MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation

Qiang Cui, Shu Wu, *Member, IEEE*, Qiang Liu, Wen Zhong, and Liang Wang, *Fellow, IEEE*

Abstract—Sequential recommendation is a fundamental task for network applications, and it usually suffers from the item cold start problem due to the insufficiency of user feedbacks. There are currently three kinds of popular approaches which are respectively based on matrix factorization (MF) of collaborative filtering, Markov chain (MC), and recurrent neural network (RNN). Although widely used, they have some limitations. MF based methods could not capture dynamic user's interest. The strong Markov assumption greatly limits the performance of MC based methods. RNN based methods are still in the early stage of incorporating additional information. Based on these basic models, many methods with additional information only validate incorporating one modality in a separate way. In this work, to make the sequential recommendation and deal with the item cold start problem, we propose a **M**ulti-**V**iew **R**recurrent **N**eural **N**etwork (*MV-RNN*) model. Given the latent feature, MV-RNN can alleviate the item cold start problem by incorporating visual and textual information. First, At the input of MV-RNN, three different combinations of multi-view features are studied, like concatenation, fusion by addition and fusion by reconstructing the original multi-modal data. MV-RNN applies the recurrent structure to dynamically capture the user's interest. Second, we design a separate structure and a united structure on the hidden state of MV-RNN to explore a more effective way to handle multi-view features. Experiments on two real-world datasets show that MV-RNN can effectively generate the personalized ranking list, tackle the missing modalities problem, and significantly alleviate the item cold start problem.

Index Terms—Multi-view, sequential recommendation, recurrent neural network, cold start

IEEE Proof

1 INTRODUCTION

Recently, with the development of Internet, applications with sequential information have become numerous and multilateral, such as web page recommendation and click prediction. Based on sequential recommendation methods, these applications could predict a user's following behaviors to improve user experience. Taking online shopping as an example, after a user buys an item, the application would predict a list of items that the user might buy in the near future. Further, we can consider the purchase behaviors as a sequence in the time order. Due to sparse user feedbacks, sequential recommendation usually encounters the item cold start problem. Thus, our task here concentrates on the sequential recommendation based on user historical implicit feedback and alleviating the item cold start problem. As shown in Fig. 1, we observe that a user will look at corresponding images and text descriptions before he or she buys items. Intuitively, we can alleviate the item cold start problem by modeling additional multi-modal information like images and text descriptions. Besides, we try to find a more effective way of incorporating additional information into sequence modeling.

As for the recommendation, collaborative filtering methods are widely used. Matrix Factorization (MF) methods [1], [2], [3] become the first choice, and learn latent representations of users and items. In order to alleviate the cold start problem, multiple additional information can be adopted, such as attribute information [22], text [4], images [5], [6], and so on. Although these methods can utilize different types of features, they usually capture the user's static interest and have much difficulty in capturing sequential information. Long-term interest should be weakened while short-term interest should become prominent relatively [7].

On the other hand, Markov Chain (MC) methods [7], [8] are widely studied for sequential recommendation by learning the transition matrix. They predict the next behavior based on recent behaviors as the transition matrix gives the probability among different states. However, MC methods could not well build the user's long-term interest due to the Markov assumption. They usually consider recent behaviors and ignore the long-term interest. Besides, after constructing the real world dataset of sequential scenarios like shopping and clicking, the transition probability among different states is established. The additional information no longer has any effect on this probability.

Recently, Recurrent Neural Network (RNN) methods have shown great achievements in machine translation [9], sequential click prediction [10], location prediction [11], next basket recommendation [12], multi-behavioral sequential prediction [13], and so on. Besides, long short-term memory [14] and gated recurrent unit [15] are developed because of the gradient vanishing and explosion problem. They can hold the

- *Q. Cui, S. Wu, Q. Liu, and L. Wang are with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100864, China, and the University of Chinese Academy of Sciences (UCAS), Beijing 100190, China.*
E-mail: cuiqiang2013@ia.ac.cn, {shu.wu, qiang.liu, wangliang}@nlpr.ia.ac.cn.
- *W. Zhong is with the University of Southern California, Los Angeles, CA 90089. E-mail: wenzhong@usc.edu.*

Manuscript received 31 May 2017; revised 20 June 2018; accepted 2 Nov. 2018. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Shu Wu.)
Recommended for acceptance by E. Terzi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2881260

1041-4347 © 2018 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.

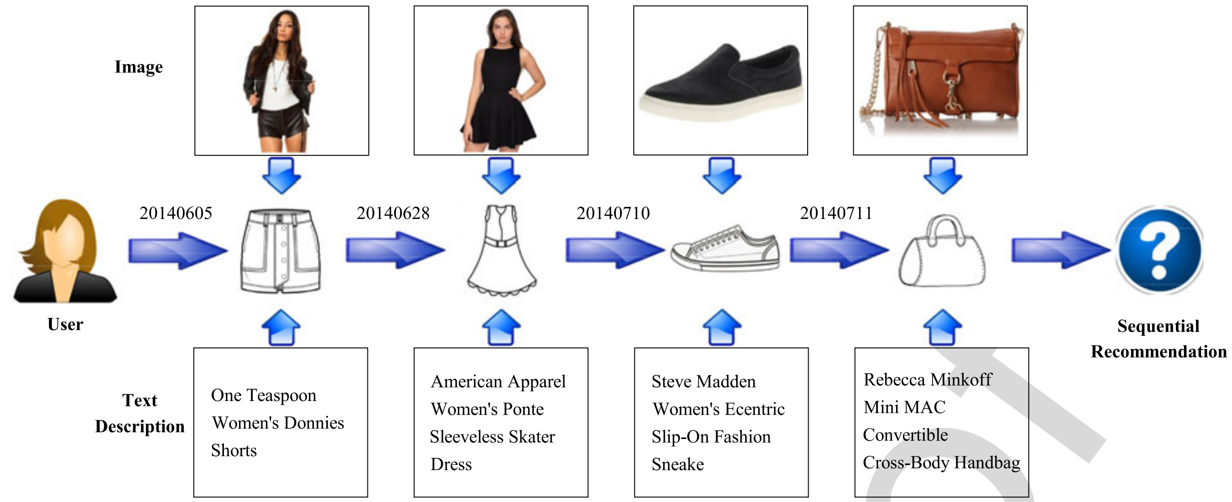


Fig. 1. Diagram of a user's purchase sequence. A user buys different items at different times. We make use of the image and text description associated with each item to build the sequential recommendation model. The goal is to recommend items a user would buy in the near future, and alleviate item cold start by incorporating multiple additional information into sequence modeling.

long-term dependency and have been applied to many tasks [16], [17], [18]. These RNN methods [11], [12], [13] are more promising than factorizing personalized markov chains [8] and other conventional MC methods.

The existing sequential recommendation methods have difficulty in alleviating the problem of item cold start. A good choice is to apply RNN and incorporate additional multi-modal features, like images and text descriptions. Recently, the parallel RNNs model (p-RNNs) [19] incorporates additional information for session-based recommendation. The p-RNNs model deals with multi-source data by separate subnets which are trained one by one. It builds multiple user's interests based on different views and combines the results at the end of each subset together. This way may not well leverage the advantage of multi-view data. We need to consider how to more effectively incorporate additional information to model sequential behaviors.

In view of the above analysis, we propose a model called **M**ulti-**V**iew **R**ecurrent **N**eural **N**etwork (*MV-RNN*) for sequential recommendation and alleviating the item cold start problem. First, we gain visual and textual features from images and text descriptions respectively. These multi-modal features are complementary to understand the item and user's interest. A latent vector is defined for each item to represent the indirectly observable representation. These multi-view features are used as the input of MV-RNN, and three different combinations are explored. Feature concatenation and fusion naturally come to mind. More importantly, we introduce a multi-modal fusion model, called multi-modal Marginalized Denoising AutoEncoder (3mDAE). This model can help to learn more robust features and handle items with missing modalities. Next, we design a separate structure and a united structure for MV-RNN to explore an effective way to handle multi-view features. One applies multiple RNN units separately at every input time, and multiple hidden states of these units are concatenated together at the same time. The other employs a single RNN unit to deal with the multi-view features at once to learn a united hidden state. The MV-RNN model adopts the recurrent structure to capture dynamic changes in user's interest. Finally, we employ the Bayesian personalized ranking framework [2] and the backpropagation through time algorithm [20] to learn parameters. The main contributions are listed as follows:

- We design a representation of item with multi-view features. These features comprise of indirectly observable (latent) feature and directly observable (e.g., visual and textual) feature. Three combinations of multi-view features are developed, especially our 3mDAE.
- To explore a more effective way to handle multi-view inputs, MV-RNN applies a separate structure and a united structure. Compared to dealing with each view separately, handling multi-view features by a united structure can better leverage the advantage of different views.
- Experiments on two large real-world datasets reveal that MV-RNN is effective and outperforms the state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 reviews previous work on sequential recommendation, cold start, and multi-modal representation learning. MV-RNN is detailly introduced in Section 3 from the perspective of input, hidden state, and output. In Section 4, we conduct extensive experiments. At last, we conclude the paper in Section 5.

2 Related Work

In this section, we review several related works including collaborative filtering, Markov chain based methods, recurrent neural networks, and multi-modal representation learning.

2.1 Collaborative Filtering

There are two main methods of Collaborative Filtering (CF): neighborhood models and latent factor models [21]. Neighborhood models have practical benefits, but they usually focus on a small subset of items or users. Latent factor models have the global perspective, and thus they tend to be more accurate. Recently, Matrix Factorization models belonging to latent factor models become fundamental because of its scalability and accuracy. MF absorbs rich additional information to alleviate the cold start problem, like item's attribute [22]. Text such as reviews is used along with the development of online searching [23]. Zhao et al. extend MF by combining visual data like posters and still frames of a movie to understand the movie and user's interest [6]. Besides MF, we can

IEEE Proof

also apply other methods to incorporate contextual information [51], [52]. However, none of these methods could reflect the changes in user's interest over time.

In recent years, pairwise methods become the state-of-the-art for implicit feedback [5]. These methods can directly optimize the ranking of feedbacks and assume positive items are preferable than negative items. Rendle et al. [2] propose a Bayesian Personalized Ranking (BPR) framework to maximize the difference of user's preferences between positive and negative items. Recently, BPR is extended to combine more information like users' social relations [24]. Other information like visual signals is accommodated by VBPR [5], which applies visual features of product images to discover user's visual interest and better understand items. Similar to MF methods, they only learn general tastes of users.

2.2 Markov Chain Based Methods

In addition to conventional CF methods, sequential methods are popular for the recommendation and they mostly rely on Markov Chains. Rendle et al. [8] make a combination of MF and MC to learn both general taste and current effect for the next-basket recommendation. Chen et al. [7] build a Markov model integrated with the forgetting mechanism to weaken long-term interest and highlight short-term interest for item recommendation. However, the Markov assumption hinders learning the long-term dependency because it assumes the next state only related to the last state. The high/variable-order MC models can make the next state related to multiple previous states, which results in a high computational cost. This problem can be solved by only considering the state-to-state probability with balancing parameters, which ignores the set-to-state probability [7], [25]. It is difficult for MC methods to model the long-term dependency.

