Multi-info Fusion Based Video Recommendation System

Yang Liu¹,*, Guijuan Zhang¹,*, Xiaoning Jin¹,² and Haifeng Yuan²,³

¹Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Technology, Beijing, China
²School of Information, Renmin University of China, Beijing, China

¹yangliubjut@outlook.com; ²zhangguijuan@163.com; ³jinxn@bjut.edu.cn; ³yuanhaifeng@ruc.edu.cn

Abstract. The great progress in recommendation system help users discover more interesting items that satisfy their appetites. Considering the video recommendation is an increasing popular sub-field of recommendation, but the traditional recommendation techniques such as Collaborative Filtering and Content-based model simply exploit one information source that limits its performance. In this paper, we proposed a Multi-info fusion based recommendation system which integrates several different information sources to comprehensively model the similarity between videos. The information sources including the common user-item rating data and video’s textual content that consists of video’s genres and textual description. Experimental results on a public dataset show that the proposed system is of high quality and achieves significant improvements over the traditional Collaborative Filtering techniques.

1. Introduction

The intelligent recommendation system aims to recommend or suggest items or products to the customers by exploiting his/her preferences, item’s content and user-item interaction history [1].

In general, the recommendation models are classified on the basis of different knowledge sources. The Collaborative Filtering (CF) mainly uses the user-item interaction data to capture user or item profiles which are applied to seek the nearest neighbors for user or item. The content-based recommendation system only takes the information of item or features of user into account while hybrid recommendation system exploits the both to make the recommendations for users.

The CF [2][3] has been widely used in many areas including book recommendation, E-commerce product recommendation and so on, but the CF solely rely on the user-item rating data limits the performance of the recommendation, which fails to capture the comprehensive features of users and items. For video recommendation, in addition to the ratings of users to their watched videos, the video itself is with rich genre information, text information which are vital to researchers to depict the features of items. Meanwhile, the various information sources fusion will alleviate the Cold start and rating data sparsity which are common problems in CF model.

In recent years, the deep learning has achieved great success in many domains such as speech recognition, computer vision and natural language processing. Particularly, the text processing techniques changed quickly and overtook the traditional techniques are contributed to the deep learning. [4] adopts the standard Siamese Network to access for semantic similarity between sentences, which is a siamese adaption of the LSTM network for labeled data comprised of pairs of variable-length sequences. In [5], the author proposes an elaborate convolutional network variant
which infers sentences similarity by integrating various difference across many convolutions at various scale. These state-of-art text similarity computing techniques motivates us to process the video’s content and integrates the similarity produces by CF model to acquire more accurate neighbors to improve the video recommendation system.

In this paper, we propose a novel Multi-info fusion model for video recommendation, which combines user-video rating data and video’s content to be more comprehensive to model the similarity between videos. The proposed model exploits the genres and descriptions of videos to compute the content similarity between videos, in this step, we adopt the Doc2Vec technique to map the video’s textual description into distributed representation which extends the skip-gram approach of word2vec from the word to sentence. Then integrates with the similarity computed by the item-based CF model to acquire the integrated similarity between videos. For an active user, we compute the most similar unwatched videos to high rated videos of him/her to make the video recommendation.

The rest of the paper is organized as following. The preliminary works of this paper are described in section 2. Section 3 demonstrates the procedure of our recommendation system in details and section 4 introduces the dataset and the experimental results are analyzed. Finally, conclusions of the study are discussed in section 5.

2. Preliminaries

2.1. Collaborative Filtering

Collaborative Filtering algorithm follows an assumption that if two users rate n items with similar scores or like items in the same category, they will have same ratings or attitudes to other items in a great probability. This association essentially reflects the interests between users and predicts the preference of user to unrated items. CF techniques mainly embrace memory-based and model-based two approaches.

