two categories.

The first category involved the recommendation models which use items' structured attribute values to extract their features [7-11]. No item auxiliary information was considered for the models in this category.

Cheng et al. [7] proposed a wide & deep model in which the wide component was used to exploit item features from history data and the deep component was used to generalize new feature combinations of items. By doing so, benefits of memorization and generalization were combined simultaneously. Guo et al. [8] introduced a Deep Factorization Machines (DeepFM) model, in which FM was used for recommendation and deep learning was used for feature learning. Compared with the wide & deep model, the DeepFM model has a shared raw feature input to both its wide and deep components, with no need of feature engineering besides raw features. Inspired by [8], Lian et al. [9] proposed the eXtreme Deep Factorization Machine (xDeepFM) model. It could learn certain bounded-degree feature interactions explicitly. Meanwhile, it was able to learn arbitrary low- and high-order feature interactions implicitly. Covington et al. [10] presented a video recommendation model, by which it could can learn to produce a score to each video based on the features of videos and users extracted from their structured attribute values. In addition, Ying et al. [11] proposed a two-layer attention network to realize recommendation. The first layer learned user long-term preferences based on the features of the latest interacted items, and the second one generated the final user representation by combining user long-term and short-term

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

preferences.

The second category consisted of the recommendation models in which items' auxiliary information was used. Yet, most of them used items' text description, i.e., single modality [12-18], and few studies utilized modal description information such as image [21-23], audio [24, 25], and knowledge base [23].

Kim et al. [12] introduced a Convolutional Matrix Factorization (ConvMF) model in which texts of items were used as auxiliary information. In ConvMF model, first, CNN was used to extract the features of items from their text description, and then, the matrix factorization algorithm [28] was used to calculate the scores of users on items. Okura et al. [13] utilized denoising autoencoder to learn news' features from news texts and adopted Recurrent Neural Network (RNN) to learn users' features from their historical data. The similarities between news and user features were calculated to obtain users' preferences on news. Seo et al. [14] utilized CNN and dual local and global attentions to realize modeling for user preferences from review texts of items. Similarly, Tay et al. [15] used the review information to represent items through two-level attention mechanism. The review-level attention was used to get the importance of each review, and the word-level attention was employed to calculate the importance of each word in reviews. The final features of items were obtained after the two-level attention operation. Chen et al. [16] presented NARRE, an efficient recommendation model based on neural attentive regression. NARRE learnt the importance of item's each review and realized the prediction of ratings through review-level explanations. Xing et al. [17] proposed HAUP, a hierarchical attention model by using product reviews. In particular, HAUP jointly learnt user and product information from ratings and review texts of products in recommendation. Xu et al. [18] proposed an opinion mining model based on CNNs to improve the recommendation effectiveness. This model used a two-step training neural network and tried to employ both review texts and ratings to capture users' true opinions in unbalanced data.

Zhang et al. [21] presented a co-attention network incorporating texts and images for recommending hashtags to tweets. Inspired by [21], Ma et al. [22] designed an efficient Cross-Attention Memory Network (CAMN) to carry out the mention recommendation task for tweets also by utilizing texts and images. Further, Zhang et al. [23] proposed Collaborative Knowledge Base Embedding (CKE) to improve the recommendation effectiveness. In particular, they designed three novel components to extract items' feature representations from texts, images and semantic information in knowledge base, respectively. Oramas et al. [24] addressed the cold-start problem for music recommendation by combining texts and audios with user feedback data through deep neural networks. Moreover, Bougiatiotis et al. [25] proposed MRTA, an efficient model to perform the movie recommendation task by extracting movies' feature representations from subtitles (i.e., texts) and audios.

As discussed in Section 1, the aforementioned models are not suitable for recommendation in IoT since they are likely to lead to a poor performance.

3 Methodology and Implementation for MRLM Model

In this section, first, MRLM is overviewed, and then, implementation details of MRLM is presented.

3.1 Overview of MRLM

MRLM mainly consists of two phases: Data Preprocessing and Model Training.

(1) Data Preprocessing. It produces input and auxiliary information for Model Training, and mainly realizes two tasks:

(*a*) xDeepFM [8], one of state-of-the-art deep learning-based recommendation models, is utilized to preliminarily extract features for all users and items. The structured attribute values of all users and items are embedded into corresponding initial feature vectors. The dimensionality of user feature vector is equal to that of item feature vector, defining as d_0.

(b) Let the number of item's modalities be r. For each modality of an item, the following procedures are carried out: First, state-of-the-art methods for this modality is employed to extract each item's auxiliary features from corresponding description information. The auxiliary feature vector of each item is then obtained, with respect to this modality. For example, the models of Transformer [29], ResNet (Residual Neural Network)-50 [30], and ECNN (Embedding Convolutional Neural Network) [31] can be used to realize feature extraction for the modalities of text, image, and video, respectively. Then, the MKC (Multiple Kernel k-means Clustering) algorithm [26] is adopted to cluster all auxiliary feature vectors into k clusters. When the procedure is finished, each item has a unique cluster ID on this modality.

In summary, after data preprocessing d_0-dimensional initial feature vector of each user and item can be obtained. For each item, r cluster IDs is produced, and each cluster ID corresponds to its modality.

(2) Model Training. It is the core of MRLM, shown in Fig. 1. In MRLM, two closely related modules, GFRL and MFRL, are trained jointly.

(*a*) GFRL takes a triplet of initial feature vectors (user, positive item, negative item) as an input sample and uses a triplet network to extract global vectors and low-level feature matrices for the triplet, respectively. Conceptually, a triplet network [32, 33] contains three copies of a base network with shared parameters. In this work, it consists of three five-layer CNN-based neural networks that have the same structure and network parameters. Global feature vectors of user and two items are extracted from the last layer, a fully-connected layer (FC), of the triplet network. They are utilized to simultaneously optimize two tasks: triple metric learning and softmax classification. For improving the accuracy of global feature representation, a microscopic verification task is introduced to assist the global feature extraction. This task is performed by using low-level feature matrices of user and two items, which are extracted from the first layer (i.e., conv1) of the

