Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 9-12 July, 2017

FULLY CONTENT-BASED MOVIE RECOMMENDER SYSTEM WITH FEATURE EXTRACTION USING NEURAL NETWORK

HUNG-WEI CHEN[1], YI-LEH WU[1], MAW-KAE HOR[2], CHENG-YUAN TANG[3]

[1]Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[2]School of Informatics, Kainan University, Taoyuan, Taiwan
[3]Department of Information Management, Huafan University, New Taipei, Taiwan

Abstract:

In recent years, movie industry is getting more and more prosperous. There are hundreds of movies released every year. However, it is difficult to notice the releasing of every movie, not to mention actually seeing it. Therefore, movie recommender system has become more and more popular as a research topic. Among a variety of movie recommender systems, content-based methods always ring a bell when it comes to recommending new movies. Content-based method uses the content of the movie as input so that it does not suffer from the "cold-start" problem.

In this paper, we propose the Fully Content-based Movie Recommender System (FCMR) to recommend movies to users. The proposed method trains a neural network model, Word2Vec CBOW, with content information (e.g., cast, crew, etc.) as the training data to obtain vector form features of each element, and then take advantage of the linear relationship of learned feature to calculate the similarity between each movie. In the end, the proposed FCMR recommends movies based on the similarity. The experiments are conducted on a massive real world dataset, and the intuition behind our proposed method has been proven by the experiment results.

Keywords:

Recommender System; Content-based; Neural Network; Feature Extraction

1. Introduction

1.1 Motivation

In recent year, the movie industry has become more and more prosperous, there are hundreds of movies released every year, which means that it is hard to find a movie in which you may be interested. As a result, the movie recommendation has become a popular research topic. There already exist many applications of recommender system in our life, such as music, books, movie, and it can even recommend suitable friends for users on the social network.

Most movie recommender systems rely on the collaborative filter (CF) to recommend movies which users may be interested in. The CF can predict user preferences by analyzing user's browsing history and other users' preferences. However, the CF methods suffer from the cold-start problem: it fails when no usage data of items is available. Another category of methods named as the content-based methods do not have such drawbacks. The content-based methods usually use additional information about movies as input so the system treats new movies just like the old ones. Despite the performance of the CF methods is better than the content-based methods in most cases, the content-based methods are still very crucial to recommender systems when it comes to cold-start problem.

Nowadays, when people want to see a movie, most of them check the plots or others' reviews. Therefore, some content-based methods use these two data as the input of their proposed model to capture the preferences of people as much as precisely by making the trained model think just like people. However, the cast, crew and genre of movies sometimes are also considered while choosing movies. Furthermore, some may be fond of the movie released in certain years so they take the release year of movies into account. Even the number of awards and nominations of movies may affect decisions of people. Despite the fact that there are so many factors may influence how people select movies, most of them are not usually considered when speaking of movie recommendation research.

In order to confirm the statement claimed above, we propose a new method which called "Fully Content-based Movie Recommender System (FCMR)" to recommend movies in content-based way completely using only content information (e.g., Directors, Actors, Genres, etc.) as our training data. The proposed method takes advantage of the neural network model, Word2Vec CBOW, to extract feature vectors and the similarity between movies can be calculated based on the extracted features. In the end, the system recommends movies to users according to the similarity between movies. The experiments are conducted on a

978-1-5386-0408-3/17/$31.00 ©2017 IEEE

massive real world dataset Movielens-20M, and the intuition behind our proposed method has been proven by the experiment results.

1.2 Related Work

Collaborative filtering (CF) [6, 7, 8] is a common method to deal with recommendation problems. It can predict users' preferences based on other users' preferences which can be obtained by analyzing the user's browsing history or item ratings given by the user, and then recommend item according to preferences similarity. For instance, if user A and B have similar preferences, then items liked by A but not yet considered by B will be recommended to B. The state-of-the-art methods for performing CF are based on matrix factorization (MF), which is well summarized by [9]

Content-based method is another popular way to solve the recommendation problem. Contrary to CF which is mostly using user-item (or user-user, item-item) matrix as input, this kind of method takes some information which can descript the characteristic of item as its training data. Some content-based movie recommender systems use metadata of movie (e.g., plot, reviews, cast, crew and genres) as its input [10, 11] and some take advantage of the audio and visual feature [12]. The performance of the content-based methods is poorer than the CF methods in general, but the content-based methods do not suffer the cold-start problem like the CF methods. Recently, content-based methods usually combine collaborative filtering method because it is much easier to improve performance generally and overcome the cold-start problem than only using content information.

