Challenging Personalized Video Recommendation

ABSTRACT

The online videos are generated at an unprecedented speed in recent years. As a result, how to generate personalized recommendation from the large volume of videos becomes more and more challenging. In this paper, we propose to extract the non-textual contents from the videos themselves to enhance the personalized video recommendation. The change of the content types makes us study three issues in this paper. The first issue is what non-textual contents are helpful. Considering the users are attracted by the videos in different aspects, multiple audio and visual features are extracted, encoded and transformed to represent the video contents in the recommender system for the first time. The second issue is how to use the non-textual contents to generate accurate personalized recommendation. We reproduce the existing methods and find that they do not perform well with the non-textual contents due to the mismatch between the features and the learning methods. To address this problem, we propose the collaborative embedding regression (CER) method in this paper. Our experiments show that CER is more accurate whether the video content features are non-textual or textual. The last issue is how to fuse multiple content features to further improve the video recommendation accuracy. We develop a new feature fusion method to differentiate the impacts of different contents in the final recommendations. Compared to average and learning-to-rank fusion methods, the proposed method is more accurate.

1. INTRODUCTION

Watching videos online has been one of the indispensable entertainment activities in human's daily life. Many famous video websites, such as YouTube [1], Netflix[2], Hulu[3], host tremendous videos to meet such demand. However, the huge video repository increases the users' retrieval burdens when they try to watch the unseen videos [20]. To improve this situation, the recommender system has been proposed alongside the retrieval system. It leverages the implicit inputs of the users to uncover their interests and generates the personalized video recommendations for each user from her or his unseen videos [10]. As an effective way to help the users browse the unseen videos, the recommender system has been widely deployed in today's video websites [5,6].

The main input of a general recommender system is the rating matrix and the content features [22]. The rating matrix records the users' preferences on the items such as {dislike, like}. It is always of the same form [9,16,25,27] across different recommender systems. The content features record the main information delivered by the items. Existing methods rely on the content features to perform the out-of-matrix recommendation (i.e. cold-start problem) where the recommended videos have not been rated by any user [25]. It is worth noting that most of the content features in use are textual. In video recommendation scenarios, it is far from enough to reveal the video contents by only text. For example, on YouTube, plenty of videos only have the titles. The scarcity of the textual contents makes the existing methods fail to generate accurate out-of-matrix video recommendation in most cases. The systems in [9,16,29] try to use non-textual content features to recommend videos, music and products respectively. However, these systems use the non-textual content features only for in-matrix recommendation where the rating matrix plays a more important role during the learning. Whether the non-textual content features can really benefit the out-of-matrix recommendation, which is more important to explore, is still unclear.

In this paper, we firstly introduce several new non-textual content features to represent the videos. We do not use the normalized color histogram and the aural tempos which are applied by the system in [29]. This is because the results reported in [29] indicate these two non-textual contents are helpless to improve the video recommendation. Instead, we propose to use MFCC [1], SIFT [3,24], Improved Dense Trajectory [26] as well as Convolutional Neural Network (CNN) [13] to quantize the audio, scene and action contents in the videos respectively. This is based on the fact that the users are attracted by the videos from different aspects. Encoding these non-textual contents with the state-of-the-art methods [11,18], the generated content features are more accurate and complete to represent the video contents [1,26,28].

With the new non-textual content features as well as the widely used textual content features, we reproduce the ex-

[1]http://www.youtube.com/
[2]http://www.netflix.com/
[3]http://www.hulu.com/

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

isting methods [9, 10, 16, 19, 25, 27] and test them in both in-matrix and out-of-matrix recommendation scenarios. The results show that none of the existing methods can achieve high accuracy in both tests: the weighted matrix factorization (WMF) based methods are more accurate in the in-matrix test, while the Bayesian personalized ranking (BPR) based methods are more accurate in the out-of-matrix test. Furthermore, we find that the mismatch between the features and the non-linear learning methods makes the WMF based methods [16, 25, 27] fail to generate accurate out-of-matrix recommendation. To improve that, we propose collaborative embedding regression (CER) in this paper. Compared to the recent methods [16, 25, 27], CER applies the linear learning method on the content features to obtain the latent vectors. The experimental results show that, for an arbitrary single content feature (no matter non-textual or textual), CER has the same accuracy as the WMF based methods in the in-matrix test, while significantly outperforms the BPR based methods in the out-matrix test. Besides, the model training of CER is efficient, thus can be easily scalable to the large-scale datasets.

In addition, we also study how to fuse multiple content features to generate more accurate top-k recommendation in the out-of-matrix scenario. We do not apply the feature fusion technique to the in-matrix scenario, because the effect of content features is negligible in the warm-start recommendation setting. For the fusion in out-of-matrix scenario, the practical methods are average and learning-to-rank fusion [1, 14, 28]. All of them use the linear combination to calculate the final ratings for the videos. The difference is how the weights are set. The average method sets the weights equally, while the learning-to-rank method sets high weights for the contents of high accuracy during the training. However, in video recommendation, there exist big accuracy gaps between different kinds of content features. Average or learning-to-rank methods fail to pay enough attention to such gaps. We therefore design another way to set the weights: firstly, the content features are ranked according to their accuracy in the out-matrix test; then, the content feature achieving high recommendation accuracy will get a weight which is larger than the sum of the remaining content features' weights. We try to use such weights to make the most powerful contents have the conclusive impacts on the final ratings. Our experiments show that the proposed fusion method not only fuses the textual and non-textual contents together to achieve more accurate top-k recommendation, but also fuses the non-textual content features together to achieve close accuracy as the textual content features. The results indicate that the recommender system can generate accurate out-of-matrix recommendation even without the textual contents. As more informative non-textual content features emerge in the future, the accuracy is very promising to exceed those achieved by the textual content features.