On the other hand, there are few Markov models involving multiple features. Chen et al. [26], [27] propose a two-view latent subspace Markov network to do image retrieval, annotation and so on. Their model is more like multi-view data fusion and is not suitable for sequential recommendation. MC is based on the probability among different states. In the sequential scenario, this probability is independent of the additional content information.

2.3 Recurrent Neural Networks

Recently, recurrent neural networks become more and more powerful. Owing to its recurrent structure, RNN can better extract the temporal dependencies. RNN based sequential click prediction [10] gains the state-of-the-art performance. Yu et al. [12] take the representation of a basket acquired by pooling operation as the input of RNN, which is most effective for next basket recommendation. Liu et al. [11] incorporate time-specific and distance-specific transition matrices into RNN to predict next location. Liu et al. [13] combine RNN and the Log-BiLinear model [28] to make multi-behavioral prediction. Compared with traditional sequential methods, RNN is more promising.

Due to the gradient vanishing and explosion problem [29], standard RNN fails to hold the long-term dependency. Lots of work have been done to alleviate this problem, and the gated activation function achieves a success, like long short-term memory (LSTM) [14] and gated recurrent unit (GRU) [15]. Sutskever et al. [16] apply a multilayered LSTM to encode the input sequence and another LSTM to decode the target sequence in translation task. Their work also demonstrates LSTM can easily handle long sentences. Chung et al. [17] propose gated feedback RNNs to investigate the character-level language modeling. Bengio's work finds that GRU/LSTM are both certainly better than the basic RNN and GRU is comparable to LSTM on sequence modeling [30].

Recently, RNN is developed to model multi-view features. Hidasi et al. introduce the basic RNN model to do the session-based recommendation task [18], then develop the p-RNNs model to incorporate rich features [19]. The p-RNNs model builds subnets for each view separately. This is similar to the latent interest and visual interest in VBPR [5]. Two RNNs are used to make video recommendation by using the image and make product recommendation by using text description. Compared with the basic RNN model with only ID feature, the performance improvement of p-RNNs is not significant. Cao et al. model multi-view features collected by the mobile phone to predict the mood score [31]. Obviously, there are large differences between features in their work, and they apply the late fusion to explore interactions.

2.4 Multi-Modal Representation Learning

There are several main multi-modal representation learning methods: probabilistic graphical models, kernel-based methods and neural networks [32]. It is often intractable and complicated to obtain exact inference for probabilistic models. Because of the eigenvalue problem, kernel-based methods occupy a lot of memory and time. On the contrary, neural networks are tractable to handle the high-dimensional data. Recently, due to the success of Deep Neural Networks (DNNs), traditional methods tend to combine deep structures.

For methods based on DNNs, two main training strategies are widely used: Canonical Correlation Analysis (CCA) and AutoEncoder (AE) [33]. CCA based methods can make the two modalities maximally correlated. Recently, Deep CCA is proposed but it needs a large minibatch to optimize [34]. Based on CCA and AE, a deep canonically correlated autoencoder model is proposed [33] for feature learning. The constraint conditions would be too complicated if CCA based methods are used in our work. Accordingly, AE based methods would be promising.

AE based methods are very powerful to learn compact representations. AE could reproduce the input signal as far as possible and find the principal component. Vincent et al. design the denoising AE (dAE) by setting some input data to zero in a probabilistic manner [35]. After that, Vincent et al. design the stacked denoising AE (sDAE) and find that a single matrix is enough to do the encoding and decoding steps [36]. Ngiam et al. introduce the bimodal deep denoising autoencoder [37]. In this way, the hidden layer could learn the shared representation from different modalities. Later, Chen et al. [38] propose the marginalized denoising AE (mDAE) model, which finishes off the nonlinear transfer function and learns a linear transfer matrix. Furthermore, Wang et al. [39] propose a coupled mDAE model to deal with cross-domain learning problems. We introduce a 3mDAE model to generate multi-modal fusion representation.

3 Proposed MV-RNN Model

In this section, we propose a Multi-View Recurrent Neural Network (MV-RNN) model. We first formulate the problem. Next, we explore 3 strategies to combine multi-

IEEE Proof

TABLE 1
Notation

Notation	Explanation
$\mathcal{U}, \mathcal{I}, \mathcal{I}^u$	set of users, set of items, sequence of user u
$\mathcal{P}^u, \mathcal{V}^u, \mathcal{T}^u$	sequences of training, validation and test of user u
p, q	positive item, negative item
$\hat{x}^t_{upq}$	difference of preference of u towards p and q at the tth time
$\mathbf{f}, \mathbf{g}$	high-dimensional visual and textual features of an item
E, V	embedding matrices for $\mathbf{f}, \mathbf{g}$
$\boldsymbol{i}_\mathrm{f}, \boldsymbol{i}_\mathrm{g}$	low-dimensional visual and textual features of an item
$\boldsymbol{i}_\mathrm{x}, \boldsymbol{i}_\mathrm{m}$	latent feature, multi-modal fusion feature built by $\boldsymbol{i}_\mathrm{f}$ and $\boldsymbol{i}_\mathrm{g}$
$d, d_\mathrm{f}, d_\mathrm{g}$	dimensions of $\boldsymbol{i}_\mathrm{x}, \mathbf{f}, \mathbf{g}$
$\boldsymbol{h}_\mathrm{x}, \boldsymbol{h}_\mathrm{m}$	latent and multi-modal fusion features of a user
$\boldsymbol{U}, \boldsymbol{W}, \boldsymbol{b}$	transition matrices and bias for recurrent neural network

view features at the input to represent the item. Then we investigate 2 structures to model multi-view features at the hidden state to build user representation. Finally, all the variants of MV-RNN can be trained with the Bayesian Personalized Ranking framework and the Back Propagation Through Time (BPTT) algorithm.

3.1 Problem Formulation

In order to simplify the problem formulation of sequential recommendation, we take purchase histories of online shopping for instance. Let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{I} = \{i_1, \ldots i_{|\mathcal{I}|}\}$ represent the sets of users and items respectively. Use $\mathcal{I}^u = (i^u_1, \ldots, i^u_{|\mathcal{I}^u|})$ to denote the items that the user u has purchased in chronological order, and the tth item $i^u_t \in \mathcal{I}$. Additionally, an image and a text description are available for each item $i \in \mathcal{I}$. Given each user's history $\mathcal{I}^u$, our goal is to recommend a list of items that a user may purchase. The notation is listed in Table 1 for clarity.

3.2 Representation of Item with Multi-View Features

Representation of item is used as the input of our MV-RNN model. Three different combinations of multi-view features are shown in Fig. 2, and details are as follows.

3.2.1 Multi-View Features

There are two basic types of multi-view features of an item: indirectly observable view and directly observable view. The former view is latent feature, which is widely-used in recommender systems. The latent feature of an item is defined by a vector

$$\boldsymbol{i}_\mathrm{x} = \boldsymbol{x}, \qquad \boldsymbol{i}_\mathrm{x} \in \mathbb{R}^d. \tag{1}$$

The latter view refers to the additional multi-modal information that is presented externally, like image, text description, category label, video, and so on. They can provide very important information for the item. For example, image can directly show the color, text description can provide the clothing size.

The multi-modal features consist of visual and textual features ($\mathbf{f}$ and $\mathbf{g}$) in our work. They are obtained by GoogLeNet [40] and GloVe [41] weighted by TF-IDF respectively. The two kinds of features are 1024-dimensional and 100-dimensional vectors respectively. Due to the difference of $\mathbf{f}$ and $\mathbf{g}$, we learn two linear embedding matrices E and V to transform the original high-dimensional features to embedded low-dimensional visual and textual features ($\boldsymbol{i}_\mathrm{f}$ and $\boldsymbol{i}_\mathrm{g}$)

$$\boldsymbol{i}_\mathrm{f} = \boldsymbol{E}\mathbf{f}, \qquad \boldsymbol{i}_\mathrm{f} \in \mathbb{R}^d \tag{2}$$

$$\boldsymbol{i}_\mathrm{g} = \boldsymbol{V}\mathbf{g}, \qquad \boldsymbol{i}_\mathrm{g} \in \mathbb{R}^d. \tag{3}$$

Sequential recommendation usually encounters the cold start problem as feedbacks are too sparse to learn fine representations of users and items. Modeling multi-view features is an effective way to alleviate this issue. These features are usually obtained from different data sources, and have different numerical ranges as well as different dimensions. Therefore, the raw features need be normalized to a same range to obtain $\boldsymbol{x}$, $\mathbf{f}$ and $\mathbf{g}$, and should better be embedded to d-dimensional vectors to obtain $\boldsymbol{i}_\mathrm{x}$, $\boldsymbol{i}_\mathrm{f}$ and $\boldsymbol{i}_\mathrm{g}$. None of them is sequence data and they are aligned with each other by the item ID.

3.2.2 Feature Concatenation

The most natural method to combine multi-view features is concatenation. Intuitively, the item representation is $\boldsymbol{i} = [\boldsymbol{i}_\mathrm{x}; \boldsymbol{i}_\mathrm{f}; \boldsymbol{i}_\mathrm{g}]$. The $\boldsymbol{i}$ is a $3d$-dimensional vector, and its dimension will increase with the number of features. The capacity and complexity of this method will also increase subsequently.

