Constrained Probabilistic Matrix Factorization with Neural Network for Recommendation System

Guoyong Cai, Nannan Chen

Constrained Probabilistic Matrix Factorization with Neural Network for Recommendation System

Guoyong Cai, Nannan Chen

Guilin University of Electronic Technology
ccgycai@gmail.com,cnn0816@126.com

Abstract. In order to alleviate the problem of rating sparsity in recommendation system, this paper proposes a model called Constrained Probabilistic Matrix Factorization with Neural Network (CPMF-NN). In user modeling, it takes the influence of users interaction items into consideration. In item modeling, it utilizes convolutional neural network to extract the item latent features from the corresponding documents. In the process of fusion of latent feature vectors, multi-layer perceptron is used to grasp the nonlinear structural characteristics of user-item interactions. Through extensive experiments on three real-world datasets, the results show that CPMF-NN achieves good performance on different sparse data sets.

Keywords: Collaborative filtering, user preference modeling, document modeling, nonlinear fusion

1 Introduction

Recommendation is one of the effective methods to solve the problem of information overload and realize personalized information service. Collaborative Filtering (CF) is a commonly used technology for recommendation. However, with the increasing number of users and items, the user-item ratings used in collaborative filtering is becoming more and more sparse which hinder the application of CF [1].

In recent studies, researchers usually try to alleviate the problem of rating sparsity from the view of user and item latent feature modeling. Salakhutdinov et al. [2] proposed a model called Constrained PMF (CPMF) on the basis of Probabilistic Matrix Factorization (PMF) who integrates the items that the users have rated into user latent feature modeling in order to obtain a more accurate user latent feature vector, and thus get a better recommendation result in the condition of sparse datasets. Wang et al. [3] combined collaborative filtering and probabilistic topic model together and proposed a model called collaborative Topic Regression (CTR) which extracts item latent features from the item documents by Latent Dirichlet Allocation (LDA). Wang et al. [4] think that CTR cannot extract item latent feature effectively. Therefore, a Collaborative

Deep Leaning (CDL) is proposed by combining Bayesian stacked denoising autoencoder (Bayesian SDAE) and PMF. In the view of Kim et al. [5], CTR and CDL cannot fully capture document information as they assume the bag-of-word model that ignores the contextual information of documents. So, Convolutional Matrix Factorization (ConvMF) which integrates Convolutional Neural Network (CNN) into PMF was proposed. ConvMF leveraged CNN to capture the contextual information of documents, so as to obtain more accuracy representation of item latent features and more accuracy predicted ratings. The above researches show that integrating item documents into item modeling can improve the recommendation effect.

In the above studies, although CPMF took the items that the users have rated into user modeling, it still placed spherical Gaussian priors on item latent feature vectors with the same parameters as PMF does, so its item modeling can still be further improved. In the other hand, CTR, CDL and ConvMF made some advance in item modeling by extract item latent features from item document, but they also placed spherical Gaussian priors on user latent feature vectors with the same parameters as PMF does. As a result, it always leads to inaccurate predicted ratings of some users on sparse datasets. Salakhutdinov et al. [2] points out that over such a spherical Gaussian priors, once the model has been fitted, the users with few ratings will have feature vectors that are close to the prior mean, or the average user, so the predicted ratings for those users will be close to the item average ratings. As a result, it still leads to inaccurate predicted ratings of some users on sparse datasets.

In view of the fact that the above researches cannot make improvement in user and item modeling at the same time, this paper proposes a model called Constrained Probabilistic Matrix Factorization with Neural Network (CPMF-NN), which achieves some enhancement in the follow three aspects. In user latent feature modeling, CPMF-NN takes the items that the users have rated into account so that users with different rated items will own Gaussian priors with different parameters. In item latent feature modeling, CNN is used to extract item latent features from the item documents. In the fusion of user and item latent feature, different from linear fusion method of traditional matrix factorization, CPMF-NN takes the advantage of Multi-Layer Perceptron (MLP) to realize a nonlinear fusion method that ultimately improves the accuracy of the predicted ratings.