In summary, the contributions of this paper include:

1. To our best knowledge, we are the first to use MFCC, SIFT, IDT as well as CNN to represent the non-textual contents of the videos in the recommender system. Compared to those used in the previous works [29], the novel video content features introduced in this paper capture more meaningful information from the videos. They are beneficial to the out-of-matrix video recommendation and verified by our experiments;
2. We propose a novel video recommender model CER (collaborative embedding regression) to work with the textual and non-textual content features uniformly. Our experiments conducted in both in-matrix and out-of-matrix settings certify that CER is more accurate than all the existing methods [9, 16, 25, 27], whatever type of content features is used;
3. We study how to fuse multiple types of content features to generate more accurate out-of-matrix recommendation and propose a novel fusion method. The experimental results show that our fusion method makes the out-of-matrix recommendations more accurate than the average fusion and learning-to-rank fusion.

2. BACKGROUND AND RELATED WORK

The basic elements of a recommender system are users and items. In video recommendation scenario, the items are the videos. For convenience, we assume there are m users and n videos in total. In addition to that, we use $r_{ij} \in \{0, 1\}$ to denote the ith user's rating on the ith video: $r_{ij} = 1$ means the ith user likes the jth video; $r_{ij} = 0$ means the ith user dislikes the jth video or has never rated the jth video.

For a single user, the task of the video recommender system is to recommend the videos which have not been rated by the user but potentially attractive to the user. To accomplish this task, the video recommender system predicts the user ratings and recommends the personalized top-k videos according to the predicted ratings. The video recommendation is further divided into two situations, namely, in-matrix and out-of-matrix recommendation. The in-matrix recommendation recommends the videos which have not been rated by the target user but rated by other users [25]. Based on the co-rating behaviors between the similar users, the state-of-the-art methods [10, 16, 25, 27] apply collaborative filtering (CF) to generate accurate recommendation. The out-of-matrix recommendation recommends the videos which have not been rated by any users [25]. In other words, there is no rating information to exploit and leverage. This makes CF based methods ineffective. Alternatively, the content based methods are applied to generate the recommendation.

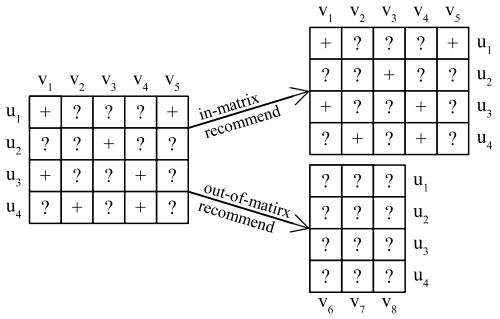


Figure 1: Rating matrix for in-matrix and out-of-matrix recommendation. In the figure, red $\surd$ denotes "like", blue $\times$ denotes "dislike" and black ? denotes "have not been rated".

The in-matrix recommendation is mainly generated by the weighted matrix factorization (WMF) based methods [12] and Bayesian personalized ranking (BPR) based methods [19]. They have three common points: first, both of them

are derived from collaborative filtering (CF); second, they learn the user and item latent vectors to predict ratings for the videos; third, they select the top ranked videos as recommendation according to the predicted ratings. The major difference between them is the optimization objective. The WMF based methods [12] learn the latent vectors by minimizing the rating prediction loss on the training data., while the BPR based methods [19] learn the latent vectors by preserving the personalized ranking.

Recent studies on WMF based methods and BPR based methods try to make the latent vector learning incorporate with the content features. As a result, the learned latent vectors are not only applicable for the in-matrix recommendation but also for the out-of-matrix recommendation. The representative WMF based methods are collaborative topic regression (CTR) [25], deep content-based music recommendation (DPM) [16] and collaborative deep learning (CDL) [27]. CTR and CDL only integrate textual contents, while DPM only considers non-textual contents. The representative BPR based method is visual Bayesian personalized ranking (VBPR) [9]. Similar to DPM, VBPR is designed for incorporating non-textual contents.

3. VIDEO CONTENT FEATURES

This section introduces the content features we will use in our video recommender system. These features are divided into two groups. The first group is the textual content features which are mainly extracted from the descriptions of the videos. The second group is the non-textual content features which are mainly extracted from the videos themselves. These non-textual content features are certified to be more powerful than the normalized color histogram and the aural tempos in many tasks, such as event detection [1], action recognition [26], scene classification [24], object classification [23] and so on. This inspires us to apply them to video recommender system.

