Multimed Tools Appl (2018) 77:2991–3008
DOI 10.1007/s11042-017-4827-2

LGA: latent genre aware micro-video recommendation on social media

Jingwei Ma[1] · Guang Li[2] · Mingyang Zhong[1] · Xin Zhao[1] · Lei Zhu[1] · Xue Li[1]

Received: 15 March 2017 / Revised: 2 May 2017 / Accepted: 10 May 2017 /
Published online: 23 May 2017

Abstract Social media has evolved into one of the most important channels to share micro-videos nowadays. The sheer volume of micro-videos available in social networks often undermines users' capability to choose the micro-videos that best fit their interests. Recommendation appear as a natural solution to this problem. However, existing video recommendation methods only consider the users' historical preferences on videos, without exploring any video contents. In this paper, we develop a novel latent genre aware micro-video recommendation model to solve the problem. First, we extract user-item interaction features, and auxiliary features describing both contextual and visual contents of micro-videos. Second, these features are fed into the neural recommendation model that simultaneously learns the latent genres of micro-videos and the optimal recommendation scores. Experiments on real-world dataset demonstrate the effectiveness and the efficiency of our proposed method compared with several state-of-the-art approaches.

Keywords Micro-video recommendation · Genre aware · Neural network

1 Introduction

Over the past few decades, the popularity of online video sharing platforms has surged to an unprecedented scale [4, 6, 7]. Since 2012, a new type of Internet media has received close attention in the entertainment area: micro-videos (or bite-sized video) generated by a new form of users with the length from 6 seconds to 300 seconds approximately. The micro-videos can be shared, commented and reposted by users when they are published.

✉ Mingyang Zhong
m.zhong1@uq.edu.au

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

[2] School of Computer Science and Technology, Tianjin University, Tianjin, China

Vine,[1] for instance, has approximately 200 million active users monthly and 1.5 billion videos in a daily loops at the end of December 2015,[2] while Snapchat hit 7 billion daily video views in 2016 [9]. The sheer volume of micro-videos available in social networks often undermines user capability to choose the micro-videos that best fit their interests.

Recommendation appears as a natural solution to this problem. However, existing video recommendation methods only consider the users' historical preferences on videos, without exploring any video contents. Compared with long videos, micro-videos are short but more focused on certain interesting topics. On the one hand, it will be technically possible to detect more specific genres of micro-videos by jointly analysing visual contents and contextual semantics. On the other hand, users may have more clear preferences on the specific micro-video genres. Hence, it is promising to capture the specific genres of micro-videos to facilitate recommendation. Nevertheless, the current micro-videos are only categorised into noisy coarse categories. For example, animals, art, comedy, edits, music&dance, sports, and weird. This freely available coarse-grain categorisation cannot reflect users' interests accurately for recommendation. Taking category, sports, as an example, people who like basketball may have no interest in football.

Motivated by the aforementioned considerations, in this paper, we propose a novel latent genre aware micro-video recommendation within deep neural network framework. Its basic structure is shown in Fig. 1. The key idea is simultaneously detecting auxiliary fine-grain latent genres of micro-videos that better match users' interests, and learning the optimal recommendation scores with user-item embedding. Our model mainly works as the following two steps: Firstly, we extract user-item features from the user and micro-video pairs. Specifically, we perform user-item embedding and represent users and micro-videos with latent feature vectors. In addition, we extract visual features with pre-trained convolution neural networks (CNNs) [8, 30] to describe visual contents of micro-videos, and textual features with bidirectional recurrent neural networks (BRNNs)[28] to represent contextual semantics. Secondly, these features are fed into the neural recommendation model that simultaneously learns the latent genres of micro-videos and the optimal recommendation scores. A loss function is defined based on the idea of learning to rank [2] with the objective that the similarity score of positive user micro-video pairs is expected to be larger than that of negative user micro-video pairs. In optimisation, these losses are propagated back to the neural network and the parameters are updated accordingly.

The main contribution of this paper can be summarised as follows:

– We propose a latent genre aware neural recommendation model for micro-videos. It improves over 5% of the $Accuracy@k$ compared with the state-of-the-art baselines. Our model specifically considers the auxiliary fine-grain latent genres of micro-videos to facilitate the recommendation. To the best of our knowledge, there is still no similar work.
– We build a large-scale micro-video dataset including 51,837 users and 147,378 micro-videos. Experiments on it demonstrate the superior performance of the proposed model compared with state-of-the-art approaches.

The rest of the paper is organised as follows. Section 2 describes the related works. In Section 3, we illustrate the proposed methodology, followed by Section 4 that details the experimental results and discussions. Finally, Section 5 concludes this paper.