The work of this paper is organized as follows. Section 2 introduces the framework, optimization methodology and the method of parameter updating. Section 3 introduces the datasets and experiments. Finally, in section 4, we summarize the work of this paper and look to the future work.

2 Constrained probabilistic matrix factorization with neural network

Like PMF, CPMF-NN obtains user and item latent feature vectors by factorizing the user-item rating matrix, and it decomposes user and item latent feature

vectors as well. The framework of CPMF-NN is shown in Fig 1. It consists of three parts and is briefly described as follows.

The first part, which is similar to PMF and is the basis of CPMF-NN, is shown in part (a) of Fig 1. Suppose there are n users and m items. Let $R \in \mathbb{R}^{n \times m}$ denotes the user-item rating matrix, and the integer ratings $R_{ij} \in \{1, 2, 3, 4, 5\}$ refers to the rating of user i for item j. The purpose of CPMF-NN is to factorize the rating matrix into the user latent feature matrix $U \in \mathbb{R}^{d \times n}$ and item latent feature matrix $V \in \mathbb{R}^{d \times m}$, and it hopes that $R_{ij} \approx \hat{R}_{ij} = f(U_i, V_j)$, where d denotes the dimension of latent feature vectors, R_{ij} denotes the predicted rating of user i for item j and $f()$ denotes the fusion function. Similar to the idea of PMF, CPMF-NN decomposes user and item latent feature vectors as well. But different from PMF, CPMF-NN takes a nonlinear fusion function rather than a linear function.

The second part is shown in part (b) of Fig 1. CPMF-NN decomposes each user latent feature vector U_i into a sum of 2 terms: offset term X_i [2, 6] and preference term P_i. P_i is the mean of constrained vectors of items that user i has rated. Let $I \in \mathbb{R}^{n \times m}$ denotes the indicator matrix with elements I_{ih} is equal 1 if user i rated item h and 0 otherwise.

The third part is shown in part (c) of Fig 1. CPMF-NN decomposes each item latent feature vector V_j into a sum of 2 terms. The first term is item latent feature term F_j that extracted from the corresponding item document via CNN. The second term is Gaussian noise which enables us to further optimize the item latent feature vector for predicting ratings.

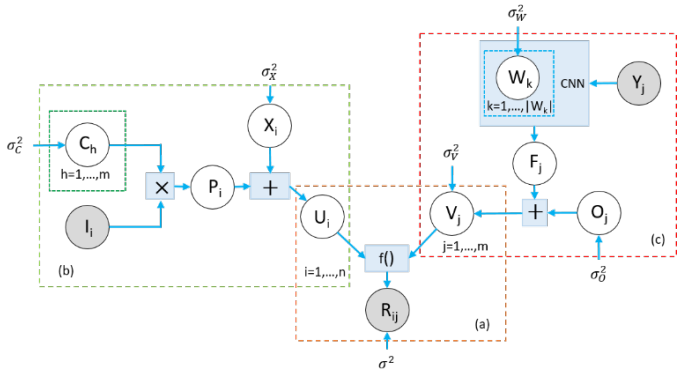


Fig. 1. The graphical model of CPMF-NN

The conditional distribution over the observed ratings is defined as Eq.(1)

$$p(R|U, V, \sigma^2) = \prod_{i=1}^{n} \prod_{j=1}^{m} [N(R_{ij}|U_i^T V_j)] \,. \tag{1}$$

2.1 User latent feature modeling

In user latent feature modeling, literatures [3-5] all placed a zero mean spherical Gaussian prior on the user latent feature vectors. However, such assumptions often lead to inaccurate predicted ratings for some users due to the problem of rating sparsity. The items that one user rated usually reflect the preference of the user. In order to get a more accurate user latent feature, we define U_i as the sum of two terms: 1) offset term X_i , which is the basic representation of user i. 2) preference term P_i, which is another representation part of user i constructed by the users whole rating items. CPMF-NN gives each item another representation besides the latent feature vector, called item constrained vector C_h, which is used to construct preference term P_i. Specially, the preference term P_i of user i is defined as the mean of item constrained vectors of items that user i has rated. With the two terms, we can get the user latent feature vector.