3.1 Textual Content Features

In the previous content-based video recommender systems [5,6,27], the video contents are described by the texts. These texts include titles, descriptions, reviews, annotations as well as meta information. Based on these texts, two kinds of content vectors are extracted. The first kind is the word vector which stores the word distribution over the associated texts. To construct the word vectors, the texts belonging to the same videos are concatenated into one. After removing stop words and stemming [27], the top discriminative words are selected by the TF-IDF values to form the vocabulary. Then, the word vectors are generated by word count according to the vocabulary. The second kind is the meta vector. It stores the meta data about the videos such as producers, countries, languages, release dates, actors, genres and so on. The top discriminative meta items are selected by the global frequency to form the codebook. Different from the word vector where a word may appear more than once, the meta item in the meta vector just appears once. Accordingly, the meta vector is binary and usually even sparser.

3.2 Non-Textual Content Features

In addition to the texts, the videos are another source to extract the content features for the content-based methods. In previous work [29], the authors extract the normalized color histogram and aural tempos to describe the video contents. The results reported in [29] show that these content features are not significantly beneficial to the video recommendation. This is reasonable, because these non-textual content features fail to discriminate videos even if they are very irrelevant. For example, when a video is about the sky and another video is about the sea, the normalized color histogram will consider these two video to be very similar due to the color blue. In this case, the recommender system is most likely to recommend the sky-related videos to the users who love the sea.

However, the limitations of the normalized color histogram and aural tempos does not mean that all the non-textual content features are useless to the video recommendation. To prove the value of the non-textual content features, in our work, we introduce some novel non-textual content features for the video recommendation. They have been proved very helpful in event detection [1], scene classification [24], action recognition [26] and object classification [23]. Based on their recent successes in other areas, we think they will help the video recommender capture more information about the videos than before.

The users are attracted by the videos from multiple ways. Considering that, we propose to extract non-textual the content features from the videos in terms of audios, scenes and actions. The details of the extracted features are as follow:

1. MFCC (Mel-Frequency Cepstral Coefficients)

MFCC measures the audio changes in the sound. The MFCC features in this paper are extracted by the following process: first, down-sampling the audio track of the videos to 16 kHz with 16 bit resolution; second, using the window size of 25 ms and the step size of 10 ms to set MFCC extractor and setting the number of the channels to 13; third, concatenating the first, second derivatives and the energy of the MFCC during the extraction to form a 40 dimensional feature step by step. After that, the whole audio track of a video will be transformed into a feature array.

2. SIFT (Scale Invariant Feature Transform)

SIFT [15] captures the texture information in the images. Since the SIFT features can match the same visual objects of different scales [15], it has been widely applied in scene classification [24] and image retrieval [11]. In this paper, we use two variants of SIFT to extract the texture information from the sampled frames in the videos. They are OSIFT (Opponent SIFT) [24] and MoSIFT (Motion SIFT) [3]. OSIFT transforms the original RGB color space by light color change and shift, which provides more robust SIFT features. The dimension of OSIFT feature is 384. MoSIFT leverages the optical flow between frames to select SIFT features. The dimension of MoSIFT feature is 256.

3. IDT (Improved Dense Trajectory)

IDT [26] captures the motion information in the video. It leverages the dense sampling and camera motion removing to extract state-of-the-art motion features. The dimension of IDT feature is 426.

4. CNN (Convolutional Neutral Network)

CNN captures the semantic information in the images. In recent years, the CNN has shown its advantage over the other models on ImageNet object classification competition [21]. Some follow-up researches show that using the pre-trained CNN on ImageNet to extract features from images is beneficial to video retrieval [28]. As a result, given the sampled frames in the video, we use the pre-trained CNN model from VGG group [23] to extract the features from the

$pool_5$ layer. The original $pool_5$ features are tensor. In order to transform them into vectors, we apply spatial pooling on them, inspired by [8]. This results in each sampled frame having 49 feature points whose dimension is 512.

Unlike MFCC, MoSIFT and IDT which take the whole audio or video file as input, the OSIFT and CNN should be applied on the frames sampled from the videos. We follow [1] [28] to fetch 5 frames per second from the videos. On the other hand, there is usually a normalization process on the raw features. According to [1, 2], we apply SSR (signed squared root) to normalize all the raw features.

Feature	MFCC	OSIFT	MoSIFT	IDT	CNN	
Encoder	FV	FV	FV	FV	VLAD	FV
Dimension	10240	98304	68608	128304	65536	131072

Table 1: The dimensions of the non-textual content vectors before PCA

After one feature extracted, each video generates a group of vectors. These grouped feature vectors need to be transformed into one vector for the learning in the recommender system [9, 16, 25, 27]. An intuitive transformation method is to average the feature vectors by dimension directly. But it is not recommended due to the limited representative capacity [17]. Recent studies show that it is better to transform the feature vectors by an encoding process [1, 17, 18, 28]. As a result, we apply the state-of-the-art encoding methods, namely Fisher vector (FV) [18] and VLAD [11], to transform a group of feature vectors into one vector. The details of video different encoded non-textual content vectors are listed in Table 1. The column "dimension" denotes the vector dimension after encoding. In Table 1, all the encoded feature vectors' dimensions are very high. The high dimensionality makes it hard to figure out which kind of content vectors is more effective and powerful. To control the impact from the high dimensionality, we apply PCA to reduce the dimension to 4000 in this paper.

