Data Science and Engineering (2019) 4:240–253
https://doi.org/10.1007/s41019-019-00101-4

MMM: Multi-source Multi-net Micro-video Recommendation with Clustered Hidden Item Representation Learning

Jingwei Ma[1] · Jiahui Wen[2] · Mingyang Zhong[3,4] · Weitong Chen[1] · Xue Li[1]

Received: 13 May 2019 / Revised: 22 August 2019 / Accepted: 22 August 2019 / Published online: 6 September 2019

Abstract
Unlike traditional video recommendations, micro-video inherits the characteristics of social platforms, such as social relation. A large amount of micro-videos showing explosive growth is badly affecting the user's choice. In this paper, we propose a multi-source multi-net micro-video recommendation model that recommends micro-videos fitting users' best interests. Different from existing works, as micro-video inherits the characteristics of social platforms, we simultaneously incorporate multi-source content data of items and multi-networks of users to learn user and item representations for recommendation. This information can be complementary to each other in a way that multi-modality data can bridge the semantic gap among items, while multi-type user networks, such as following and reposting, are able to propagate the preferences among users. Furthermore, to discover the hidden categories of micro-videos that properly match users' interests, we interactively learn the user–item representations and perform the hidden item category clustering. The resulted categorical representations are interacted with user representations to model user preferences at different levels of hierarchies. Finally, multi-source content item data, multi-type user networks and hidden item categories are jointly modelled in a unified recommender, and the parameters of the model are collaboratively learned to boost the recommendation performance. Experiments on a real dataset demonstrate the effectiveness of the proposed model and its advantage over the state-of-the-art baselines.

Keywords Micro-video recommendation · Clustering · Social recommendation

1 Introduction

The explosion of micro-videos has arisen as a problem on social media in recent years, as the sheer volume of available micro-videos can often undermine a users' capability to choose the micro-videos that best fit their interests. Recommender systems appear as a natural solution to this problem, helping social applications to precisely determine the information offering to consumers and allowing users to quickly find the most useful information [12]. Most of the previous methods for recommendation are based on collaborative filtering (CF) [22, 29] that assumes users who show similar preferences in the past are supposed to make similar choices in the future. However, CF-based methods suffer from data sparseness and cold-start problem [41], as the only information they employed is the user–item interaction matrix that is extremely sparse, and it is impossible to make recommendation for an unseen user or item unless its latent representation is learned beforehand. To approach those problems, many previous works incorporate additional information sources by joint learning users, items and auxiliary information such as texts [1, 10, 38, 41], images [5, 7, 30, 38], video [5, 18, 19] and social connections [17, 37].

The basic idea of collaborative filtering is that users who show similar preferences in the past are supposed to make similar choices in the future. One of the most popular collaborative filtering techniques is matrix factorization [22] that factorizes a user–item interaction matrix into user and item hidden vectors. Thereby, the users and items are mapped to the same latent space, and their similarities can be measured based on the hidden representations. Although

✉ Jingwei Ma
Jingwei.ma@uq.edu.au

1 School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia

2 National University of Defense Technology, Changsha, China

3 Centre for Intelligent Systems, Central Queensland University, Brisbane, Australia

4 Department of Automation and Robotic Engineering, Chongqing Meiqi Industry Co., Ltd., Chongqing, China

those methods that consider auxiliary information can be effective for recommendation task are not applicable for micro-video recommendation due to the following reasons. First, micro-videos usually involve multiple modalities such as textual features and visual features, and each modality reveals different characteristics of micro-videos. For example, the descriptive text associated with a micro-video indicates the main content of the micro-video, while frames in a micro-video carrying the partial information of the content are more intuitive than text. Therefore, it is necessary to design a recommender system that can cater for different characteristics of auxiliary information and incorporate high-level representations of modality information into a unified framework for micro-video recommendation. Most of the existing works [7, 38] only leverage data of a single modality or from a single source, and they usually specifically design the process of information extraction from the single-source data and tightly couple it with the recommend frameworks, which limits the scalability of the frameworks, such as integrating heterogeneous data sources. Second, many previous works employ social connections for improving recommendation accuracy based on the idea that social friends interrelated with strong social connections are more likely to have similar interests. However, many of them [31, 37] simply utilize the following relationship as social regularization, while discarding other user–user graphs. As demonstrated in the previous research [3], information from a single social network may conflict with the true reasons underlying the user–item interactions, while clues from multiple graphs can be complementary for each other for accurate recommendation. For example, family members may be ”friends” in a social network but with completely different item preferences, while users consume the same item explicitly indicating the similarity of their interests.

To address those problems, we propose a multi-source multi-net method for micro-video recommendation. Beside user–item interactions, we leverage heterogeneous information sources for better item profiling and integrate the latent representations of the multiple sources into a unified recommender framework to promote personalized recommendation. Information from different domains can be complementary to each other and reveal the true factors of user preferences over items. For example, some people view a micro-video because they are attracted by the overall description of the video, while other people may be interested in the visual images of the video. Furthermore, we employ multiple social networks for propagating user representations in a shared latent space based on the idea that strong social ties are supposed to be located in a proximity close to each other. We learn the hidden representations of users and items in an end-to-end neural network and back-propagate the recommendation errors to update the parameters of the network jointly; thus, the user and item representations can best explain the user preferences over the items by considering heterogeneous data sources and multiple social networks. The contributions of this paper are listed as follows:

- We incorporate multiple information sources and multiple user networks in a unified model for micro-video recommendation. Each modality can be complementary to each other for better modelling user and item representations.
- We propose a micro-video recommender framework that leverages heterogeneous information sources. Data source of difference modality is processed differently, and their representations are jointly learned with user representations in the unified framework to best explain the user–item interactions.
- We propose to discover hidden item category information based on clustered hidden item representation for the recommendation task. The hierarchy of item-level and category-level information can reflect the underlying reason of user preferences over items. We exploit user–user relations based on multiple social networks to regularize user representations. User representations are propagated through the social connections, so that users with strong social ties are closed to each other in the shared latent space.
- We demonstrate the effectiveness of the proposed solution with a real dataset and present insights and its advantage over state-of-the-art recommender methods with comprehensive experiments and analysis.

The remainder sections are organized as follows. Section 2 describes the related work. Section 3 presents the details of the proposed method, followed by Sect. 4 that describes the evaluation on a real dataset. Section 4 concludes this paper.

2 Related Work

2.1 Video Recommendation

Many existing video-oriented sites, such as YouTube[1] and MSN Video,[2] have provided video recommendation services. Content-based filtering approaches [25, 27] are exploited in YouTube, where videos are recommended to users based on the past viewed videos. In VideoReach [22], video recommendation is performed based on the multimodal relevance and users' click-through data. It integrates textual, visual and acoustic modalities with an attention

[1] https://www.youtube.com/?hl=zh-cn.

[2] https://www.msn.com/en-us/video.

function. The main limitation of these methods is that they only consider the video–video relations, ignoring the related users' preferences that are important. In MovieLens system [24], the users are required to rate films that they have seen from 1 (Awful) to 5 (Must Watch) stars. The films are recommended to other users who have a high correlation coefficient (stars) with the current user. Collaborative filtering ignores the video attributes such as descriptions and keywords. Its performance may be undesirable when the ratings are insufficient.

2.2 Neural Recommendation

In the past decade, deep learning techniques have been applied in several fields such as computer vision, speech recognition and natural language processing. As for recommendation, many previous works have tried to combine various neural network structures with collaborative filtering to boost the recommendation performance [13]. For example, the works [9, 21] combine generalized matrix factorization with multi-layer perceptron to capture the interrelations between users and items. Some others [14, 32] apply auto-encoders to model user–item interactions, and the recommendations are made by reconstructing the user preferences over the item with the pre-trained de-noising auto-encoders. Even though these are demonstrated to be effective in previous works, they are inapplicable in our cases as the models are learned solely based on the user–item interactions. Deep learning methods are able to extract abstract features of data automatically through many layers of nonlinear transformation [35]. The high-level features are extracted in a way towards the optimization of some pre-defined objective functions [2], and hence, the extracted features are highly representative for the data in a specific learning task.

2.3 Social Recommendation

Some works consider social connections to jointly model user representations based on the idea that users with strong social connections are more likely to have similar interests, and they need to be close to each other in the projected low-dimensional continuous latent space. For example, [17, 31] explicitly regularize social friends to have similar representations. Yang et al. [37] exploit the second-order information by predicting user–item contexts with user–item representations; hence, user–item with similar contexts is constrained to be similar to each other in the latent space. In [36], the trust network is leveraged to propagate user representations. The adjacency matrix in the trust network is factorized in the same way as the rating matrix, and truster representations are limited to share the same feature space with the active users in the rating matrix to bridge them. However, most of those works leverage a single-user network to propagate user similarities; therefore, discovering similar users with information of single modality may provide conflict evidence and compromise recommendation accuracy [26]. On the contrary, we model user representations with multiple user networks, where information from the user networks can be complementary with each other to best characterize the user similarity in the shared latent space.