3.2.3 Feature Fusion

Fusion can be directly established by the addition operation without nonlinear transformation

$$\boldsymbol{i}_\mathrm{m} = \boldsymbol{i}_\mathrm{f} + \boldsymbol{i}_\mathrm{g}, \qquad \boldsymbol{i}_\mathrm{m} \in \mathbb{R}^d. \tag{4}$$

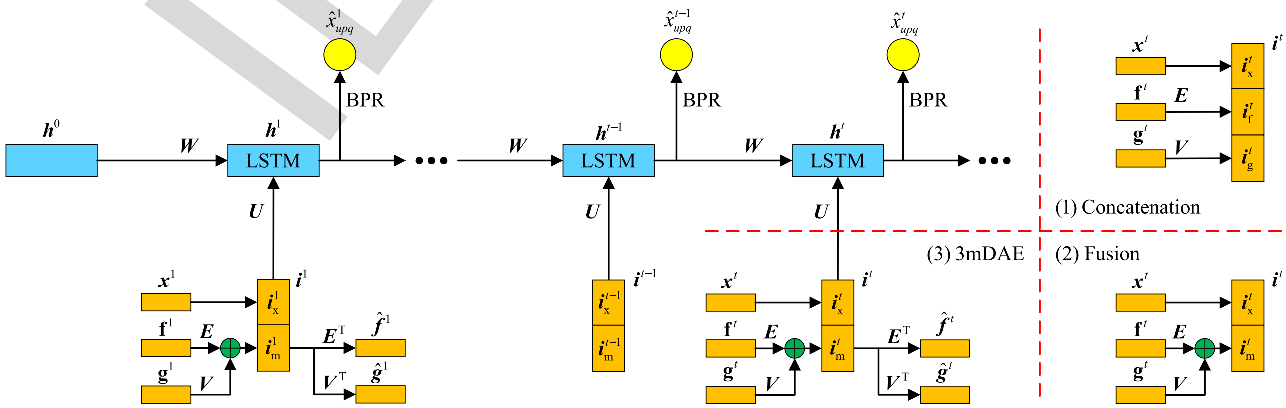
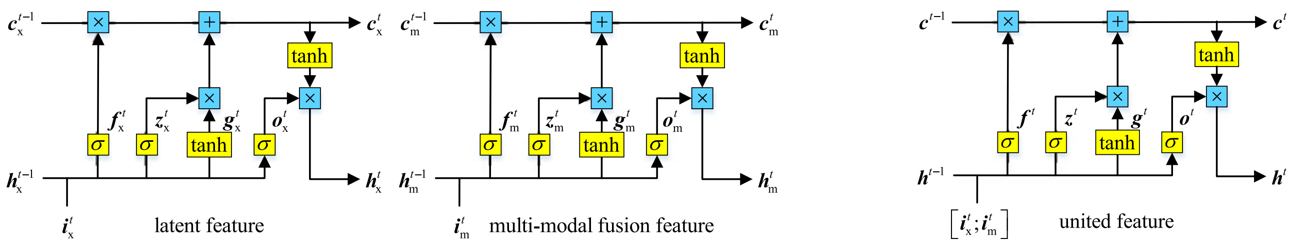


Fig. 2. Diagram of the MV-RNN model. The multi-view input consists of latent feature and additional visual and textual features. Concatenation, Fusion, and 3mDAE are three kinds of combinations of multi-view features. The hidden state captures dynamic changes in the user's interest.

IEEE Proof



(a) Separate Structure (b) Unified Structure

Fig. 3. Diagram of hidden state structures of the MV-RNN model. We devise a separate structure and a united structure. The two structures handle the multi-view input features at the input by multiple RNN units and by one RNN unit each time, respectively.

Please note that features with similar contents are suitable for fusion. Therefore, i_{f} and i_{g} are fused as the multi-modal fusion feature $\boldsymbol{i}_m$, and this process can make the model more concise. Benefiting from linear embedding and linear transformation, $\boldsymbol{i}_{\mathrm{m}}$ can hold all the information from $\mathbf{f}$ and $\mathbf{g}$. Then we obtain item representation $\boldsymbol{i} = [\boldsymbol{i}_{\mathrm{x}}; \boldsymbol{i}_{\mathrm{m}}]$ by concatenation.

Although concatenation and fusion are easy to utilize, they still have three issues. First, both concatenation and fusion do not have an explicit objective which is able to explore correlations across modalities [37]. Second, they are unhandy to use in such a situation where items in the test set have missing modalities [37]. Third, no matter the combination of $\boldsymbol{i}_{\mathrm{f}}, \boldsymbol{i}_{\mathrm{g}}$ is concatenation or fusion, useful information is entered into the model as well as noise. Therefore, more robust structures and parameters (E, V) need to be learned.

3.2.4 Multi-Modal Marginalized Denoising AutoEncoder

We introduce a new fusion method to combine the multi-modal information to learn fusion feature. This method can go further to leverage the advantage of different modalities, learn more robust features and tackle the missing modalities problem.

This method is based on the mDAE model [38]. It learns a linear mapping M and minimizes the reconstruction loss $l(t, M\tilde{t})$, where $\tilde{t}$ is the corrupted version of original feature t. However, mDAE has no hidden layer. Later, the coupled mDAE [39] modifies the original mDAE with two mappings in a linear way $l(t, M^{\mathrm{T}}M\tilde{t})$. $M\tilde{t}$ and $M^{\mathrm{T}}M\tilde{t}$ represent the encoding and decoding processes respectively. Based on these works, we introduce a **m**ulti-**m**odal *mDAE* model, called *3mDAE*, to learn fusion feature. Details are as follows.

Encoder-Decoder. The encoding process is represented by Eqs. (2) and (3), and the corresponding hidden layer is built by Eq. (4). In the decoding process, we need to reconstruct the multi-modal input features. The mapping matrix in decoding process is just the transpose of the mapping matrix in encoding process [36]

$$\begin{aligned} \hat{f} &= E^{\mathrm{T}}\boldsymbol{i}_{\mathrm{m}} \\ \hat{g} &= V^{\mathrm{T}}\boldsymbol{i}_{\mathrm{m}}. \end{aligned} \tag{5}$$

In our introduced 3mDAE model, we omit bias term and apply original features $\mathbf{f}$ and $\mathbf{g}$ instead of corrupted version as input. The denoising operation is discussed in Section 4.3. The final representation of an item is also $\boldsymbol{i} = [\boldsymbol{i}_{\mathrm{x}}; \boldsymbol{i}_{\mathrm{m}}]$.

Objective Function. The mDAE model minimizes the overall quadratic reconstruction loss for one modality [39]

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} \left\| t_i - M^{\mathrm{T}} M \tilde{t}_i \right\|^2, \tag{6}$$

where m is the number of samples. We extend this to form the objective function of 3mDAE

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} \left(\frac{\|\mathbf{f}_i - \hat{f}_i\|^2}{|d_{\mathrm{f}}|} + \frac{\left\|\mathbf{g}_i - \hat{g}_i\right\|^2}{|d_{\mathrm{g}}|} \right). \tag{7}$$

The d_{f} and d_{g} are the original dimensions of visual and textual features respectively, where $|d_{\mathrm{f}}| = 1024$ and $|d_{\mathrm{g}}| = 100$ in our work. They are used as balance factors.

3.3 Modeling of Multi-View Features on Hidden State

User representation is expressed by the hidden state of our MV-RNN model. Two different ways are explored to model the multi-view features built at the input. In detail, Figs. 3a and 3b reveal the separate and united hidden state structures respectively. Specifically, the illustration is based on $\boldsymbol{i}_{\mathrm{x}}$ and $\boldsymbol{i}_{\mathrm{m}}$.

3.3.1 Long Short-Term Memory

Conventional RNN suffers from the gradient vanishing and explosion problem, so that it fails to learn long-term dependencies [29]. Gated activation function is proposed to solve this issue. We chose the widely-used LSTM [14] and it is denoted by

$$\begin{aligned} \boldsymbol{f}^t &= \sigma\big(U_1\boldsymbol{x}^t + W_1\boldsymbol{h}^{t-1} + \boldsymbol{b}_1\big), \\ \boldsymbol{z}^t &= \sigma\big(U_2\boldsymbol{x}^t + W_2\boldsymbol{h}^{t-1} + \boldsymbol{b}_2\big), \\ \boldsymbol{g}^t &= \tanh\big(U_3\boldsymbol{x}^t + W_3\boldsymbol{h}^{t-1} + \boldsymbol{b}_3\big), \\ \boldsymbol{c}^t &= \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{z}^t \odot \boldsymbol{g}^t \\ \boldsymbol{o}^t &= \sigma\big(U_4\boldsymbol{x}^t + W_4\boldsymbol{h}^{t-1} + \boldsymbol{b}_4\big), \\ \boldsymbol{h}^t &= \boldsymbol{o}^t \odot \tanh\big(\boldsymbol{c}^t\big) \end{aligned} \tag{8}$$

where $\odot$ means element-wise product between two variables, t is the time step, $\boldsymbol{x}^t \in \mathbb{R}^d$ is the input feature. Transition matrices $U_{1\sim4} \in \mathbb{R}^{d\times d}$ transfer the current input. Recurrent connections $W_{1\sim4} \in \mathbb{R}^{d\times d}$ delivers the sequential information. $\boldsymbol{b}_{1\sim4} \in \mathbb{R}^d$ are bias terms. The $\boldsymbol{f}^t, \boldsymbol{z}^t, \boldsymbol{g}^t, \boldsymbol{c}^t, \boldsymbol{o}^t, \boldsymbol{h}^t$ are the *forget* gate, *input* gate, *update* gate, *cell*, *output* gate and the hidden state, respectively. In our work, we apply a $Lstm(\cdot)$ function to substitute the original formulas in Eq. (8)

$$\boldsymbol{h}^t = Lstm\big(U\boldsymbol{x}^t, W\boldsymbol{h}^{t-1}, \boldsymbol{b}\big), \qquad \boldsymbol{h}^t \in \mathbb{R}^d, \tag{9}$$

where U is a set of four matrices $U_{1\sim4}$, and so do the $W, \boldsymbol{b}$.

3.3.2 Separate Multi-View RNN

A natural way to handle multi-view features is to apply separate RNN units. Each unit is used for each kind of feature.

IEEE Proof

In this stage, our MV-RNN is a two-unit model, as shown in Fig. 3a.

We apply one RNN unit to model the latent feature and apply another RNN unit to model the multi-modal fusion feature. Formulation is defined by

$$\boldsymbol{h}_{\mathrm{x}}^t = Lstm\big(\boldsymbol{U}_{\mathrm{x}}\boldsymbol{i}_{\mathrm{x}}^t, \boldsymbol{W}_{\mathrm{x}}\boldsymbol{h}_{\mathrm{x}}^{t-1}, \boldsymbol{b}_{\mathrm{x}}\big), \boldsymbol{h}_{\mathrm{x}}^t \in \mathbb{R}^d, \tag{10a}$$

$$\boldsymbol{h}_{\mathrm{m}}^t = Lstm\big(\boldsymbol{U}_{\mathrm{m}}\boldsymbol{i}_{\mathrm{m}}^t, \boldsymbol{W}_{\mathrm{m}}\boldsymbol{h}_{\mathrm{m}}^{t-1}, \boldsymbol{b}_{\mathrm{m}}\big), \boldsymbol{h}_{\mathrm{m}}^t \in \mathbb{R}^d, \tag{10b}$$

where $\boldsymbol{h}_{\mathrm{x}}^t$ and $\boldsymbol{h}_{\mathrm{m}}^t$ are defined as a user's latent interest and multi-modal fusion interest at the tth input. $\boldsymbol{U}_{\mathrm{x}}$ is a set of four matrices: $\boldsymbol{U}_{\mathrm{x}1\sim 4} \in \mathbb{R}^{d\times d}$. Similarly, $\boldsymbol{W}_{\mathrm{x}}, \boldsymbol{b}_{\mathrm{x}}, \boldsymbol{U}_{\mathrm{m}}, \boldsymbol{W}_{\mathrm{m}}$ and $\boldsymbol{b}_{\mathrm{m}}$ are sets of three matrices or vectors, where subscripts x and m represent the latent modeling and multi-modal modeling.