[1] Vine: https://vine.co

[2] Vine report: http://blog.vine.co

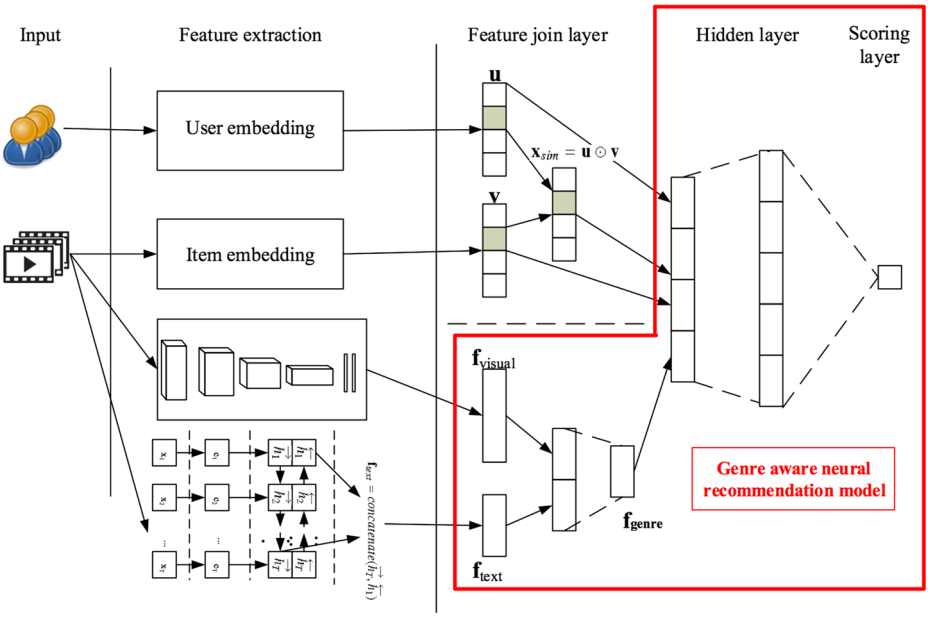


Fig. 1 The basic structure of the proposed approach

2 Related Work

In this section, we describe the literatures that are most related to this paper. Specifically, we review existing video recommendation and neural recommendation methods.

2.1 Video recommendation

Many existing video-oriented sites, such as YouTube,[3] MSN Video[4] and Yahoo! Video,[5] have already provided video recommendation services. Content-based filtering approaches [20, 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 multi-modal relevance and users' click-through data. It integrates textual, visual and aural modalities with an attention function. The main limitation of these methods is that they only consider the video-video relations, ignoring the related users' preferences that are important when recommending videos. Collaborative filtering [23, 34] is also explored in video recommendation. [34] compares users' rating of videos previously given by others, the videos are then recommended to users who share the similar preferences. In MovieLens system [23], the users are required to rate films that they have seen from 1 (Awful) to 5 (Must See) 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

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

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

[5] https://video.search.yahoo.com/

such as descriptions and keywords. Its performance may be undesirable when the ratings are insufficient.

2.2 Neural recommendation

There are many works using a deep neural network for recommendations. Wang et al. [34] proposed the collaborative deep learning (CDL) recommendation model to bridge the gap between the deep learning model and recommendation system. They demonstrated the advanced recommendation performance by jointly performing deep learning on the content information and collaborative filtering with the ratings matrix. The CDL model presents a good performance for text content recommendation but ignores to analyse other features, such as visual features, which are essential to micro-video recommendation.

He et al. [16] developed a neural recommendation framework based on the collaborative filtering. The authors first modelled the interaction of user-item features through collaborative filtering. Then, they applied the deep learning to learn the latent features of users and items instead of inner product of them. The experiment results show that promising performance can be achieved using deeper neural networks. However, the method only explores the interaction between user and item features, without considering any valuable auxiliary micro-video information.

In [37], the authors exploit recurrent neural networks (RNNs) [28] to learn the similarity between each query and advertisement. They proposed RNN module to model the word sequence of queries and advertisement in online advertising. In [15], Guo et al. leverage deep neural network for ad-hoc retrieval task. They formalise the retrieval task as a matching problem using deep models.

Our approach also advocates on neural recommendation. But, we are different from the aforementioned approaches on exploiting the contextual semantics and visual contents of micro-videos as auxiliary information to facilitate the recommendation. Moreover, our proposed model can simultaneously detect the fine-grain latent genres of micro-videos that may possibly match the users' interests, and learn the optimal ranking scores for final recommendation. To the best of our knowledge, there is still no similar work.

3 The proposed method

In this section, we first give the problem definition. Then, we describe the methods of extracting the features of user-item and micro-videos. Finally, we present the latent genre aware neural recommendation model.

3.1 Problem formulation

Given a set of user $U = \{u = 1, 2, \cdots, N\}$, a set of micro-videos $I = \{1, 2, \cdots, M\}$, and a log of the users' preferences (retweeting, commenting and liking) on items $O = (u, i, y_{ui})$. y_{ui} can be represented by a binary value [0,1], with 1 indicating the preferred item of user and 0 otherwise. We aim to recommend each user a list of micro-videos that maximise user's satisfaction by recommending micro-videos that match user's interests and are more likely to be retweeted or commented. For each user u, we use O_u to denote user's preferred items in the training set, and $\overline{O_u}$ denotes user's unobserved preferences.

In this paper, we formulate the micro-video recommendation as a pair-wise learning to ranking problem. For a user u, items in O_u are assumed positive samples and unobserved

items in $\overline{O_u}$ are considered as the negative samples. The basic idea is to learn a ranking function h, such that the positive samples are ranked higher that the negative samples.

$$h(\mathbf{w}, \psi(u, i)) > h(\mathbf{w}, \psi(u, j)) + \varepsilon \tag{1}$$

where $i \in O_u$ is the positive item and $j \in \overline{O_u}$ is the negative item, while $\mathbf{w}$ is a vector of model weights and $\psi(\cdot)$ is the function that maps the user-item pair into a feature vector. Later in this section, we describe how to extract multinomial features from the user-item pairs and finally describe the ranking model in Section 3.3.