Some previous works [8, 34] uncover latent item groups for recommendation. It draws user–item group from a multinomial distribution parameterized by user–item representations and then generates the rating score from exponential family parameterized by the co-clustered user–item group pair. The generative characteristic limits the scalability of the model, as it is unable to incorporate other information sources for modelling user–item ratings. The modelling of rating scores as the interactions between user–item groups means that it is unable to incorporate the hierarchy of items and item categories for better user preferences profiling.

2.4 Joint Recommendation

Recently, many works propose to jointly model the auxiliary information associated with users or items for recommendation. The underlying reason of incorporating additional data sources is that they are highly interrelated with user–item interactions and can help to alleviate the data sparseness and cold-start problem. The data modalities under consideration include text, audio and visual. Those observable attributes are processed into abstract representations with popular deep learning techniques (e.g. CNNs, RNNs) and interact them with user or item representations to compensate sparse user–item interactions. For example, Zhang et al. [38] model textual and visual representations for the items with stacked de-noising auto-encoders and assume the item representations to be Gaussian distribution on the residual noise of the linear combination of textual and visual representations. In [40], user and item representations are modelled in each information source (e.g. rating, image and text), and a joint hidden layer is applied to integrate the representations for the recommendation task. However, most of those methods linearly combine representations from different sources, without considering the hierarchy of items and item categories for better modelling user–item interactions. Another limitation of those methods is that the underlying user networks that interlink the user have not been explicitly exploited. The projection of users into low-dimensional representations needs to preserve local structures in the user networks. However, most of those methods customize the modelling process for data source of different modalities and tightly couple them with the key collaborative filtering effect, which negatively affects the scalability of the methods.

Moreover, when it comes to the interactions between user and item representation, they resort to latent factor model that is vulnerable to data sparsity.

3 The Proposed Model

This section describes the details of the proposed model. Section 3.1 provides the problem formulation. After that, Sects. 3.2 and 3.5 present the modelling of the content of micro-videos and the modelling of user networks that consider multiple types of user networks, respectively, and multi-source data contents are utilized in our model, such as textual, visual features of micro-videos and user networks. Section 3.3 details the modelling of hidden category that learns the hidden categories of micro-videos that properly match users' interests. Section 3.6 describes the unified model that incorporates multiple data sources.

3.1 Problem Formulation

We denote a user–item interaction matrix as $\mathbf{R} \in \mathbb{R}^{M \times N}$, where M and N denote the number of users and items, respectively. The non-empty entries $\boldsymbol{R}_{ij}$ refer to the positive interaction between user $u_i \in \mathbb{U}$ and item $v_j \in \mathbb{V}$. In our case, each item v_j is associated with a textual description and a key frame $(\mathbf{x}_j, \mathbf{g}_j)$. More importantly, there are multiple user networks, representing different relations such as following and liking. For a user network $G^k = (\mathbb{U}, \mathbb{E}^k)$, where $\mathbb{U}, \mathbb{E}^k$ is the set of nodes and edges, $(u_i, u_l) \in \mathbb{E}^k$ means there is a positive connection between u_i and u_l. In this work, our framework considers following and liking user networks as examples, which are easily extended to incorporate other user networks such as reposting. Given the interaction matrix $\mathbf{R}$, the set of user.$\mathbb{U}$ and item $\mathbb{V}$, the observable texts and images $(\mathbf{x}_j, \mathbf{g}_j, j = 1, \ldots, N)$ associated with the items and the user network $G^k, k = 1, 2$, our task is to predict the missing values in $\mathbf{R}$.

3.2 Content Modelling

In this subsection, we describe the modelling of content information (i.e. textual and visual information) for the micro-videos. Since similar items are more likely to have similar textual descriptions and visual information, the latent representations of those items are supposed to be in a proximity close to each other in the shared latent space. Therefore, the content information is able to bridge the semantic gaps between items, and we can learn better item latent representations by exploiting content information.

3.2.1 Textual Representations

The descriptive text $\mathbf{x}_j$ associated with a micro-video v_j summarizes the overall content of the item, and underlying reason of modelling textual data sources is that a user's preference over an item can be explained by the fact that the user is attracted by the overall content of a micro-video.

First, we transform each text, $\mathbf{x}_j = \{w_j^n\}_{n=1}^{|\mathbf{x}_j|}$ where w_j^n is the nth word in $\mathbf{x}_j$, into embedding vectors with Glove: $\mathbf{E}_j = \{\mathbf{e}_j^n\}_{n=1}^{|\mathbf{x}_j|}$. The embedding vectors are then fed into a convolution layer and a max-pooling layer to obtain the representation of each document:

$$\mathbf{o}_j = \text{CNN} - \text{maxpooling}(\mathbf{E}_j) \tag{1}$$

where $\mathbf{o}_j \in \mathbb{R}^{n1}$ is the output of the CNN and max-pooling layer. More details of this process are referred to [41].

Specifically, for each filter map $K_j \in \mathbb{R}^{(k \times m)}$, we generate a feature map when moving the filter map through the embedding vectors. The feature map is a vector as shown in Eq. (2), where k is the embedding size and m is the filter size.

$$z_t = f(\mathbf{E}^u_{[1:k,t:t+m-1]} * K_j + b_j) \tag{2}$$

where $*$ is convolution operator and f is an activation function (i.e. relu). Each filter map produces a feature map of different lengths depend on the size of filter map. We then apply max-pooling over each feature map to extract the most significant feature as presented in Eq. (3), and reduce the feature map into a scalar, o_j.

$$o_j = \max\{z_1, z_2, \ldots, z_{|\mathbf{x}_j|-m+1}\} \tag{3}$$

In practice, we apply multiple filter maps of various sizes onto the embedding vectors, so that we can extract local features of different n-gram. Since each filter map produces a scalar through the convolution and max-pooling operation, we concatenate all the scalars produced by the filter maps of different sizes into a vector, as shown in Eq. (4).

$$\mathbf{o}_j = \{o_1, o_2, \ldots, o_{n1}\} \tag{4}$$

where $n1$ is the number of filter maps in the CNNs. Therefore, the length of the hidden representation of an item depends on the hyperparameter of the neural network rather than the length of the input length. In this way, this convolution and pooling scheme can naturally deal with the descriptive texts of varied length. The concatenated vector $\mathbf{o}_j$ is then passed to a fully connected layer to obtain the hidden representation for the descriptive text $\mathbf{x}_j$, as shown in Eq. (5).

$$\mathbf{h}_j^{text} = f(\mathbf{W}_{text} \times \mathbf{o}_j + \mathbf{b}_{text}) \tag{5}$$

where $\mathbf{W}_{text} \in \mathbb{R}^{n1 \times d}$ and $\mathbf{b}_{text} \in \mathbb{R}^d$ are the transformation matrix and the biases, in which d is the pre-defined hidden size. Notice that, unlike previous works that use RNNs to process texts, we employ CNNs to extract hidden text

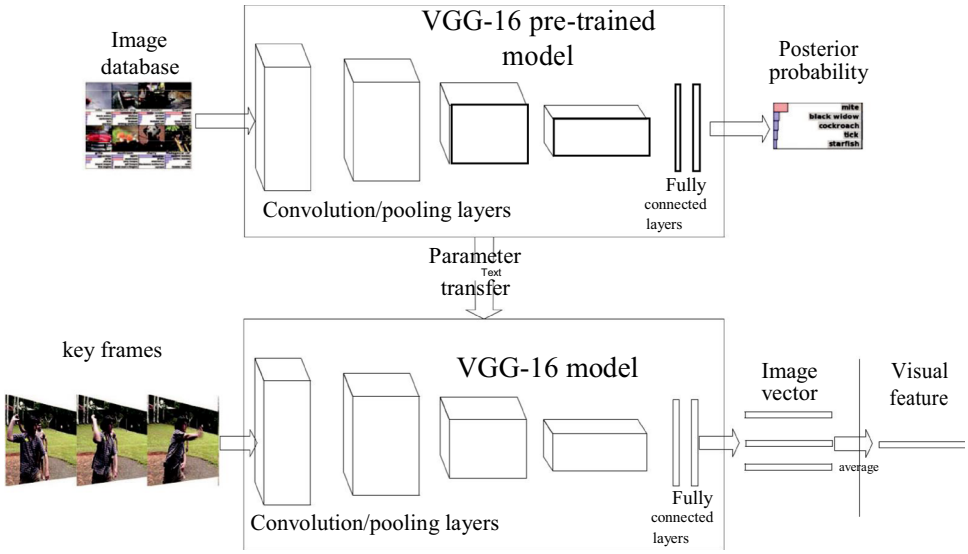


Fig. 1 An illustration of visual features extraction process

representations. The underlying reason is that the key words in the descriptive text of a micro-video are more important for attracting users than the word orders. Since the max-pooling scheme in CNNs is able to capture the most significant features and the embedding method can bridge the semantic gaps among words, CNNs are able to locate the most informative features for recommendation by combining these two mechanisms together.

