A Content-Based Deep Hybrid Approach with Segmented Max-Pooling

Dapeng Zhang, Liu Yajun, Jiancheng Liu

A Content-based Deep Hybrid Approach with Segmented Max-pooling

ZhangDapeng[1], LiuYajun[2], LiuJiancheng[1]

[1] The College of Information Science and Engineering,
Yanshan University, Qinhuangdao 066004, China
[2] Hebei Institute of Architectural Engineering Information Engineering Institute,
Zhangjiakou, 075000, China
459817216@qq.com

Abstract. Convolutional matrix factorization (ConvMF), which integrates convolutional neural network(CNN) into probabilistic matrix factorization(PMF), has been recently proposed to utilize the contextual information and achieve higher rating prediction accuracy of model-based collaborative filtering (CF) recommender systems. While ConvMF uses max-pooling, which may lose the feature's location and frequency information. In order to solve this problem, a novel approach with segmented max-pooling(ConvMF-S) has been proposed in this paper. ConvMF-S can extract multiple features and keep their location and frequency information. Experiments show that the rating prediction accuracy has been improved.

Keywords: ConvMF, CNN, PMF, Max-pooling.

1 Introduction

Recommender systems have drawn more and more attention in the last decade. They can help people get useful information from "the ocean of information", and can be found in many fields of our life. For example, Alibaba and Amazon use recommender systems to recommend products to their users in their e-commerce platforms. Facebook and Tencent Weibo apply recommender systems in their social networks.

Collaborative filtering(CF) is one of the main methods to build recommender systems[1]. Recently, combined with CF, there are more and more efforts to apply deep learning in recommender systems[2-8]. Due to the exploding growth of the number of users and items, the sparseness of relationships between users and items can be extremely high, which deteriorates the prediction accuracy of the CF recommender systems. In order to alleviate this problem, auxiliary information such as description documents of items, which are easily available from various sources, have been utilized to enhance the rating prediction accuracy. Especially, convolutional neural network(CNN) has been integrated into probabilistic matrix factorization(PMF) to develop convolutional matrix factorization model (Con-

vMF).

Convolution and pooling are of the most important stages in CNN. And max-pooling is the most common sub-sampling operation of pooling layer. It only keeps the maximum feature from each feature vector obtained from convolution layer, which has the following disadvantage: (1) The location information of the features is totally lost. In fact, the location information is kept in convolution layer.(2) Sometimes, certain features may appear frequently. The more frequently it appears, the stronger it is. But max-pooling also loses this frequency information.

In order to address this problem, we propose a new approach with segmented max-pooling, which is called ConvMF-S to improve the ConvMF.

2 Related Work

The great success achieved by convolutional neural network in computer vision has inspired the recent effort to apply deep learning method in NLP. Since 2014, significant work in this field have been published.

Kalchbrenner[9] has proposed a CNN model for sentence modeling, which uses dynamic k-max pooling as a global pooling operation over linear sequences. Besides, he[10] has also proposed an extended CNN for processing sequences. The resulting network has two core properties: it runs in time that is linear in the length of the sequences and it sidesteps the need for excessive memorization, which can solve the problem that the pooling layer may lose some information (whether the information is useful or useless). Chen[11] has proposed a CNN model for event extraction, which uses a dynamic multi-pooling layer according to event triggers and arguments to reserve more crucial information. Lei[12] has proposed a non-linear discontinuous CNN for text modeling, which nonlinearly transforms the convolutional layer. The multi-column CNN model introduced by Dong[13] uses multiple columns of CNN to learn the representations of different aspects of questions. Ma[14] exploits various long-distance relationships between words, and presents a dependency-based convolution framework. Johnson[15] studies CNN on text categorization, the author directly applies CNN to high-dimensional text data, which leads to directly learning embedding of small text regions for use in classification.

More recently, CNNs have also been applied in recommender systems. Several hybrid methods have been proposed for recommender systems that utilize auxiliary information, particularly, the reviews and abstracts of items. Kim[16] has presented ConvMF, a robust document context-aware hybrid method which seamlessly integrates CNN into probabilistic matrix factorization(PMF) in order to capture contextual information in description documents for the rating prediction while considering Gaussian noise differently through using the statistics of items. While its max-pooling layer extracts only the maximum contextual feature from each contextual feature vector. So the information of feature strength is lost. Meanwhile, the location that feature appears is also important, which is also ignored in ConvMF. In order to address the former limitation of ConvMF, we pro-

pose an approach with segmented max-pooling, which can keep multiple features when pooling and reflect the location information of features.