3.2 Feature extraction

We extract two kinds of features of micro-videos as the input of subsequent neural recommend model. One of them describes the interaction between users and micro-videos and the other characterises the micro-videos. Specifically, we extract the user features, item features and user-item features to represent users, items and their interactions. To exploit micro-videos, textual and visual features are extracted from visual contents and contextual texts.

3.2.1 User-item feature extraction

The first kind of features is extracted from social media platforms such as Twitter representing the characteristics of users, items and their interactions. We use word embedding model [14] to learn user and item embeddings ($\mathbf{u}$ and $\mathbf{v}$). Specifically, we use the user's historical tweets and the tweet associated with the item (micro-video) to compute $\mathbf{u}$ and $\mathbf{v}$ respectively. Then using the low-level representation of a user $\mathbf{u}$ and a item $\mathbf{v}$, we define the similarity score between them as follows to capture their interactions that are such as similar topics and interests:

$$\mathbf{x}_{sim} = \mathbf{u} \odot \mathbf{v} \tag{2}$$

where $\odot$ is the element-wise product. Although many previous works [1, 29] use the matrix transform method to compute the similarity score between two vectors, we found that the element-wise product is more effective.

3.2.2 Textual feature extraction

After extracting user-item features, we model the textual and visual features of micro-videos. The extraction of textual features goes through the embedding step and the RNN step.

1) Embedding Step We tokenise a collection of all the texts associated with micro-videos and create a vocabulary. After that, each text is represented with a word sequence $\mathbf{x} = \{x_1, \cdots, x_T\}$, where x_t is the index of the t-th word, representing the ordering of the word in the vocabulary. We pad all the word sequences into the same length by padding 0 to the end of those word sequences that have the length shorter than the specified length. Each word in the sequence is mapped into a continuous vector by indexing the word embedding look-up table:

$$e_t = W_{emb}[x_t], s.t. W_{emb} \in R^{V \times d_{emb}} \tag{3}$$

where V is the size of the vocabulary, d_{emb} is the manually specified embedding size, and W_{emb} is the word embedding matrix where each row corresponds to a word in the vocabulary. The word embedding matrix transforms each word into a low-dimensional dense vector

e_t that comes to represent in some abstract way the "meaning" of a word. The embedding matrix is updated jointly when optimising the object function in the learning process.

2) RNN Step The RNN step transforms the sequences of word embedding vectors into latent vectors, it takes the output of word embedding step $\{e_1, \cdots, e_T\}$ as input. Each word embedding vector e_t is regarded as input in the corresponding t-th time step, and the latent state (i.e. latent vector) h_t is non-linear transformation of the word embedding vector at that step and the latent state of the previous step, h_{t-1}. For example, the simplest RNN updates the latent state with:

$$h_t = \sigma(W_{eh}e_t + W_{hh}h_{t-1} + b_h) \tag{4}$$

where $W_{eh} \in R^{d_h \times d_{emb}}$, $W_{hh} \in R^{d_h \times d_h}$, $b_h \in R^{d_h}$ are the weights and bias to be learned, and d_h is the dimension of the latent states. The σ is the non-linear transform (e.g. $relu(x) = max(0, x)$), and (4) can be abstracted as $h_t = H(h_{t-1}, e_t)$. The final state h_T at step T can be considered as the encoding of the whole word sequence. However, the update of the latent state only consider the previous state may lack the ability of representing long word sequence. Therefore, more sophisticated RNN such as LSTM [17] are proposed, it is able to catch long term dependency in sequences and updates the latent state with:

$$\begin{aligned} f_t &= \sigma(W_{ef}e_t + W_{hf}h_{t-1} + b_f) \\ i_t &= \sigma(W_{ei}e_t + W_{hi}h_{t-1} + b_i) \\ o_t &= \sigma(W_{eo}e_t + W_{ho}h_{t-1} + b_o) \\ g_t &= tanh(W_{ec}e_t + W_{hc}h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{5}$$

where $\odot$ represents element-wise product between vectors. A LSTM cell consists of an input gate i, a forget gate f, a memory cell c and an output gate o. The input gate decides whether to block the input signal, the forget gate decides whether to remember the previous cell state, and the output gate can allow new output or prevent it. By catching the long-term dependencies, LSTM is able to avoid quick vanishing and exploding problems that simple RNN suffers from back propagation optimisation [33].

In this paper, we use BRNN to encode the textual features. BRNN is implemented with two separate LSTM layers that work in opposite direction. The forward LSTM starts at time=1 that is the begin of a sequence, while the backward LSTM starts at time=T that is the end of a sequence. The latent state of BRNN can be represented with the concatenating of the latent state of the two LSTMs:

$$\begin{aligned} \overrightarrow{h_t} &= \overrightarrow{H}(e_t, h_{t-1}), \overleftarrow{h_t} = \overleftarrow{H}(e_t, h_{t+1}) \\ h_t &= concatenate(\overrightarrow{h_t}, \overleftarrow{h_t}) \end{aligned} \tag{6}$$

Therefore, the latent representation of a word sequence is $concatenate(\overrightarrow{h_T}, \overleftarrow{h_1})$. BRNN is able to leverage both the past and future context information of a word sequence, and result in a better representation of a sequence. The process of extracting textual features is shown in Fig. 2.