4. VIDEO RECOMMENDATION

In this section, we reproduce the existing methods and analyze their accuracy in in-matrix and out-of-matrix scenarios. We study what make the existing methods disable to generate the accuracy recommendation with the non-textual features. The study inspires us to propose an improved framework to deal with the textual and non-textual content features in a unified way. After that, we study how to fuse the recommendations from different content features to achieve higher accuracy and propose a new fusion method.

4.1 Recent Methods on Various Contents

Given the video content features we extracted, an interesting question arises. That is, how the recent methods perform with the various content features in top-k video recommendation tasks. To answer this question, we reproduce the recent WMF based methods as well as VBPR [9] with 50 latent factors on Movielens 10M dataset [7]. We exactly adopt the settings proposed in [9, 10, 16, 19, 25, 27]. Besides that, the reproduction also modifies CDL [27] and VBPR [9] to make them work with MFCC and CNNFV vectors. The list of the methods and the corresponding content features are summarized in Table 2:

Methods	Contents
WMF, BPR	N/A
CDL, VBPR, CTR	WORD
CDL, VBPR, CTR	META
CDL, VBPR, DPM	MFCC
CDL, VBPR	CNNFV

Table 2: The reproduced methods as well as the content features in use

The methods in Table 2 are tested in both in-matrix and out-of-matrix settings with their optimal parameters. More details about the data split and evaluation metric can refer to our experiment section. The results are depicted in Figure 2 where the subscripts denote the content vectors in use. To clearly show the differences between these methods, we adopt the values of Accuracy@30.

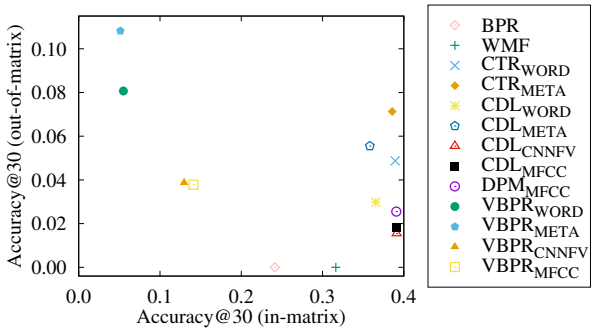


Figure 2: Performance of state-of-the-art methods in both in-matrix and out-of-matrix settings. To clearly display the methods which only support in-matrix recommendation, we shift the origin of vertical axis.

We make the following observations from the results illustrated in Figure 2:

1. WMF based methods achieve more accurate recommendation than BPR based methods in the in-matrix test. In Figure 2, all the WMF based methods locate at the right side of the BPR based methods except DPM. This shows that the accuracy of WMF based methods is superior in the in-matrix test. Besides that, CDL's in-matrix accuracy does not significantly vary with different content vectors. All the facts indicate that **the in-matrix accuracy is mainly dominated by the latent vectors generated by WMF**.

2. VBPR achieves the highest accuracy in out-of-matrix test. In Figure 2, when the input feature is fixed, VBPR's position is always higher than all the other methods. This shows that VBPR is the most effective method in the out-of-matrix test, which indicates that **the content-based part of the existing WMF methods is not suitable for the out-of-matrix recommendation**.

In summary, our reproduction experiment shows that none of the existing methods achieve high accuracy in both in-matrix and out-of-matrix scenarios. To address this prob-

lem, we propose a new WMF based method in this paper.

4.2 Collaborative Embedding Regression

All the recent WMF based methods [16, 25, 27] follow the similar rating generation process. The major difference between them is how to generate the content latent vectors. CTR [25] takes word vectors as input. It generates the content latent vectors by latent Dirichlet allocation (LDA). Since the optimization of LDA is based on word count only, CTR naturally fails to support the non-textual content features which are real values. Compared to CTR, DPM [16] and CDL [27] can generate content latent vectors from various contents. They achieve this by applying the multiple layer perception (MLP) and the stacked de-nosing auto-encoder (SDAE) as the generation functions respectively. However, the results in Figure 2 show that none of them perform well in the out-of-matrix scenario. This is because the non-textual content features are more suitable for linear models [11, 18]. MLP and SDAE which apply the non-linear transformation on the content features will corrupts the original information in the non-textual content features. Based on above analysis, we propose a novel recommender model CER, collaborative embedding regression, to work with various content vectors including both textual and non-textual content features. We denote d as the dimension of the content vector and k as the dimension of the latent vector. The whole generation process of CER on a single content feature is as follows.

1. For each user i, draw a user latent vector $w_i \in \mathcal{R}^{k\times 1}$:
$$w_i \sim \mathcal{N}(0, \lambda_u^{-1} I_k).$$
2. For each video j,
   (a) Generate a video content latent vector $h'_j \in \mathcal{R}^{k\times 1}$:
   $$h'_j = E^T f_j.$$
   (b) Draw a video content latent offset $\epsilon_j \in \mathcal{R}^{k\times 1}$:
   $$\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_k)$$
   , then set latent vector as:
   $$h_j = h'_j + \epsilon_j.$$
3. For each user-video pair (i, j), draw the rating:
$$r_{ij} \sim \mathcal{N}(w_i^T h_j, c_{ij}^{-1})$$

where I_k is a identity matrix whose rank is k, $f_j \in \mathcal{R}^{d\times 1}$ is the video content vector, $E \in \mathcal{R}^{d\times k}$ is the embedding matrix, and c_{ij} is the confidence parameter for the user-item pair (i, j). The value of c_{ij} is defined as below [25] [27]:

$$c_{ij} = \begin{cases} 1, & \text{if } r_{ij} = 1 \\ 0.01, & \text{if } r_{ij} = 0 \end{cases}$$

Noticing that, in step $2(a)$, we use linear embedding instead of the non-linear methods. This is more suitable for learning the content latent vectors from the non-textual features as we discussed.