3.2.2 Visual Representations

The idea of extracting visual features from video frames for micro-video recommendation is that frames can reflect the user specific interests in a straightforward way, which are usually difficult to be described by text [7]. Therefore, visual and textual features reveal user preferences from different point of views and they are complementary with each other for micro-video recommendation.

In this paper, we choose CNNs for extracting visual features, as CNNs are powerful in learning high-level visual representations for image classification and object detection. We utilize the pre-trained VGG-16 [28] net to obtain the visual features of video frames. The VGG-16 model is trained on large-scale images and classifies an image into one of the 1000 classes. For each image as input into the VGG-16 model, we extract the 4096-dimension activations of the fully connected layers prior to the last layer as high-level features for each of the key frames and then take the average of the high-level features of the key frames as the visual features [39]. For each micro-video frame $\mathbf{g}_j$, we use the output of the third-to-last layer and pass it to a fully connected layer to obtain the representation of the frame, as shown in Eq. (6):

$$\mathbf{h}_j^{image} = f(\mathbf{W}_{image} \times \mathrm{CNN}(\mathbf{g}_j) + \mathbf{b}_{image}) \tag{6}$$

where $\mathrm{CNN}(\mathbf{g}_j) \in \mathbb{R}^{4096}$ is the output of the third-to-last layer of the VGG-16 model. The VGG-16 model consists of 13 convolution, five max-pooling, three fully connect and one softmax layers. It takes images of size 244 * 244 * 3 as input and classifies each of them into one of the predefined 1000 classes. The parameters of VGG-16 are pre-trained with large-scale image database, and the intermediate representations for an input image can be obtained by performing convolution and max-pooling operations with the available parameters. The output of the last two layers is ignored as they are used for image classification purpose. $\mathbf{W}_{image} \in \mathbb{R}^{4096 \times d}$, $\mathbf{b}_{image} \in \mathbb{R}^d$ are the transformation matrix and bias vector, respectively. The process of extracting textual features is shown in Fig. 1.

3.2.3 Latent-Content Modelling

The idea of modelling item content is to learn mapping function to project item latent representations and content representations (i.e. $\mathbf{h}_j^{text}, \mathbf{h}_j^{image}$) of different modalities into a common space so that the similarities between them can be directly measured. We denote the latent representation of item v_j as $\mathbf{v}_j$ and then the conditional probability of observing an item latent representation given its content representations as follows:

$$\begin{aligned} P\left(\mathbf{h}_j^{text} | \mathbf{v}_j\right) &= \frac{1}{1 + e^{-\mathbf{v}_j^\top \mathbf{M}^{text} \mathbf{h}_j^{text}}} \\ &= \sigma\left(\mathbf{v}_j^\top \mathbf{M}^{text} \mathbf{h}_j^{text}\right) \\ P\left(\mathbf{h}_j^{image} | \mathbf{v}_j\right) &= \frac{1}{1 + e^{-\mathbf{v}_j^\top \mathbf{M}^{image} \mathbf{h}_j^{image}}} \\ &= \sigma\left(\mathbf{v}_j^\top \mathbf{M}^{image} \mathbf{h}_j^{image}\right) \end{aligned} \tag{7}$$

where $\mathbf{M}^{text}, \mathbf{M}^{image}$ are two linear transformation matrices that map an item latent representation and its content representations into a common space. The basic idea of Eq. (7) is that the latent representation of an item is similar to its content representation in the shared common space. The employment of sigmoid function for measuring the similarity between two objects is commonly used in previous works [4, 33].

3.3 Hidden Category Modelling

Existing works [6, 12] have demonstrated that information of different hierarchies can be incorporated to boost recommendation performance. For example, the category information of an item can help to capture user preferences in a more general granularity. While a user's interest in a specific item is unclear, his/her general taste on the item category is obvious. Therefore, category and item representations can be complementary with each other for better modelling user preferences. However, the category information in our case is not explicitly available, so we propose to discover category-level information for the items, and model user preferences on both item and category levels.

Specifically, we perform clustering over the hidden item representations and results in several cluster centroids. Instead of regarding the centroids as category representations, we draw the category representations from a Gaussian distribution parameterized by the centroids. Thus, the category representations can be used for modelling user preference in a more general granularity, which is detailed in the next subsection. Notice that the clustering process is iteratively performed with the item representations learning process, and they mutually benefit each other. As items consumed by similar users share similar characteristics, the learning of item representations drives similar items to have similar representations, and it benefits the clustering process. Furthermore, the category information can help to reveal real user preferences, which in return benefits the learning of representative item latent feature vectors.

In this paper, we employ K-means for discovering latent categories and denote the category id for item v_j as c_j. The clustering process aims to minimize the following objects.

$$argmin \sum_{k=1}^{K} \sum_{c_j=k} ||\mathbf{v}_j - \boldsymbol{\nu}_k||^2 \tag{8}$$

where K is the pre-specified category number and $\boldsymbol{\nu}_k$ is the centroid for category k. The parameters $\{c_j\}_{j=1}^{N}$ and $\{\boldsymbol{\nu}_k\}_{k=1}^{K}$ are iteratively updated as in Eq. (9).

$$\begin{aligned} c_j &= argmin_k ||\mathbf{v}_j - \boldsymbol{\nu}_k||^2 \\ \boldsymbol{\nu}_k &= \frac{\sum_{c_j=k} \mathbf{v}_j}{\sum_j I(c_j = k)} \end{aligned} \tag{9}$$

where $I(x) = 1$ if x holds, and 0 otherwise. Instead of using $\{\boldsymbol{\nu}_k\}_{k=1}^{K}$ as the category representations, we introduce extra vectors $\{\boldsymbol{\theta}_k\}_{k=1}^{K}$ for representing the categories and assume Gaussian distribution on the residual noise of the clustered centroids as

$$P(\boldsymbol{\nu}_k|\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{\nu}_k|\boldsymbol{\theta}_k, \sigma_c^2\mathbf{I})\mathcal{N}(\boldsymbol{\theta}_k|0, \sigma_\theta^2\mathbf{I}) \tag{10}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2 and $\mathbf{I}$ is the identity matrix. $\mathcal{N}(\boldsymbol{\theta}_k|0, \sigma^2\mathbf{I})$ is the Gaussian prior we place on the category representations to avoid overfitting.

3.4 Interaction Modelling

In this subsection, we introduce the modelling of user–item interaction. We choose probabilistic matrix factorization (PMF) [23] as our basic model for micro-video recommendation, as it is one of the most popular collaborative filtering models and is commonly used for recommendation tasks [30]. In this work, we have hierarchy of item- and category-level representations, and to capture user preferences at different levels of granularity, we assume Gaussian distribution over observed interaction data as,

$$\begin{aligned} P(\mathbf{R}_{ij}|\mathbf{u}_i, \mathbf{v}_j, \boldsymbol{\theta}_{c_j}) &= \mathcal{N}(\mathbf{u}_i|0, \sigma_u\mathbf{I})\mathcal{N}(\mathbf{v}_j|0, \sigma_v^2\mathbf{I}) \\ &= \mathcal{N}\Big(\mathbf{R}_{ij}|\mathbf{u}_i^T(\mathbf{v}_j + \boldsymbol{\theta}_{c_j}), \sigma^2\mathbf{I}\Big) \end{aligned} \tag{11}$$

where $\mathcal{N}(\mathbf{u}_i|0, \sigma_u\mathbf{I}), \mathcal{N}(\mathbf{v}_j|0, \sigma_v^2)$ are the Gaussian priors we place on the user and item representations $\mathbf{u}_i, \mathbf{v}_j$, respectively. In Eq. (11), $\mathbf{u}_i^T\boldsymbol{\theta}_{c_j}$ explicitly model user's general preferences over the item categories and $\mathbf{u}_i^T\mathbf{v}_j$ model user preferences over the item content, as item representations are collaborative modelled with the raw content information in the common latent space (shown in Eq. (7)). Therefore, we model user interests with a hierarchy of item- and category-level information, which can be complementary with each other for promoting recommendation performance.