3.2.3 Visual feature extraction

There are variety of ways to represent the visual contents.[3, 5, 21]. In this paper, we exploit deep convolutional neural networks (CNNs) [8, 30] to extract high-level features from the

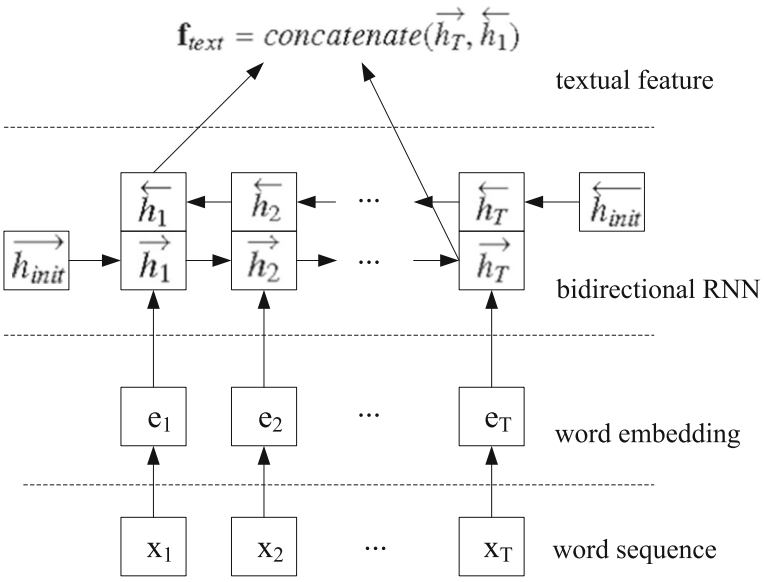


Fig. 2 An illustration of textual features extraction process

key frames of a micro-video. The CNNs extract hierarchies of features ranging from low-lever features (e.g. color blobs, lines) in the low-level layers, to high-level features (e.g. objects) in the high-level layers. The powerfulness of high-level features from the CNNs has been demonstrated in many applications such as image captioning [32, 33, 35], visual question answering [10], image style transfer [13].

Training the CNNs requires a large amount of training data, a potential solution to this problem is to transfer parameters from a CNN that has already been trained with large scale image database [11]. In this paper, we use the pre-trained 19-layer VGG net [8, 31] as a fixed high-level image features extractor. The VGG-19 model is trained on large scale images and classify an image into one of the 1000 classes. For each image as input into the VGG-19 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]. We believe that firing of the neurones immediately prior to the classification task that contains high-level knowledge is useful to the recommendation task. The process of extracting textual features is shown in Fig. 3.

3.3 Genre aware neural recommendation model

The previous section introduces the extraction of the user-item feature, the textual and visual feature. In this section, we detail the genre aware micro-video recommendation model. Our model is developed based on the deep neural network framework. It has six layers in total which include the latent genre layers (one input layer, one hidden layer and one output genre layer) and the fully concatenating neural network (one input layer, one hidden layer and one scoring layer). The main objective is to jointly learns the latent genres of micro-videos and the optimal recommendation scores. The basic idea of pair-wise learning to rank is to calculate a similarity score for each user-item pair, and ensure that the score of user-positive item should be higher than the user-negative item.

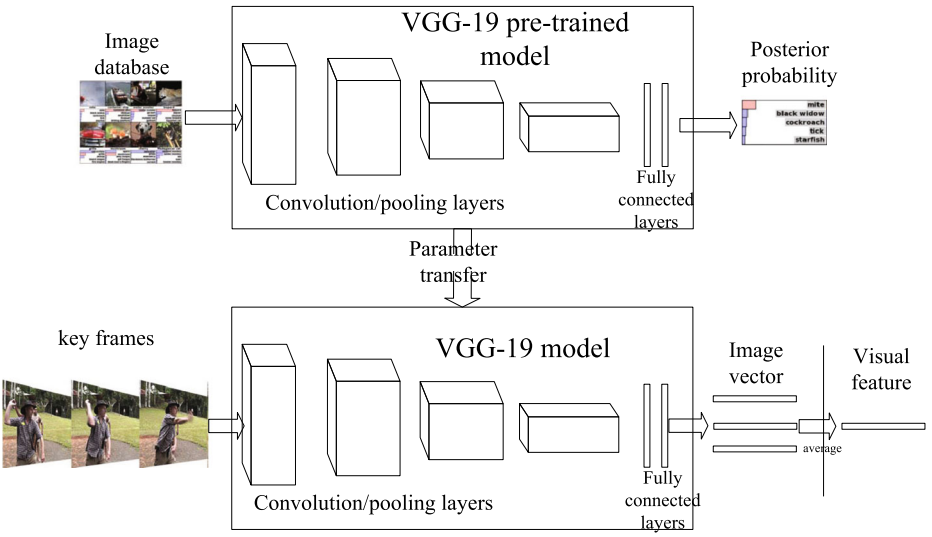


Fig. 3 An illustration of visual features extraction process

Note that, in traditional collaborative filtering methods such as [27], the similarity score is the inner product of the user and item vectors. However, to achieve desirable micro-video recommendation, we need consider extra auxiliary information of micro-videos. Therefore, the traditional collaborative filtering such as matrix factorisation is inapplicable in our scenario.

In the following subsections, we detail the layers of our latent genre aware neural recommendation model and the calculation of the similarity score for each user micro-video pair.