Memory-based CF [6][7][8] approaches firstly calculate the similarity between users (user-based) or items (item-based) based user-item ratings matrix, then make recommendations or predictions based on the similarity values. Commonly, the Pearson correlation coefficient is used to measure the similarity between user i and user j:

$$sim(i, j) = \frac{\sum_{p \in \mathcal{P}} (r_{i,p} - \bar{r}_i)(r_{j,p} - \bar{r}_j)}{\sqrt{\sum_{p \in \mathcal{P}} (r_{i,p} - \bar{r}_i)^2} \sqrt{\sum_{p \in \mathcal{P}} (r_{j,p} - \bar{r}_j)^2}}$$

where $r_{i,p}$ and $r_{j,p}$ respectively indicate user i and user j give rating on item p. $\bar{r}_i$ and $\bar{r}_j$ respectively represent the average rating of user i and user j on all co-rated items. Similarly, we get the similarities between items like above.

Model-based CF [9] predicts the ratings by utilizing the machine learning approaches to recognize the patterns or features of users and items. Unlike the memory-based CF, model-based CF analyzes the meaning behind user-item ratings in detail, and models a rating one user gave to one item at a time as the similarity between the user's and the item's features. As a representative of model-based CF, Matrix Factorization (MF) was proposed by Koren et al. [10], which decomposes the user-item ratings matrix into low-dimensional user latent vectors and item latent vectors to find common factors interpreting ratings given by users.

2.2. Word Embedding

Word Embedding is a general term for a group of language modeling and feature learning techniques in Natural Language Processing (NLP) that maps words or phrases into a vector of real numbers. The Word2Vec [11] proposed by Tomas Mikolov has become a general and effective approach of Word Embedding, it uses the Artificial Neural Network technique to train the vocabulary and acquires the vector representation of words. Inspired by the Word2Vec, Quoc Le and Tomas Mikolov put forward the Doc2Vec [12] that maps each document into a dense vector which is trained to predict words in the
documents. The Doc2Vec model can be trained in two approaches: Distributed Memory version of Paragraph Vector (PV-DM) and Distributed Bag of Words version of Paragraph Vector (PV-DBOW). Figure 1 and Figure 2 demonstrate the framework of PV-DM and PV-DBOW respectively.

In Figure 1, the PV-DM model maps each paragraph or document into a vector which can be represented as a column of Paragraph matrix, each word is also mapped into a column of Word matrix, then computes the average features of the paragraph vector and word vectors to predict the next word. The ID of paragraph remains unchanged to share the same paragraph vector during training this paragraph. Differ from PV-DM, Figure 2 shows that the PV-DBOW model only takes the paragraph vector as the input while the context information are ignored, then let the model predict the words in this paragraph randomly. After the training procedure is terminated, the input paragraph matrix is the learned paragraph embeddings.

3. System Overview

The proposed Multi-info fusion video recommendation system integrates multiple various information sources to model the video profile, which facilitates to compute the video similarity comprehensively. In addition, the system is also designed to address the common Cold start problem and with great scalability. Meanwhile, it provides users with excellent user experience by explaining why the system recommend a list of videos to him/her.

As shown in the Figure 3, the Multi-info fusion recommendation system is comprised of two modules: Collaborative filtering technique takes the user-video rating data as the input to compute the similarity from interaction history between users and videos, text processing technique computes the video content similarity by mining the video information including video genres and textual descriptions. Here are some advantages of our system in design: 1) The decoupled submodules make the system easy to develop and debug. 2) Modularization of task improves the scalability of system, it precomputes the two part of similarity offline which makes the system scale to very large data sets.
3.1. **Word Embedding**

The Multi-info fusion model mainly integrates two types of data sources: the video ratings history and the detailed profiles of videos. However, the video corpora that we crawled from different video websites exist some drawbacks such as information incomplete, data missing and data inconsistency, thereby the quality of raw data is not high.

We first clean the “dirty data” out of raw data and normalize the same dimensional features like video genres such that the system properly computes similarity between video profiles. In detail, we remove the user’s records if he/she watched or rated less than 20 videos. And video profiles are comprised of video’s ID, video’s title, video’s genres and video’s textual description. As for lacking of video’s textual description, we use Web Crawler to crawl the detailed video’s textual description from different video websites or the third-party authoritative data such as Wikipedia to enrich the video profile.