Multi-view user representation is the concatenation of hidden states from the two RNN units. They are linked together at every time step in our work

$$\boldsymbol{h}^t = \big[\boldsymbol{h}_{\mathrm{x}}^t; \boldsymbol{h}_{\mathrm{m}}^t\big], \qquad \boldsymbol{h}^t \in \mathbb{R}^{2d}, \tag{11}$$

where $\boldsymbol{h}^t$ is the user's general interest. But it may not be able to leverage the connection between multi-view features, as we model them in two RNN units separately and build discrete user's interests. Thus we tend to develop a single RNN unit to handle multi-view features simultaneously.

3.3.3 United Multi-View RNN

We incorporate the multi-modal fusion feature into one RNN unit together with the latent feature. In such situation, our MV-RNN is a one-unit model, as shown in Fig. 3b. This structure can capture the relation between multi-view features and construct the united user's interest, which promotes the model to have more promising performance

$$\boldsymbol{h}^t = Lstm\big(\boldsymbol{U}\big[\boldsymbol{i}_{\mathrm{x}}^t; \boldsymbol{i}_{\mathrm{m}}^t\big], \boldsymbol{W}\boldsymbol{h}^{t-1}, \boldsymbol{b}\big), \qquad \boldsymbol{h}^t \in \mathbb{R}^{2d}, \tag{12}$$

where $\boldsymbol{h}^t$ is the complete user's interest, not a simple combination of a user's different interests in Eq. (11). We apply one factor $\boldsymbol{U}$ consisting of $\boldsymbol{U}_{1\sim 4} \in \mathbb{R}^{2d\times 2d}$ because we have $\big[\boldsymbol{i}_{\mathrm{x}}^t; \boldsymbol{i}_{\mathrm{m}}^t\big] \in \mathbb{R}^{2d}$, and so do the $\boldsymbol{W}, \boldsymbol{b}$.

Via the 3mDAE model and the united structure, we finally model the item's multiple (latent, visual and textual) features and the user's interest in the same feature space. Our MV-RNN model benefits from this united viewpoint.

3.4 Model Learning

After discussing the input and hidden state of the MV-RNN model, we introduce the training procedure on output. No matter what kind of combinations of features at input or structures of hidden state, the BPR [2] framework is always suitable. BPR is a powerful pairwise method for implicit feedback, and it has been widely used in many works [5], [11], [12], [13], [18], [42]. Besides, as a 3mDAE model is introduced, we need to carefully consider the multi-modal reconstruction loss. A united objective function needs to be constructed. The description is also based on $\boldsymbol{i}_{\mathrm{x}}$ and $\boldsymbol{i}_{\mathrm{m}}$.

The training set $\mathcal{S}$ is made by (u, p, q) triples, where u represents the user, p and q denote the positive and negative items respectively. Item p is selected from a user's purchase history $\mathcal{I}^u$, while item q is randomly chosen from the rest items ($\mathcal{I} \setminus \mathcal{I}^u$). A negative item is regenerated for each positive item in each epoch

$$\mathcal{S} = \{(u, p, q) | u \in \mathcal{U} \wedge p \in \mathcal{I}^u \wedge q \in \mathcal{I} \setminus \mathcal{I}^u\}. \tag{13}$$

Given the training set, we calculate the difference of user's preferences between positive and negative items on output at every time step. At the tth time step, it can be computed by

$$\begin{aligned} \hat{x}_{upq}^t &= \hat{x}_{up}^t - \hat{x}_{uq}^t \\ &= \big(\boldsymbol{h}^t\big)^{\mathrm{T}}\Big(\boldsymbol{i}_p^{t+1} - \boldsymbol{i}_q^{t+1}\Big), \end{aligned} \tag{14}$$

where $\boldsymbol{i}_p^{t+1}$ and $\boldsymbol{i}_q^{t+1}$ represent positive and negative inputs respectively: $\boldsymbol{i}_p^{t+1} = \Big[\boldsymbol{i}_{\mathrm{x}p}^{t+1}; \boldsymbol{i}_{\mathrm{m}p}^{t+1}\Big], \boldsymbol{i}_q^{t+1} = \Big[\boldsymbol{i}_{\mathrm{x}q}^{t+1}; \boldsymbol{i}_{\mathrm{m}q}^{t+1}\Big]$.

The objective function combines BPR and our 3mDAE by a minimal form. The MV-RNN can simultaneously model these two kinds of losses. BPR maximizes the following formula:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \sum_{(u,p,q)\in S} \ln \sigma\big(\hat{x}_{upq}\big) - \frac{\lambda_\Theta}{2}\|\Theta\|^2. \tag{15}$$

It is transformed to the minimal form in our work. Next, 3mDAE loss represented in Eq. (7) is extended along with the BPR. Because we compute preference at every output using positive and negative items, we need to minimize all the visual and textual encoder-decoder losses. Last, we introduce a multiplicator r_{a} to leverage the preference of BPR and the reconstruction loss of our 3mDAE model. The final objective function is defined as

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{(u,p,q)\in S} \left(\begin{array}{l} -\ln \sigma(\hat{x}_{upq}) \\ + \frac{r_{\mathrm{a}}}{2|d_{\mathrm{f}}|}\Big(\|\mathbf{f}_p - \hat{\boldsymbol{f}}_p\|^2 + \|\mathbf{f}_q - \hat{\boldsymbol{f}}_q\|^2\Big) \\ + \frac{r_{\mathrm{a}}}{2|d_{\mathrm{g}}|}\Big(\Big\|\mathbf{g}_p - \hat{\boldsymbol{g}}_p\Big\|^2 + \Big\|\mathbf{g}_q - \hat{\boldsymbol{g}}_q\Big\|^2\Big) \end{array} \right) + \frac{\lambda_\Theta}{2}\|\Theta\|^2, \tag{16}$$

where Θ denotes a set of parameters $\Theta = \{\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{V}, \boldsymbol{U}, \boldsymbol{W}, \boldsymbol{b}\}$. $\boldsymbol{X}$ is the set of all items' latent features. $\boldsymbol{U}, \boldsymbol{W}$ and $\boldsymbol{b}$ are the sets of the matrices or vectors represented in previous equations. $\lambda_\Theta \geqslant 0$ is the regularization parameter. Please note that λ_{ev} is introduced to regularize embedding matrices $\boldsymbol{E}$ and $\boldsymbol{V}$. Then, MV-RNN can be learned by the mini-batch gradient descent and parameters are updated by classical BPTT [20].

After the training, we obtain the fixed representations of Θ. Then $\boldsymbol{X}, \boldsymbol{E}$ and $\boldsymbol{V}$ are reused to obtain each item's final representation. We recalculate each user's sequential hidden states, and the last hidden state denotes a user's final representation.

4 Experimental Results and Analysis

In this section, we conduct experiments on two real-world datasets. First, experimental settings are introduced. Then a hyperparameter optimization is performed. Next, we make a comparison between MV-RNN and baselines, and a denoising experiment is conducted for our 3mDAE. The last section is cold start analysis on items.

4.1 Experimental Settings

4.1.1 Datasets

Experiments are conducted on two datasets collected from Taobao[1] and Amazon.[2] The basic statistics are listed in

1. https://tianchi.shuju.aliyun.com/datalab/dataSet.htm?id=13
2. http://jmcauley.ucsd.edu/data/amazon

IEEE Proof

TABLE 2
Datasets

(a) Datasets (5-core) used throughout the experiment.

dataset	users	items	feedbacks	sparsity
Taobao	1,003,331	343,134	12,613,815	99.996%
Amazon	38,840	22,586	272,949	99.969%

(b) Sub-datesets for the controlled study in Section 4.3.2.

dataset	users	items	feedbacks	sparsity
Taobao (10-core)	478,391	145,867	7,558,233	99.989%
Taobao (15-core)	89,634	34,903	1,912,708	99.939%
Taobao (20-core)	3,536	1,843	124,453	98.090%

We list the numbers of users, items, feedbacks, and sparsity of each dataset, respectively.

Table 2. Both datasets have massive sequential implicit feedbacks, and each item contains an image and a text description. We apply the filtering strategy called k-*core* [8], [12], [42]. Each user purchases at least k items and each item is bought by at least k users. We set k=5 and also hold users with no more than 100 items, because users with very long sequences ($|\mathcal{I}^u| > 100$) may scalp items.

- *Taobao* is a dataset for clothing matching competition on *TianChi*[3] platform. We use user historical data and item features (image, text) to make the sequential recommendation. Its time span is from 14-Jun-2014 to 15-Jun-2015.
- *Amazon* contains many reviews and product metadata [43], [44]. We use one large category *Clothing, Shoes and Jewelry* located in the second half of the website. We acquire the sequential implicit feedback from review histories where the ratings range from 1 to 5, obtain the images and text data from product metadata. The original time span is between 29-Sep-2000 and 23-Jul-2014. As feedbacks in previous years are too sparse, we only keep feedbacks within the most recent two years.

4.1.2 Multi-Modal Features

Multi-modal features are obtained by using the existing methods. They are normalized to the same range by min-max normalization. Then, they are used as the input features ($\mathbf{f}$ and $\mathbf{g}$).

The visual feature is obtained by the GoogLeNet [40] implemented by BVLC Caffe deep learning framework [45]. This network has 22 layers and has been pre-trained on 1.2M ImageNet ILSVRC2014 images [46]. We apply the output of layer $pool5/7x7\_s1$ to obtain 1024-dimensional visual features. They are all positive and are normalized to range $[0, 0.5]$.

To generate the textual features of items, a text description of each item is collected first. On Taobao, we directly use item titles which have already been segmented and disordered by the data provider. On Amazon, we combine each item's category and title as its text data. Then we adopt the GloVe model [41] weighted by TF-IDF [47] to obtain each word's feature and weight. Finally, the weighted feature for each item is computed to obtain 100-dimensional textual features. Their values are in the vicinity of zero and are normalized to range $[-0.5, 0.5]$.

4.1.3 Evaluation Metrics

Performance is evaluated on test set by Recall, Mean Average Precision (MAP) [48] and Normalized Discounted Cumulative Gain (NDCG) [42]. The former one is an evaluation of unranked retrieval sets, while the latter two reflect the order of items. Here we consider top-k (e.g., k =20, 30) recommendations. Besides, the Area Under the ROC Curve (AUC) [2], [5] is introduced to evaluate the overall performance.

Data is divided by time. We use feedbacks in first 60 percent of the time for training, 20 percent for validation and the rest 20 percent for test. Same as p-RNNs, hyperparameters are optimized on the validation set, and all models are retrained on the full training set (training and validation sets) before obtaining final results on the test set.