3 Improved Convolutional Matrix Factorization:ConvMF-s

3.1 Convolutional Matrix Factorization

In essence, CNN is a classifier because its object is to address classification task, such as image recognition, label predicting for words, phrases or documents. While the object of recommender is a regressive task. So traditional CNN is not suitable for recommender tasks.

Convolutional matrix factorization can address the above issue through seamlessly integrating CNN into PMF. The probabilistic model of ConvMF is shown in figure 1.

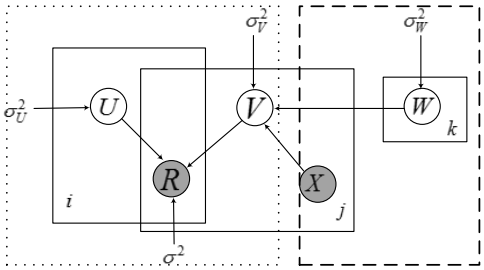

Fig. 1. Probabilistic model of ConvMF

The left dotted part is PMF and the right dashed part is CNN. Suppose we have N users and M items, and observed ratings are represented by $\mathrm{R} \in \mathrm{R}^{\mathrm{N*M}}$ matrix. Then the conditional distribution over observed ratings is given by formular 1.

$$P(R \mid U, V, \sigma^2) = \prod_{i}^{N} \prod_{j}^{M} N(r_{ij} \mid u_i^T v_j, \sigma^2)^{I_{ij}} \tag{1}$$

Figure 2 illustrates the CNN architecture for ConvMF, which is composed of four layers: embedding layer, convolution layer, pooling layer and output layer.

(1)Embedding layer

The object of the embedding layer is to transform a raw document into a dense numeric matrix for the convolution layer. The document matrix $D \in R^{p*l}$ can be represented by:

$$D = [\cdots w_{\mathrm{i-1}}, w_{\mathrm{i}}, w_{\mathrm{i+1}} \cdots] \tag{2}$$

where *l* is the length of the document, and *p* is the size of embedding dimension for each word *w*.

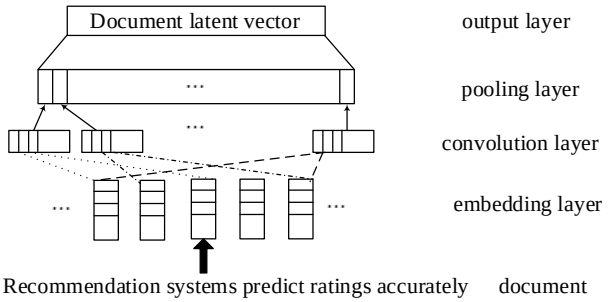

Fig. 2. CNN architecture for ConvMF

(2)Convolution layer

The convolution layer is responsible for extracting contextual features. A contextual feature $c_i^j \in R$ is extracted by *j*th shared weight $W_c^j \in R^{p*ws}$ whose window size *ws* determines the number of surrounding words:

$$c_i^j = f(\mathrm{W}_\mathrm{c}^j * \mathrm{D}_{(:,i:(i+ws-1))} + \mathrm{b}_\mathrm{c}^j) \tag{3}$$

where $*$ is a convolution operator, $b_c^j \in R$ is a bias for W_c^j and f is a non-linear activation function. Then, a contextual feature vector $\mathrm{c}^\mathrm{j} \in R^{l-ws+1}$ of a document with W_c^j is constructed by:

$$c^j = [c_1^j, c_2^j, \cdots, c_i^j, \cdots, c_{l-ws+1}^j] \tag{4}$$

(3)Pooling layer

The pooling layer extracts representative features from the convolution layer, and also deals with variable lengths of documents via pooling operation that constructs a fixed-length feature vector. Max-pooling is utilized here to reduce the representation of a document into a fixed-length vector. The maximum contextual feature from each contextual feature vector can be expressed as:

$$d_f = [max(c^1), max(c^2), \ldots, max(c^j), \ldots, max(c^{n_c})] \tag{5}$$

(4)Output layer

High-level features obtained from the previous layer could be converted at output layer. The produced document latent vector can be expressed as:

$$s = \tanh(W_{f2}\{\tanh(W_{f1}d_f + b_{f1})\} + b_{f2}) \tag{6}$$

where $W_{f1} \in R^{f*n_c}$, $W_{f2} \in R^{k*f}$ are projection matrices, and $b_{f1} \in R^f$, $b_{f2} \in R^k$ are bias vectors for W_{f1}, W_{f2}.