3.5 User Network Modelling

In social recommendation, the behaviour of a user is affected by his social neighbours through social inference; hence, user preferences can be propagated through the social ties and encourage users with strong social connections to have the similar interests. Following [11], we employ probabilistic social matrix factorization to propagate preferences among social users. The advantage of our model is that we utilize multiple user networks to accurately model user's interests. We denote $\mathbf{u}_i^k$ as the representation of user u_i in kth user network. A user's representation depends on the latent representations of the connected social friends.

$$\mathbf{u}_i^k = \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k \tag{12}$$

where N_i^k is set of social ties of the user u_i in kth user network, and $T_{in}^k = \frac{1}{N_i^k}$. The above equation implies that the representation of a user is the average of the representations of his/her connected neighbours. Similarly, the unified representation of a user across the user networks is dependent on the representations of the user in each network. We formulate the unified representation $\mathbf{u}_i$ as the weighted sum of the representations $\mathbf{u}_i^k$ across the networks.

$$\mathbf{u}_i = \sum_{k=1}^{K} \delta(\pi_{ik}) \mathbf{u}_i^k \tag{13}$$

where $\delta(\pi_{ik})$ is a softmax function in the form of $\delta(\pi_{ik}) = \frac{exp(\pi_{ik})}{\sum_k exp(\pi_{ik})}$. $\delta(\pi_{ik})$ can be explained as the impact of user representation in each individual network on the unified user representation. Considering the conditional probability of unified user representations, we have:

$$\begin{aligned}
&P\Big(\mathbf{u}_i, \{\mathbf{u}_i^k\}_{k=1}^K | \{T^k\}_{k=1}^K, \sigma_{U_1}^2, \sigma_{U_2}^2, \sigma_T^2\Big) \\
&\quad \propto P\Big(\mathbf{u}_i | \{\mathbf{u}_i\}_{k=1}^k, \sigma_{U_1}^2 \mathbf{I}\Big) \prod_k \Big(P\Big(\mathbf{u}_i^k | 0, \sigma_{U_2}^2 \mathbf{I}\Big) \\
&\qquad \times P\left(\mathbf{u}_i^k | T^k, \sigma_T^2 \mathbf{I}\right)\Big) \\
&\quad = \mathcal{N}\left(\mathbf{u}_i | \sum_{k=1}^{K} \delta(\pi_{ik}) \mathbf{u}_i^k, \sigma_{U_1}^2 \mathbf{I}\right) \\
&\qquad \times \prod_{k=1}^{K} \left(\mathcal{N}\Big(\mathbf{u}_i^k | 0, \sigma_{U_2}^2 \mathbf{I}\Big) \right. \\
&\qquad \left. \times \mathcal{N}\left(\mathbf{u}_i^k | \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k, \sigma_T^2 \mathbf{I}\right)\right)
\end{aligned} \tag{14}$$

The objective function for user regularization can be obtained by taking negative log-likelihood of Eq. (14). By taking the negative log-likelihood of Eq. (14) and ignoring the constant, we have the objective function for the user regularization:

$$\begin{aligned}
O_r = &\frac{1}{2} \sum_{i=1}^{N} \left(\left(\mathbf{u}_i - \sum_{k=1}^{K} \pi_{ik} \mathbf{u}_i^k \right)^T \left(\mathbf{u}_i - \sum_{k=1}^{K} \pi_{ik} \mathbf{u}_i^k \right) \right) \\
&+ \frac{\lambda_U}{2} \sum_{k=1}^{K} \sum_{i=1}^{N} (\mathbf{u}_i^k)^T (\mathbf{u}_i^k) \\
&+ \frac{\lambda_T}{2} \sum_{k=1}^{K} \sum_{i=1}^{N} \left(\mathbf{u}_i^k - \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k \right)^T \\
&\times \left(\mathbf{u}_i^k - \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k \right)
\end{aligned} \tag{15}$$

where $\lambda_U = \sigma_{U1}^2 / \sigma_{U2}^2$, $\lambda_T = \sigma_{U1}^2 / \sigma_{U2}^2$.

3.6 The Unified Model

With the aforementioned information modelling, the conditional probability of the latent parameters given the observed data can be modelled as follows:

$$\begin{aligned}
&P\Big(\mathbb{U}, \mathbb{V}, \{\theta\}_{k=1}^K | \mathbf{R}, \{\mathbf{x}_j\}_{j=1}^N, \{\mathbf{g}_j\}_{j=1}^N, \{G^k\}_{k=1}^2, \{\boldsymbol{\nu}_k\}_{k=1}^K\Big) \\
&\quad \propto P\Big(\mathbf{R} | \mathbb{U}, \mathbb{V}, \{\theta\}_{k=1}^K\Big) P(G|\mathbb{U}) P\Big(\{\mathbf{x}_j\}_{j=1}^N | \mathbb{V}\Big) P\Big(\{\mathbf{g}_j\}_{j=1}^N | \mathbb{V}\Big) \\
&P\Big(\{\boldsymbol{\nu}_k\}_{k=1}^K | \{\boldsymbol{\theta}_k\}_{k=1}^K\Big) P(\mathbb{U}) P(\mathbb{V}) P\Big(\{\boldsymbol{\theta}\}_{k=1}^K\Big) \\
&\quad = \prod_{(u_i, v_j)} \left\{ \mathcal{N}\Big(\mathbf{R}_{ij} | \mathbf{u}_i^T (\mathbf{v}_j + \boldsymbol{\theta}_{c_j}), \sigma^2 \mathbf{I}\Big) \mathcal{N}\left(\mathbf{u}_i | \sum_{k=1}^{K} \delta(\pi_{ik}) \mathbf{u}_i^k, \sigma_{U_1}^2 \mathbf{I}\right) \right. \\
&\qquad \times \prod_{k=1}^{K} \left[\mathcal{N}\Big(\mathbf{u}_i^k | 0, \sigma_{U_2}^2 \mathbf{I}\Big) \mathcal{N}\left(\mathbf{u}_i^k | \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k, \sigma_T^2 \mathbf{I}\right) \right] \\
&\qquad \times \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{text} \mathbf{h}_j^{text}\Big) \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{image} \mathbf{h}_j^{image}\Big) \\
&\qquad \left. \times \mathcal{N}(\boldsymbol{\nu}_{c_j} | \boldsymbol{\theta}_{c_j}, \sigma_c^2 \mathbf{I}) \mathcal{N}(\boldsymbol{\theta}_{c_j} | 0, \sigma_\theta^2 \mathbf{I}) \mathcal{N}(\mathbf{u}_i | 0, \sigma_u \mathbf{I}) \mathcal{N}(\mathbf{v}_j | 0, \sigma_v^2 \mathbf{I}) \right\}
\end{aligned} \tag{16}$$

By taking the negative likelihood of the above conditional probability, the loss function that needs to be minimized is as follows:

$$\begin{aligned}
\mathcal{L} = &\sum_{(u_i, v_j)} \left\{ \frac{1}{2} [\mathbf{R}_{ij} - \mathbf{u}_i^T (\mathbf{v}_j + \boldsymbol{\theta}_{c_j})]^2 \right. \\
&+ \sum_k \left[\frac{\lambda_T}{2} \left(\mathbf{u}_i^k - \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k \right)^2 + \frac{\lambda_{U_2}}{2} (\mathbf{u}_i^k)^2 \right] \\
&+ \frac{\lambda_{U_1}}{2} \left(\mathbf{u}_i - \sum_k \delta(\pi_{ik}) \mathbf{u}_i^k \right)^2 \\
&- \alpha \log \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{text} \mathbf{h}_j^{text}\Big) - \alpha \log \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{image} \mathbf{h}_j^{image}\Big) \\
&\left. + \frac{\lambda_c}{2} (\boldsymbol{\nu}_{c_j} - \boldsymbol{\theta}_{c_j})^2 + \frac{\lambda_\theta}{2} (\boldsymbol{\theta}_{c_j})^2 + \frac{\lambda_u}{2} (\mathbf{u}_i)^2 + \frac{\lambda_v}{2} (\mathbf{v}_j)^2 \right\}
\end{aligned} \tag{17}$$

where $\lambda_T = \sigma_T^2 / \sigma^2$, $\lambda_{U_1} = \sigma_{U_1}^2 / \sigma^2$, $\lambda_{U_2} = \sigma_{U_2}^2 / \sigma^2$, $\lambda_c = \sigma_c^2 / \sigma^2$, $\lambda_\theta = \sigma_\theta^2 / \sigma^2$, $\lambda_u = \sigma_u^2 / \sigma^2$, $\lambda_v = \sigma_v^2 / \sigma^2$, $\alpha = 2\sigma^2$.