3.3.1 Latent genre aware layers

The features that are related to micro-videos are then fed into the latent genre aware layers. The rationale of concatenating the textual and visual features of micro-videos and adding hidden layers upon the concatenated features is to explicitly model the interactions among low-level features of the micro-videos and capture complex high-level latent features (such as genre or category of micro-videos). Furthermore, transforming low-level features into high-level latent features can significantly improve the training efficiency. Specifically, compared with using the textual feature (256-dimension) and visual feature (4096-dimension) as the input of a fully concatenating neural network that requires tuning over 4000 parameters, the dimension of the high-level latent features generated by the latent genre layers is significantly reduced. The parameters in these layers are gradually tuned to capture complex high-level features during the learning process for optimising the object function with annotated data. The output is formulated as follows:

$$\mathbf{x}_{genre} = \mu(\mathbf{w}_g \cdot concatenate(\mathbf{x}_{visual}, \mathbf{x}_{text}) + \mathbf{a}) \tag{7}$$

where $\mathbf{w}_g$ and $\mathbf{a}$ are the weights and biases of the hidden layers respectively, and μ is the activation function.

3.3.2 Full join layer

In this join layer, we join the multinomial features, including users and micro-videos with latent representations of latent genre, user-item pair and their similarity score. By considering these multinomial features, our model naturally alleviates the cold-start problem for the micro-video recommendation.

$$\mathbf{x}_{join} = concatenate(\mathbf{u}, \mathbf{v}, \mathbf{x}_{sim}, \mathbf{x}_{genre}) \tag{8}$$

3.3.3 Hidden layers

The joined feature is then fed into the fully connected hidden layers. Notice that the number of hidden layers is not limited, and is tuned by experiments.

$$\mathbf{x}_{hidden} = \sigma(\mathbf{w}_h \cdot \mathbf{x}_{join} + \mathbf{b}) \tag{9}$$

where $\mathbf{w}_h$ and $\mathbf{b}$ are the weights and biases of the hidden layer respectively, and σ is the activation function. The motivation of adding a hidden layer is that we need to further capture the interactions among the latent represent of different factors (e.g. user, item, latent genre features).

3.3.4 Scoring layer

In this layer, we compute the score for each user-item pair, representing the similarity score by considering the multinomial features.

$$score_{u,v} = \mathbf{w}_s \cdot \mathbf{x}_{hidden} \tag{10}$$

where $\mathbf{w}_s$ is the weight vector of this final layer. At this point, $\mathbf{x}_{hidden}$ can be regarded as the high-level representation of the user-item pair obtained through a series transformations from the input layers. The information flow of our model can be illustrated in Fig. 1. Notice that the score of an user-item pair represents the similarity of the pair, and the scores are used to define the loss function described in the later section. The losses based on the scores are propagated back to the neural network for fitting the recommendation model into the annotated data. Unlike traditional similarity measurement, we are able to incorporate different raw features into the neural network, and the parameters for extracting the latent features are tuned collaboratively during the learning process. In this light, our model is not limited by the visual and textual features. Instead, it is generic and can integrate different kinds of features.

3.3.5 Loss function

As described previously, we calculated a score for each item-user pair based on the vector representations of the users and items. After that, we define a proper loss function to conduct learning. As we can mine the implicit interactions between the users and items, we adopt the supervised pair-wise learning to ranking learning method, and the loss function is similar to previous works [19, 38] that maximise the similarity of the users and the micro-videos that the users have implicitly interacted with. Since the score in (10) measures the similarity between the user interests and item attributes, minimising the loss function reflects the fact that positive items have higher scores than the negative ones. Formally, let (u, i) be the positive user-item pair (i.e. the user has implicit feedback on the item), and (u, j) be the

negative user-item pair (i.e. unobserved item for this user). We formulate our loss function as follows:

$$J(\boldsymbol{\theta}) = -\sum_{u} \sum_{i \in O_u, j \in \overline{O}_u} log\sigma(score(u,i) - score(u,j)) \tag{11}$$

where $\sigma(x)$ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ that maps the difference between the score of the user-(positive item) and the score of the user-(negative item) into the range of [0,1]. $\boldsymbol{\theta}$ is the union of the parameters that needs to be learned from the training set, and they include the embedding matrix for the users and items, the parameters in the network for processing the latent genre features, and the parameters after the join layer. In this way, all the parameters defined previously are learned with the goal to correctly match related user-item pairs. In (11), negative items are randomly sampled from the unobserved items, which allow us to greatly reduce the overhead of computing the loss function. Therefore, we can directly adopt maximum log-likelihood estimation and minimise the loss function with standard gradient descent, and train our model on large a large training set. The σ function denotes the probability that positive items are ranked higher than negative items for a user, hence maximising the loss function is equivalent to increasing the similarity score of the user-positive item and decreasing that of the user-negative item.

4 Experiments

In this section, we first describe the settings of experiments and then demonstrate the experimental results.

4.1 Dataset collection and preprocessing

A real dataset is collected for our experiment. First, we used Twitter Streaming APIs[6] to collect a year's data from 2015 to 2016. Then, for alleviating data sparsity, we filtered out the users who posted less than 5 micro-videos. After that, we wrote a web crawler to collect all micro-videos in our testing dataset. The dataset is finally constructed with 51,837 users and 147,378 micro-videos.

4.2 Experimental setup

In order to evaluate the recommendation effectiveness, we compare our model with several state-of-the-art recommendation approaches. In addition, to provide more insights, three variants of our models are further designed and evaluated to investigate the effects of different modalities on micro-video recommendation performance. For evaluation, the dataset D is split into the training set D_{train} and the test set D_{test} according to the proportion of 70% and 30%, respectively. In the following, we detail the compared approaches and evaluation metric.