4.1.4 Comparisons

We compare MV-RNN with several comparative baselines:

- *Random*: Items are randomly ranked for all users. The AUC of this method is 0.5 [2].
- *POP*: This baseline recommends the most popular items in the training set for each user u.
- *BPR*: This method refers to the BPR-MF for implicit feedback [2]. It optimizes the difference of user's preferences for positive and negative items. The corresponding pairwise training procedure has been applied to many sequential tasks [11], [12], [13], [18].
- *VBPR*: Introduced in [5], this is an extended method with visual features based on BPR. It first incorporates visual information to build the user's interest.
- *LSTM*: This sequential baseline trained with BPR is developed for next basket recommendation [12]. Instead of basic RNN, LSTM is used in our work. Both BPR and LSTM only model the latent feature.
- *p-RNN*: The p-RNNs is a feature-rich model for session-based recommendation [19]. It has 3 structures and 4 training strategies. According to its experiments, we choose the best variant 'Parallel (res)'.

We design 3 combinations of input and 2 structures for the hidden state. There are 4 variants implemented as MV-RNN-Con., MV-RNN-Fus., MV-RNN-3mDAE-1U and MV-RNN-3mDAE-2U. The former 3 variants are built by the united structure, while the last one has the separate structure. The prefix 'MV-RNN-' can be omitted, and the 4 variants can be abbreviated as *Con.*, *Fus.*, *3mDAE-1U* and *3mDAE-2U* respectively. The Con. has the highest dimension of hidden state ($\boldsymbol{h} \in \mathbb{R}^{3d}$), while the rest has the same dimension ($\boldsymbol{h} \in \mathbb{R}^{2d}$). Additionally, we need to initialize parameters Θ to the same range, e.g., uniform distribution $[-0.5, 0.5]$. The initial hidden state $\boldsymbol{h}^0$ of each sequence is always zero. The learning rate is fixed at $\alpha = 0.1$ for all methods. Besides, the mini-batch size for training is set as 4 and users with similar lengths are grouped into one batch. This length-adjustment can greatly speed up training [49]. Complete codes for all models are written by using Theano and are available on GitHub.[4] All experimental results are also listed on this website.

3. https://tianchi.shuju.aliyun.com/

4. https://github.com/cuiqiang1990/MV-RNN

IEEE Proof

TABLE 3
The Best Parameters Acquired on the Validation Set for All Methods

dataset	parameter	BPR	VBPR	GRU/LSTM	p-RNN	based on GRU/LSTM		based on GRU		based on GRU/LSTM	
						Con.	Fus.	3mDAE-1U	3mDAE-2U	3mDAE-1U	3mDAE-2U
Taobao	λ_Θ	0.0	0.0	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	$\lambda_{\rm ev}$	-	0.00001	-	-	0.0	0.0	0.0	0.0	0.0	0.0
	$r_{\rm a}$	-	-	-	-	-	-	0.0001	0.001	0.00001	0.00001
Amazon	λ_Θ	0.0001	0.0001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	$\lambda_{\rm ev}$	-	0.0001	-	-	0.0	0.0	0.0	0.00001	0.00001	0.00001
	$r_{\rm a}$	-	-	-	-	-	-	0.0001	0.0001	0.001	0.001

TABLE 4
The Performance Difference of Our MV-RNN on Validation Set between Using Different Baselines (GRU, LSTM)

dataset	Based on GRU					Based on LSTM				
	method	@30 (%)			AUC	method	@30 (%)			AUC
		Recall	MAP	NDCG			Recall	MAP	NDCG	
Taobao	GRU	1.141	0.283	0.622	0.608	LSTM	1.124	0.287	0.603	0.610
	Con.	1.410	0.372	0.786	0.679	**Con.**	1.372	0.358	0.761	0.685
	Fus.	1.360	0.362	0.762	0.680	**Fus.**	1.309	0.332	0.718	0.678
	3mDAE-1U	1.362	0.334	0.735	0.675	**3mDAE-1U**	1.349	0.342	0.738	0.678
	3mDAE-2U	1.186	0.338	0.690	0.675	**3mDAE-2U**	1.196	0.353	0.709	0.676
Amazon	GRU	1.494	0.249	0.657	0.577	LSTM	1.604	0.305	0.717	0.583
	Con.	2.210	0.421	1.012	0.687	**Con.**	2.250	0.433	1.049	0.685
	Fus.	2.091	0.418	0.962	0.687	**Fus.**	2.248	0.415	0.998	0.687
	3mDAE-1U	2.237	0.410	1.013	0.688	**3mDAE-1U**	2.237	0.430	1.038	0.685
	3mDAE-2U	2.104	0.401	0.955	0.687	**3mDAE-2U**	2.283	0.425	1.035	0.690

4.2 Optimization on Validation Set

4.2.1 Regularization Parameter

The best parameters for regularization are listed in Table 3. They are chosen by the evaluations of all the metrics on validation set under the dimension $d = 20$.

In this optimization process, λ_Θ is first selected based on basic methods (BPR, GRU and LSTM), then $\lambda_{\rm ev}, r_{\rm a}$ are chosen by grid search. The ranges of these three parameters are $\lambda_\Theta, \lambda_{\rm ev} \in [0.001, 0.0001, 0.00001, 0.0]$ and $r_{\rm a} \in [0.001, 0.0001, 0.00001]$. With the reduction of data size from Taobao to Amazon, the best $\lambda_\Theta, \lambda_{\rm ev}, r_{\rm a}$ almost all get bigger.

4.2.2 Baseline Selection

Although several studies explore the difference between GRU and LSTM [17], [30], few people do comparisons for sequential recommendation. This part aims for completeness. Shown in Table 4, the result is the performance by using the best parameters obtained in Section 4.2.1. Please note that all values of Recall, MAP and NDCG in Tables 4, 5, 6, 7, 8 and Fig. 4 are represented in percentage.

Obviously, the performance of MV-RNN based on LSTM is better than that based on GRU in most cases, except the Con. and Fus. based on LSTM on Taobao. Although LSTM has more parameters, it also has the better model capacity. As a long as the model size is not significantly bigger, we should always consider the model with the best architecture. Therefore, in all the following experiments, we consider LSTM as the baseline instead of GRU and our MV-RNN is based on LSTM.

4.2.3 Dimension Analysis

The dimension analysis is investigated in Fig. 4. We illustrate the performances of top-30 and AUC on the validation set. The dimensions are set as $d = [10, 15, 20, 25]$.

With the increasing of dimension, performances of top-30 metrics have similar trends with each other on both datasets. BPR and VBPR tend to get worse. They have similar trends as well as absolute values. It is difficult to tell the difference between VBPR and BPR on Recall, MAP, and NDCG, especially on Taobao. The p-RNN model is not sensitive to dimension. The LSTM and MV-RNN models obtain better performance with the increasing of dimension on Taobao, while they almost do not change with the dimension on Amazon. On the other hand, AUCs of all models are much stable with different dimensions. VBPR has obviously better performance than BPR on both datasets. The 4 variants of MV-RNN are nearly coincident with each other. The AUC is not sensitive to the dimension.

Generally, it is obvious that LSTM is a very strong baseline. Apparently, our MV-RNN model is the best. The optimal dimension is chosen as $d = 20$ and it is applied to other experiments.

4.3 Analysis of Experimental Results

Table 5 illustrates all performances on two datasets with four evaluation metrics. Recall, MAP and NDCG focus on local performance, while AUC reflects global performance.

IEEE Proof

TABLE 5
Evaluation of Different Methods on the Test Set with the Dimension of Latent Vector $d = 20$

	Taobao								Amazon							
method	p	@20 (%)			@30 (%)			AUC	p	@20 (%)			@30 (%)			AUC
		Recall	MAP	NDCG	Recall	MAP	NDCG			Recall	MAP	NDCG	Recall	MAP	NDCG	
Random	-	0.004	0.001	0.002	0.006	0.001	0.003	0.500	-	0.083	0.016	0.040	0.137	0.018	0.056	0.500
POP	-	0.113	0.016	0.051	0.218	0.020	0.085	0.441	-	1.418	0.299	0.697	1.993	0.321	0.847	0.553
BPR	-	0.191	0.038	0.101	0.274	0.041	0.127	0.573	-	0.641	0.168	0.340	0.812	0.176	0.390	0.511
VBPR	-	0.196	0.042	0.106	0.283	0.045	0.131	0.577	-	0.700	0.181	0.368	0.922	0.190	0.423	0.584
LSTM	-	0.666	0.162	0.386	0.884	0.171	0.453	0.567	-	1.443	0.283	0.671	1.982	0.301	0.820	0.608
p-RNN	-	0.537	0.149	0.335	0.688	0.156	0.382	0.553	-	1.484	0.301	0.708	1.939	0.320	0.831	0.609
Con.	-	0.863	0.212	0.502	1.164	0.224	0.592	**0.690**	-	2.113	0.522	1.092	2.827	0.554	1.294	**0.723**
Fus.	-	0.808	0.212	0.481	1.082	0.223	0.559	**0.690**	-	2.157	0.508	1.096	2.867	0.538	1.285	0.722
3mDAE-1U	0.0	0.849	0.213	0.499	1.140	0.225	0.586	0.680	0.0	2.190	0.517	1.116	2.869	0.549	1.309	0.722
	0.2	0.802	0.205	0.472	1.075	0.216	0.555	0.687	0.1	**2.243**	**0.541**	**1.149**	**2.995**	**0.570**	**1.352**	0.722
	0.3	**0.881**	0.228	**0.523**	**1.174**	0.240	**0.612**	0.680	0.2	2.211	0.529	1.136	2.892	0.558	1.322	0.720
	0.4	0.807	0.219	0.488	1.075	0.230	0.570	0.679	0.3	2.217	0.521	1.117	2.968	0.552	1.317	0.721
3mDAE-2U	0.0	0.676	0.208	0.440	0.892	0.217	0.506	0.685	0.0	2.227	0.524	1.108	2.856	0.550	1.286	0.721
	0.2	0.750	0.234	0.491	0.971	0.243	0.558	0.683	0.1	2.227	0.528	1.128	2.883	0.555	1.301	0.720
	0.3	0.760	0.235	0.494	1.001	0.246	0.568	0.677	0.2	2.162	0.517	1.107	2.906	0.544	1.292	0.722
	0.4	0.792	**0.243**	0.514	1.029	**0.253**	0.586	0.681	0.3	2.134	0.512	1.104	2.838	0.544	1.305	0.720

We generate top-20 and 30 items for each user. Because of the structure of concatenation, the hidden state dimension of Con. is much larger than the others.

4.3.1 Performance Comparison

From a global perspective, additional multi-modal information of items (e.g, image and text description) is indeed beneficial. VBPR beats BPR. MV-RNN outperforms LSTM model. Our MV-RNN can effectively model the additional information. For example, the Con. has almost more than 30 percent and more than 40 percent improvements over LSTM on Taobao and Amazon respectively with respect to Recall, MAP and NDCG. Its improvements of AUC over LSTM are both around 20 percent on two datasets. As for the rest 3 variants which have hidden states of the same length, 3mDAE-1U performs best. In a perspective of statics and dynamics, although both trained by the BPR framework to maximize the difference of user's preferences towards positive and negative items, LSTM beats BPR by a large margin. The recurrent structure of LSTM can capture sequential information which is helpful for the recommendation.