$$U_i = X_i + P_i \tag{2}$$

here

$$P_i = \frac{\sum_h^m I_h C_h}{\sum_h^m I_h} \tag{3}$$

and we also place spherical Gaussian priors on offset terms and item constrained terms as CPMF does.

$$p(X|\sigma_X^2) = \prod_i^n N(X_i|0, \sigma_X^2 I),\ p(C|\sigma_C^2) = \prod_i^n N(C_h|0, \sigma_C^2 I) \tag{4}$$

Taking Eq. [3] and [4] into Eq. [2], for each user we can draw a user latent feature vector $U_i \sim N(U_i|0, \sigma_X^2 + \sigma_C^2 / \sum_h^m I_h)$. The variance of U_i will getting more close to the variance of X_i when user i have more rating items. That is to say, the influence of the item constrained vectors will be smaller and even eliminate. On the contrary, the influence will be strong on the users who have rated a few items.

2.2 Item latent feature modeling

In item latent feature modeling, considering the sparsity problem, we leverage item documents to obtain item latent feature vectors. Similar to user latent feature modeling, we decompose each item latent feature vector into a sum of two parts: 1) item latent feature term F_j which is extracted from the corresponding item document Y_j by CNN. 2) Gaussian noise O_j for the more accurate representation of item latent feature. With these two parts, we can get the item latent feature vector

$$V_j = F_j + O_j \tag{5}$$

where $F_j = CNN(W, Y_j)$ and we also place spherical Gaussian priors on the weight of CNN and Gaussian prior on the Gaussian noise:

$$p(W|\sigma_W^2) = \prod_k N(w_k|0, \sigma_W^2), O_j \sim N(0, \sigma_O^2) \tag{6}$$

Accordingly, the conditional distribution over item latent feature vectors is given by

$$p(V|W, Y, \sigma_V^2) = \prod_j^m N(V_j|CNN(W, Y_j), \sigma_V^2 I). \tag{7}$$

We use the CNN architecture Kim [7] proposed to analyse item documents. Specially, for each item, we take its document $Y_j = [y_1, y_2, \ldots, y_t]$ as the input of CNN in which t denotes the length of document and $y_.$ is the word embedding vector of a word in the documents. Then, with a shared weight $W_e^j \in \mathbb{R}^{|y_.| \times x}$ whose window size is x, a convolution feature $e^j = [e_1, e_2, \ldots, e_{t-x+1}]$ is generated. In the pooling layer, we use max-pooling to get the document feature representation $e = [max(e^2), max(e^2), \ldots, max(e^j)]$. Finally, projecting e by a nonlinear activation and we can get the feature presentation of each document $F_j = tanh(W_2(tanh(W_1 e + b_1)) + b_2) = CNN(W, y_j)$.

2.3 Fusion of latent features

In order to get the predicted ratings, we define a fusion function $f()$ to fuse the user and item latent feature vectors. The framework of fusion is shown in table 2. The process of fusion can be given in the form of $\hat{R}_{ij} = f(U_i, V_j)$ with user and item latent feature vectors as the input and rating as the output. Different from the traditional linear fusion method such as inner product, CPMF-NN realizes a nonlinear fusion method based on MLP: $\hat{R}_{ij} = f(U_i, V_j) = mlp(mlp(U_i) \odot mlp(V_j)))$, where $\odot$ denote element-wise product.

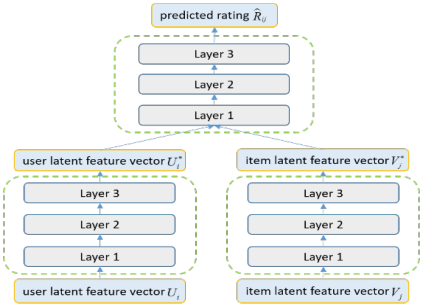


Fig. 2. The framework of fusion

Firstly, taking user latent feature vector U_i and item latent feature vector V_j as the input of MLP respectively. In particular, it can be formulated as follows.

$$\begin{aligned} L_1 &= a_1(W_1^T x + b_1), \\ L_2 &= a_2(W_2^T L_1 + b_2), \\ x^* = L_3 &= a_3(W_3^T L_2 + b_3). \end{aligned} \tag{8}$$

where x denotes the input (U_i or V_j) of MLP. L_k, a_k, W_k and b_k respectively denote the output, activation function, weight and bias of hidden layer k where $k = 1, 2, 3$ and x^* (U_i^* or V_j^*) denotes the output of MLP. Then, taking $x = U_i^* \odot V_j^*$ as the input of MLP for the purpose of predicting rating. Finally, we can get the output of the last layer as the predicted rating $\hat{R}_{ij}$.