1.3 Contributions

The main contributions in this study are listed below:

1. The proposed method recommends movies in a fully content-based way, which means that no user browsing history is necessary while training models.
2. The proposed method is the first method to use the Word2Vec CBOW model to extract feature vectors from textual data of movie content and then apply the extracted feature vectors to the movie recommender system.
3. The experiments on a massive movie dataset prove the correctness of the intuitions behind our proposed method.

The study is organized as follow: Section 2 describes the details of Word2Vec model. Section 3 describes the MovieLens-20M dataset, the preprocessing and our proposed method "Fully Content-based Movie Recommender System". Section 4 describes the evaluation of our proposed system. Finally, Section 5 is the conclusions and future work of this whole study

2. Word2Vec Model

Recently, deep learning has been the most popular research topic in the data mining domain due to its fabulous performance. Some recommender system research [21, 21, 22] has pointed out that the performance of recommender system can be improved by using the features extracted by a proper deep neural network model.

2.1 Model Architecture

The Word2Vec Model is an open source neural network model proposed in [1]. It is designed for training textual data and extracting a set of vector features corresponded to every single word in the input dataset. There are two different models proposed in [1], the Continuous Bag-of-words (CBOW) and the Continuous Skip-gram (Skip-gram). These models have different architecture and functionality respectively.

The CBOW model is capable of predicting the current word based on the context. The architecture of CBOW model consists of input, projection and output layer. It takes the sentences of documents as training data. Every sentence is told apart into words and each single word is corresponded to its own word vector. At the input layer, the CBOW model uses the context of the word as input. The context is converted into vector form. The input layer is then projected to the projection layer using the projection matrix. In the end, the sum of all the projected word vectors of context would be the output, the word vector of the predicted word.

The Skip-gram model uses each current word as an input to a log-linear classifier with projection layer, and predicts words within a certain range before and after the current word. The range is a random number and it changes for each training word.

The Similarity between words can be measured easily by calculating the cosine distance between two word vectors. Furthermore, the extracted features can keep the semantically linear relationship between the original words.

3. Proposed Method

3.1 Dataset and Preprocessing

We use the MovieLens-20M [23] dataset for exper

iments. The MovieLens-20M dataset contains about 27 K movies, 138K users, and 20M ratings. However, the original dataset does not include any content informatio n of the movies so we must access the content data o f movies from other sources like IMDb or Wikipedia.

The Open Movie Database (OMDb) provides the APIs to access the data of movies in the IMDb. Among the 27K movies in the MovieLens-20M dataset, we retrieve about 23K movies' data from the IMDb through the APIs. The crawled data of movies are well-organized JSON files and these data are composed of a variety of information including directors, actors, writers, genres, ratings, etc.

To generate a dataset which Word2Vec model can use as input, the movie data must be represented in sentence form. We treat part of the movie metadata as words and compose them into a movie sentence that can describe the movie itself. Furthermore, the occurrence of movie sentences are their own rounding IMDb rating which is included in the JSON data of movies. In the end, we denote this collection of movie sentence as "Movie Sentence Dataset" in the following paragraphs.

3.2 Feature Extraction

The feature extraction of our proposed method is based on a trained word2vec CBOW model which uses the Weighted Movie Sentence dataset as the training dataset. The feature vectors are available while the model is well-trained.

The reason why we choose the CBOW model instead of the Skip-gram model is that we conjecture predicting the current word (e.g., an actor) based on its context (all the other words in the movie sentence) is more reasonable and simpler than predicting the surrounding words given the current word. Furthermore, our dataset, the Weighted Movie Sentence Dataset, is much smaller than the dataset used in [1]. We are not sure if the Skip-gram model will work with relatively small dataset. Therefore, we choose the CBOW model to extract feature in our proposed method.

Most of the parameters used are default value except for the window size. It is set to be a relatively larger value so that the proposed model can take the entire movie sentence as context while training a word of the sentence.