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

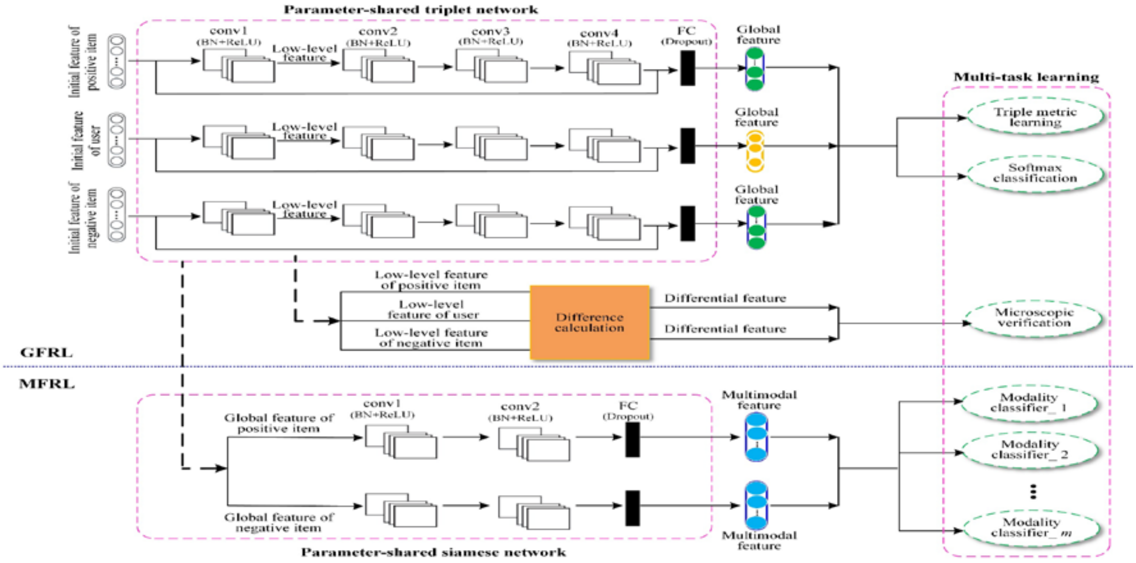


Fig. 1. The overall framework of Model Training in MRLM.

triplet network.

(*b*) MFRL takes the global feature vectors of two items from GFRL as input and uses a Siamese network to extract their multimodal features. Conceptually, a Siamese network [34] contains two copies of a base network with shared parameters. In this work, it consists of two three-layer CNN-based neural networks that have the same structure and parameters. For either item, its multimodal feature vector is extracted from the last layer (i.e., a fully-connected one) of the Siamese network and used to simultaneously optimize *m* modality classifiers. Once these modality classifiers achieve optimal performance, the obtained multimodal feature vector is the most accurate one.

It is noteworthy that the positions of the triplet network and Siamese network cannot be swapped, as shown in Fig. 1, since GFRL and MFRL require different input format (a triplet for GFRL and a two-tuple for MFRL, respectively).

After Model Training is completed, for each item, the multimodal feature vector is used as its final feature representation. And for each user, the concatenation of global feature vector and latest interacted items' multimodal feature vectors are used as his final feature representation. For a given a pair of user and item, the preference of user on item can be obtained by calculating the cosine similarity between their feature representations. Note that the cosine similarity is adopted here since it is also used as a preference distance measure in the triple metric learning task (see Equation (3) for details). The similarity calculation method used in training and testing phases should be consistent.

3.2 GFRL

The process of global feature representation learning is described in detail. (u, v^+, v^-) is used as a triplet example, where u is a user, v^+ is a positive item, and v^- is a negative item. Its corresponding triplet of initial feature vectors are denoted as ($\mathbf{u}_0$, $\mathbf{v}_0^+$, $\mathbf{v}_0^-$), input of the triplet network.

As shown in Fig. 1, the triplet network contains four parallel CNN-based neural networks that have the same structure and parameters. Each neural network consists of five layers: the first four layers are convolutional ones (i.e., conv1~conv4) and the fifth layer is a fully-connected one (i.e., FC). To avoid gradient vanishing and to speed up training, each convolutional layer is followed by a BN (Batch Normalization) operation and a ReLU (Rectified Linear Unit) activation function. While a dropout operation is executed before FC.

Specially, the output of the i-th ($1 \le i \le 4$) convolutional layer can be expressed as:

$$\mathbf{c}_i = \mathrm{ReLU}(\mathbf{k}_{it} * \mathbf{c}_{i-1}), t = 1, 2, \dots, \mu_i, \qquad (1)$$

where μ_i is the number of convolution kernels in this layer, and $\mathbf{k}_{it}$ is the t-th convolution kernel. Note that, $\mathbf{c}_0$ is the input vector of the triplet network (i.e., $\mathbf{u}_0$, $\mathbf{v}_0^+$, or $\mathbf{v}_0^-$).

The output of FC (i.e., global feature vector) can be expressed as:

$$\mathbf{g} = f(\mathbf{W}\mathbf{c}_4 + \mathbf{b}), \qquad (2)$$

where f is the Sigmoid activation function, $\mathbf{W}$ and $\mathbf{b}$ are the parameters to be learned through training.

Let $\mathbf{g}_u$, $\mathbf{g}_v^+$, and $\mathbf{g}_v^-$ denote the global feature vectors of u, v^+, and v^-, respectively, and their dimensionality be d_1. These three global feature vectors are then employed to simultaneously carry out two tasks, i.e., triple metric learning and softmax classification, respectively.

The triple metric learning task is designed to minimize the preference distance between $\mathbf{g}_u$ and $\mathbf{g}_v^+$, and to maximize the preference distance between $\mathbf{g}_u$ and $\mathbf{g}_v^-$. Therefore, the loss function of this task can be defined as:

$$\mathcal{L}_{g1} = \frac{1}{|B|}\sum_{(u,v^+,v^-)\in B}(cos(\mathbf{g}_u, \mathbf{g}_v^+) - cos(\mathbf{g}_u, \mathbf{g}_v^-)), \qquad (3)$$

where B is a mini-batch of training samples, $|B|$ is the size of B, and $cos(\cdot)$ is the cosine distance formula.

The softmax classification task is given for ID recognition of u, v^+, and v^- based on their global feature vectors $\mathbf{g}_u$, $\mathbf{g}_v^+$, and $\mathbf{g}_v^-$, respectively. User u is used in the follow-

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

ing as an example to demonstrate this task.

Assume that there are η users in the recommender system. Let u be the i-th ($i\in[1, \eta]$) user among them. First, $\mathbf{g}_u$ is fed into a softmax layer to produce η output values $<s_1, s_2,\dots, s_\eta>$, and each output value corresponds to one user. Then, s_i is normalized to get the probability value:

$$\tilde{s}_i = \frac{e^{s_i - max(<s_1,s_2,\dots,s_\eta>)}}{\sum_{y=1}^{\eta} e^{s_y - max(<s_1,s_2,\dots,s_\eta>)}}, \tag{4}$$

where $max(\cdot)$ is the max function. Hence, the sample-level loss of u is equal to $-log\tilde{s}_i$.