3.2. **Video Content Similarity Computing**

The system computes the video content similarity according to two types of information in video’s profile, one is genres of video and the other is video’s textual description. Assuming $v_i$ and $v_j$ are two videos, genres of two videos are represented by a invariant-length binarized sets $c_i$ and $c_j$, then computing the genre similarity between videos which noted as $sim_{\text{genre}}$ by Jaccard index formula:

$$sim_{\text{genre}} = \frac{c_i \cap c_j}{c_i \cup c_j}$$

For video’s textual description, the system adopts Doc2Vec technique elaborated in section 2.2 to computes the textual description similarity $sim_{\text{text}}$ between videos. Firstly, each video textual description is split into a group of independent words by using Stanford NLP toolkit which is a well-known Word Segmentation tool, noting that those content-irrelevant stop words will be removed. In consideration of Embedding quality, we adopt PV-DBOW model to train the segmented paragraphs. The training procedure is organized by format of paragraph-wise, in one paragraph training step, every word in this paragraph segmented words group is selected as the input to the neural network and the neural network outputs potential context words of the input word. Meanwhile, the number of potential context words is set to 3 or 5 according training experiences. After a large number of training epochs, we acquire a learned matrix that each column in it indicates the fixed-size valued vector of each video textual description. Defining the $ParaV_i$ and $ParaV_j$ indicate learned vectorized representations of $v_i$ and $v_j$, then computes the textual description similarity noted as $sim_{\text{text}}$ by the Cosine Similarity formula:

$$sim_{\text{text}} = \frac{ParaV_i \cdot ParaV_j}{|ParaV_i| \times |ParaV_j|}$$

Then the system merges the above two types of content similarity into a comprehensive content similarity, we have designed a adjust weighted average function for merging similarity:
where the value of \( \alpha \) depends on the importance of genres and textual description to the video content.

3.3. Rating-based Video Similarity

As described in section 2.1, Collaborative Filtering techniques take the user-video rating data as the input to model the similarity between users or videos and recommend videos to active user. In consideration of capture the video’s features, the Multi-info fusion model adopts item-based CF to calculate the similarity between videos.

For video \( v_i \), the system describes it by a fixed-length vector which depends on the total number of users, each element value of the vector is set to user’s rating score if user \( u_k \) has rated this video, else set to 0. Different from Pearson correlation coefficient, we the system adopts a new technique based weighted information entropy to compute the rating-based video similarity, which calculates the weighted information entropy of the rating score difference on the basis of video’s score difference from the same user.

\[
H(v_i, v_j) = -\frac{1}{n} \sum_{k=1}^{n} p(d_k) \log_2 p(d_k) \times |d_k|
\]

where \( p(d_k) \) indicates the probability of k-th element in video’s rating score differential vector. The larger the score difference of videos from same user, the more different the videos are. Thereby the weighted term \( |d_k| \) is introduced into formula. Furthermore, \( n \) represents the number of same users on videos \( v_i \) and \( v_j \). Then the system normalizes the above results such that the similarity value ranges from 0 to 1:

\[
sim_{rat}(v_i, v_j) = \frac{\max(H(v_i, :)) - H(v_i, v_j)}{\max(H(v_i, :)) - \min(H(v_i, :))}
\]

where \( \max(H(v_i, :)) \) and \( \min(H(v_i, :)) \) respectively indicate maximum and minimum of \( H(v_i, v_j) \).

3.4. Similarity Fusion for Recommendation

After computing the video similarity-based rating data and content, the system combines these two similarities of different information sources for video recommendation. The traditional technique of similarity simply computes the weighted sum of two similarity values such that the weight of each part of similarity is set to a fixed value which achieve the optimal metric value. However, it neglects a truth that the importance of user-video rating data is changing with the sparsity of rating matrix. The Multi-info fusion model designs an Adjusted weighted sum technique to adjust the variant importance of rating data, the specific similarity fusion computational formula as follows:

\[
sim_{total}(v_i, v_j) = \frac{\alpha}{1 + e^{-|N-20|}} \cdot sim_{rat}(v_i, v_j) + \left(1 - \frac{\alpha}{1 + e^{-|N-20|}}\right) \cdot sim_{text}(v_i, v_j)
\]

where the \( N \) is the number of common users have rated on both video \( v_i \) and \( v_j \), we see that the larger \( N \) is, the rating-based similarity hold a more important part in total similarity.

The system next recommends videos for users on the basis of computed total videos’ similarity. For an active user \( u_k \), the system firstly finds \( M \) highest score videos rated by \( u_k \), then uses the K-Nearest Neighbors technique to seek \( K \) videos most similar to aforementioned \( M \) highest score videos and recommend to the active user.