3.7 Parameter Learning

With the constructed objective function, its local minimum is targeted by taking the derivative with respect to the parameters that are updated along the gradient direction. The derivatives of the objective function are shown as follows:

3.7.1 Update $\mathbf{u}_i$

The partial derivative of $\mathcal{L}$ w.r.t unified user representation $\mathbf{u}_i$ is given as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_i} = \sum_{(u_i,v_j)} \Bigg\{ -\Big[\mathbf{R}_{ij} - \mathbf{u}_i^T(\mathbf{v}_j + \boldsymbol{\theta}_{c_j})\Big](\mathbf{v}_j + \boldsymbol{\theta}_{c_j}) + \lambda_{U_1}\Bigg(\mathbf{u}_i - \sum_k \delta(\pi_{ik})\mathbf{u}_i^k\Bigg) + \lambda_U \mathbf{u}_i \Bigg\} \tag{18}$$

3.7.2 Update $\mathbf{v}_j$

The partial derivative of $\mathcal{L}$ w.r.t $\mathbf{v}_j$ is given as

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{v}_j} = \sum_{(u_i,v_j)} \Bigg\{ &-\Big[\mathbf{R}_{ij} - \mathbf{u}_i^T(\mathbf{v}_j + \boldsymbol{\theta}_{c_j})\Big]\mathbf{u}_i + \lambda_v \mathbf{v}_j \\ &- \alpha\Big[1 - \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{text}\mathbf{h}_j^{text}\Big)\Big]\mathbf{M}^{text}\mathbf{h}_j^{text} \\ &- \alpha\Big[1 - \sigma\Big(\mathbf{v}_j^\top \mathbf{M}^{image}\mathbf{h}_j^{image}\Big)\Big]\mathbf{M}^{image}\mathbf{h}_j^{image}\Bigg\}\end{aligned} \tag{19}$$

3.7.3 Update $\boldsymbol{\theta}_{c_j}$

The partial derivative of $\mathcal{L}$ w.r.t category representation corresponding to v_j is given as

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_{c_j}} = \sum_{(u_i,v_j)} \Bigg\{ -\Big[\mathbf{R}_{ij} - \mathbf{u}_i^T(\mathbf{v}_j + \boldsymbol{\theta}_{c_j})\Big]\mathbf{u}_i - \lambda_c(\boldsymbol{\nu}_{c_j} - \boldsymbol{\theta}_{c_j}) + \lambda_\theta \boldsymbol{\theta}_{c_j} \Bigg\} \tag{20}$$

3.7.4 Update $\mathbf{u}_i^k$

The partial derivative of $\mathcal{L}$ w.r.t user representation in network k $\mathbf{u}_i^k$ is given as

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{u}_i^k} = \sum_{(u_i,v_j)} &\Bigg\{ -\lambda_{U_1}\Bigg(\mathbf{u}_i - \sum_{k=1}^{K}\delta(\pi_{ik})\mathbf{u}_i^k\Bigg)\delta(\pi_{ik}) \\ &+ \lambda_T\Bigg(\mathbf{u}_i^k - \sum_{u_n^k \in N_i^k} T_{in}^k \mathbf{u}_n^k\Bigg) + \lambda_{U_2}\mathbf{u}_i^k\Bigg\} \\ &- \lambda_T \sum_{(u_w,v_j)|u_i^k \in N_w^k}\Bigg(\mathbf{u}_w^k - \sum_{u_v^k \in N_w^k} T_{wv}^k \mathbf{u}_v^k\Bigg)T_{wi}\end{aligned} \tag{21}$$

where $(\mathbf{u}_w^k - \sum_{u_v^k \in N_w^k} T_{wv}^k \mathbf{u}_v^k)T_{wi}$ is the partial derivative of the loss function w.r.t $\mathbf{u}_i^k$ when u_i acts as social neighbour of other users (e.g. u_w) in network k.

3.7.5 Update Other Parameters

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{M}^s} &= \sum_{(u_i,v_j)} \Big\{ -\alpha\Big[1 - \sigma(\mathbf{v}_j^\top \mathbf{M}^s \mathbf{h}_j^s)\Big]\mathbf{v}_j(\mathbf{h}_j^s)^T \Big\} \\ \frac{\partial \mathcal{L}}{\partial \Theta^s} &= \sum_{(u_i,v_j)} \Bigg\{ -\alpha\Big[1 - \sigma(\mathbf{v}_j^\top \mathbf{M}^s \mathbf{h}_j^s)\Big]\sum_k (\mathbf{M}_{:,k}^s)^T \mathbf{u}_i \frac{\partial \mathbf{h}_{j,k}^s}{\partial \Theta^s} \Bigg\} \\ \frac{\partial \mathcal{L}}{\partial \pi_{ik}} &= \sum_{(u_i,v_j)} \Bigg\{ \lambda_{U_1}\delta(\pi_{ik})(1 - \delta(\pi_{ik}))\Bigg(\mathbf{u}_i - \sum_k \delta(\pi_{ik})\mathbf{u}_i^k\Bigg)^T \mathbf{u}_i^k \Bigg\} \\ s &\in \{text, image\}\end{aligned} \tag{22}$$

where Θ^{text} and Θ^{image} are the parameters in the CNNs for extracting textual and visual features, respectively. $\mathbf{h}_{j,k}^s$ is the kth element of $\mathbf{h}_j^s$, and $\mathbf{M}_{:,k}^s$ is the kth column of matrix $\mathbf{M}^s$.

Algorithm 1 Training algorithm of the proposed model.

Require:
Parameters learning steps, T_1;
Item clustering steps, T_2;
Batch size:B; Learning rate:lr; Training set:$\{(u_i, v_j)\}$
repeat
  for $i = 0; i < T_1; i++$ **do**
    Sample a batch of training data of size B
    Take a gradient step for all the parameters:
    $\mathbf{u}_i = \mathbf{u}_i - lr \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{u}_i}$
    $\mathbf{v}_j = \mathbf{v}_j - lr \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{v}_j}$
    $\boldsymbol{\theta}_{c_j} = \boldsymbol{\theta}_{c_j} - lr \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_{c_j}}$
    $\mathbf{u}_i^k = \mathbf{u}_i^k - lr \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{u}_i^k}$
    $\mathbf{M}^s = \mathbf{M}^s - lr \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{M}^s}$
    $\Theta^s = \Theta^s - lr \cdot \frac{\partial \mathcal{L}}{\partial \Theta^s}$
    $\pi_{ik} = \pi_{ik} - lr \cdot \frac{\partial \mathcal{L}}{\partial \pi_{ik}}$
    $\Omega = \Omega - lr \cdot \frac{\partial \mathcal{L}}{\partial \Omega}$,
    $\Omega = \{\mathbf{u}_i, \mathbf{v}_j, \boldsymbol{\theta}_{c_j}, \mathbf{u}_i^k, \mathbf{M}^s, \Theta^s, \pi_{ik}\}$
  end for
  for $i = 0; i < T_2; i++$ **do**
    Item clustering:
    $c_j = argmin_k ||\mathbf{v}_j - \boldsymbol{\nu}_k||^2$
    $\boldsymbol{\nu}_k = \frac{\sum_{c_j = k} \mathbf{v}_j}{\sum_j I(c_j = k)}$
  end for
until *loss* converges or is sufficiently small

4 Evaluation

In this section, we first describe the dataset followed by the experiment setup. After that, we compare the proposed model with the state-of-the-art methods and then study the structure of the model. Finally, we study the parameters of the model.

4.1 Dataset

As the existing datasets (video datasets) either do not contain the social information or do not fit for micro-video scenarios, to validate the effectiveness of the proposed model, a real dataset is collected for our experiments. First, we used Twitter Streaming APIs[3] to collect a year's data from 2015 to 2016. Inspired by the existing works such as [15, 16], we filter out the users with fewer than five review records and the micro-videos with fewer than five viewers. Finally, the processed dataset includes 9412 users, 19058 micro-videos and 109433 interactions. On average, each user has 11.6 records and each micro-video has 5.7 viewers.

4.2 Setup

We set the hyper-parameters of the proposed model to the following default values: for the regularization terms in the loss function, the λs are the hyper-parameters and we set the default values to 0.01; for the texts associated with the items, we initialize the embedding matrix with Glove and tune it during the training process; for parameters in CNNs processing the texts, we set the number of filters to be 100 for each of the filter sizes in the range of 2 to 4; for the training configuration, we set the initial learning rate to 0.001 and decay it by 0.95 for every 1000 steps. The batch size is set to 1024, and the model is trained for a maximum of 1000 epochs. For each user, we randomly sample five missing interactions as negative samples for each positive sample. In addition, for each user 70% of the respective positive and negative samples are used for training. Beside the following social network, another social network is created by connecting users if they like the same micro-videos. In addition to the aforementioned default values, we also evaluate the proposed model with different network structures.