Similarly, assume that there are δ items in the recommender system. Let v^+ and v be the x-th and z-th items among them, respectively. Then, the sample-level losses of v^+ and v^- can be obtained as $-log\tilde{p}_x$ and $-log\tilde{p}_z$, where $\tilde{p}_x$ and $\tilde{p}_z$ are their corresponding probability values, respectively.

Thereby, the loss function of this task is:

$$\mathcal{L}_{g2} = -\frac{1}{|B|}\sum_{(u,v^+,v^-)\in B}(log\tilde{s}_u + log\tilde{p}_{v^+} + log\tilde{p}_{v^-}), \tag{5}$$

where $\tilde{s}_u$, $\tilde{p}_{v^+}$, and $\tilde{p}_{v^-}$ are the corresponding probability values of u, v^+, and v^-, respectively.

For improving the accuracy of the global feature representation, inspired by [35], the microscopic verification task is designed to jointly optimize the aforementioned two tasks. Its core is a Difference Calculation Procedure (DCP).

DCP employs low-level feature matrices of a user and two items as inputs, which are extracted from the first convolutional layer of the triplet network in GFRL. Let them be denoted by $\mathbf{h}_u$, $\mathbf{h}_v^+$, and $\mathbf{h}_v^-$, respectively. First, it constructs two preference pairs, P^+=($\mathbf{h}_u$, $\mathbf{h}_v^+$) and P^-=($\mathbf{h}_u$, $\mathbf{h}_v^-$). Then, it calculates the element-level difference of either pair. P^+=($\mathbf{h}_u$, $\mathbf{h}_v^+$) is used in the following as an example to demonstrate the calculation process.

It is easy to see that $\mathbf{h}_u$ and $\mathbf{h}_v^+$ have the same size. Let its dimensionality be $d_2 \times d_3$, where d_2 and d_3 are the number of rows and columns, respectively. For each element $\mathbf{h}_u[i,j]$ ($1\leq i\leq d_2$, $1\leq j\leq d_3$), DCP firstly expands its value ε to a matrix of 3×3, denoted as $\mathbf{R}_u^{ij}$. The value of each element in $\mathbf{R}_u^{ij}$ equals ε. And then, DCP gets a submatrix of 3×3 in $\mathbf{h}_v^+$ whose center is the element $\mathbf{h}_v^+$ $[i,j]$, denoted as $\mathbf{R}_{v^+}^{ij}$. Thereby, the difference of P^+at the (i, j)-th element equals $\mathbf{R}_u^{ij} - \mathbf{R}_{v^+}^{ij}$. Fig. 2 demonstrates an example for calculating the element-level difference of P^+.

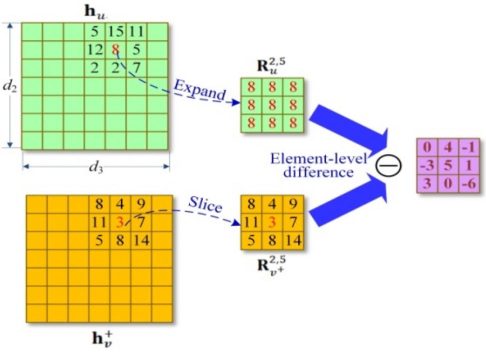


Fig. 2. An example of calculating element-level difference.

On this basis, DCP can eventually achieve the overall difference $\mathbf{R}_{P^+} \in \mathbb{R}^{d_2\times d_3\times 3\times 3}$ of P^+:

$$\mathbf{R}_{P^+} = \left(\mathbf{R}_u^{ij} - \mathbf{R}_{v^+}^{ij}\right)_{1\leq i\leq d_2, 1\leq j\leq d_3}. \tag{6}$$

Similarly, DCP can obtain the overall difference $\mathbf{R}_{P^-} \in \mathbb{R}^{d_2\times d_3\times 3\times 3}$ of P^-:

$$\mathbf{R}_{P^-} = \left(\mathbf{R}_u^{ij} - \mathbf{R}_{v^-}^{ij}\right)_{1\leq i\leq d_2, 1\leq j\leq d_3}. \tag{7}$$

Finally, DCP flattens $\mathbf{R}_{P^+}$and $\mathbf{R}_{P^-}$ as a one-dimensional vector $\overline{\mathbf{R}}_{P^+}$and $\overline{\mathbf{R}}_{P^-}$, respectively, and compresses them by using a same fully-connected layer:

$$\mathbf{C}_{P^+} = \mathbf{W}_c\overline{\mathbf{R}}_{P^+} + \mathbf{b}_c. \tag{8}$$

$$\mathbf{C}_{P^-} = \mathbf{W}_c\overline{\mathbf{R}}_{P^-} + \mathbf{b}_c. \tag{9}$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are the parameters to be learned through training. $\mathbf{C}_{P^+}$ and $\mathbf{C}_{P^-}$ are the output of DCP.

After obtaining $\mathbf{C}_{P^+}$ and $\mathbf{C}_{P^-}$, two input samples ($\mathbf{C}_{P^+}$, 1) and ($\mathbf{C}_{P^-}$, 1) are then constructed and fed into a fully-connected layer, respectively, to complete the microscopic verification task. This fully-connected layer is used as a two-class classifier. The output predicted values of classifier for the two samples are denoted as p_{u,v^+} and p_{u,v^-}, respectively. In particular, the binary cross-entropy is used as its loss function:

$$\mathcal{L}_{g3} = -\frac{1}{|B|}\sum_{(u,v^+,v^-)\in B}\left(logp_{u,v^+} + \log(1-p_{u,v^-})\right). \tag{10}$$

The above three loss functions are jointly optimized to realize the process of global feature representation learning in GFRL:

$$\mathcal{L}_{GFRL} = \lambda_1\mathcal{L}_{g1}+\lambda_2\mathcal{L}_{g2}+(1-\lambda_1-\lambda_2)\mathcal{L}_{g3}, \tag{11}$$

where λ_1, $\lambda_2\in(0, 1)$ are two hyper-parameters which control the importance of the three tasks.

3.3 MFRL

In this section, the process of multimodal feature representation learning is described in detail. MFRL aims to refine items' global feature vectors by using their multimodal description information, thus accurately capturing their deep semantic features.

It takes the global feature vectors $\mathbf{g}_v^+$ and $\mathbf{g}_v^-$ of two items from GFRL as input and uses a Siamese network to extract their multimodal feature vectors. As shown in Fig. 1, the Siamese network contains two parallel CNN-based neural networks that have the same structure and parameters. Each neural network includes three layers: the first two layers are convolutional ones and the third layer is a FC. Like the triplet network in GFRL, each convolutional layer is followed by a BN operation and a ReLU activation function. While a dropout operation is executed before FC.