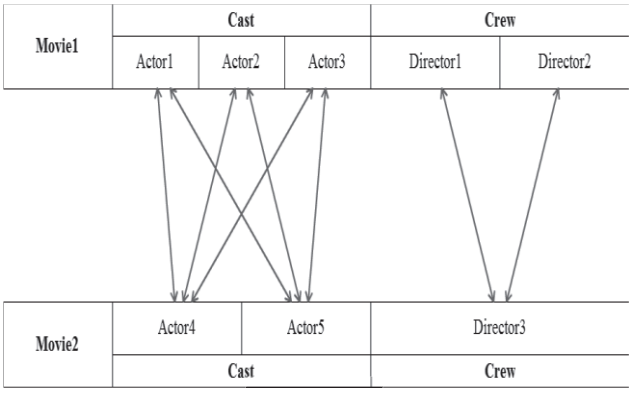
3.3 Similarity Measurement

The extracted word vector can keep the semantically linear relationship between two words. Furthermore, the semantical similarity between two words can be measured by the cosine similarity between two word vectors. In other words, the similarity between two movie sentences can be represented as a combination of cosine similarity between word pairs from the movie sentences. We propose a method to measure the similarity between two movie using movie sentences. The detail of measuring similarity between two movies is shown below as in Table 1.

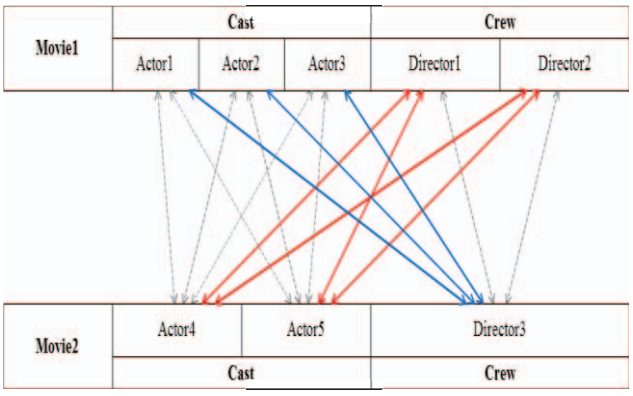
TABLE 1. The process to measure similarity between movies

Measuring the similarity between movies	
1	For every distinct pair of movie sentence
2	Turn words in sentence into word vector
3	For every distinct pair of word vector
4	Calculate cosine similarity between word vector
5	Add up the chosen cosine similarity
6	Averaged sum is the similarity between two movies

In the process of measuring the similarity, there are two adjustable details which may affect the performance of proposed method. We will introduce these variations in the following paragraphs.



(a)



(b)

FIGURE 1. (a) Same Metadata Only, (b) Fully-Connected

The first adjustable detail is how to choose the cosine similarity. We propose two different ways to select which cosine similarity should be added into the sum. One is named as the Same Metadata Only (SMO), which means that only the cosine similarity belongs to same type of metadata is covered, as shown in Figure 4(a). Another one is called the Fully-Connected, which means that every cosine similarity is covered, as shown in Figure 4(b).

The second adjustable detail is how to average the sum of cosine similarity. We proposed two ways to average the sum. One is called the Total Average and the other is named as the Metadata-Based (MB).

The Total Average adds up all the cosine similarities and treats the averaged sum as the similarity between two movie sentences. The formula of the Total Average is shown below.

$$Total\ Average\ Similarity_{movie_{sentences}} = \frac{\sum cosine\ similarity_{word_{pair}}}{number\ of\ word\ pairs}$$

However, even a movie is very similar to another one in the aspect of director, the Total Average may lead that both are not recommended to each other due to other lower cosine similarities between different pair of metadata (e.g., actors-actors, genres-year, etc.).

To overcome the drawback of the Total Average, we propose the Metadata-Based (MB).The MB add up the cosine similarity according to metadata type of word pair and the averaged sum is denoted as similarity of metadata. In the end, the MB adds up every similarity of metadata and treats the sum as the similarity of two movie sentences. The formula of the Metadata-Based (MB) is shown below.

$$Metadata-Based\ Similarity_{movie\_sentences} = \sum Similarity_{metadata\_pair}$$

$$Similarity_{metadata\_pair} = \frac{\sum cosine\ similarity_{word\_pair\ in\ same\ metadta\_pair}}{number\ of\ word\ pair\ in\ same\ metadta\ pair}$$

3.4 Recommendation List Generation

After measuring the similarity between movies, our proposed system generates a recommendation list to every movie in dataset according to the similarity between movies. The detail of recommendation list generation is shown below in Table 2.

TABLE 2. The process to generate recommendation list

Generating recommendation lists of movies	
1	For every single movie
2	Rank similarity of other movies in descending order
3	The ranking is the recommendation list of movie

It is worth mentioning that the recommendation lists generated by our proposed method are for movies not users, which means that the list itself does not change no matter who uses the list. However, we still take this property as superiority despite the lack of personalized recommendation. Our method only needs a movie which user is interested in then we can start to recommend user other movies. Because of this property, our system works even in the new-movie scenario.