4. Experiments and Results

The experiments aim to test whether the Multi-info fusion based video recommendation system proposed in this paper makes improvements in metric, compared to the classical CF recommendation.
system. And fusion of different information sources work will be considered prominent if the results are better than the baselines.

4.1. Similarity Fusion for Recommendation

We evaluated the above-described Multi-info fusion recommendation model on MovieLens ml-latest-small dataset, which is commonly used for evaluating Collaborative Filtering techniques. The dataset contains 100,004 rating records from 671 users on 9066 movies and the rating score ranges from 1 to 5. The side information of movies includes the release date and video’s genres, we crawl the movies’ textual descriptions from the API interfaces of OMDB or IMDb. We split the training data sets into 5 folds for 5-fold cross-validation and test the model on the hold out 20% of the dataset.

4.2. Similarity Fusion for Recommendation

In our experiments, we adopt three different common metrics to evaluate the performance of the proposed recommendation system: The Root Mean Square Error (RMSE), the Precision@K and the Recall@K. RMSE is used to measure the deviation between the predicted ratings and their real rating scores given by users. RMSE\( (u_i) \) for active user \( u_i \) is defined as follows:

\[
RMSE(u_i) = \sqrt{\frac{1}{N} \sum_{j}^{N} (\hat{r}_{i,j} - r_{i,j})^2}
\]

where the \( N \) indicates the total number of ratings of active user \( u_i \) in testing datasets and \( \hat{r}_{i,j} \) represents the predicted score. The RMSE over all users is the average of RMSE of individual active users.

\[
RMSE = \sqrt{\frac{1}{S} \sum_{i=1}^{S} RMSE(u_i)}
\]

where \( S \) represents the total number of users.

Like most recommendation system evaluation, we compute the predicting precision at cut-off \( k \) in the Top-\( N \) list. The Precision@\( K \) is defined as follows:

\[
Precision@K = \frac{\text{number of items that are relevant}}{\text{total number of recommended items}}
\]

The Recall@\( K \) evaluates model’s performance by calculates the recommendation coverage rate of top \( K \) movies.

\[
Recall@K = \frac{\text{number of items that are relevant}}{\text{Top K related items}}
\]

4.3. Similarity Fusion for Recommendation

We begin by splitting the training data sets into 5 folds for 5-fold cross-validation and test the model on the hold out 20% of the dataset. During the Doc2Vec model training procedure, we set 100-dimensional length of vector space which represents the movie textual descriptions. Then the constant \( d \) in equation (4) is set to 0.1 and the parameter \( \alpha \) in equation (7) is set to 0.8 which reflects the high importance rating history. We set the \( K \) to 5, 10, 15, 20 respectively to compare our system with baselines at precision metric. Furthermore, we also set an extra parameter \( C \) in order to guarantee that the system provides high-quality recommendations.

We select three popular recommending algorithms in the following to justify the effectiveness of our proposed Multi-info fusion recommendation model:

- User-CF: User-based Collaborative Filtering calculates the user-user similarity by modeling the users' ratings as their appetites or interests,
• Item-CF: Item-based Collaborative Filtering calculates the item-item similarity to seek similar items for recommending.
• MF: Matrix Factorization is a model learning the latent factor vectors of users and items by factorizing the user-item ratings matrix.

4.4. Similarity Fusion for Recommendation
Table 1 shows the RMSE recommendation performance of our Multi-info fusion recommendation model and four baselines, Figure 4 and Figure 5 respectively illustrate the Precision@K and the Recall@K performance of our model and baselines at K = 5, 10, 15, 20. Noting that we set the parameter C = 3.5 to make sure that the recommending videos’ predicted rating score are equal or greater than 3.5 which guarantee the high-quality recommending videos. Furthermore, the four different lines in Figure 4 and Figure 5 represent four comparative experiments respectively.

Table 1. RMSE for Multi-info model and baselines.

| benchmark | RMSE  |
|-----------|-------|
| Multi-info| 0.9183|
| User-CF   | 0.9986|
| Item-CF   | 0.9893|
| MF        | 0.9460|

Figure 4. Precision@K of benchmarks on test datasets.