3mDAE and Denoising. In this part, we analyze the four variants of MV-RNN and focus on the 3mDAE. The Con. almost always beats the Fus. but not too much. The highest hidden state dimension of Con. improves its capacity. This phenomenon also shows that feature addition has no great damage to multi-modal modeling. Then, we embody the advantage of 3mDAE and introduce a training setting called $denoising$. It can help to learn more robust features and acquire the best performance.

The denoising AE is first proposed for image classification on the MNIST database. It can make features more robust and avoid learning the identity function by using corrupted input. Identity function means just mapping the original input to its copy, which happens in the encoding process in AE (e.g., $\mathbf{f} \rightarrow \boldsymbol{E}\mathbf{f}$). It is easy to obtain a denoising AE just by a stochastic corruption operation on input. The original corruption mechanism randomly sets some of an input feature to zero with probability $0 \leqslant p < 1$. While in our experiment, we make feature itself corrupted.

This $denoising$ is conducted for 3mDAE. In this setting, we make some multi-modal data corrupted in the encoding process and still reconstruct both modalities in the decoding step. Training 3mDAE still requires all the data in Table 2 a. The corruption levels are set as $p = [0.0, 0.2, 0.3, 0.4]$ and $p = [0.0, 0.1, 0.2, 0.3]$ for Taobao and Amazon respectively. If $p = 0.0$, the input data in the encoding process is complete. The results are still obtained on the original test set where all items have all features. Results are shown in eight rows at the bottom of the Table 5.

Obviously, performance can become better than the original ($p = 0\%$) by $denoising$, especially the Recall, MAP and NDCG. More importantly, 3mDAE-1U performs best. It is able to be better than Con., although Con. has the highest hidden state dimension. When we randomly reset some features to zero in the encoding process, the noise in the whole input data is reduced. However, by reconstructing both modalities in the decoding step, the fusion feature of our 3mDAE can still keep the useful information in both modalities. Our 3mDAE can acquire more robust features. The best corruption levels for 3mDAE-1U/2U are $p = 0.3/0.4$ and $p = 0.1/0.1$ on two datasets respectively.

The 3mDAE-1U/2U are a one-unit model with the united structure and a two-unit model with the separate structure respectively. In Table 5, the one-unit model outperforms the

TABLE 6
Results of the Controlled Study in Section 4.3.2

dataset	method	@30 (%)			AUC
		Recall	MAP	NDCG	
Taobao (10-core)	LSTM	1.366	0.305	0.794	0.603
	Con.	1.635	0.365	0.946	0.689
Taobao (15-core)	LSTM	2.343	0.752	1.742	0.591
	Con.	2.801	0.868	2.040	0.678
Taobao (20-core)	LSTM	4.681	13.795	16.651	0.536
	Con.	5.449	16.701	19.118	0.623

IEEE Proof

TABLE 7
A Setting Called *missing* is Introduced and Measured on an Artificial Test Set, where Some Items' Multi-Modal Features are Missing (Deleted)

MV-RNN	*missing* - Taobao								*missing* - Amazon							
	p	@20 (%)			@30 (%)			AUC	p	@20 (%)			@30 (%)			AUC
		Recall	MAP	NDCG	Recall	MAP	NDCG			Recall	MAP	NDCG	Recall	MAP	NDCG	
Con.	-	0.784	0.189	0.453	1.042	0.199	0.531	0.665	-	1.903	0.448	0.946	2.537	0.473	1.118	0.692
Fus.	-	0.748	0.187	0.439	0.986	0.197	0.511	0.649	-	1.775	0.423	0.913	2.265	0.444	1.054	0.696
3mDAE-1U	0.0	0.732	0.199	0.447	0.975	0.209	0.521	0.653	0.0	1.823	0.430	0.924	2.431	0.457	1.101	0.691
	0.2	0.743	0.181	0.427	0.999	0.191	0.504	**0.679**	**0.1**	**2.059**	**0.491**	**1.040**	**2.696**	0.517	**1.217**	0.703
	0.3	**0.832**	**0.216**	**0.496**	**1.102**	**0.228**	**0.578**	0.671	0.2	2.003	0.488	1.028	2.561	0.510	1.176	0.702
	0.4	0.746	0.191	0.440	1.000	0.202	0.517	0.666	0.3	2.001	0.470	0.995	2.645	0.498	1.171	**0.705**
3mDAE-2U	0.0	0.605	0.181	0.388	0.791	0.188	0.444	0.652	0.0	1.779	0.392	0.864	2.414	0.414	1.042	0.688
	0.2	0.643	0.180	0.400	0.851	0.189	0.464	0.673	0.1	1.964	0.452	0.979	2.624	0.479	1.163	0.704
	0.3	0.676	0.194	0.424	0.897	0.204	0.492	0.670	0.2	1.858	0.482	0.986	2.476	0.506	1.145	0.702
	0.4	0.701	0.201	0.441	0.920	0.210	0.508	0.674	0.3	1.827	0.496	1.003	2.435	**0.522**	1.171	0.704

This setting aims to study the ability of MV-RNN to handle missing modalities.

two-unit model. A united inner structure can better leverage the advantage of multi-view features. The separate structure may be not able to well model the connection between different views.

p-RNN versus MV-RNN. The session-based p-RNN model also incorporates additional features, but it is comparable to LSTM. If we carefully examine the results of p-RNN in its original paper [19], we find that most results of p-RNN are also close to the basic model ('ID only' in their paper). The reason is varied as p-RNN is substantially different from our MV-RNN. The first one is feature normalization. Multi-view features must be normalized to the same range, but only visual features are normalized in their work. Next, different from our strategy in Eq. (14), p-RNN uses output weight matrix to compute the user's scores on items. This matrix improves the capacity of a model but increases the learning difficulty, especially for the modeling of visual and textual features. We experimented with using this matrix on our Con., but its performance is very close to that of LSTM. Then, different subnets within p-RNN are trained one by one, which can not well construct the connection among multi-view features.

4.3.2 A Controlled Study

In Table 5, the metrics (Recall, MAP and NDCG) seem to be low, especially on Taobao. Therefore, we conduct a controlled study to explore the factors that influence the metrics.

Reducing the number of items (search space) may be helpful. We extract three sub-datasets from Taobao by increasing the filtering strategy as $[10, 15, 20]$-core. The statistics are shown in Table 2 b. In this way, the search space is greatly reduced. Then, we perform experiments by using LSTM and Con. Accordingly, we need to re-select the best parameters and the results are shown in Table 6. With the increasing of k, the three metrics get bigger. Metrics of Taobao (20-core) are obviously bigger than that of the other datasts. This may be because the sparsity of Taobao (20-core) is clearly small. At the same time, our method Con. is always better than LSTM, which shows the effectiveness of our MV-RNN. Therefore, although the absolute values on Taobao are small, they are related to the dataset itself (e.g., sparsity).

In summary, our MV-RNN model is better than the others. MV-RNN can well model multi-view features and achieves the best and stable performance in different situations. The *denoising* of 3mDAE is a good setting to improve performance. Besides, special strategies used in p-RNN are not necessary for handling multi-view features. Feature concatenation is natural but very useful. A united structure with simultaneous training strategy is easy to use and is better than the separate subnets built for each view in p-RNN. These conclusions of joint learning are also confirmed by the previous works, like a multi-view model for cross-domain user modeling [50].

4.4 Analysis of Missing Modalities in Test Set

Multi-modal methods usually hold an assumption that all modalities are available. However, in practice, certain modality is often missing, like an item without the visual feature. In such case, our 3mDAE is theoretically better than the concatenation and fusion. To verify this, we introduce a setting of test set called *missing*. First, we artificially modify the test set. We set one-third of items without visual features, one-third without textual features, and the last one-third with all the multi-modal features. Then, the training procedure also applies the *denoising*, and the only difference between Sections 4.3 and 4.4 is that *missing* here is evaluated on our artificial test set. The result is shown in Table 7.

Experimental results indicate that our 3mDAE is very promising for tackling missing modalities problem. Both 3mDAE-1U/2U perform very well and 3mDAE-1U is more successful. For example, 3mDAE-1U under $p = 0.3$ increases by about 10 percent with respect to Con. on Recall, MAP and NDCG on Taobao. This improvement acquired by 3mDAE-1U under $p = 0.1$ on Amazon is about 9 percent. Besides, 3mDAE-1U/2U also have some increases on AUC over Con. and Fus.. Our 3mDAE is greatly better than others in this *missing* setting and it can effectively handle the items with missing modalities.

4.5 Analysis of Cold Start

We investigate the performance of MV-RNN on cold start items in the test set. These items usually account for a large proportion and cold start is an intractable problem in practical recommender systems. Previous works like VBPR [5] usually only consider cold start items and neglect the rest.

IEEE Proof

TABLE 8
Cold Start Performance on Two Datasets Under the Evaluation of Recall@30 and AUC with Dimension of Latent Feature $d = 20$

(a) Numbers of items in each subset and each bin of the test set. Numbers of feedbacks are also counted.

dataset		subsets of test set		bins of test set									
		cold-start	*active*	$[1,2]$	$[3,4]$	$[5,6]$	$[7,8]$	$[9,10]$	$[11,12]$	$[13,14]$	$[15,16]$	$[17,18]$	$[19,]$
Taobao	items	72,273	106,001	46,919	25,354	24,807	16,776	11,286	8,170	5,958	4,648	3,626	30,730
	feedbacks	152,623	2,918,957	64,363	88,260	135,031	124,920	106,703	93,649	80,135	71,916	63,380	2,243,223
Amazon	items	12,399	2,826	8,970	3,429	1,422	525	340	184	98	64	49	144
	feedbacks	24,122	24,054	12,548	11,574	7,662	3,885	3,203	2,100	1,312	990	855	4,047

(b) Evaluation of cold start performance on Taobao. The interval is the accumulation of several bins.

eva.	method	p	subsets of test set (%)			intervals of test set (%)									
			cold-start	*active*	*whole*	$[1,2]$	$[1,4]$	$[1,6]$	$[1,8]$	$[1,10]$	$[1,12]$	$[1,14]$	$[1,16]$	$[1,18]$	*all*
Recall @30	LSTM	-	0.184	0.920	0.884	0.242	0.184	0.133	0.115	0.106	0.103	0.100	0.098	0.100	0.884
	Con.	-	0.153	1.216	1.164	0.173	0.153	0.114	0.101	0.098	0.098	0.097	0.097	0.097	1.164
	Fus.	-	0.144	1.131	1.082	0.174	0.144	0.109	0.105	0.103	0.103	0.106	0.105	0.109	1.082
	3mDAE-1U	0.3	0.269	1.221	1.174	0.362	0.269	0.195	0.171	0.165	0.160	0.157	0.157	0.158	1.174
	3mDAE-2U	0.4	0.621	1.050	1.029	0.839	0.621	0.437	0.378	0.354	0.340	0.333	0.328	0.324	1.029
AUC	LSTM	-	0.608	0.565	0.567	0.657	0.608	0.519	0.487	0.473	0.467	0.463	0.462	0.462	0.567
	Con.	-	0.659	0.691	0.690	0.681	0.659	0.631	0.623	0.620	0.620	0.619	0.621	0.621	0.690
	Fus.	-	0.714	0.688	0.690	0.742	0.714	0.652	0.631	0.622	0.618	0.616	0.616	0.616	0.690
	3mDAE-1U	0.3	0.651	0.681	0.680	0.676	0.651	0.614	0.603	0.600	0.600	0.600	0.601	0.603	0.680
	3mDAE-2U	0.4	0.649	0.683	0.681	0.671	0.649	0.620	0.611	0.607	0.606	0.606	0.607	0.608	0.681