For rating prediction, the latent vectors and the embedding matrix need to be learned. This requires maximizing the log likelihood of latent vector generation and applying the regularization on E simultaneously. Putting them together, the objective function of CER is formulated in Eq. 1:

$$\begin{aligned} &\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{c_{ij}}{2}(w_i^T h_j - r_{ij})^2 + \frac{\lambda_u}{2}\sum_{i=1}^{m} w_i^T w_i + \\ &\frac{\lambda_v}{2}\sum_{j=1}^{n}(h_j - E^T f_j)^T(h_j - E^T f_j) + \frac{\lambda_e}{2}||E||_F^2. \end{aligned} \tag{1}$$

where k, c_{ij}, λ_u, λ_v and λ_e are the hyper parameters and $||\cdot||_F$ denotes the Frobenius norm. When these hyper parameters are fixed, the optimal latent vectors w and h as well as the embedding matrix E are obtained by performing the coordinate descend which is similar to [25, 27]. Specifically, in each iteration, given the current E, the derivatives with regard to w_i and h_j are computed and set to zero. This gives us the following updates:

$$\begin{aligned} w_i &\leftarrow (HC_iH^T + \lambda_u I_k)^{-1} HC_iR_i \\ h_j &\leftarrow (WC_jW^T + \lambda_v I_k)^{-1}(WC_jR_j + \lambda_v E^T f_j) \end{aligned}$$

where $W = (w_i)_{i=1}^m \in \mathcal{R}^{k\times m}$ is the user latent matrix, $H = (h_j)_{j=1}^n \in \mathcal{R}^{k\times n}$ is the video latent matrix, and $F = (f_j)_{j=1}^n \in \mathcal{R}^{d\times n}$ is the content matrix. For user i, $C_i \in \mathcal{R}^{n\times n}$ is a diagonal matrix with c_{ij}, $j = 1\cdots, n$ as the diagonal elements, $R_i \in \{0, 1\}^{n\times 1}$ is a vector with r_{ij}, $j = 1\cdots, n$ as its elements. For video j, C_j and R_j are similar defined.

Then, given the learned H in current iteration, the derivatives regard to E are computed and set to zero. And this gives us the following update:

$$E \leftarrow (\lambda_v FF^T + \lambda_e I_d)^{-1}(\lambda_v FH^T)$$

Similar to CTR and CDL, CER supports both in-matrix and out-of-matrix rating prediction. For in-matrix predictions, given a user-video pair (i, j), the rating $\hat{r}_{ij}$ is estimated as $w_i^T(E^T f_j + \epsilon_j)$. For out-matrix prediction, the rating $\hat{r}_{ij}$ is predicted as $w_i^T E^T f_j$ since there is no offset observed. In summary, the rating predictor is defined as:

$$\hat{r}_{ij} = \begin{cases} w_i^T h_j, & \text{in matrix} \\ w_i^T E^T f_j, & \text{out of matrix} \end{cases} \tag{2}$$

4.3 Multiple Content Fusion

In WMF based methods, the out-matrix recommendation accuracy varies significantly when different contents are in use. This indicates the top-k video recommendations diverge from different content features. In recent video retrieval system [1], such divergence is leveraged to fuse more accurate ranking list. We think the fusion strategy can benefit the video recommender system as well. So we study how to fuse the top-k recommendations from multiple content features in this paper as well.

We apply the common weighted sum strategy to implement the fusion in this paper. The fused estimate rating is as follow given the user-video pair (i, j):

$$\bar{r}_{ij} = \sum_{l=1}^{L} w_l \hat{r}_{ij}^l,$$

where L is the number of different content features, w_l is the weight of the corresponding content feature, $\hat{r}_{ij}^l$ is the estimate rating based on the corresponding content. The

challenge of fusion is how to tune the weights. A naive solution is treating the content features equally, namely average fusion. However, the relative accuracy divergences between different content features are very large as shown in Figure 2. The average fusion neglects the divergence, which leads to inferior result. Another solution is learning the weights by the learning-to-rank methods [14]. But the computation of feature weights by learning to rank methods is time-consuming. In this paper, we find an efficient way to decide the weights. The input of our method is the content ranking which is based on their out-of-matrix accuracies. After the content ranking list is obtained from cross validation, the weight of lth content feature is defined as:

$$w_l = p(1-p)^{l-1}, \tag{3}$$

where $p \in [0.5, 1)$ is a hyper parameter. Noticing that, for $\forall t > 0$, $\sum_{l=t+1}^{L} w_l \leqslant w_t$. This ensures that the lth content feature has higher weight than the remaining less powerful content features. Eq. 3 makes the more powerful feature have more impact in the final estimate rating. To clearly illustrate the calculation of the weights, we use Table 3 to display an example when four features are given and ranked.