Compared to the traditional linear fusion, the nonlinear fusion method proposed in this paper can catch the nonlinear feature of interactions between users and items and enhance the accuracy of predicted ratings. We take back propagation algorithm to optimize the weight and bias of hidden layers and take ReLU as activation function since it is proved to be non-saturated [8]. In addition, ReLU encourages sparse activations, being well-suited for sparse data and making the model less likely to be overfitting [9].

Optimization methodology To optimize the parameters such as U, V and the weight of CNN, maximum a posterior (MAP) estimation is employed. Since computing the full MAP is intractable, maximizing MAP is equivalent to minimizing the log-likelihood as follows.

$$\begin{aligned} minE = \frac{1}{2}\sum_{i}^{n}\sum_{j}^{m} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_X}{2}\sum_{i}^{n}\|X_i\|_F^2 + \frac{\lambda_C}{2}\sum_{h}^{m}\|C_h\|_F^2 \\ + \frac{\lambda_V}{2}\sum_{j}^{m}\|V_j - CNN(W, Y_j)\|_F^2 + \frac{\lambda_W}{2}\sum_{k}^{|w_k|}\|W_k\|_F^2 \end{aligned} \tag{9}$$

where $\lambda_X = \frac{\sigma_X^2}{\sigma^2}$,$\lambda_V = \frac{\sigma_V^2}{\sigma^2}$, $\lambda_C = \frac{\sigma_C^2}{\sigma^2}$, $\lambda_W = \frac{\sigma_W^2}{\sigma^2}$.

Similar to Kim et al.[4], we adopt coordinate descent to optimize X_i, C_h and V_j. It optimizes one variable while fixing the remaining variables. As a result, the variables can be updated as follows.

$$X_i = (VI_iV + \lambda_X I_d)^{-1}(VR_i - VI_iV^T\frac{\sum_h^m I_{ih}C_h}{\sum_h^m I_{ih}}) \tag{10}$$

$$\begin{aligned} C_h = (\sum_{i}^{n} VI_hV^T\frac{1}{\sum_h^m I_{ih}} + \lambda_C I_d)^{-1}(\sum_{i}^{n}(VR_i\frac{1}{\sum_h^m I_h} \\ -VI_hV^T(X_i + \frac{\sum_h^m I_{ih}C_h}{(\sum_h^m I_{ih})^2})\frac{1}{\sum_h^m I_{ih}})) \end{aligned} \tag{11}$$

$$V_j = (UI_jU^T)^{-1}(UR_j + \lambda_V CNN(W, Y_j)) \tag{12}$$

where I_i is a diagonal matrix with I_{ij} as its diagonal elements, and I_d, I_j and I_h are same defined as I_i.

As for the weight W of CNN, we use back propagation algorithm to optimize as E can be seen as a squared error function with L_2 regularized terms when other variables are temporarily constant.

3 Experiments

3.1 Experimental environment and datasets

The experiments are implemented on a E5-2620 CPUs work station and a Tesla P100-PCIE GPU work station. The development environment are Python 2.7, Tensorflow 1.3.0 and Keras 2.0.5, and the development tool is PyCharm.

We experimented with three publicly accessible datasets: Movielens 1m (ML-1m), Movielens 100k (ML-100k) and Amazon Instant Video (AIV). The value of user-item ratings in each dataset is 1 to 5. ML-1m and ML-100k are movie ratings datasets widely used in recommendation, we obtained the plot summary from IMDB as the document of each movie, and removed some movies from the datasets cause the absence of plot summary of these movies on IMDB. AIV is an instant video ratings dataset with reviews on each video. Because of the large scale of the AIV, we removed the videos with less 5 ratings and with reviews more than 10000 words. We randomly split each dataset into a training set (80%), a validation set (10%) and a test set (10%). As a result, the Statistics of each dataset are showed in table 1.