(c) Evaluation of cold start performance on Amazon. The interval is the accumulation of several bins.

eva.	method	p	subsets of test set (%)			intervals of test set (%)									
			cold-start	*active*	*whole*	$[1,2]$	$[1,4]$	$[1,6]$	$[1,8]$	$[1,10]$	$[1,12]$	$[1,14]$	$[1,16]$	$[1,18]$	*all*
Recall @30	LSTM	-	0.000	3.970	1.982	0.000	0.000	0.000	0.003	0.033	0.034	0.066	0.074	0.165	1.982
	Con.	-	0.398	5.263	2.827	0.215	0.398	0.538	0.676	0.826	0.996	1.135	1.192	1.398	2.827
	Fus.	-	0.328	5.413	2.867	0.215	0.328	0.463	0.558	0.702	0.876	1.017	1.114	1.276	2.867
	3mDAE-1U	0.1	0.623	5.675	2.995	0.207	0.323	0.434	0.547	0.692	0.869	1.038	1.144	1.337	2.995
	3mDAE-2U	0.1	0.319	5.454	2.883	0.199	0.319	0.450	0.552	0.705	0.874	1.034	1.149	1.335	2.883
AUC	LSTM	-	0.496	0.721	0.608	0.471	0.496	0.514	0.531	0.549	0.561	0.569	0.576	0.582	0.608
	Con.	-	0.660	0.787	0.723	0.647	0.660	0.669	0.678	0.688	0.696	0.700	0.703	0.707	0.723
	Fus.	-	0.667	0.777	0.722	0.654	0.667	0.676	0.683	0.691	0.697	0.701	0.704	0.707	0.722
	3mDAE-1U	0.1	0.656	0.788	0.722	0.640	0.656	0.667	0.677	0.687	0.694	0.698	0.702	0.705	0.722
	3mDAE-2U	0.1	0.658	0.783	0.720	0.645	0.658	0.666	0.675	0.685	0.692	0.697	0.700	0.703	0.720

While in our work, we expand this general setting because the rest items may produce a large volume of feedbacks. Two new experimental settings are designed, Recall@30 and AUC are applied to test the performance, as shown in Table 8. Furthermore, we compute the improvement to analyze the effect of multi-modal information on cold start items. The improvements are shown in Table 9 and Fig. 5.

4.5.1 Subsets of Test Set

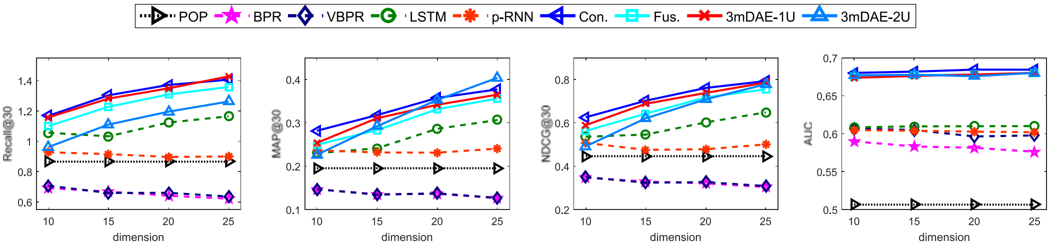
According to each item's support number in the test set, we divide items into three subsets: *cold-start* ($\leqslant 4$), *active* ($\geqslant 5$) and *whole* (test set). Numbers of items of each subset are listed in Table 8a. The cold start items account for 40.5 and 81.4 percent on Taobao and Amazon respectively.

From the perspective of basic performance, as shown in Tables 8b and 8c, the best values are scattered in four variants. It is difficult to draw a consistent conclusion.

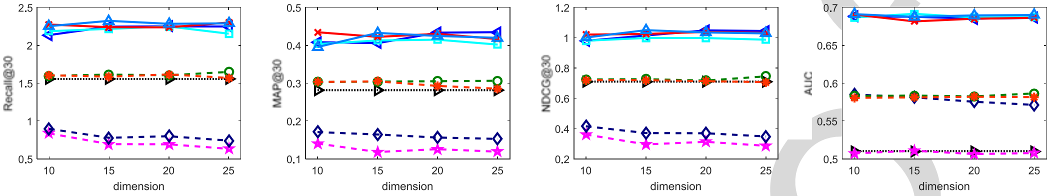
As for the improvement shown in Table 9, most improvements on *cold-start* are higher than those on *whole*, and are much higher than those on *active*. Comparatively, the basic model like LSTM has difficulty in predicting cold start items, while it is easier to obtain good performance on active items. Thus on the contrast, it is easy to design a model to substantially enhance the performance on *cold-start*, while it is more difficult to acquire obvious improvement on *active*. Under such situation, our MV-RNN still performs very well on *active*. For example, most improvements of MV-RNN are over 10 percent on *active*. MV-RNN not only has a significant improvement on cold start items but also has a sufficient improvement on active items.

In Table 9, there are some surprising improvements about Recall@30 on Amazon. We specify the improvement of MV-RNN over LSTM as $3 \times 10^4\%$, because the performance of LSTM on *cold-start* is zero. This poor performance of LSTM can be explained from the perspective of probability. When

EPROOF



(a) Taobao



(b) Amazon

Fig. 4. Recall@30, MAP@30, NDCG@30, and AUC performances on validation set with varied dimensions of latent feature $d = [10, 15, 20, 25]$.

we train a sequence, we practically apply LSTM to model a joint probability $p(x_1, \ldots, x_t)$, where x_i represents an item. When we predict n items in corresponding test sequence, we actually predict a conditional probability $p(x_{t+1}, \ldots, x_{t+n} | x_1, \ldots, x_t)$. Because the 81.4 percent cold start items and the corresponding 50.1 percent feedbacks on Amazon result in limited interactions among users and items, both probabilities are very small. Therefore, it is very hard to make accurate recommendation under Recall@30 on Amazon. After we incorporate the additional content information, 4 variants of MV-RNN have performances of 0.398, 0.328, 0.623 and 0.319 percent respectively. The absolute values are small, but we obtain very large but reasonable improvements. This strange and extreme phenomenon exactly reflects the great power of additional content information and the powerful modeling capability of MV-RNN.

4.5.2 Intervals of Test Set

According to the support number of each item in the test set, we divide items into ten bins (e.g., $[1, 2]$, $[3, 4]$, $[5, 6]$). For example, bin $[1, 2]$ has the items that appear for 1 or 2 times. Numbers of items in each bin are listed in Table 8a. In order to alleviate the fluctuation of performance on each bin, performance is recorded on cumulative bins (e.g., $[1, 4]$) which are called intervals.

When the bin number increases, performance becomes better, as seen from Tables 8 b and 8 c. That is because it is easier to predict frequent items. On Taobao, there is a strange phenomenon. Performance decreases first on a few bins in the front and then increases. As the decrement is not significant, we can still think the performance is growing. Then we mainly focus on the analysis of improvements. For better representation, improvements are illustrated by curves in Fig. 5.

These growth curves do not always have the same change on two datasets. On Taobao, curves tend to be flat. On Amazon, as the bin has a larger proportion of cold start items (seeing from the right side of a figure to its left side), the improvement almost becomes larger. This indicates that multi-modal information is indeed beneficial to relieve cold start. In other words, when the cold start problem gets worse on small bins with a bigger proportion of cold start items, multi-modal information can significantly relieve this problem. Because cold start items have few interactions with users, directly related multi-modal information would effectively represent the item's characteristics and the user's interest.

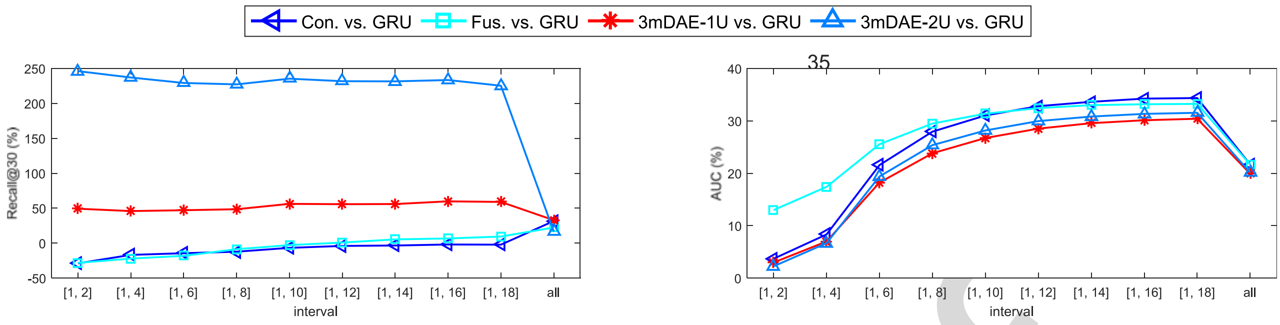
AUC is much more stable than Recall@30. We consider the difference of user's preferences towards positive and negative items in AUC, and the BPR training process exactly maximizes this difference. For Recall@30 curves, there is a large difference between Taobao and Amazon. These curves on Taobao are separate from each other, but they almost come together in the last interval *all*. Perhaps because of the small proportion of feedbacks on the interval $[1, 18]$ (27.0 percent), there would be some fluctuations in the performance of each model. These curves on Amazon have an obvious increasing law when the bin number gets smaller. For AUC curves, the situation is much better. On Taobao, most improvements are stable. For example, improvements of MV-RNN are around 30 percent. On Amazon, the smaller the bin number, the larger the improvement.