As for the validation process, we adopt the widely used leave-one-out scheme. For each ground-truth item for a user, we mix it with 100 random items, then rank the ground truth together with the 100 items and measure the recommendation performance of the proposed model. Finally, we apply three commonly used metrics, including precision@k, recall@k and nDCG@k, to evaluate our model from different point of views.

4.3 Baselines

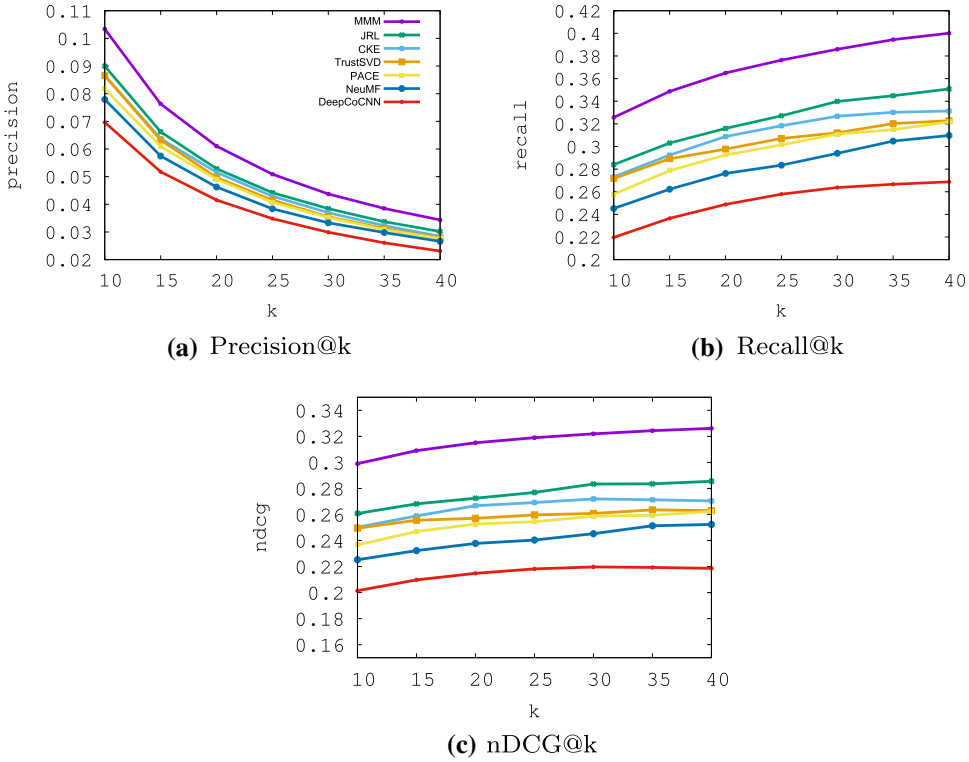
We compare the proposed model with the following three state-of-the-art baselines.

- NeuMF [9]: It generalizes matrix factorization with an end-to-end neural network. It embeds users and items into a latent space and employs a multi-layer neural network to model their interactions. The embedding of users and items is learned jointly with the objective function in the model.
- PACE [37]: In addition to modelling of user–item interactions, it introduces contexts to bridge the semantic gap between users and items. By predicting contexts with user–item embedding, users with similar contexts are forced to be similar in the latent space.
- TrustSVD [8]: It learns user and item representations from user–item interaction data. In addition, adjacency matrix of trust network is leveraged to factorize truster and trustee representations. User and truster representations are limited to share the same latent feature vector in order to bridge them together.
- DeepCoNN [41]: Instead of embedding users and items, it uses user-generated and item-generated texts to represent users and items and applies a deep neural network to model their interactions.
- CKE [38]: It applies deep learning methods (stacked de-noising auto-encoders and stacked convolution auto-encoders) to joint learn item textual and visual representations for recommendation.
- JRL [40]: It learns user and item representations in each individual information source (e.g. review text, image, numerical rating). Representations from different sources are integrated to obtain the joint representations for users and items.
- MMM: The proposed model incorporates multiple data sources and multiple user networks for micro-video recommendation with item category discovery.

We select these baselines as they constitute the state-of-the-art methods that consider information of different sources. For example, NeuMF models latent user and item representations from rating information with deep neural network, and DeepCoNN models the interaction between user–item pairs with the similarity measurement between their respective textual features. PACE and TrustSVD both consider user interaction and social information, and the difference is the way that they exploit social information for regularizing user representations. CKE and JRL incorporate data source of multiple modalities, and the different is that CKE extracts item features in each data source (e.g. text, image) and then generates unified item representations for modelling user–item interaction data, while JRL learns user and item representations in each data space and then combines them for recommendation.

We select these baselines as they constitute the start-of-the-art recommendation models that consider item attributes and regularization. Moreover, different structures of the

[3] Twitter Streaming APIs: https://dev.twitter.com/streaming/overview.



(a) Precision@k **(b)** Recall@k

(c) nDCG@k

Fig. 2 Performance comparisons

proposed models can be regarded as the previously proposed model (e.g. single social network regularization), and they are evaluated later in this section. For fair comparison, our model only considers following social network and textual information of items in this subsection, as *PACE* involves friendship information, while *DeepCoNN* has textual representations for users and items. For the baseline *NeuMF*, we fine-tune the parameters (e.g. learning rates and batch normalization) to achieve the best result, while for others we set the parameters to the values suggested in the original works.

4.4 Performance Comparison

The performance comparisons among those models on different metrics are presented in Fig. 2. From the figures, we have the following observations.

First, we find that JRL performs better than CKE across different metrics, which has been demonstrated by previous work [40]. This is because JRL models user–item representations and adds up their interaction scores from each of the information sources to rerank the top-N recommendations. Therefore, items can be ranked higher in the final recommendation lists as long as the user preferences over the items are properly profiled in any one of the information sources. On the contrary, CKE linearly combines the item representations over the information sources and assumes the unified item representations to be drawn from a Gaussian distribution parameterized by the linear combinations. As a result, evidence from different sources can negatively affect each other for reranking recommendation lists.

Second, the reason that JRL and CKE outperform DeepCoCNN with large margin is that the DeepCoCNN only considers the textual information for modelling the user–item interactions, while JRL and CKE include multiple data sources (e.g. interactions, texts and images) to learn user–item representations for the recommendation.

Third, PACE performs slightly better than NeuMF across different metrics, which demonstrates the benefit of incorporating social context for propagating user preferences. However, its improvement over NeuMF is marginal, and the reason is twofold. The original PACE integrates geographic contexts for regularizing items, but they are not available in our dataset, and this component is ignored when implementing PACE. Moreover, we fine-tune the parameters for NeuMF as the original model faces the overfitting problem in our dataset due to the data spareseness. To do this, we add l2-norm for all the parameters and apply batch normalization on the output of fully connected layers, and we finally grid search the learning rate to achieve the best result.

Also, TrustSVD outperforms PACE with large margin, especially on recall and nDCG. PACE leverages social contexts to regularize user latent vectors. The user latent vectors are used to predict their social contexts and the losses are back-propagated to update the vectors so that users who share the similar social friends are forced to be in a proximity that is close to each other. However, the regularization

of users is detached from the objective function; namely, the user latent features are not updated in a way towards the optimization of the recommendation performance. On the contrary, TrustSVD jointly optimizes the objective function and regularizes user latent vectors together. More importantly, TrustSVD explicitly regularizes users with matrix factorization, while PACE implicitly propagates user preferences based on similar contexts, which may account for its insufficiency in learning real user preferences.

NeuMF uses deep neural networks to model user–item interactions. The advantage of the model is that the nonlinear transformations in the neural network are able to capture informative user–item semantics for the recommendation, and the informative user–item semantics cannot be modelled with simple dot product in traditional matrix factorization. Moreover, due to the limited information sources, NeuMF still faces the problem of data spareseness, and the other models such as CKE and JRL significantly outperform NeuMF, as they are able to incorporate rich information from both users and items.

Unexpectedly, DeepCoCNN performs the worst of all baselines. The model exploits user-generated and item-generated texts for modelling their respective latent features. The basic idea is that user-generated texts reflect users preference and texts associated with items indicate their characteristics. However, the features learned from texts are basically biased to the textual semantic and cannot be generalized to reflect user latent vectors. One possible solution to this problem is to add user and item embedding along with the textual features, and this result can be regarded as a variant of JRL.