Finally, latent vectors of each document are returned as output:

$$s_j = cnn(W, X_j) \tag{7}$$

where W denotes all the weight and bias variables and X_j denotes a raw docu-

ment of item j, and s_j denotes a document latent vector of item j.

3.2 Improved ConvMF with Segmented Max-pooling(ConvMF-S)

Convolution and pooling are of the most important stages in CNN. And max-pooling is the most common sub-sampling operation of pooling layer. It only keeps the maximum feature from each feature vector obtained from convolution layer. One of the advantage of max-pooling is that it can reduce the number of the features to enhance performance and it can also keep the length of the feature vectors the same which makes it easy to construct the following layers. The architecture of max-pooling is shown in figure 3.

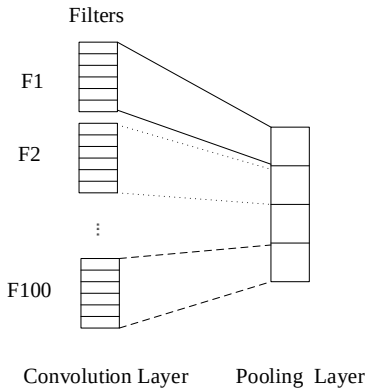

Fig. 3. Architecture of max-pooling

The disadvantage of max-pooling has been stated in section 1. In order to deal with this problem, we propose a new approach with segmented max-pooling, which is called ConvMF-S to improve the ConvMF. It divides each feature vector obtained from convolution layer into segments as required and extracts the maximum value from each segments. The architecture of segmented max-pooling is shown in figure 4.

In ConvMF-S, the embedding layer, convolution layer and output layer are the same with ConvMF. The only improvement is in pooling layer, which is described as follows.

Suppose W_c^j is the weight matrix and c_i^j is a contextual feature extracted by the *j*th filter in convolution layer. The length of the document is *l*. Processed by the convolution layer, a document is represented as n_c contextual feature vectors, and each contextual feature vector has variable length, which is represented by $l-ws+1$.

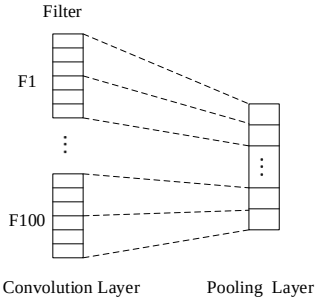

Fig. 4. Architecture of segmented max-pooling

In order to keep more information in pooling layer, we need to divide each contextual feature vector into segments, and extract the maximum contextual feature from these segments. If a contextual feature is divided into *s* segments, the length of each segmented contextual is represented by Eqn:

$$n = \left\lceil \frac{l - ws + 1}{s + 1} \right\rceil \tag{8}$$

Then the fixed-length contextual feature vector is converted by extracting maximun contextual features from *s* segments:

$$d_f = [s^1, s^2, \ldots, s^j, \ldots, s^{n_c}] \tag{9}$$

Where:

$$s^j = \left[\max\left(c_1^j, \cdots, c_n^j\right), \max\left(c_{n+1}^j, \cdots, c_{2n}^j\right) \cdots \max(c_{(s-1)*n+1}^j, \cdots, c_{sn}^j) \right] \tag{10}$$

3.3 ConvMF-S Algorithm

Integrating CNN into PMF, our ConvMF-S algorithm can be described as follows.

Table 1. ConvMF-S Algorithm

Input: R : user-item rating matrix, X: description documents of items
1: Embed the one-hot encoded word vectors to generate word sequence $D \in \mathbb{R}^{p*l}$ from X. 2: Process D with filters of three different window size(3,4,5) to extract contextual feature $c^j \in R^{l-ws+1}$. 3: For each $c^j \in R^{l-ws+1}$, extracts feature values using segmented max-pooling to create the contextual feature vector d_f . 4: Flatten the pooling results to make it to be one-dimensional . 5: Output the document latent vectors. 6: Use the document latent vectors as the mean of Gaussian noise of an item to initialize the item feature matrix . 7: Initialize the user feature matrix and fit the rating matrix R with item feature matrix. 8: Output the result of RSME.