In particular, the output of the i-th (i=1, 2) convolutional layer can be expressed as:

$$\mathbf{n}_i = \mathrm{ReLU}(\mathbf{e}_{it} * \mathbf{n}_{i-1}), t = 1,2,\dots,\kappa_i, \tag{12}$$

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

where κ_i is the number of convolution kernels in this layer, and $\mathbf{e}_{it}$ is the t-th convolution kernel. Note that, $\mathbf{n}_0$ is the input vector of the Siamese network (i.e., $\mathbf{g}_v^+$, or $\mathbf{g}_v^-$).

The output of FC (i.e., multimodal feature vector) can be expressed as:

$$\mathbf{m} = \mathrm{Sigmoid}(\mathbf{W}'\mathbf{n}_2 + \mathbf{b}'), \tag{13}$$

where $\mathbf{W}'$ and $\mathbf{b}'$ are the parameters to be learned through training.

Let $\mathbf{m}_v^+$ and $\mathbf{m}_v^-$ denote the multimodal feature vectors of v^+ and v^-, respectively, let their dimensionality be d_4. For accurately representing semantic features of items, m modality classifiers are jointly optimized to fine-tune $\mathbf{m}_v^+$ and $\mathbf{m}_v^-$, respectively. $\mathbf{m}_v^+$ is used in the following as an example to demonstrate this process.

Referring to Subsection 3.1 (Data Preprocessing), $\mathbf{m}_v^+$ has r modalities, and it has a unique cluster ID on each modality. Each classifier is designed to correspond to the modality of $\mathbf{m}_v^+$, which performs a multi-classification task on this modality. Specifically, for the i-th ($1 \le i \le r$) classifier, $\mathbf{m}_v^+$ is fed into a softmax layer to generate τ_i output values $< o_1^+, o_2^+, \ldots, o_{\tau_i}^+ >$, and each output value corresponds to one cluster ID on the modality. Let the cluster ID of v^+ be l ($1 \le l \le \tau_i$). Then, similar to Equation (4), o_l is normalized and the probability value is obtained:

$$o_l^+ = \frac{e^{o_l^+ - max\left(<o_1^+, o_2^+, \ldots, o_{\tau_i}^+>\right)}}{\sum_{y=1}^{\tau_i} e^{o_y^+ - max\left(<o_1^+, o_2^+, \ldots, o_{\tau_i}^+>\right)}}. \tag{14}$$

Hence, the loss with respect to $\mathbf{m}_v^+$ is equal to $-log o_l^+$. Similarly, the loss with respect to $\mathbf{m}_v^-$ is $-log o_c^-$, where c is the cluster ID of v^-.

On this basis, the loss function of the i-th classifier is:

$$\mathcal{L}_{mi} = -\frac{1}{|B|}\sum_{(u,v^+,v^-)\in B}(log o_l^+ + log o_c^-). \tag{15}$$

Thereby, the joint loss function of multimodal feature representation learning in MFRL can be defined as:

$$\mathcal{L}_{MFRL} = \upsilon_1\mathcal{L}_{m1} + \upsilon_2\mathcal{L}_{m2} + \ldots + \upsilon_r\mathcal{L}_{mr}, \tag{16}$$

where $\upsilon_1 \sim \upsilon_r \in (0, 1)$ are r hyper-parameters which control the importance of r modalities, and $\sum_{z=1}^{r} \upsilon_z = 1$.

Finally, the loss functions of GFRL and MFRL are combined to realize the training of MRLM:

$$\mathcal{L}_{MRLM} = \theta_1\mathcal{L}_{GFRL} + (1 - \theta_1)\,\mathcal{L}_{MFRL}, \tag{17}$$

where $\theta_1 \in (0, 1)$ is the hyper-parameter that control the importance of two modules .

3.4 Complexity Analysis

Suppose that there are m users and n items having r modalities in a given recommender system, and on average, every user has rated $\bar{n}$ items ($r \ll n, \bar{n} \ll n$). Then, MRLM needs time $O(m + n + rn + rn^2) \approx O(m + rn^2)$ for data preprocessing, and time $O((2 + 3 + 2 + 2r)m\bar{n}^2))$= $O((2r + 7)m\bar{n}^2))$ for model training. Hence, its time complexity is $O(rn^2 + (2r + 7)m\bar{n}^2))$.

Let the dimensionality of auxiliary feature vectors of items be $\bar{d}$. Then, MRLM needs space O_{DP}=max{ $O(w_0)$, $O(w_1)$,…, $O(w_r)$, $O(n\bar{d})$} for data preprocessing, where w_0 is the number of parameters in xDeepFM [8], and w_i ($1 \le i \le r$) is the number of parameters in the model used for the i-th modality of an item. On the other hand, let the dimensionalities of initial feature vectors, global feature vectors, and multimodal feature vectors be d_0, d_1, and d_4, respectively. Then, MRLM needs space 0_{MT}=$O(3\left(\sum_{i=1}^{4} k_i^2 \mu_{i-1}\,\mu_i + (\mu_4\psi + d_0)d_1\right) + 2(\sum_{t=1}^{2} e_t^2 \kappa_{t-1}\,\mu_t + \kappa_2\phi d_4))$ for model training, where k_i and μ_i are the size and the number of kernels in the i-th convolutional layer in GFRL, respectively, ψ is the size of feature maps in the fourth convolutional layer in GFRL, e_t and κ_t are the size and the number of kernels in the t-th convolutional layer in MFRL, respectively, and ϕ is the size of feature maps in the second convolutional layer in MFRL. Thus, its space complexity is max{O_{DP}, O_{MT}}.

4 Experimental Evaluation

In this section, an empirical study on the proposed MRLM model with two real-world datasets in IoT was conducted.

4.1 Experimental setup

Two real-world datasets were employed in the experiments: Movielens-20M[1] and BookCrossing[2]:

(1) MovieLens-20M. It is one of the most widely used datasets in recommendation system domain, containing information about users, movies, and ratings that users give to movies. In the experiments, according to previous works, the ratings greater than or equal to 4 were treated as positive feedback, and the ratings less than 4 were treated as negative feedback. Only the users with more than 20 ratings were considered in the experiments.

(2) BookCrossing. It is a prevalent book dataset, containing the information about users, books, and ratings. According to previous works, the ratings greater than or equal to 5 were treated as positive feedback, and the ratings less than 5 were treated as negative feedback.

In MovieLens-20M, description information of four modalities was introduced for each movie: (*a*) For text modality, the text summary extracted from its plot was used; (*b*) For image modality, its poster image was used; (*c*) For video modality, its promotional video was used; and (*d*) For knowledge-base modality, its directly adjacent entities and relationships were utilized in KB4Rec [36].