Furthermore, we can conclude that JRL and CKE perform better than TrustSVD and PACE. However, these two types of recommenders are not comparable, as JRL and CKE leverage content information for modelling item representations, while TrustSVD and PACE exploit user network for propagating user preferences. For datasets where the social information is discriminative, TrustSVD and PACE may outperform JRL and CKE. And for datasets where content data are informative, JRL and CKE may be superior.

Finally, the proposed model MMM outperforms the state-of-the-art baselines with a large margin. The advantage of the proposed over JRL and CKE demonstrates the effectiveness of incorporating user networks for recommendations. The information from multiple user graphs can be exploited for better user profiling and network local structure preservation. The superiority of MMM over PACE and TrustSVD shows the benefit of extracting content information from item side for better item representations learning, as the data of different modalities associated with items can bridge similar items in the shared latent space. However, it may be unfair to compare with those baselines since we are able to integrate different data modalities for joint modelling user–item interactions, even though this is one of the major contributions of this paper. For a fair comparison, we examine the effectiveness of the proposed model with each modelling component (e.g. content, category and user network) available, when compared with the baselines.

Overall, the comparison experiments demonstrate the effectiveness of incorporating information of multiple sources in an end-to-end neural network for micro-video recommendation, and the importance of regularizing user latent features for ranking optimization. The social connections and the item textual information can help to bridge semantic gap among users and items by propagating user preferences and item characteristics.

4.5 Structure Study

In this subsection, we study different variants of the proposed model to investigate the effect of each modelling component (e.g. content, category and user networks). The variants of our model are listed as follows:

- MMM-con: This variant only incorporates item content information and user–item interaction data for modelling user–item representations and recommendation.
- MMM-cat: This variant models user–item interactions and discovers item category iteratively, and it then incorporates category representations for predicting the rating scores.
- MMM-net: This variant only integrates user–item interaction and user networks for regularizing user representations and recommendation simultaneously.

We compare MMM-con with JRL, CKE and DeepCoCNN since all of them explore item content information for the recommendation. MMM-cat is compared with NeuMF, as both of them only consider rating matrix for user–item representations learning and recommendation, the difference is that MMM-cat dynamically discovers item category with clustering technique and incorporate the hierarchy of item- and category-level information for modelling user preference over the item at different granularities. Finally, MMM-net is compared with TrustSVD and PACE, because all of them take advantage of the user network for preserving the local structures in the network when learning user representations.

According to the comparison results presented in Tables 1, 2 and 3, we have the following observations. First, for models that leverage content information for recommendations, MMM-con, JRL and CKE achieve better performance than DeepCoCNN, this is because they exploit both textual and visual features for recommendations, while DeepCoCNN only considers textual information. The reason of performance variance among MMM-con, JRL and CKE is the way they employ to process the content data. We adopt convolution and

Table 1 Precision@k of different comparable models

Models	pre@10	pre@20	pre@30	pre@40
DeepCoCNN	0.06964	0.04154	0.02989	0.02309
CKE	0.08656	0.05156	0.03697	0.02843
JRL	0.08999	0.05286	0.03843	0.03017
MMM-con	0.09076	0.05319	0.03827	0.03048
NeuMF	0.07791	0.04625	0.03332	0.02661
MMM-cat	0.0822	0.04848	0.03457	0.02712
PACE	0.08183	0.04903	0.03523	0.02764
TrustSVD	0.08654	0.0497	0.03551	0.02781
MMM-net	0.09008	0.05347	0.0379	0.02962
MMM	0.1034	0.06102	0.04374	0.03436

Table 2 Recall@k of different comparable models

Models	rec@10	rec@20	rec@30	rec@40
DeepCoCNN	0.21963	0.24878	0.26371	0.26873
CKE	0.27311	0.30879	0.32677	0.33137
JRL	0.28397	0.31594	0.33985	0.35082
MMM-con	0.28724	0.31861	0.33825	0.35456
NeuMF	0.24517	0.27618	0.29394	0.30975
MMM-cat	0.25879	0.2903	0.30585	0.31537
PACE	0.25785	0.29252	0.31101	0.32182
TrustSVD	0.27174	0.29767	0.31212	0.32283
MMM-net	0.28384	0.31969	0.33382	0.34477
MMM	0.3257	0.365	0.38595	0.40015

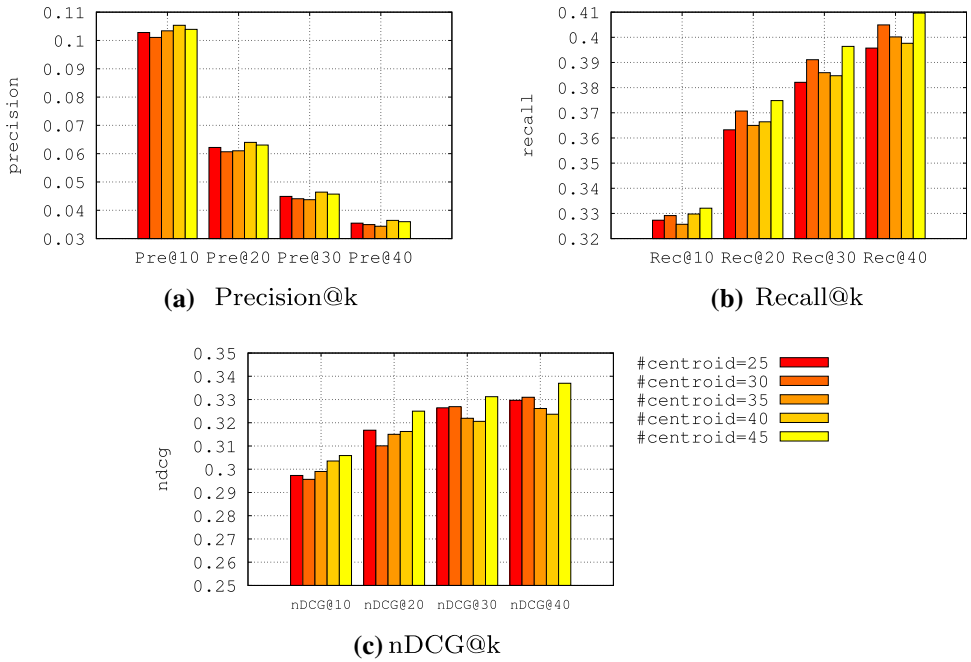
Table 3 nDCG@k of different comparable models

Models	ndcg@10	ndcg@20	ndcg@30	ndcg@40
DeepCoCNN	0.20145	0.2148	0.21967	0.21862
CKE	0.25021	0.26671	0.27194	0.27043
JRL	0.26079	0.27245	0.28339	0.28546
MMM-con	0.26327	0.27566	0.28258	0.28838
NeuMF	0.22526	0.23775	0.24526	0.25235
MMM-cat	0.23736	0.25024	0.25484	0.25772
PACE	0.23659	0.25268	0.25865	0.2623
TrustSVD	0.24948	0.257	0.26079	0.26296
MMM-net	0.26095	0.27566	0.27881	0.28082
MMM	0.29904	0.31502	0.32192	0.32613

max-pooling mechanisms for processing texts and images, and they are proved to be able to capture the most informative features for information retrieval tasks [38]. CKE utilizes stacked de-noising auto-encoders for extracting textual and visual features, and JRL employs word embedding and convolution techniques to process texts and images. Even though the performance improvement of MMM-con is not noticeable, the proposed model MMM is able to outperform JRL and CKE significantly by incorporating user network and item category information. Also, by comparing MMM-cat with NeuMF, we can observe the benefit of discovering latent item categories and incorporating them for modelling hierarchical user preferences. MMM-cat and NeuMF are comparable since both of them leverage user interaction data for learning user–item representations and recommendation, except that MMM-cat draws category representations from the centroids of clustered item representations and incorporates them to model user–item interactions. The hierarchy of item- and the category-level information is able to capture user interests at different levels of granularity and improve recommendation performance. Finally, the advantage of MMM-net over TrustSVD and PACE demonstrates the effectiveness of incorporating multiple user networks for propagating user preferences, since a single-user network may contain conflicting evidence against the real factors underlying user–item interaction, while information from multiple user networks can be complementary to each other for better regularizing user representations. Comparing different variants of the proposed models, we can find that MMM-cat is inferior to MMM-con and MMM-net; this is because the only information available for MMM-cat is the rating matrix. Therefore, MMM-cat still faces the problem of data sparseness.