Feature	WORD	CNNFV	IDT	MFCC
l	1	2	3	4
w_l	0.5	0.25	0.125	0.0625

Table 3: An example of the weights generated by the proposed fusion method when p equals 0.5.

5. EXPERIMENT

5.1 Dataset

In this paper, we use MovieLens 10M [7] as the base dataset to conduct the experiments. The Movielens dataset does not include the videos or their download links. So we try to collect the videos from the Youtube [4] by ourselves. However, there are no full length videos free for downloading due to the copyright. Instead, we download the trailer videos according to the movie titles. After manual check, we ensure these trailers are from the original full-length videos. In addition to that, a small fraction of the movies do not have trailers on Youtube, we use their available clips instead. By this means, we collect 10380 videos in the end. The Movielens 10M dataset has 10682 movies in total. Therefore, the ratings not associated with the collected 10380 videos are removed. This slightly decreases the amount of the ratings from 10000054 to 9988676 (drops 0.1% of the total). Besides, the collect videos are resized to accelerate the content feature extraction: their widths are reduced to 240 pixels and their heights are adjusted proportionally.

The Movielens dataset also provides the corresponding movie ids on IMDB[5]. Based on these ids, we crawl the movie plots, actors, directors, companies, languages and genres. Each movie's the title and plot are concatenated into a document. The top 20000 words are selected according to the global TF-IDF values as [25, 27]. Then, the word vector of each movie is generated by word counting. The other textual information which include actors, directors, languages, companies and genres are made into meta vectors. To make the textual content vectors have the same dimension, top 20000 items are selected as the word vector generation.

The ratings in Movielens 10m dataset range from 1.0 to 5.0 with interval 0.5. They need to transform to $\{0, 1\}$ to represent {dislike, like}. Refer to CTR [25] and CDL [27], we treat the rating 5.0 as like. This makes the ratings whose values are 5.0 transform to 1, while all the other ratings become 0. After that, 1543593 likes are generated in total. Compared to the whole elements in the rating matrix which consists of 69878 users and 10380 videos, the amount of likes only occupies 0.2%. Besides, we notice that the data preparation step in CTR and CDL remove the videos and the users whose rating amounts are below some thresholds. In this paper, we do not perform such process.

5.2 Settings

5.2.1 Data Split

Following the previous works [25,27], we apply 5 fold cross validation to test the accuracy of each method in both the in-matrix and the out-of-matrix scenarios. Specifically, the training set, in-matrix test set and out-of-matrix test set have 60%, 20%, 20% of the total likes. To achieve this, the video ids are first split into five folds uniformly. Then the ratings associated with the split video ids are split correspondingly. When one rating fold is used as the out-of-matrix test set, the rest four rating folds are mixed together and re-split into four folds uniformly. After that, three re-split rating folds will be used as the training set and the rest re-split rating fold will be used as the in-matrix test set.

We split dataset in this way because comparing the accuracy of a recommendation method in both in-matrix and out-of-matrix scenarios in a unified way that is more reasonable. In previous works [25], the in-matrix and out-of-matrix tests are separately conducted. The training data changes when the scenario switches. It actually makes the two recommendation scenarios incomparable. Our split protocol improves that situation so that we can provide more exact comparison in our analysis.

5.2.2 Comparison Methods

We compared the proposed CER method with six state-of-the-art methods. Their descriptions and settings as follow:

Weighted Matrix Factorization (WMF) [12] is for in-matrix recommendation alone. We use it as the baseline [12]. Its accuracy is best when $\lambda_u = 0.01, \lambda_v = 0.01$

Collaborative Topic Regression (CTR) [25] learns the content latent vectors from the word vectors by LDA. It recommends videos in both in-matrix and out-of-matrix scenarios. In our experiments, we train CTR with both the word vectors and the meta vectors. The training of CTR requires applying LDA on the textual vectors for initialization. We use a python version [6] to do that. After that, we tune the parameters of CTR and find its performance is best when $\lambda_u = 0.1, \lambda_v = 10$.

DeepMusic (DPM) [16] uses MLP to learn the content latent vectors from MFCC. It recommends videos in both in-matrix and out-of-matrix scenarios. Since MLP can accept all kinds of the content features as input, we make DPM

[4]https://www.youtube.com/

[5]http://www.imdb.com/

[6]https://pypi.python.org/pypi/lda

work with all the content vectors generated in this paper. In DPM, $\lambda_u = 0.1, \lambda_v = 10$ achieve the best performance. The rest parameters are tuned according to [16].

Collaborative Deep Learning (CDL) [27] learns the video content latent vectors by SDAE from the word vectors of the videos. It can recommend videos in both in-matrix and out-of-matrix scenarios. Replacing the binary visible layer with Gaussian visible layer, SDAE can accept the non-textual content vectors as input. We therefore modify CDL when the input content vectors are non-textual. After the parameter tuning, we find CDL achieves its best performance when $\lambda_u = 0.1, \lambda_v = 10, \lambda_n = 1000$.

Bayesian Personalized Ranking (BPR) [19] is the very first version of BPR based method. It is only applicable in in-matrix scenario. We use the results from BPR as the baseline like WMF. The parameters are $\lambda_u = 0.0025, \lambda_i = 0.0025, \lambda_j = 0.00025, \lambda_b = 0.0$. BPR is sensitive to the learning rate. This value is set to 1^{-4} in this paper.