4.2.1 Evaluated baselines

– MP: A standard "most popular" baseline, which ranks micro-videos in descending order of posting times performed by Twitter users.

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

- IBCF: Item-based collaborative filtering is a classic recommendation method proposed in [27]. IBCF first analyse the user-item matrix to identify relationships between different items, and then uses it for recommendation.
- WRMF: Weighted regularised matrix factorisation [18] is a regularised least-square optimisation with weights to reduce the impact of negative examples.
- BPR: Bayesian personalised ranking model [26] imposes a pairwise ranking criterion, where the latent factors of users, items, and tags are jointly optimised for recomendation.
- PPR: Pairwise preference regression model [24] is a predictive feature-based regression model that leverages the available information of users and item content features for tackling the cold-start problem. The texts in PPR are determined as the contextual texts associated with the micro-videos.
- CDL: Collaborative deep learning [34] jointly performs deep representation learning for the content information and collaborative filtering for the ratings matrix. The authors have evaluated CDL on movie recommendation. Similar to PPR, the contextual texts in CDL are also determined as the texts associated with the micro-videos.

In addition three variants of our model are compared to further investigate the effects of different modalities. They are

- LGA-UI: In this approach, we remove the textual feature and the visual feature. The features of users, items and their interactions are used for recommendation. Formally, (8) is modified to:

$$\mathbf{x}_{join} = concatenate(\mathbf{u}, \mathbf{v}, \mathbf{x}_{sim}) \tag{12}$$

  It simulates the application scenario that targets recommending micro-videos to new users.
- LGA-Text: To simulate the application scenarios where users post text descriptions but without any micro-videos, we remove the visual feature. Hence, in recommendation, the features of users, items, user-item interactions, and visual features are used. Formally, the (8) is modified to:

$$\mathbf{x}_{join} = concatenate(\mathbf{u}, \mathbf{v}, \mathbf{x}_{sim}, \mathbf{x}_{text}) \tag{13}$$

- LGA-Visual: To simulate the application scenario where the users post many micro-videos but without text descriptions, we remove the textual features. The features of users, items, user-item interactions, and the visual feature are used for recommendation. Formally, the (8) is modified to:

$$\mathbf{x}_{join} = concatenate(\mathbf{u}, \mathbf{v}, \mathbf{x}_{sim}, \mathbf{x}_{visual}) \tag{14}$$

4.2.2 Evaluation metric

For evaluation metric, we adopt a commonly-used measurement $Accuracy@k$ as [12, 36]. It is formulated by averaging all test cases:

$$Accuracy@k = \frac{\#match@k}{|D_{test}|} \tag{15}$$

where $match@k$ is defined for a single test case as either 1, if the micro-video appears in the top-k results, or 0, otherwise. $\#match@k$ denotes the number of matches in the test set, and $|D_{test}|$ is the number of all test cases. In experiments, the presented $Accuracy@k$ are the average of ten results.

4.3 Experimental results

In this section, we present the experimental results of the compared approaches with well-tuned parameters. We only show the performance where k is set to 1, 3, 5, 7, 9. The reasons are that great values of k are usually ignored for recommending micro-video to social media users. Furthermore, the total number of micro-videos posted by a user in our dataset is limited, as the Twitter Streaming APIs do not provide users' complete timeline data but sampling data.

4.3.1 Comparison results

Figure 4 presents the recommendation accuracy of our model compared with the selected baselines. The $Accuracy@k$ of LGA is about 0.29 when $k = 3$ and 0.31 when $k = 5$ (i.e., the model has a probability of 29% of placing an appealing micro-video in the top-3 and 31% of placing it in the top-5). Obviously, our proposed LGA model outperforms other competitor models significantly (improved over 5% of the $Accuracy@k$ compared with the four baselines). Several observations are made from the results: i) CDL shows similar performance as LGA-Text plotted in Fig. 5 as they are both based on deep learning model and utilised similar features e.g. text feature. ii) All algorithms perform significantly better than MP because few micro-videos are widely posted by users in the real world.

4.3.2 Effects of feature modalities

In Fig. 5, we illustrate the effectiveness of our proposed model in terms of its constituent features. Firstly, the LGA model with all features being enabled outperforms all incomplete feature combinations, as it can explicitly model the interactions among the low-level features of the micro-videos and capture the complex high-level latent features, such as genre. Secondly, LGA-UI slightly outperforms UI-CF in Fig. 4 since they use similar user-item features. By adding text and visual features individually, LGA-Text and LGA-Visual both achieve better results compared with LGA-UI. Furthermore, LGA-Text shows better results than LGA-Visual, probably due to the similarity of keyframes extracted from micro-videos. In our current implementation, we extract one key frame from each micro-video (the first frame), and we will continue to investigate the visual feature in the future.