In BookCrossing description information of two modalities was produced for each book: (*a*) For text modality, its brief introduction was used; (*b*) For image modality, its front cover image was used. In addition, Transformer [29], ResNet-50 [30], ECNN [31], and TransG [37] were used to realize feature extraction for the modalities of text, image, video, and knowledge-base, respectively.

Table 1 shows the statistical data of two real-world datasets used in the experiments.

In the experiments, the datasets were randomly divided into training (70%), validation (20%), and test (10%)

[1] https://grouplens.org/datasets/movielens/20m

[2] https://grouplens.org/datasets/book-crossing

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

TABLE 1
THE STATISTICS OF TWO DATASETS USED IN EXPERIMENTS

	Movielens-20M	BookCrossing
Users	138,493	278,858
Items	27,278	271,379
Ratings	20,000,263	1,149,780
Ratings per user	144.4	4.1
Ratings per item	733.2	4.2
Items having texts	26,012	259,385
Items having images	27,084	263,194
Items having videos	24,616	0
Items in KB	25,982	0

sets, where the training set for model training, the validation set for hyper-parameters tuning and the test set for performance evaluation, respectively.

Grid search was carried out for MRLM to find the hyper-parameters to achieve the best recommendation performance in the validation set. Table 2 shows the hyper-parameter settings of MRLM in the experiments (DP: Data Preprocessing; MT: Model Training).

TABLE 2
THE HYPER-PARAMETER SETTINGS OF MRLM OF TWO IN THE EXPERIMENTS

	Movielens-20M	BookCrossing
DP	d_0=250, τ_1=25, τ_2=20, τ_3=25, τ_4=20	d_0=200, τ_1=25, τ_2=20
MT	μ_1=20, μ_2=30, μ_3=30, μ_3=15, d_1=300, d_2=20, d_3=125, λ_1=0.4, λ_2=0.2, κ_1=20, κ_2=30, d_4=250, υ_1=0.3, υ_2=0.25, υ_3=0.25, υ_4=0.2, θ_1=0.55, θ_2=0.45	μ_1=20, μ_2=30, μ_3=30, μ_3=15, d_1=250, d_2=20, d_3=100, λ_1=0.5, λ_2=0.2, κ_1=20, κ_2=30, d_4=250, υ_1=0.6, υ_2=0.4, θ_1=0.5, θ_2=0.5
Optimizer	B=128, *Learning rate*=0.001	B=64, *Learning rate*=0.001

MRLM was implemented in Tensorflow. All the models were trained with GPU acceleration.

As for evaluation metrics, two well-known metrics, widely applied for Top-*N* recommendation evaluation, Recall@*N* [38] and AUC@*N* [39], were adopted. AUC represented the area under ROC (Receiver Operating Characteristic) curve. Recall represented the percentage of correctly predicted true positive items in the samples:

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}), \tag{18}$$

where TP is the number of positive items that are correctly predicted to be true, and FN is the number of positive items that are falsely predicted to be false.

For verifying the effectiveness of MRLM, the experimental results by MRLM were compared with the results by ten state-of-the-art models, i.e., xDeepFM [8], Covington [9], SRSHAN [11], ConvMF [12], NARRE [16], HAUP [17], CAMN [22], CKE [23], ORAMAS [24], and MRTA [25]. For comparison, the extended versions of these ten models were considered, which were incorporated with all modalities via a straightforward way presented in Section 1. For simplicity, these extended models were denoted as xDeepFM+, Covington+, SRSHAN+, ConvMF+, NARRE+, HAUP+, CAMN+, CKE+, ORAMAS+, and MRTA+, respectively.

4.2 Performance Comparison

The performance of MRLM was compared with that of its peers. Tables 3 and 4 show the values of Recall@*N* and AUC@*N* for the two datasets with *N*={10, 20, 50, 100}.

From Tables 3 and 4, MRLM achieves the superior recommendation accuracy. For example, in Table 3, it outperforms xDeepFM, Covington, SRSHAN, ConvMF, NARRE, HAUP, CAMN, CKE, ORAMAS, and MRTA by 62.39%, 74.27%, 49.76%, 78.20%, 50.65%, 43.37%, 35.72%, 27.62%, 40.66%, and 41.59%, respectively, for Recall@100 on the dataset MovieLens-20M. While it outperforms their extended models by 57.91%, 62.19%, 38.58%, 68.83%, 44.83%, 24.37%, 24.07%, 21.21%, 22.38%, and 22.74%, respectively. The main reasons are two-folds. (*a*) MRLM can accurately represent the global features of items and users by jointly training three tasks: triplet metric learning, softmax classification, and microscopic verification; and (*b*) more important, it can efficiently refine global features of items through deeply fusing their multimodal description information, and therefore, can accurately capture their deep semantic features.

Furthermore, for each state-of-the-art model, it slightly underperforms its extended version in most cases. For example, in Table 4, xDeepFM underperforms xDeepFM+ by 2.45% for AUC@100 on the dataset BookCrossing. It is mainly due to the fact that its extended model simply integrates description information of items' different modalities, thus slightly improving the accuracy of item feature representation. Please note that for either CAMN or CKE, its accuracy is the same as that of its extended version on the dataset BookCrossing. It is mainly due to the fact that BookCrossing only has two modalities (i.e., text and image), which is not helpful for their extended versions.

4.3 Effectiveness of Item Multimodal Features

The overall performance comparison shows that MRLM has a higher recommendation effectiveness in IoT. For further understanding the importance of item multimodal features, the "ablation" study was carried out.

Firstly, MRLM was compared with its variant U-GFRL representing that only the GFRL module was used, i.e., only the global features of items were used. Recall@*N* and AUC@*N* were used as the experimental evaluation metrics for two datasets with *N*={50, 100}. The experimental results are shown in Fig. 3.

Figure 3 depicts that MRLM outperforms U-GFRL in all cases. For example, MRLM outperforms U-GFRL by 49.35% and 66.27% for Recall@50 on MovieLens-20M and BookCrossing, respectively, as shown in Fig. 3 (*a*). It clearly indicates that recommendation is not effective if only item global features were used. The final recommendation effectiveness can be effectively improved by fusing item multimodal features.