4.6 Parameter Study

In this paper, the number of centroids needs to be pre-specified for clustering items; thus, we study the effect of the centroid number on the recommendation performance in this subsection. Few item clusters can make the category representations not discriminative enough to model hierarchical user preferences, while many centroids can make the category representations less informative for bridging items having similar characteristics, considering the special case where each item is regarded as a cluster. We present in Fig. 3 the recommendation performance across different metrics by varying the number of cluster centroids. We can see that the number of pre-specified cluster centroids has little impact on our recommendation model, as we are able to achieve similar recommendation performance when we change the centroids number. This is because with the different pre-specified number of centroids, we can cluster the items into different levels of hierarchies. Therefore, as long as the categorical structure of the items data is preserved, the item categorical representations can be informative and discriminative for distinguishing items with different characteristics. The underlying reason that the discovered category representations can boost accurate recommendation performance is twofold. From the items' point of view, the discovered categorical information encourages items with similar characteristics to have a unified category representation, and the representation is able to bridge the semantic gap among items in the category. From



(a) Precision@k **(b)** Recall@k

(c) nDCG@k

Fig. 3 Recommendation performance as a function of the cluster centroids

the users' perspective, by interacting users with both items and categories, we are able to capture user preferences at different levels of granularity, and an item tends to be ranked higher in the recommendation list as long as the user preference on the item is properly profiled either on the item level or on the category level. This experiment demonstrates that it is flexible to specify the item category number for achieving competitive recommendation accuracy.

We visualize the item categories with t-SNE [20] in Fig. 4, where different item categories are indicated with different colours. We can observe the hidden categorical structure among the items; namely, intra-cluster items are close to each other, while inter-cluster items are far away from each other in the projected latent space.

5 Conclusion

By leveraging the multi-modality information sources in micro-videos and the multinomial networks among users, we propose to incorporate the latent representations of the multiple sources into a unified model to facilitate recommendation, and employ multiple user networks for propagating user representations in a shared latent space. The multiple information sources act as bridges to interrelate items with similar content, while the user networks regularize users having strong social ties to have similar preferences. In addition, we propose to discover hidden categorical representations of micro-videos and interact them with user representations for boosting recommendation. The hidden categorical information can help to bridge items in a cluster and capture user preference at different levels of granularity. The modelling of hidden category and the user–item representations learning are iteratively performed in a unified model, and the recommendation losses are back-propagated to update the parameters in a way that can best optimize the recommendation performance. Finally, we validate the proposed model on a real dataset and study different variants of the proposed model, which demonstrates its advantage over the state-of-the-art baselines.

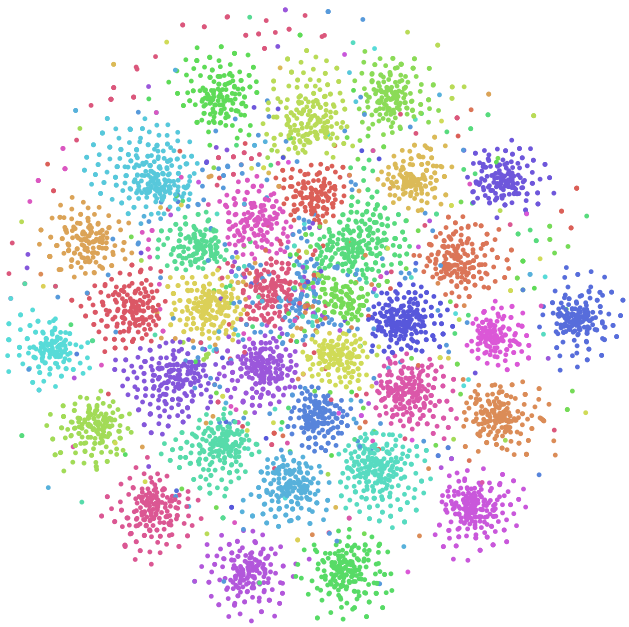


Fig. 4 Visualization of item clustering

References

1. Ai Q, Zhang Y, Bi K, Chen X, Croft WB (2017) Learning a hierarchical embedding model for personalized product search. In: SIGIR
2. Bengio Y et al (2009) Learning deep architectures for AI. Foundations and trends® in Machine Learning, Hanover
3. Cao C, Ge H, Lu H, Hu X, Caverlee J (2017) What are you known for?: Learning user topical profiles with implicit and explicit footprints. In: SIGIR
4. Chang S, Han W, Tang J, Qi GJ, Aggarwal CC, Huang TS (2015) Heterogeneous network embedding via deep architectures. In: KDD
5. Chen J, Zhang H, He X, Nie L, Liu W, Chua TS (2017) Attentive collaborative filtering: multimedia recommendation with item-and component-level attention. In: SIGIR
6. Chen X, Qin Z, Zhang Y, Xu T (2016) Learning to rank features for recommendation over multiple categories. In: SIGIR
7. Chen X, Zhang Y, Ai Q, Xu H, Yan J, Qin Z (2017) Personalized key frame recommendation. In: SIGIR
8. Guo G, Zhang J, Yorke-Smith N (2015) TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI
9. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: WWW
10. Huang H, Zhang Q, Wu J, Huang X (2017) Predicting which topics you will join in the future on social media. In: SIGIR
11. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Recsys
12. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 426–434
13. Li P, Wang Z, Ren Z, Bing L, Lam W (2017) Neural rating regression with abstractive tips generation for recommendation. In: SIGIR
14. Li S, Kawale J, Fu Y (2015) Deep collaborative filtering via marginalized denoising auto-encoder. In: CIKM
15. Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: KDD
16. Liang D, Charlin L, McInerney J, Blei DM (2016) Modeling user exposure in recommendation. In: WWW
17. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the fourth ACM international conference on Web search and data mining. ACM, pp 287–296
18. Ma J, Li G, Zhong M, Zhao X, Zhu L, Li X (2018) LGA: latent genre aware micro-video recommendation on social media. Multimed Tools Appl 77(3):2991–3008
19. Ma J, Wen J, Zhong M, Chen W, Zhou X, Indulska J (2019) Multi-source multi-net micro-video recommendation with hidden item category discovery. In: Database systems for advanced applications - 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, 22–25 April 2019, proceedings, part II, pp. 384–400. https://doi.org/10.1007/978-3-030-18579-4_23
20. Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579
21. Manotumruksa J, Macdonald C, Ounis I (2017) A deep recurrent collaborative filtering framework for venue recommendation. In: CIKM
22. Mei T, Yang B, Hua XS, Yang L, Yang SQ, Li S (2007) Videoreach: an online video recommendation system. In: SIGIR
23. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: NIPS
24. Park J, Lee SJ, Lee SJ, Kim K, Chung BS, Lee YK (2010) An online video recommendation framework using view based tag cloud aggregation. In: MM
25. Pazzani MJ, Billsus D (2007) Content-based recommendation systems. In: The adaptive web. Springer, Berlin, pp 325–341
26. Qu M, Tang J, Shang J, Ren X, Zhang M, Han J (2017) An attention-based collaboration framework for multi-view network representation learning. arXiv preprint arXiv:1709.06636
27. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. ACM, pp 285–295
28. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
29. Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1235–1244
30. Wang S, Wang Y, Tang J, Shu K, Ranganath S, Liu H (2017) What your images reveal: exploiting visual contents for point-of-interest recommendation. In: WWW
31. Wang X, He X, Nie L, Chua TS (2017) Item silk road: recommending items from information domains to social users. arXiv preprint arXiv:1706.03205
32. Wu Y, DuBois C, Zheng AX, Ester M (2016) Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM
33. Xu L, Wei X, Cao J, Yu PS (2017) Embedding of embedding (eoe): joint embedding for coupled heterogeneous networks. In: WSDM
34. Xu Y, Lam W, Lin T (2014) Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management. ACM, pp 251–260
35. Yan R, Zhao D et al (2017) Joint learning of response ranking and next utterance suggestion in human–computer conversation system. In: SIGIR
36. Yang B, Lei Y, Liu J, Li W (2017) Social collaborative filtering by trust. IEEE Trans Pattern Anal Mach Intell 39(8):1633–1647
37. Yang C, Bai L, Zhang C, Yuan Q, Han J (2017) Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: KDD. ACM
38. Zhang F, Yuan NJ, Lian D, Xie X, Ma WY (2016) Collaborative knowledge base embedding for recommender systems. In: KDD
39. Zhang J, Nie L, Wang X, He X, Huang X, Chua TS (2016) Shorter-is-better: venue category estimation from micro-video. In: Proceedings of the 2016 ACM on multimedia conference. ACM, pp 1415–1424
40. Zhang Y, Ai Q, Chen X, Croft W (2017) Joint representation learning for top-n recommendation with heterogeneous information sources. In: CIKM
41. Zheng L, Noroozi V, Yu PS (2017) Joint deep modeling of users and items using reviews for recommendation. In: WSDM