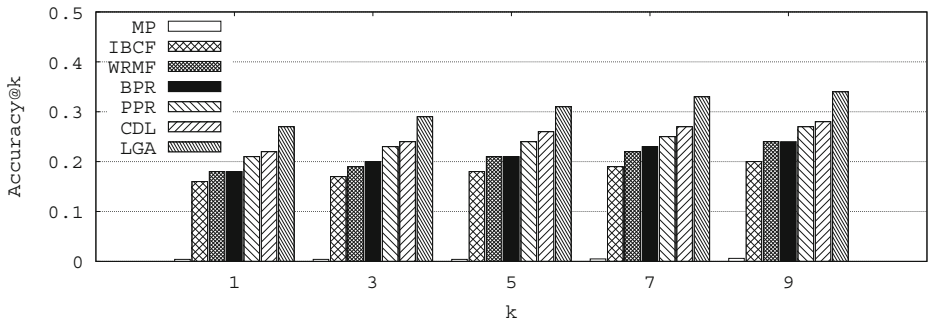


Fig. 4 Accuracy@k: comparison of different algorithms

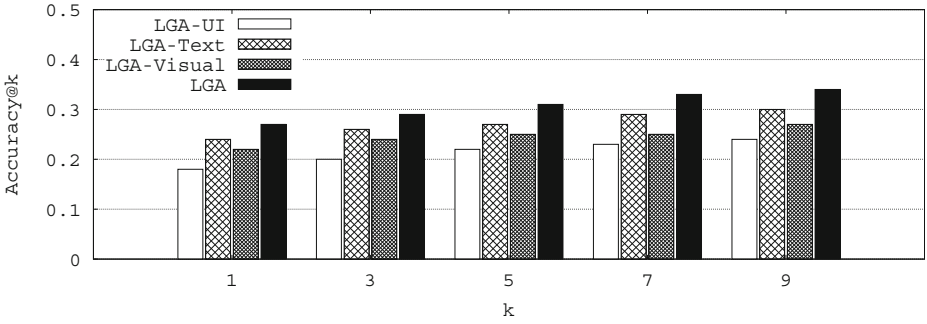


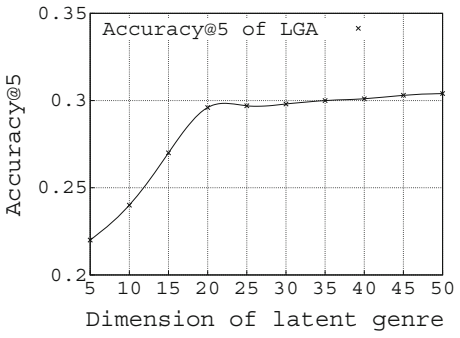
Fig. 5 Accuracy@k: representativeness of different modalities

4.4 Impact of latent genre

Tuning model parameters, such as the dimension of the high-level latent feature of micro-videos described in Section 3.3.1, is critical to the performance of LGA model. Therefore, we study the impact of the dimension of latent genre on our dataset in this section. In this set of experiments, we test $Accuracy@5$ by varying the dimension of the latent genre from 5 to 50 with an interval 5. As shown in Fig. 6, we can see that with the dimension increasing $Accuracy@5$ rises and keeps steady after the dimension reaches around 20. An interesting finding is that the current genre of Vine (seven categories, including Animals, Art, Comedy, Edits, Music&Dance, Sports, and Weird) cannot represent the genres of micro-videos effectively for the recommendation. For example in Sports category, people who like basketball may not be interested in football. Therefore, the dimension of the latent genre is tuned to 20.

Moreover, we conduct another set of experiments to verify the effectiveness and the efficiency of our LGA model compared with a fully concatenating neural network. In the fully concatenating neural network, the textual and the visual features are fed into the neutral network directly with their original dimensions that are 256 and 4096 respectively. Compared with the fully concatenating neural network model, $Accuracy@k$ of the LGA model is 0.009 lower. However, the time of training fully concatenating neural network model requires around 28971 seconds while that of the LGA model is 19158 seconds. Therefore, the LGA model significantly improves the efficiency of training time around 33.9%.

Fig. 6 Impact of dimension of latent genre



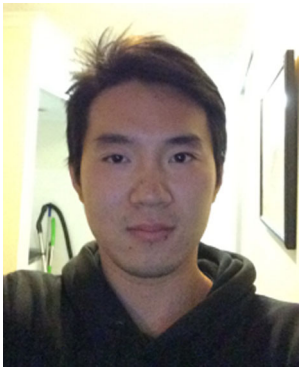
5 Conclusion

In this paper, we tackle the micro-video recommendation problem with a novel deep neural recommendation model that takes multi-modal information as input. We also exploit the latent integration of micro-video features as well as user and item. A latent genre aware recommendation is proposed to learn the ranking model from the output scores of multi-modal features by simultaneously learning the latent genres of micro-videos and the optimal recommendation scores. Experiments on real-world micro-video dataset demonstrate the superior performance regarding both effectiveness and efficiency of our proposed model.