Next, the importance of different modalities was studied. Specifically, MRLM was compared with its four variants:

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

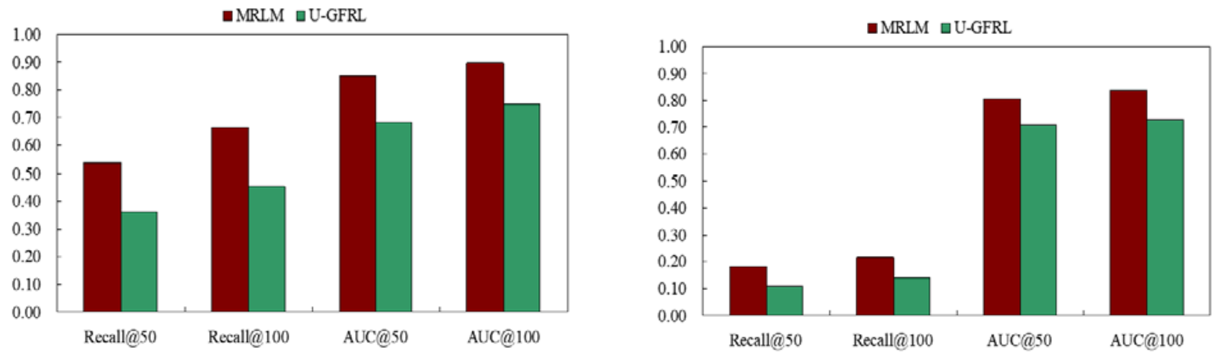
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

TABLE 3
THE RECALL@*N* VALUES OF MRLM AND 20 BASELINES (BEST RESULTS ARE BOLD-FACED)

Model	Dataset							
	MovieLens-20M				BookCrossing			
	N=10	*N*=20	*N*=50	*N*=100	*N*=10	*N*=20	*N*=50	*N*=100
xDeepFM	0.2422	0.2685	0.3217	0.4092	0.0574	0.0689	0.0773	0.0988
Covington	0.2301	0.2570	0.3094	0.3813	0.0491	0.0622	0.0696	0.0893
SRSHAN	0.2596	0.2834	0.3503	0.4437	0.0652	0.0770	0.0837	0.1092
ConvMF	0.2237	0.2496	0.3018	0.3729	0.0401	0.0508	0.0632	0.0925
NARRE	0.2525	0.2801	0.3447	0.4411	0.0615	0.0717	0.0798	0.1067
HAUP	0.2608	0.2923	0.3701	0.4635	0.0728	0.0947	0.1121	0.1453
CAMN	0.2651	0.3014	0.3950	0.4896	0.1013	0.1372	0.1509	0.1785
CKE	0.2929	0.3472	0.4173	0.5207	0.1026	0.1431	0.1586	0.1863
ORAMAS	0.2633	0.2985	0.3776	0.4724	0.0798	0.1015	0.1193	0.1506
MRTA	0.2620	0.2961	0.3722	0.4693	0.0785	0.1004	0.1178	0.1491
xDeepFM+	0.2594	0.2818	0.3472	0.4208	0.0649	0.0701	0.0792	0.1044
Covington+	0.2466	0.2691	0.3310	0.4097	0.0519	0.0690	0.0743	0.0976
SRSHAN+	0.2712	0.3044	0.3813	0.4795	0.0824	0.1205	0.1281	0.1504
ConvMF+	0.2339	0.2607	0.3179	0.3936	0.0502	0.0675	0.0765	0.0910
NARRE+	0.2625	0.2916	0.3685	0.4588	0.0783	0.1162	0.1208	0.1425
HAUP+	0.2842	0.3495	0.4218	0.5343	0.1008	0.1315	0.1474	0.1716
CAMN+	0.2897	0.3582	0.4274	0.5356	0.1013	0.1372	0.1509	0.1785
CKE+	0.3005	0.3697	0.4361	0.5482	0.1026	0.1431	0.1586	0.1863
ORAMAS+	0.2913	0.3624	0.4319	0.5430	0.1019	0.1390	0.1531	0.1768
MRTA+	0.2901	0.3605	0.4302	0.5414	0.1016	0.1387	0.1510	0.1742
MRLM	**0.3516**	**0.4483**	**0.5402**	**0.6645**	**0.1204**	**0.1638**	**0.1819**	**0.2172**

TABLE 4
THE AUC@*N* VALUES OF MRLM AND 20 BASELINES (BEST RESULTS ARE BOLD-FACED)

Model	Dataset							
	MovieLens-20M				BookCrossing			
	N=10	*N*=20	*N*=50	*N*=100	*N*=10	*N*=20	*N*=50	*N*=100
xDeepFM	0.6182	0.6305	0.6551	0.6936	0.5795	0.5923	0.6031	0.6283
Covington	0.5750	0.6029	0.6186	0.6628	0.5637	0.5801	0.5905	0.6047
SRSHAN	0.6619	0.6697	0.6940	0.7502	0.6318	0.6537	0.6700	0.6891
ConvMF	0.5624	05982	0.6101	0.6457	0.5356	0.5614	0.5726	0.5984
NARRE	0.6437	0.6503	0.6684	0.7288	0.6243	0.6492	0.6605	0.6801
HAUP	0.6725	0.6892	0.7120	0.7601	0.6652	0.7030	0.7112	0.7346
CAMN	0.6812	0.7091	0.7218	0.7436	0.6697	0.7102	0.7194	0.7405
CKE	0.7321	0.7705	0.7895	0.8147	0.7028	0.7263	0.7352	0.7582
ORAMAS	0.6785	0.7013	0.7166	0.7350	0.6601	0.7025	0.7106	0.7324
MRTA	0.6741	0.6926	0.7104	0.7315	0.6553	0.6962	0.7058	0.7240
xDeepFM+	0.6302	0.6495	0.6752	0.7275	0.5984	0.6158	0.6215	0.6437
Covington+	0.6081	0.6208	0.6437	0.7094	0.5892	0.5986	0.6141	0.6295
SRSHAN+	0.7024	0.7286	0.7498	0.8042	0.6715	0.6932	0.7183	0.7349
ConvMF+	0.5993	0.6114	0.6295	0.6835	0.5704	0.5875	0.5994	0.6182
NARRE+	0.6726	0.7035	0.7212	0.7731	0.6528	0.6804	0.7010	0.7159
HAUP+	0.7185	0.7411	0.7606	0.8215	0.7027	0.7319	0.7472	0.7667
CAMN+	0.7248	0.7530	0.7751	0.8289	0.6697	0.7102	0.7194	0.7405
CKE+	0.7394	0.7798	0.8024	0.8465	0.7028	0.7263	0.7352	0.7582
ORAMAS+	0.7286	0.7617	0.7845	0.8373	0.7074	0.7169	0.7297	0.7491
MRTA+	0.7245	0.7592	0.7809	0.8324	0.7040	0.7151	0.7216	0.7435
MRLM	**0.7862**	**0.8237**	**0.8514**	**0.8976**	**0.7611**	**0.7895**	**0.8066**	**0.8372**



(*a*) The dataset MovieLens-20M

(*b*) The dataset BookCrossing

Fig.3. Experimental evaluation for item multimodal features.