Visual Bayesian Personalized Ranking (VBPR) [9] is the extension of BPR to combine visual contents in the predictor. It can recommend videos in both in-matrix and out-of-matrix scenarios. VBPR can work with all the contents vectors. After tuning, the parameters of VBPR are set to $\lambda_u = 0.0025, \lambda_p = 0.0025, \lambda_i = 0.0025, \lambda_j = 0.00025, \lambda_b = 0.0, \lambda_e = 0.0$. The learning rate of VBPR is same as BPR.

We uniformly set the max iteration and the dimension of the latent vectors for all the methods. The max iteration of each method is set to 200 while the dimension of the latent vectors is set to 50. Besides that, the training of DPM and CDL needs to set the layers of MLP and SDAE respectively. According to [27], we configure MLP and SDEA with 3 layers. The amounts of neutrons in each layer are 20000, 2000 and 50 respectively. The parameters of our proposed CER is set to $\lambda_u = 0.1, \lambda_v = 10, \lambda_e = 1000$.

We compare our proposed fusion method with three state-of-the-art methods. The descriptions of them are as follow:

Average Fusion (AF) averages the predicted ratings from different video content features.

Ranking SVM (SF) gets the weights in a learning-to-rank way by ranking SVM [14]. SF achieves its best performance when $C = 0.01$.

Ranking BPR (BF) gets the weights in a learning-to-rank way by BPR [19]. BF achieves its best performance when $\lambda_u = 0.0025, \lambda_i = 0.0025, \lambda_j = 0.00025$.

Our proposed fusion method is denoted as PF. The ranking of the content features is based on the out-of-matrix accuracy on the first fold. Given the ranking list, we find PF achieves its best performance when $p = 0.5$.

5.2.3 Evaluation Metrics

We adopt the evaluation methodology and measurement Accuracy@k applied in [4, 30] to evaluate the top-k video recommendation accuracy. In details, according to our data split protocol mentioned in Section 5.2.1, each user will have roughly 8000 unrated videos in the in-matrix test and 2000 unrated videos in the out-of-matrix test. We compute the ratings based on the latent vectors or the content vectors, then generate a ranking list of the unrated videos for each user according to the predicted ratings. The evaluation takes the top-k videos from the ranking list as the personalized recommendation. For each user-video pair (i, j) in the test set D_{test}, if video j is in user i's recommendation, we have a hit (i.e., the ground truth video is recommended to the user), otherwise we have a miss.

All the methods are evaluated by Accuracy@k where a higher value means better performance. Its calculation proceeds as follows. We define Hit@k for a single test case as either the value 1, if the ground truth video is in a user's top-k video recommendation, or the value 0 if otherwise, if otherwise. The overall Accuracy@k is defined by averaging all the test cases:

$$Accuracy@k = \frac{\#Hit@k}{|D_{test}|} \tag{4}$$

where $\#Hit@k$ denotes the total number of hits in the test set, and $|D_{test}|$ is the number of all test cases. The experimental results have been validated by means of a standard 5-fold cross validation. In previous works [25] [27], k is selected from $\{50, 100, 150, 200, 250, 300\}$. According to [6], such values of k are too large for a user to receive at once in the real world recommender system. Therefore, k is selected from $\{5, 10, 15, 20, 25, 30\}$ in this paper.

5.3 Experimental Results and Analysis

In the following experiments, we evaluate the performance of our proposed CER and other comparison methods in both in-matrix and out-of-matrix settings. Besides, we also study whether the multiple content feature fusion can improve the out-of-matrix recommendation. Recommendation efficiency is also studied.

5.3.1 In-matrix Accuracy

The first experiment examines the in-matrix accuracy of different methods. For each recommendation method, the accuracy differences incurred by the different content features are actually ignorable. In Figure 3, we report the best in-matrix accuracy of each method. Overall, our proposed CER achieves the highest accuracy compared to the comparison methods although the superiority is not visually obvious in Figure 3. The differences between BPR based methods and WMF based methods are significant. This indicates WMF based methods are more suitable for in-matrix top-k recommendation. Besides, the differences between pure WMF and its variants (CTR, DPM, CDL, CER) are also non-negligible. This indicates content information is beneficial to in-matrix top-k recommendation.