References

1. Bordes A, Weston J, Usunier N (2014) Open question answering with weakly supervised embedding models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp 165–180
2. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: Proceedings of the 22nd international conference on Machine learning. ACM, pp 89–96
3. Chang X, Yang Y, Long G, Zhang C, Hauptmann AG (2016) Dynamic concept composition for zero-example event detection. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pp 3464–3470. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12499
4. Chang X, Yang Y, Xing EP, Yu Y (2015) Complex event detection using semantic saliency and nearly-isotonic SVM. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pp 1348–1357
5. Chang X, Yu Y, Yang Y, Hauptmann AG (2015) Searching persuasively: Joint event detection and evidence recounting with limited supervision. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26–30 2015, pp 581–590. [Online]. Available: doi:10.1145/2733373.2806218
6. Chang X, Yu Y, Yang Y, Xing EP (2016) They are not equally reliable: Semantic event search using differentiated concept classifiers. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 1884–1893
7. Chang X, Yu Y-L, Yang Y, Xing EP (2016) Semantic pooling for complex event analysis in untrimmed videos. IEEE Trans Pattern Anal Mach Intell
8. Chatfield K, Simonyan K, Vedaldi A, Zisserman A (2014) Return of the devil in the details: Delving deep into convolutional nets. arXiv:1405.3531
9. Chen J (2016) Multi-modal learning: Study on a large-scale micro-video data collection. In: Proceedings of the 2016 ACM on Multimedia Conference ACM, pp 1454–1458
10. Chen K, Wang J, Chen L-C, Gao H, Xu W, Nevatia R (2015) Abc-cnn: An attention based convolutional neural network for visual question answering. arXiv:1511.05960
11. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp 248–255
12. Ference G, Ye M, Lee W-C (2013) Location recommendation for out-of-town users in location-based social networks. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, pp 721–726
13. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2414–2423
14. Guàrdia-Sebaoun E, Guigue V, Gallinari P (2015) Latent trajectory modeling: a light and efficient way to introduce time in recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems. ACM, pp 281–284

15. Guo J, Fan Y, Ai Q, Croft WB (2016) A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, pp 55–64
16. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S Neural collaborative filtering
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. Ieee, pp 263–272
19. Huang P-S, He X, Gao J, Deng L, Acero A, Heck L (2013) Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, pp 2333–2338
20. Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J (1997) Grouplens: applying collaborative filtering to usenet news. Commun ACM 40(3):77–87
21. Liu L, Wiliem A, Chen S, Lovell BC (2017) What is the best way for extracting meaningful attributes from pictures? Pattern Recogn 64:314–326
22. Mei T, Yang B, Hua X.-S., Yang L, Yang S.-Q., Li S (2007) Videoreach: an online video recommendation system. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp 767–768
23. Park J, Lee S-J, Lee S-J, Kim K, Chung B-S, Lee Y-K (2010) An online video recommendation framework using view based tag cloud aggregation. IEEE Multimedia 1:99
24. Park S-T, Chu W (2009) Pairwise preference regression for cold-start recommendation. In: Proceedings of the third ACM conference on Recommender systems. ACM, pp 21–28
25. Pazzani MJ, Billsus D (2007) Content-based recommendation systems, in The adaptive web. Springer
26. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th conference on uncertainty in artificial intelligence. AUAI Press, pp 452–461
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. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
29. Severyn A, Moschitti A (2015) Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp 373–382
30. Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S (2014) Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 806–813
31. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
32. Vinyals O, Toshev A, Bengio S, Erhan D (2016) Show and tell: Lessons learned from the 2015 mscoco image captioning challenge
33. Wang C, Yang H, Bartz C, Meinel C (2016) Image captioning with deep bidirectional lstms. arXiv:1604.00790
34. Wang H, Wang N, Yeung D-Y (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
35. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhutdinov R, Zemel RS, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention 2(3):5. arXiv:1502.03044
36. Yin H, Cui B, Huang Z, Wang W, Wu X, Zhou X (2015) Joint modeling of users' interests and mobility patterns for point-of-interest recommendation. In: Proceedings of the 23rd ACM international conference on Multimedia. ACM, pp 819–822
37. Zhai S, Chang K-h, Zhang R, Zhang ZM (2016) Deepintent: Learning attentions for online advertising with recurrent neural networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 1295–1304
38. Zhai S, Chang K-h, Zhang R, Zhang ZM (2016) Deepintent: Learning attentions for online advertising with recurrent neural networks. In: Proceedings of the 22nd ACM SIGKDD conference on Knowledge Discovery and Data Mining. ACM, pp 1295–1304
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

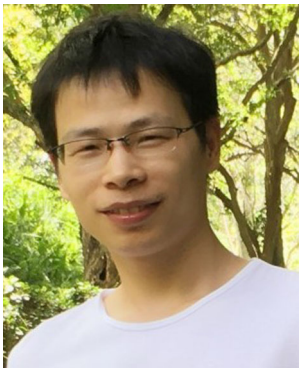


Jingwei Ma received the M.S. degree from The University of Queensland, Australia in 2016. He is currently a Ph.D student at University of Queensland. His current research interests mainly include deep learning, video analysis and recommender system.

Guang Li is currently a Ph.D student at Tianjin University, China. His current research interests mainly include data mining, deep learning and recommender system.

Mingyang Zhong received the PhD from the University of Queensland, Brisbane, Australia, in 2016. He is currently a postdoc at the University of Queensland. His current research interests include machine learning, data mining, networks, pervasive computing and mobile computing.

Xin Zhao received the PhD from the University of Queensland, Brisbane, Australia, in 2014. He is currently a postdoc at the University of Queensland. His current research interests include machine learning, data mining, healthcare analytics and social computing.



Lei Zhu received the B.S. degree (2009) at Wuhan University of Technology, the Ph.D. degree (2015) at Huazhong University of Science and Technology. He is currently a research fellow with the School of Information Technology and Electrical Engineering, University of Queensland. His research interests are in the area of image retrieval and classification.

Xue Li is honoured as one of "the most powerful people in Australia" on Big Data by the Financial Review–the Power Issue 2015. He is a Professor in the School of Information Technology and Electrical Engineering at the University of Queensland (UQ) in Brisbane, Queensland, Australia. His major areas of research interests and expertise include: Data Mining, Social Computing, Database Systems, and Intelligent Web Information Systems.