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

(*a*) MRLM-T: in MRLM, only text modality was used.

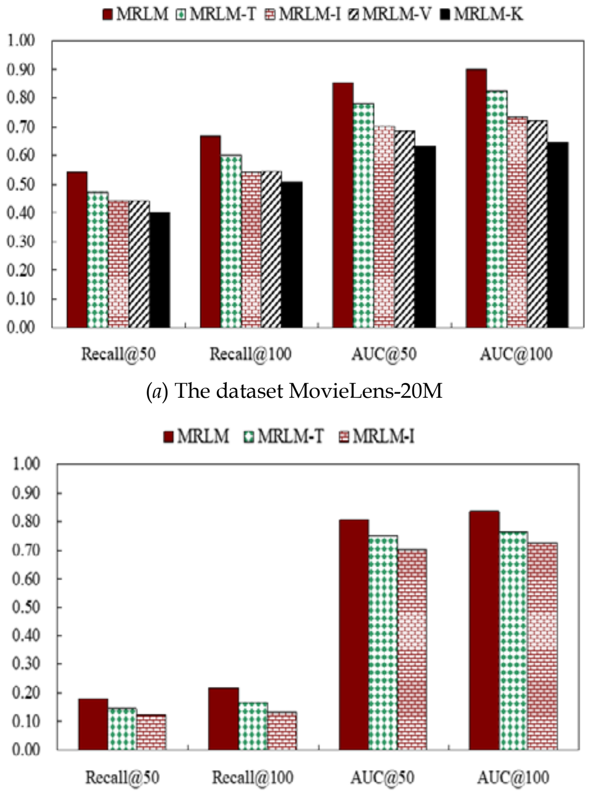
(*b*) MRLM-I: in MRLM, only image modality was used.

(*c*) MRLM-V: in MRLM, only video modality was used.

(*d*) MRLM-K: in MRLM, only knowledge-base modality was used.

Note that since BookCrossing only has two modalities: text and image, MRLM-V and MRLM-K do not be implemented in it.

Similarly, Recall@*N* and AUC@*N* were used as experimental evaluation metrics for two datasets with *N*={50, 100}. The experimental results are shown in Fig. 4.



(*a*) The dataset MovieLens-20M

(*b*) The dataset BookCrossing

Fig.4. Experimental study for the importance of different modalities.

Compared with different variants, MRLM obtains the superior recommendation accuracy. For example, it outperforms its four variants by 9.01%, 22.04%, 24.65% and 39.66%, respectively, for AUC@100 on MovieLens-20M, as shown in Fig. 4 (*a*). Item's each modality is beneficial to the final recommendation, and combining them leads to the highest effectiveness. On the other hand, MRLM-T has the highest accuracy among the four variants. For example, in Fig. 4 (*a*), MRLM-T averagely outperforms MRLM-I, MRLM-V, and MRLM-K by 6.38%, 7.16% and 12.34%, respectively, on MovieLens-20M. It indicates that in the two datasets, item's text modality can provide more useful auxiliary information than item's other modalities for final recommendation.

4.4 Study on the Ratio of Training Set

In previous experiments, the ratio of the training set was fixed to 70% and good experimental results were achieved. In this subsection, the effect of the ratio on the accuracy of the proposed model was investigated.

The ratios of the training set varied from 55% to 75% over the whole dataset at a 5% incremental, with a fixed validation set size (20%) and using the rest of data as the test set. AUC@50 and AUC@100 were used as the evaluation metrics for two datasets. The experimental results are shown in Fig. 5.

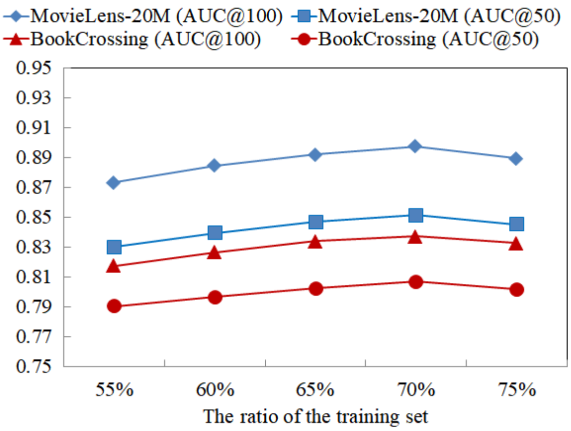


Fig.5. Experimental study for the ratio of the training set.

As the ratio of the training set increases, the recommendation accuracy first increases until reaches the maximum at 70%, then decreases. For example, the AUC@100 values equal 0.8732, 0.8845, 0.8921, 0.8976, and 0.8893 at 55%, 60%, 65%, 70%, and 75%, respectively, on MovieLens-20M. This is because insufficient training data may underfit the model (i.e., make the model less generalizable), while too much training data may lead to model overfitting.

5 Conclusion

In this paper, a novel multimodal representation learning-based model was introduced to address the challenge that item description information are heterogeneous and multimodal in IoT. During training, its two closely related modules, GFRL and MFRL were simultaneously optimized. In GFRL, a five-layer CNN-based triplet network was employed to extract global and low-level features for a user and a pair of positive and negative items. In particular, global features were used to perform two tasks of triple metric learning and softmax classification. Low-level features were used to perform the microscopic verification task. In MFRL, item's global features were taken as input and their multimodal description information were fused. Specially, a three-layer CNN-based Siamese network was used to extract multimodal features of positive and negative items, and all modality classifiers were jointly optimized to fine-tune them. The experimental results showed that MFRL can effectively improve the effectiveness of recommendation in IoT.

Future work includes using more types of item auxiliary information to further improve the recommendation effectiveness in IoT and designing more efficient neural networks to effectively improve the accuracy of feature extraction.

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 61772366) and the Natural Science Foundation of Shanghai (No. 17ZR1445900).

REFERENCES

[1] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017.

[2] Z. Huang et al., "An efficient passenger-hunting recommendation framework with multi-task deep learning," *IEEE Internet of Things Journal*, Feb. 2019. doi: 10.1109/JIOT.2019.2901759.

[3] N. Sachdeva, K. Gupta, and V. Pudi, "Attentive neural architecture incorporating song features for music recommendation," in *Proc. RecSys*, 2018, pp. 417–421.

[4] D. Cao et al., "Attentive group recommendation," in *Proc. SIGIR*, 2018, pp. 645–654.

[5] P. P. Zhao et al., "A generative model approach for geo-social group recommendation," *J. Comput. Sci. Tech.*, vol. 33, no. 4, pp. 727–738, Sep. 2018.

[6] Z. Huang et al., "TRec: An efficient recommendation system for hunting passengers with deep neural networks," *Neural Comput. Appl.*, pp. 1–14, Sep. 2018. doi: 10.1007/s00521-018-3728-2.