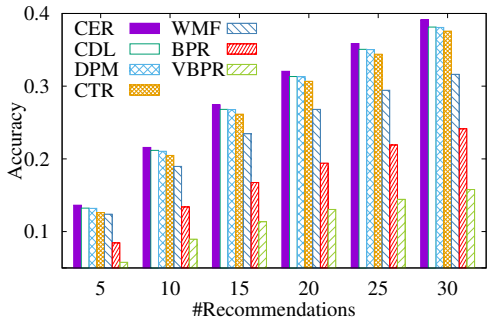
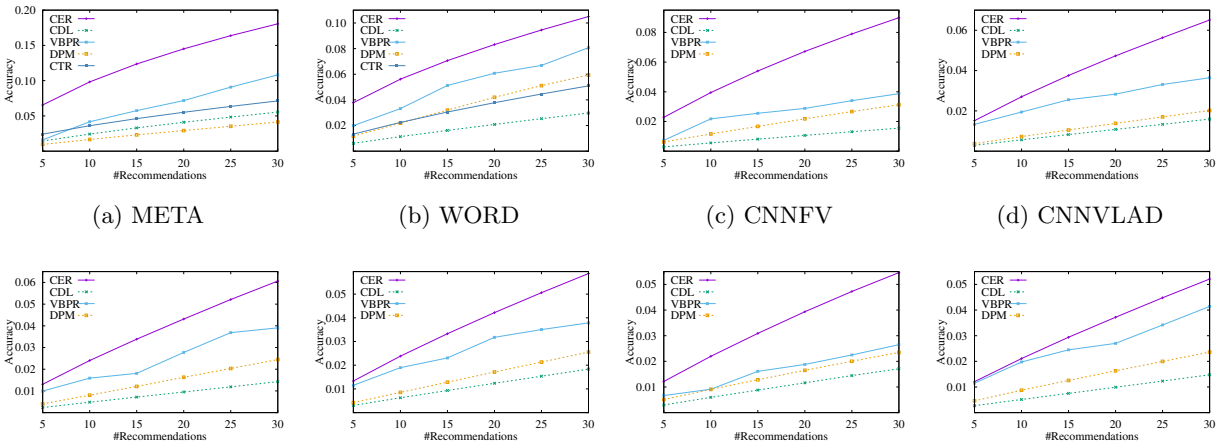


Figure 3: Accuracy@k of different methods under in-matrix setting

5.3.2 Out-of-matrix Accuracy



(a) META (b) WORD (c) CNNFV (d) CNNVLAD

(e) IDT (f) MFCC (g) OSIFT (h) MoSIFT

Figure 4: Accuracy@k of different methods and features under out-of-matrix setting

The in-matrix test shows that our proposed CER is as good as recent WMF based methods. More important issue is that whether CER can improve the out-of-matrix accuracy as well. Since out-of-matrix accuracy is heavily dependent on the types of content features, we show the performance of all recommendation methods with different types of content features in Figure 4. We sort the sub-figures in descending order according to CER's performance in the out-of-matrix scenario. In these figures, CER's out-of-matrix accuracy significantly outperforms the other WMF based methods. What's more, CER's out-of-matrix accuracy is superior to the most effective existing method VBPR as well. This indicates embedding is more suitable for generating latent content vectors in top-k video recommendation. On the other hand, the results in these figures also indicate that textual content features are still most powerful in out-of-matrix recommendation, and our proposed non-textual feature CNNFV achieves comparable performance with the textual features. We think the non-textual content features will catch up with the textual content features with the fast development of deep learning techniques. The experimental results also indicate without textual contents, the recommender system can still generate accurate recommendations by exploiting and integrating non-textual content features.

6. CONCLUSION

In this paper, we firstly propose to use non-textual content features from the videos to perform video recommendations. We study and select multiple state-of-the-art audio and video features to represent the videos, and encode them to work with the existing methods. However, we notice that none of the methods are able to generate accurate top-k recommendation in both in-matrix and out-of-matrix scenarios. We propose the collaborative embedding regression (CER) to address this problem. Our experiments show that CER is the only method which performs well in both in-matrix and out-of-matrix tests. Considering that CER is also more efficient than DPM and CDL, CER is the most powerful method for top-k video recommendation in this paper. Besides, we also study whether fusing the recommendations from multiple content features could improve the recommendation accuracy. We find that, when the accuracy gaps among different content features are large, it is beneficial to give higher weight to the content features which get better accuracy in the cross validation. As a result, we proposed a novel fusion method to achieve this. The experiments show that the proposed method outperforms the average and learning-to-rank methods in out-of-matrix test.

7. REFERENCES

[1] R. Aly, R. Arandjelovic, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. Mcguiness, N. O'Connor, D. Oneata, et al. The axes submissions at trecvid 2013. In *TRECVID Workshop*, 2013.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.

[3] M.-y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009.

[4] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pages 39–46, 2010.

[5] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. V. Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. In *RecSys*, pages 293–296, 2010.

[6] C. A. Gomez-Uribe and N. Hunt. The netflix recommender system: Algorithms, business value, and innovation. *TMIS*, 6(4):13, 2015.

[7] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *TiiS*, 5(4):19, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *PAMI*, 37(9):1904–1916, 2015.

[9] R. He and J. McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*, pages 144–150, 2016.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[11] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 34(9):1704–1716, 2012.

[12] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS.*, pages 1106–1114, 2012.

[14] T. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] A. V. D. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, pages 2643–2651, 2013.

[17] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CoRR*, abs/1405.4506, 2014.

[18] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[20] F. Ricci, L. Rokach, and B. Shapira, editors. *Recommender Systems Handbook*. Springer, 2015.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[22] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[24] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596, 2010.

[25] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *SIGKDD*, pages 448–456, 2011.

[26] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[27] H. Wang, N. Wang, and D. Yeung. Collaborative deep learning for recommender systems. In *SIGKDD*, pages 1235–1244, 2015.

[28] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, pages 1798–1807, 2015.

[29] B. Yang, T. Mei, X. Hua, L. Yang, S. Yang, and M. Li. Online video recommendation based on multimodal fusion and relevance feedback. In *CIVR*, pages 73–80, 2007.

[30] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. *PVLDB*, 5(9):896–907, 2012.