All the item documents are preprocessed as follows: 1) the maximum length of documents is set to be 300, 2) remove the stop words, 3) calculate the ti-idf value of each word, 4) remove the corpus-specific stop words with document frequency higher than 0.5, 5) selecte the top 8000 words as a vocabulary, 6) remove all non-vocabulary words from documents.

Table 1. Statistics of the datasets

Dataset	User#	item#	rating#	Sparsity
ML-100k	943	1542	91636	93.698%
ML-1m	6040	3544	993482	95.359%
AIV	1136	7065	10883	99.864%

3.2 Baselines and parameter settings

We compared CPMF-NN with the following two baselines.

• PMF [2]: Probabilistic matrix factorization is a classical collaborative filtering method. It is the basis of the ConvMF proposed by Kim et al. [5] and the basis of CPMF-NN model proposed in this paper.

• ConvMF [5]: ConvMF extracted item latent feature from item document by CNN and integrated CNN into PMF model. Compared to ConvMF, CPMF-NN involves user interaction items in user modelling and fuses the user and item feature latent vectors in a nonlinear way.

We set the dimension of latent feature vector to be 64 in experiments. Table 2 shows other parameters setting which are set according to experience.

Table 2. Statistics of the datasets

	Model						
	PMF		ConvMF			CPMF-NN	
	λ_U	λ_V	λ_U	λ_V	λ_X	λ_V	λ_C
Ml-100k	1	100	5	100	1	500	10
Ml-1m	0.01	10000	0.01	10	200	10	10
AIV	20	0.1	0.001	1e5	0.005	10000	1

3.3 Evaluation protocols

We adopt root mean squared error (RMSE) and Recall as the protocols for each model on the three real-world datasets. RMSE is a popular matric and it measures the error between the real ratings and the predicted ratings, and is defined as follows.

$$RMSE = \sqrt{\frac{\sum_{i,j \in test}^{n,m} (R_{ij} - \hat{R}_{ij})^2}{n}}, \tag{13}$$

where n is the number of user-item ratings in the test dataset.

Recall is a measure of classification accuracy, which indicates the ability of the model to predict a particular item the user like or dislike. It is defined as follows.

$$Recall = \frac{1}{N} \sum_{i}^{N} \frac{|Z_i - T_i|}{|T_i|}, \tag{14}$$

where N denotes the user number in test set, Z_i denote the set of recommendation items of user i in test set and T_i denote set of the real items of user i in test set.

3.4 Experimental results

Fig 3 and Fig 4 show the performance of each model on the three real-world datasets. As shown in Fig 3, the trend of RMSE of different models on different datasets is consistent. The classical CF method PMF is greatly influenced by the sparsity of the datasets. When it occurs to Amazon Instant Video dataset whose sparseness is much more than the other two, the RMSE value of PMF raises obviously compared to ConvMF and CPMF-NN. It indicates that using CNN to extract the item latent features from the documents can effectively alleviate the sparsity problem. On different dataset, the improvement of CPMF-NN over the best competitor are 5.5470%, 7.4747% and 9.7166%. It proves that it is helpful to alleviate the sparsity by taking users interaction items and nonlinear fusion method into consideration.

Fig 4 gives the overall Recall performance of each model on the three datasets. On the ML-100k dataset, when Top_n=2 and Top_n = 4, the corresponding Recall value of the CPMF model is slightly lower than that of the ConvMF