[7] H. T. Cheng et al., "Wide & deep learning for recommender systems," in *Proc. DLRS*, , 2016, pp.7–10.

[8] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, and Z. Dong, "DeepFM: An end-to-end wide & deep learning framework for CTR prediction," 2018. *arXiv preprint arXiv:1804.04950*.

[9] J. Lian et al., "xDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. SIGKDD*, 2018, pp. 1754–1763.

[10] P. Covington et al., "Deep neural networks for youtube recommendations," in *Proc. RecSys*, 2016, pp. 191–198.

[11] H. Ying et al., "Sequential recommender system based on hierarchical attention networks," in *Proc. IJCAI*, 2018, pp. 3926–3932.

[12] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. RecSys*, 2016, pp. 233–240.

[13] S. Okura et al., "Embedding-based news recommendation for millions of users," in *Proc. SIGKDD*, 2017, pp.1933–1942.

[14] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proc. RecSys*, 2017, pp. 297–305.

[15] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proc. SIGKDD*, 2018, pp. 2309–2318.

[16] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. WWW*, 2018, pp. 1583–1592.

[17] S. Xing, Q. Wang, X. Zhao, and T. Li, "A hierarchical attention model for rating prediction by leveraging user and product reviews," *Neurocomputing*, vol. 332, pp. 417–427, Mar. 2019.

[18] Y. Xu et al., "Exploiting the sentimental bias between ratings and reviews for enhancing recommendation," in *Proc. ICDM*, 2018, pp. 1356–1361.

[19] S. Wang et al., "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in *Proc. WWW*, 2017, pp. 391–400.

[20] S. Liu and M. Li, "Multimodal GAN for energy efficiency and cloud classification in Internet of Things," *IEEE Internet of Things Journal*, Aug. 2018. doi: 10.1109/JIOT.2018.2866328.

[21] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, "Hashtag recommendation for multimodal microblog using co-Attention network," in *Proc. IJCAI*, 2017, pp. 3420–3426.

[22] R. Ma, Q. Zhang, J. Wang, L. Cui, and X. Huang, "Mention recommendation for multimodal microblog with cross-attention memory network," in *Proc. SIGIR*, 2018, pp. 195–204.

[23] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. SIGKDD*, 2016, pp. 353–362.

[24] S. Oramas, O. Nieto, M. Sordo, X. Serra, "A deep multimodal approach for cold-start music recommendation," in *Proc. DLRS*, 2017, pp. 32–37.

[25] K. Bougiatiotis and T .Giannakopoulos, "Multimodal content representation and similarity ranking of movies," 2017. *arXiv preprint arXiv: 1702.04815*.

[26] S. Yu et al. "Optimized data fusion for kernel k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031–1039, May 2012.

[27] Z. Huang, Y. Xiang, B. Zhang, and X Liu, "A clustering based approach for skyline diversity," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 7984–7993, Jul. 2011.

[28] Z. Huang, S. E, J. Zhang, B. Zhang, and Z. Ji, "Pairwise learning to recommend with both users' and items' contextual information," *IET Commun.*, vol. 10, no. 16, pp. 2084–2090, Nov. 2016.

[29] L. Zhou et al., "End-to-end dense video captioning with masked transformer," in *Proc. CVPR*, 2018, pp. 8739–8748.

[30] Z. Wu, C. Shen, A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119 –133, Jun. 2019.

[31] X. Yang, T. Zhang, and C. Xu, "Semantic feature mining for video event understanding," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 4, pp. 1–22, Aug. 2016.

[32] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. IWSBPR*, 2015, pp. 84–92.

[33] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proc. ICCV*, 2015, pp. 2794–2802.

[34] S. Chopra, R. Hadsell, and Y. LeCun, Learning a similarity metric discriminatively, with application to face verification," in *Proc. CVPR(1)*, 2005, pp. 539–546.

[35] Z. Zhang, Y. Xu, L. Shao, and J. Yang, "Discriminative block-diagonal representation learning for image recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3111–312, Jul. 2017.

[36] W. X. Zhao, et al., "Kb4rec: A dataset for linking knowledge bases with recommender systems," 2018. *arXiv preprint arXiv:1807.11141*.

[37] H. Xiao, M. Huang, and X. Zhu, "TransG: A generative model for knowledge graph embedding," in *Proc. ACL*, 2016, pp. 2316–2325.

[38] Z. Huang, C. Yu, J. Cheng, and Z Wang, "UIContextListRank: A listwise recommendation model with social contextual information," in *Proc. APWeb -WAIM*, 2018, pp. 207–215.

[39] S. Li, A. Pandharipande, B. Masini, et al., "Automated detection of commissioning changes in connected lighting systems," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 898–905, Jan. 2019.

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2940709, IEEE Internet of Things Journal

Zhenhua Huang received the Ph.D. degree in the Computer Science from Fudan University, Shanghai, China, in 2008. From 2012 to 2016, he was an associate professor in School of Electronics and Information Engineering at Tongji University. From 2016 to 2018, he was a professor in School of Electronics and Information Engineering at Tongji University. He is currently a professor in the School of Computer Science at South China Normal University, Guangzhou, China. His research interests mainly include deep learning, Internet of Things, recommendation system, data mining, knowledge discovery and big data. Since 2004, he has published three books and more than 80 papers in various journals and conference proceedings.

Xin Xu received the B. S. degree in the Internet of Things Engineering from Shaoyang University. He is currently working toward the master's degree with Computer Science at South China Normal University. His main research interests mainly include Internet of Things, recommendation system, big data, deep learning and data mining.

Juan Ni received the M.S. degree from East China Normal University, Shanghai, China, in 2011. She is currently a teacher in the School of Politics & Administration, South China Normal University. Her main research interests include educational data mining, educational knowledge graph, big data analysis and recommendation system. She has published 10 papers in various journals and conference proceedings.

Honghao Zhu received the B.S. degree in Computer Science from Huaibei Normal University, China, in 2003, and the M.S. degree from University of Electronic Science and Technology of China, China, in 2009. He is currently pursuing the Ph.D. degree with Department of Computer Science at Tongji University, China. He is also a Lecturer with Bengbu University, China. His current research interests mainly include machine learning, evolutionary algorithms, big data and credit card fraud detection.

Cheng Wang received the M.S. degree in the Department of Applied Mathematics from Tongji University in 2006 and the PhD degree in the Department of Computer Science at Tongji University in 2011. He is currently a professor in the Department of Computer Science and Engineering at Tongji University. His research interests include deep learning, data mining, wireless privacy and intelligent transportation system.

2327-4662 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.