4 Experiments

In this section, we evaluate the performance of ConvMF-S algorithm compared with PMF and ConvMF.

4.1 Experimental Environment and Datasets

We use ml-100k dataset obtained from Movielens, which contains 100,000 ratings on 1682 movies from 943 users. And we randomly divide it into training set(80%), validation set(10%) and test set(10%).

We also obtain documents of corresponding items from IMDB. The obtained documents are preprocessed as follows: (1) Set maximum length of input documents to 300; (2) Remove stop words; (3) Calculated TF-IDF score for each word; (4) Remove corpus-specific stop words of which the document frequency are higher than 0.5; (5) Select top 8000 distinct words as a vocabulary; (6) Remove all non-vocabulary words from input documents.

4.2 Word Vectors Pre-training with Word2vec

One of the most critical issues of contextual-based deep hybrid recommender systems is how to utilize text data more efficiently to generate high-quality features. This involves text analysis tasks in NLP. Therefore, Our word embedding vectors are initialized with word2vec[17], a very popular pre-trained word embedding model. And we pre-train our word vectors on IMDB, which contains 50000 labeled comments and 50000 unlabeled comments.

Each comment in IMDB is kept as a single file. So we merge these comments as a dataset. The format of the merged dataset is shown in table 2.

Table 2. Format of the Merged Dataset

line-number	id	sentiment	review
0	5814_8	1	"With all this…kay. Visually….
1	2381_9	1	"\"The Classic War of the Worlds\" by…
2	7759_3	0	"The film starts with a manager(Nichola…
3	3630_4	0	"Superbly trashy and wondrously …

The field id in table 1 represents the file name of the comment. Left side of the underline is the movie ID, and right side is the rating of the movie from user. The contents of review are processed by removing HTML labels, punctuations and numbers, transforming them into lowercase, splitting them into individual word and rejecting repeated words.

4.3 Experimental Results

In our experiments, RMSE(Root Mean Squared Error) is adopted as the evaluation measure, which is related to the objective functions of prediction models.

Firstly, we compare the performance of these three algorithms based on the numbers of iterations, which is illustrated in figure 5.

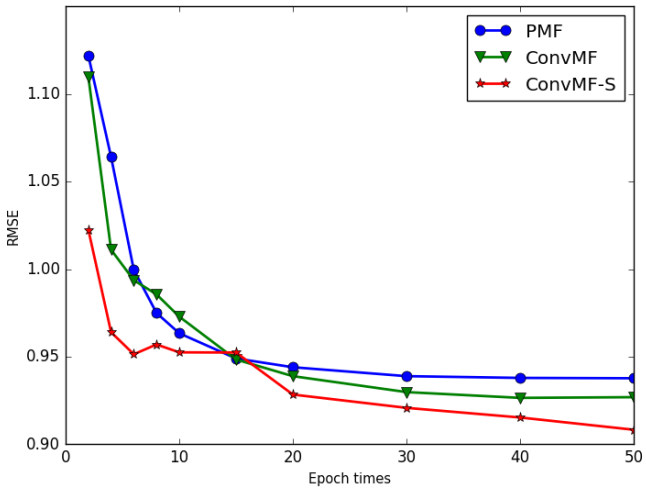

Fig. 5. Comparison of numbers of iterations

From figure 5, it can be seen that PMF converges quickly during the first 15 iterations, the RMSE value tends to be stable after 15th iterations. ConvMF converges quickly during the first 20 iterations. After the 20^{th} iteration, the model is still converging, but the convergence speed is slowed down. ConvMF tends to be stable and the RMSE value does not change when the number of iterations exceeds 30. ConvMF-S is superior to PMF and ConvMF at the beginning of the model training, indicating that the segmented max-pooling effectively improves CNN's ability to analyze document data. ConvMF-S's final iterative result is also superior to the other two algorithms which further proves that the improved method can effectively improve the recommender quality.

Secondly, we compare the performance of the three algorithms on training sets with different percentages(20%,40%,60% and 80%). The result is shown in figure 6.