TABLE 9
Based on the Cold Start Performance in Table 8, We Compute Improvements (%) on Each Subset

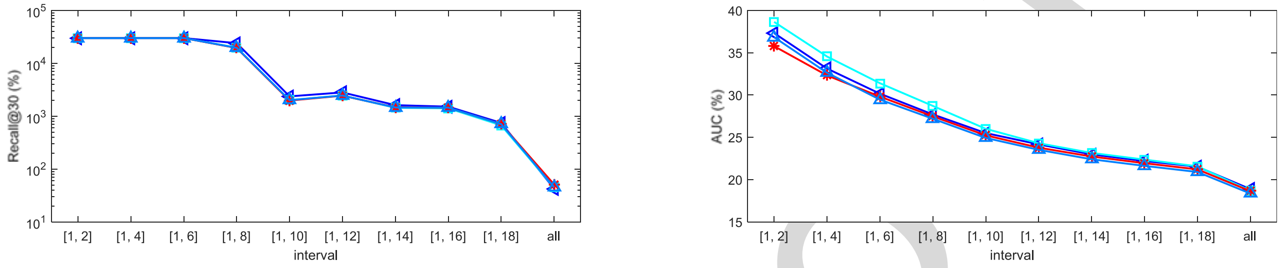
method	Taobao - Recall@30			Taobao - AUC			Amazon - Recall@30			Amazon - AUC		
	cold	*active*	*whole*	*cold*	*active*	*whole*	*cold*	*active*	*whole*	*cold*	*active*	*whole*
Con. versus LSTM	-16.73	32.20	21.70	8.33	22.42	21.67	3×10^4	32.57	42.62	33.12	9.07	18.88
Fus. versus LSTM	-22.05	22.87	22.41	17.41	21.88	21.64	3×10^4	36.34	44.61	34.55	7.76	18.69
3mDAE-1U versus LSTM	45.90	32.68	32.82	6.98	20.65	19.92	3×10^4	42.93	51.10	32.31	9.23	18.65
3mDAE-2U versus LSTM	237.05	14.14	16.46	6.67	20.88	20.13	3×10^4	37.38	45.45	32.63	8.53	18.36

The best corruption levels p for our 3mDAE-1U/2U is the same as in Table 8, and we omit the p in this table. The cold refers to the cold-start.

IEEE Proof



(a) Taobao



(b) Amazon

Fig. 5. Based on the cold start performance in Table 8, we calculate improvements (%) on each interval. The best corruption levels p for our 3mDAE-1U/2U is the same as in Table 8, and we omit the p in this figure.

These curves, especially those on Amazon, can greatly support the following conclusion. Multi-modal information can significantly relieve the item cold start problem. Besides, the worse the cold start, the more powerful the multi-modal information.

4.5.3 Visualization of Learned Features

In this part, we make the visualization of learned features by similarity retrieval to investigate whether they are correlated or complementary. There are five different input features $i_x, i_f, i_g, i_m, [i_x; i_m]$ represented in Eqs. (1), (2), (3), and (4). Given a query item, we select top-5 most similar items based on the euclidean distance for each kind of feature. The features are acquired by 3mDAE-1U under $p = 0.3$ and the results are shown in Fig. 6.

Fig. 6. Visualization of similarity retrieval based on the euclidean distance. Features are acquired by 3mDAE-1U under $p = 0.3$ on Taobao.

Obviously, the similar items under different kinds of features vary greatly, and the multi-view (latent, visual, textual) features are complementary to each other. (1) For the latent feature i_x, the similar items are greatly different from each other as i_x are just learned by the feedback. If the latent features of two items are similar, probably because they were both purchased by many people. (2) Whether it is item itself or the background in the image, the top-5 items based on the visual feature i_f are very similar in appearance. However, the second and the forth items in this line obviously belong to other categories. The visual feature is powerful but can not reflect the intrinsic characteristics of items, like material of clothes. (3) On the other hand, the textual feature i_g is acquired by the item description. It can truly reflect what the product is and can ignore the effect of the background in an image, but it is not intuitive to show the color, shape, etc. (4) The fusion feature i_m is a combination of i_f and i_g. It mainly integrates the external and intrinsic characteristics of the item, such as the style and material of clothes. However, such characteristics can not generate precise recommendation because there is no one-to-one match between each characteristic and each item. (5) The final item feature $[i_x; i_m]$ fuses i_x, i_f, i_g. It can fully reflect the characteristics of an item and help to understand the user's overall interest. In summary, multi-view features i_x, i_f, i_g used in our work are complementary.

5 Conclusion

In this work, we have proposed a novel multi-view recurrent model (MV-RNN) for sequential recommendation and alleviating the item cold start problem. First, we construct comprehensive item representation with latent, visual and textual features by three different combinations. A 3mDAE model is introduced to build the fusion feature based on visual and textual features. Then the user's interest is captured by the recurrent structure. We devise two types of inner structures to handle multi-view features. Next, we design a united objective function to combine the preference loss of BPR and the reconstruction loss of our 3mDAE. Experiments validate the state-of-the-art performance of MV-RNN. The fusion feature of 3mDAE helps to learn more robust features and tackle the missing modalities problem. Experiments confirm that a united inner structure can better leverage the advantage of multi-view features than a separate one. The multi-modal information like the image and text description could indeed significantly alleviate the item cold start problem.

In the future, we would investigate the item detection and segmentation in images. The items in images often have a large proportion of unrelated background, especially in the Taobao dataset. We would like to obtain the more accurate item representation. These can motivate the model to improve performance.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions allowing to improve the quality of this paper. This work is jointly supported by National Key Research and Development Program (2016YFB1001000), National Natural Science Foundation of China (61403390, U1435221).

References

[1] Y. Koren, R. Bell, C. Volinsky, et al., "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, no. 8, pp. 30–37, 2009.

[2] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[3] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 1–8.

[4] Y. Bao, H. Fang, and J. Zhang, "TopicMF: Simultaneously exploiting ratings and reviews for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2–8.

[5] R. He and J. McAuley, "VBPR: Visual Bayesian personalized ranking from implicit feedback," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 144–150.

[6] L. Zhao, Z. Lu, S. J. Pan, Q. Yang, W. Xu, Y. Lee, B. Gao, S. Ma, A. Zhang, S. Mondal, et al., "Matrix factorization+ for movie recommendation," *Proc. Int. Joint Conf. Artif. Intell.*, 2016, Art. no. 1.

[7] J. Chen, C. Wang, and J. Wang, "A personalized interest-forgetting Markov model for recommendations," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 16–22.

[8] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. Int. Conf. World Wide Web*, 2010, pp. 811–820.

[9] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2013, pp. 1044–1054.

[10] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu, "Sequential click prediction for sponsored search with recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1369–1376.

[11] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.

[12] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 729–732.

[13] Q. Liu, S. Wu, and L. Wang, "Multi-behavioral sequential prediction with recurrent log-bilinear model," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1254–1267, Jun. 2017.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, 1997, Art. no. 1735.

[15] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Comput. Sci., EMNLP*, pp. 1724–1734, 2014.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[17] J. Chung, C. Gülçehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2067–2075.

[18] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *ICLR*, 2016.

[19] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. ACM Conf. Recommender Syst.*, 2016, pp. 241–248.

[20] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[21] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015, pp. 77–118.

[22] J.-D. Zhang, C.-Y. Chow, and J. Xu, "Enabling kernel-based attribute-aware matrix factorization for rating prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 798–812, Apr. 2017.

[23] A. Levi, O. Mokryn, C. Diot, and N. Taft, "Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system," in *Proc. ACM Conf. Recommender Syst.*, 2012, pp. 115–122.

[24] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 261–270.

[25] A. E. Raftery, "A model for high-order Markov chains," *J. Roy. Statistical Soc.*, vol. 47, no. 3, pp. 528–539, 1985.

[26] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: A large margin approach," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 361–369.

[27] N. Chen, J. Zhu, F. Sun, and E. P. Xing, "Large-margin predictive latent subspace learning for multiview data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2365–2378, Dec. 2012.

[28] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 641–648.

[29] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop Deep Learn.*, Dec. 2014.

[31] B. Cao, L. Zheng, C. Zhang, P. S. Yu, A. Piscitello, J. Zulueta, O. Ajilore, K. Ryan, and A. D. Leow, "DeepMood: Modeling mobile phone typing dynamics for mood detection," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 747–755.

[32] Y. Li, M. Yang, and M. Z. Zhang, "A survey of multi-view representation learning," 2018.

[33] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.

[34] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 4590–4594.

[35] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[37] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.

[38] M. Chen, Z. Xu, K. Weinberger, and S. Fei, "Marginalized denoising autoencoders for domain adaptation," *ICML*, 2012.

[39] S. Wang, Z. Ding, and Y. Fu, "Coupled marginalized autoencoders for cross-domain multi-view learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2125–2131.

IEEE Proof

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[41] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2014, pp. 1532–43.

[42] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for next basket recommendation," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 403–412.

[43] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 43–52.

[44] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 785–794.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[47] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[48] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

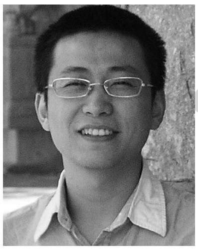
[49] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technol.*, 2016, pp. 1480–1489.

[50] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 278–288.

[51] S. Wu, W. Guo, S. Xu, Y. Huang, L. Wang, and T. Tan, "Coupled topic model for collaborative filtering with user-generated content," *THMS*, vol. 46, no. 6, pp. 908–920, 2016.

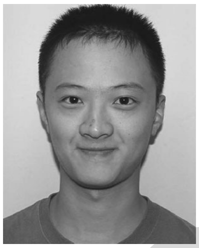
[52] S. Wu, Q. Liu, L. Wang, and T. Tan, "Contextual operation for recommender systems," *TKDE*, pp. 2000–2012, 2016.

Qiang Cui received the BS degree from Shandong University, China, in 2013. He is currently working toward the PhD degree in the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include data mining, machine learning, recommender systems, and information retrieval.



Shu Wu received the BS degree from Hunan University, China, in 2004, the MS degree from Xiamen University, China, in 2007, and the PhD degree from the University of Sherbrooke, Quebec, Canada. He is an associate professor with the Center for Research on Intelligent Perception and Computing (CRIPAC). He has published more than 20 papers in the areas of data mining and information retrieval at international journals and conferences, such as the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Human-Machine Systems*, AAAI, ICDM, SIGIR, and CIKM. He is a member of the IEEE.

Qiang Liu received the BS degree in electronic science from Yanshan University, China, in 2013. He is currently working toward the PhD degree in the Center for Research on Intelligent Perception and Computing (CRIPAC). His research interests include machine learning, data mining, user modeling, and information credibility evaluation. He has published several papers in the areas of data mining and information retrieval at international journals and conferences, such as the *IEEE Transactions on Knowledge and Data Engineering*, AAAI, SIGIR, CIKM, and ICDM.



Wen Zhong received the BS degree in software engineering from Shandong University, China, in 2015. He is currently working toward the MS degree in computer science at the University of Southern California. His research interests include machine learning and natural language processing.

Liang Wang received the BEng and MEng degrees from Anhui University, in 1997 and 2000, respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2004. Currently, he is a full professor of the Hundred Talents Program at the National Lab of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition, and computer vision. He has widely published in highly ranked international journals such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *IEEE Transactions on Image Processing* and leading international conferences such as CVPR, ICCV, and ICDM. He is a fellow of the IEEE and fellow of the IAPR.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

IEEE Proof