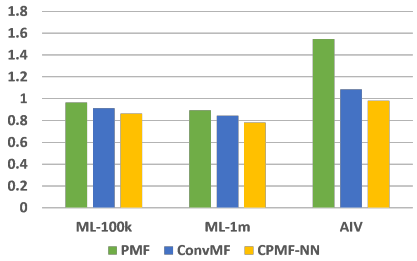
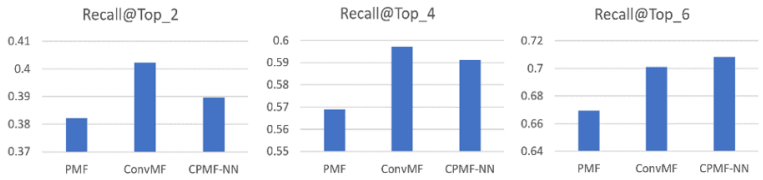
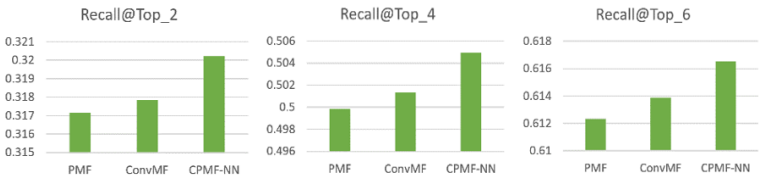


Fig. 3. The overall RMSE performance

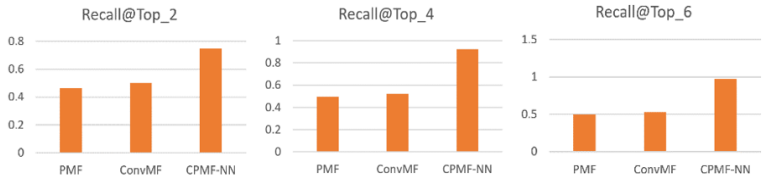
model. It should be pointed out that when $Top\_n \geq 6$, the Recall value of the CPMF-NN is higher than the ConvMF. On the ML-1m dataset, the experimental results of CPMF-NN have a slight improvement compared to PMF and ConvMF. On the AIV dataset, the improvement of CPMF-NN is obvious compared to the baselines. The results on three different sparse datasets prove that the sparsity problem can be effectively alleviated by improving user modeling, item modeling and fusion method.



(a) ML-100k



(b) ML-1m



(c) AIV

Fig. 4. The overall Recall performance

4 Conclusions and future work

CPMF-NN proposed in this paper commitment to alleviate the sparsity problem by combining the traditional PMF model with deep learning. In user modeling, it considers the influence of the items that users have rated, and realizes them by adding item constrained vectors to user latent feature vectors. In item modeling, CPMF-NN extracts item latent features from the item documents by CNN. In the last, it fuses user and item latent feature vectors to get predicted ratings with the structure of MLP.

With the development of internet, it is becoming easier and easier for us to access multimodality data such as the context, reviews and images about users and items. How to effectively take advantage of these multimodal data is the direction of our future work.

Acknowledgments. This work is supported by Chinese National Science Foundation (#61763007), Guangxi Key Lab of Trusted Software under project Kx201 503 and Innovation Project of GUET Graduate Education (#2017YJCX44).

References

1. Goldberg, D, Nichols, D,Oki B.M.,et al: Using collaborative filtering to weave an information tapestry. Communications of the ACM. 35, 61–70 (1992)
2. Mnih, A, Salakhutdinov, R.R.,: Probabilistic matrix factorization. In: Advances in neural information processing systems, pp.1257–1264. ACM Press, New York (2008)
3. Wang, C, Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In:Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.448–456. ACM Press, San Diego (2011)
4. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In:Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.1235–1244. ACM Press, Australia (2015)
5. Kim, D., Park, C., Oh, J., et al: Convolutional matrix factorization for document context-aware recommendation. In:Proceedings of the 10th ACM Conference on Recommender Systems, pp.233-240. ACM Press, Boston (2016)
6. Zhang, F., Yuan, N.J., Lian, D., et al, et al: Collaborative knowledge base embedding for recommender systems. In:Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp.353–362. ACM Press, Francisco (2016)
7. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv. 1408–5882 (2014)
8. Glorot, X., Bordes, A., Bengio, Y. : Deep sparse rectifier neural networks. In:Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp.315–323. Florida (2011)
9. He, X., Liao, L., Zhang, H., et al.: Neural collaborative filtering. In:Proceedings of the 26th International Conference on World Wide Web, pp.173–182. International World Wide Web Conferences Steering Committee, Australia (2017)