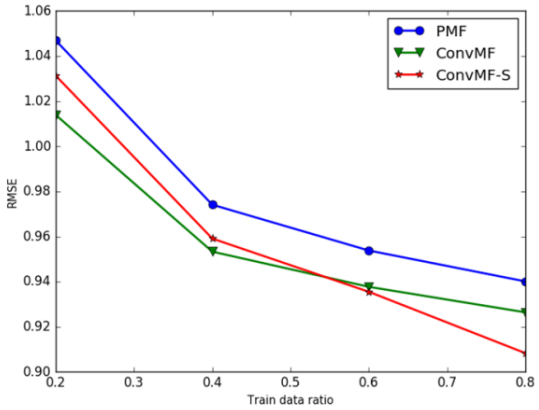

Fig. 6. Comparison on training sets with different percentages

From figure 6, it can be seen that the RMSE values of the three algorithms get smaller as the percentage of the training set becomes higher. The recommended results of the other two algorithms are better than the PMF algorithm, just because adding document information can improve accuracy. ConvMF is better than ConvMF-S when the percentage of the training set is beyond 40%. When the percentage of the training set rises exceed 40%, ConvMF-S surpasses ConvMF.

Finally, we compare the performance of ConvMF-S with embedded pre-trained word vectors and without embedded pre-trained word vectors. The result is shown in figure 7.

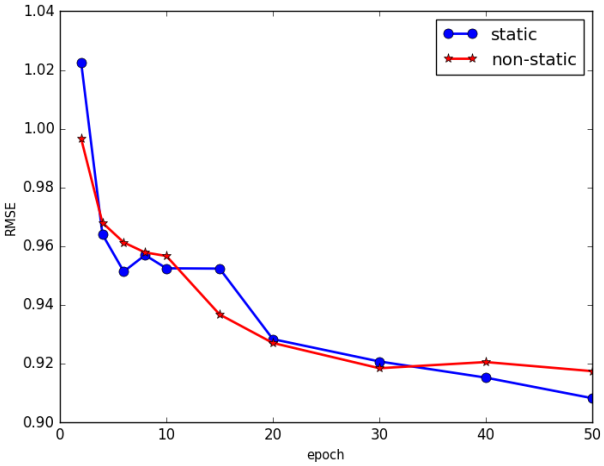

Fig. 7. Effect of the embedded word vectors

From figure 7, we can see that there is no obvious difference whether word vectors are embedded. While the model with embedded word vectors is still converging after 30 iterations. And the final result of the model with embedded word vectors also surpasses the model without embedded word vectors.

5 Conclusion

In this paper, we introduces a novel content-based deep hybrid approach with segmented max-pooling, which we call ConvMF-S. The segmented max-pooling can preserve the location information and frequency information while extracting features. Experiments show that the performance of recommendation is improved. Future work may include using distributed technology to deal with the situation in which the document data is extremely large or the selected dimension is especially high.

Acknowledgement This work was supported by the Natural Science Foundation of China (No. 61303129).

References:

1. Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD, 2008: 426-434.
2. Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering. International Conference on Machine Learning, 2007:791-798.
3. Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles. International Conference on Knowledge Discovery and Data Mining, 2011:448-456.
4. Ling G, Lyu M R, King I. Ratings meet reviews, a combined approach to recommend. RecSys'14, 2014:105-112.
5. A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. Proceedings of the 30 th International Conference on Machine Learning, 2013.
6. Salakhutdinov R, Mnih A. Probabilistic Matrix Factorization. International Conference on Neural Information Processing Systems, 2007:1257-1264.
7. Wang H, Wang N, Yeung D Y. Collaborative Deep Learning for Recommender Systems. 2014:1235-1244.
8. Dieleman S, Schrauwen B. Deep content-based music recommendation. International Conference on Neural Information Processing Systems. 2013:2643-2651.
9. Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences. Eprint ArXiv, 2014.
10. Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural Machine Translation in Linear Time. ArXiv, 2016.
11. Chen Y, Xu L, Liu K, et al. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015: 167–176.
12. Lei T, Barzilay R, Jaakkola T. Molding CNNs for text: non-linear, non-consecutive convo-

lutions. Indiana University Mathematics Journal, 2015, 58(3): 1151-1186.
13. Dong L, Wei F, Zhou M, et al. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015: 260-269.
14. Ma M, Huang L, Xiang B, et al. Dependency-based Convolutional Neural Networks for Sentence Embedding. ArXiv, 2015.
15. Johnson R, Zhang T. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. Eprint ArXiv, 2014.
16. Kim D, Park C, Oh J, et al. Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items. Information Sciences, 2017: 72-87.
17. Y Goldberg , O Levy. Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method. ArXiv, 2014.