Figure 5. Recall@K of benchmarks on test datasets.

For proving our multiple information sources fusion work is prominent to recommendation performance, the RMSE of Multi-info should be less than other baselines in Table 1 and the Multi-info line in Figure 4 and Figure 5 should be higher than other lines.

In Table 1 we see that our Multi-info fusion model outperforms other baselines especially improve the user-based CF by 8.03% and achieved 2.9% improvement compared with MF (model based CF). Figure 4 shows that our model also achieve the best precision of recommendation compared with other baselines. Our model does not get the highest recall at K=5 in Figure 5, but with the growth of K value, we finally get the best recall at k=20.

4.5. Analysis
The experiments show that our Multi-info fusion model achieves the best performance compared other baselines whatever in RMSE, Precision@K or Recall@K metric on the whole. As the representative of model based CF, MF behaves better than both User-CF and Item-CF, however our model overwhelms MF in all the above metrics, which explains the importance of multiple accurate information sources, integrating more high-quality data sources will help us accurately model the features of users and videos.
On the other hand, our Multi-info fusion model and Item-CF model almost have the same logic for recommendation except the data used for training model. Item-CF only exploits the user-video rating history to model the similarity between videos while our model adopts Doc2Vec technique to analyze the textual descriptions and integrate with genre information to enrich the calculations of video’s similarity. However, results show that our model behaves better than Item-CF in the above three metrics, especially we achieve 7.17% improvements in RMSE. It suggests that the user-video rating history limits the performance of Item-CF, one prominent disadvantage is the sparsity of rating matrix, but our model integrating multiple data sources could solve this problem well.

Note that our model acquires the best performance over other baselines by just mining the textual information of video including video’s genre and textual description, while there is abundant information about users are not used, thus integrating more high-quality information sources to model user feature would be a good choice, we will try to apply more valuable information sources to model the user-user similarity.

5. Conclusion
In this work we propose a recommendation system based on multiple information sources fusion. The system first uses the Doc2Vec technique to extract the video’s feature from textual description and integrates with video’s genre information to compute the content similarity between videos. Then we creatively proposed a similarity fusion computational formula to combine content similarity and similarity from rating data. Finally, the experiments show that our multiple information fusion work of the system is effective, and indicate that we can further improve the system by exploiting more side information of user to model user’s features.

References
[1] Ziesemer A, Batista J and Oliveira S D 2011 How to Know What Do You Want? A Survey of Recommender Systems and the Next Generation Proc. 8th Brazilian Symp. on Collaborative Systems pp 105-11
[2] Su X and Khoshgoftaar T M 2009 A Survey of Collaborative Filtering Techniques Advances in Artificial Intelligence pp 1-19
[3] Wei J, He J and Chen K 2017 Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items Expert Systems with Applications pp 29-39
[4] Thyagarajan A 2016 Siamese Recurrent Architectures for Learning Sentence Similarity 13th AAAI Conf. on Artificial Intelligence (AAAI Press)
[5] He H, Gimpel K, and Lin J 2015 Multi-perspective sentence similarity modeling with convolutional neural networks Proc. of 2015 Conf. on Empirical Methods in Natural Language Processing pp 1576-86
[6] Cacheda F, Carneiro V and Formoso V 2011 Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems ACM Trans Web pp 1-33
[7] Sunitha M and Adilakshmi T 2018 Music Recommendation System with User-based and Item-based Collaborative Filtering Techniques Indonesian. J. EE. Infs. pp 59-68
[8] Deng A L, Zhu Y Y and Shi B L 2003 A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction China. J. Software pp 54-65
[9] Miyahara K and Pazzani M J 2000 Collaborative Filtering with the Simple Bayesian Classifier Proc. 6th Pacific Rim Int. Conf. on Artificial Intelligence pp 679-89
[10] Koren Y, Bell R and Volinsky C 2009 Matrix Factorization Techniques for Recommender Systems Computer pp 30-7
[11] Mikolov T, Chen K and Corrado G 2013 Efficient Estimation of Word Representations in Vector Space Computer Science. arXiv
[12] Le Q V and Mikolov T 2014 Distributed Representations of Sentences and Documents Computer Science. arXiv