4. Experiments

4.1 Measurement

We use the Precision@K to show the performance of our system in top k recommendation; a high precision with lower K will be a better system. We defined the Precision@K as follow:

$$\text{Precision@K} = \frac{\text{number of items the user had seen in top K}}{\text{total number of prediciton}}$$

4.2 Baseline and Comparisons

A baseline and several comparisons are proposed to find out the best setup. The variations are listed below:

- Two different ways to calculate similarity, the Fully-Connected (FC) and the Same Metadata Only (SMO)
- Two different ways to average the similarity, the Total Average (TA) and the Metadata-Based (MB)
- Different feature sets are also considered.

The baseline and comparisons are shown in Table 3.

TABLE 3. The baseline and comparison in our experiments. (D, A, G means Director, Actor, Genre and * is the baseline)

Name	Feature	FC/SMO	TA/MB
DAG_F_T*	D, A, G	FC	TA

DAG_S_T	D, A, G	SMO	TA
DAG_F_M	D, A, G	FC	MB
DAG_S_M	D, A, G	SMO	MB
DAGY_F_M	D, A, G, Year	FC	MB
DAGY_S_M	D, A, G ,Y	SMO	MB

4.3 Result

To the best of our knowledge, the DAGY_S_M is the best combination we proposed with the best performance in our experiments. Table 4 shows the performance comparisons.

TABLE 4. The performance of baseline (Precision@K)

	1	**10**	**100**	**200**
DAG_F_T*	4.57%	3.95%	2.92%	2.74%
DAG_S_T	7.64%	5.03%	3.33%	3.10%
DAG_F_M	10.19%	7.10%	4.39%	3.95%
DAG_S_M	13.68%	8.02%	4.70%	4.15%
DAGY_F_M	8.69%	6.49%	4.53%	4.10%
DAGY_S_M	**15.07%**	**7.98%**	**4.67%**	**4.16%**

5. Conclusions

In this paper, we propose the Fully Content-based Movie Recommender System. The proposed system only uses the content data of movie as the training dataset. Furthermore, the proposed method takes advantages of the Word2Vec CBOW Model to extract features from the content of movies and transform the textual content data into feature vectors which can keep linear relationship semantically. We conduct the evaluation with real-world massive movielens-20M dataset. The dataset contains 138K users and 20M browsing histories. The result of experiments supports the intuition behind our proposed method and the comparison between baselines also show that our proposed variation of process details successfully simulate how the user thinks while choosing movies.

Future work includes that try to use more metadata (e.g., writer, camera crew, etc.) as input feature and use different ratings (e.g., Rotten tomatoes or Metacritic) to weight the dataset instead of IMDb ratings. The ratings of Metacritic and Rotten tomatoes are given by professional movie critic so the model may be able to simulate the performance of using award and nomination of the movie as weighting approach. Moreover, combining the proposed process to the collaborative filtering is also favorable. We will try to treat the browsing history of a user as a sentence and use the sentences to train a word2vec model. Last but not least, combining the CF and the Content-based method into a hybrid model is a promising way to improve the performance.

References

[1] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[2] Wang, H., Wang, N., & Yeung, D. Y. (2015, August). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1235-1244), 2015.

[3] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in Neural Information Processing Systems* (pp. 2643-2651), 2013.

[4] Elkahky, A. M., Song, Y., & He, X. (2015, May). A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 278-288), 2015.

[5] MovieLens-20m dataset, http://grouplens.org/datasets/, referenced on April 28th, 2016.

[6] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 230-237), 1999.

[7] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. *In Proceedings of the 10th international conference* on World Wide Web (pp. 285-295), 2001.

[8] Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000, December). Explaining collaborative filtering recommendations. *In Proceedings of the 2000 ACM conference on Computer supported cooperative work* (pp. 241-250), 2000.

[9] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30-37, 2009.

[10] Diao, Q., Qiu, M., Wu, C. Y., MBla, A. J., Jiang, J., & Wang, C. (2014, August). Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 193-202), 2014.

[11] Uluyagmur, M., Cataltepe, Z., & Tayfur, E. (2012). Content-based movie recommendation using different feature sets. *In Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1, pp. 17-24), 2012.

[12] Lehinevych, T., Kokkinis-Ntrenis, N., Siantikos, G., Dogruoz, A. S., Giannakopoulos, T., & Konst antopoulos, S. (2014, November). Discovering sim ilarities for content-based recommendation and br owsing in multimedia collections. *In Signal-Image Technology and Internet-Based Systems (SITIS)*, 2014 *Tenth International Conference* on (pp. 237-243), 2014.