Research Repository UCD

Title	Convolutional Matrix Factorization for Recommendation Explanation
Authors(s)	Lu, Yichao, Dong, Ruihai, Smyth, Barry
Publication date	2018-03-11
Publication information	Lu, Yichao, Ruihai Dong, and Barry Smyth. "Convolutional Matrix Factorization for Recommendation Explanation." ACS, 2018.
Conference details	The 23rd International Conference on Intelligent User Interfaces Companion, Tokyo, Japan, 07-11 March 2018
Series	Article No. 34
Publisher	ACS
Item record/more information	http://hdl.handle.net/10197/9941
Publisher's statement	© ACM, 2018. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in IUT'18 Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion http://doi.acm.org/10.1145/10.1145/3180308.3180343
Publisher's version (DOI)	10.1145/3180308.3180343

Downloaded 2023-04-07T10:52:29Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Convolutional Matrix Factorization for Recommendation Explanation

Yichao Lu
University of Toronto
Toronto, Canada
yichao@cs.toronto.edu

Ruihai Dong
University College Dublin
Dublin, Ireland
ruihai.dong@ucd.ie

Barry Smyth
University College Dublin
Dublin, Ireland
barry.smyth@ucd.ie

ABSTRACT

In this paper, we introduce a novel recommendation model, which harnesses a convolutional neural network to mine meaningful information from customer reviews, and integrates it with matrix factorization algorithm seamlessly. It is a valid method to improve the transparency of CF algorithms.

Author Keywords

Recommendation Explanation; Convolutional Neural Network; Matrix Factorization; Customer Reviews.

INTRODUCTION

Recommender systems have become an essential part of on-line shopping sites. Collaborative Filtering (CF) approach is one of the most popular ways in the recommendation area, widely used in industry, such as at Amazon and Netflix. CF algorithms, either neighborhood methods or latent factor models, heavily rely on user-item ratings. For example, latent factor models that based on matrix factorization map users and items to a set of latent features. These algorithms suffer from sparsity and transparency issues.

In an effort to continually improve the effectiveness and applicability of CF algorithms, researchers have sought to harness various types of textual data and knowledge, such as item descriptions and customer review text. For example, Latent Dirichlet Allocation (LDA) based topic modeling techniques are adopted to integrate customer reviews into CF algorithms [2]. Such jointly trained models have been shown to more accurately predict ratings and being capable of interpreting latent features of CF algorithms by utilizing topical information.

Convolutional Neural Networks (CNN) are widely used in the literature of computer vision. CNNs have a strong capability to learn feature representations. Similarly, a CNN can mine information from a document, such as a product description. It can be considered as an n -gram language model, see [1]. In this paper, We employ a convolutional neural network to learn representative features from customer reviews and to work with the traditional matrix factorization algorithm. The

Figure 1. High-level Convolutional Matrix Factorization.

main contribution of this work is to show we can use textual information existing in customer reviews located by CNN filters to interpret the latent features.

CONVOLUTIONAL MATRIX FACTORIZATION

Our architecture is illustrated in Figure 1. It consists of two parts: the convolutional neural network, which learns an item representation from customer review through three layers, i.e., *Embedding Layer*, *Convolutional Layer*, and *Pooling Layer*, and matrix factorization, which is used to predict ratings.

Convolutional Neural Network

The embedding layer takes as input a sequence of words ($w_1, w_2, w_3, \dots, w_T$), and maps each word into a respective k -dimensional real-valued vector $x_i \in \mathbb{R}^k$. Essentially, it embeds a document into a vector space $x_{1:T} \in \mathbb{R}^{kT}$, where $x_{i:i+j}$ refers to the concatenation of the word vectors $x_i, \dots, x_{i+j}$. We initialize the word vectors with pre-trained word embeddings¹ and then fine-tune them via back-propagation.

The convolutional layer consists of a set of filters corresponding to latent features in matrix factorization model. These filters are used to extract text features. Each filter $f \in \mathbb{R}^{hk}$ is applied to a window of h words (the *window width*) to catch a contextual feature. We denote the j -th convolutional filter as f^j , and the contextual feature c_i^j is generated by $c_i^j = f(f^j \cdot x_{i:i+h-1} + b)$. Each convolutional filter loops over the document to produce a feature map $c^j = [c_1^j, c_2^j, \dots, c_{T-h+1}^j]$.

The pooling layer performs a *max-pooling* operation over the feature map. It takes the maximum value $\hat{c}^j = \max\{c^j\}$ as the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IUI'18 Companion March 7–11, 2018, Tokyo, Japan

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5571-1/18/03.

DOI: <https://doi.org/10.1145/3180308.3180343>

¹<https://code.google.com/archive/p/word2vec/>

activation value of the feature map. The pooled activation values are concatenated into a dense vector $\hat{c} = [\hat{c}^1, \hat{c}^2, \dots, \hat{c}^j]$. We remove the final fully-connected layer in the conventional convolutional neural network, and regard the activations of pooling layer as the textual representations learned from review documents. This enables us to keep track of the corresponding features of each latent dimension.

Matrix Factorization

In matrix factorization, each user and item is represented by a k -dimensional real-valued vector. The predicted rating given by a particular user on a specific item is given by the dot product between the user and item latent factor vectors. In our approach, the user latent factor vectors are learned in the same way of previous works. That is to say, they are initialized randomly, and updated during the training process. The item latent feature vectors, on the other hand, are learned quite differently. For each item, we denote the set of all reviews written on it as its review document. In order to generate the latent feature vector associated with the item, we first extract the textual features from each review with the convolutional neural network architecture described above, and then average these feature vectors as the item latent feature vector.

Note that in our setting, the user latent factor vectors are learned from scratch, while the item latent feature vectors are dependent on the item review documents. The rationale behind it is the observation that, instead of talking about their own preferences, users are more likely to discuss the properties of the items in the reviews.

The training objective is to minimize the mean squared error between the predicted ratings and the ground-truth ratings. The user latent factor vectors and the parameters in the convolutional neural network are then learned in an end-to-end fashion through mini-batch stochastic gradient descent.

EXPLANATION EXTRACTION

This idea is borrowed from the literature of computer vision[3], which states that only the most salient features of each image are worth noticing. The pooling layer effectively subsamples the feature map to reduce its dimensionality and prevent overfitting.

To understand the meaning of each particular latent dimension, we need to look for the n -gram features that can maximally activate the corresponding convolutional filters. Briefly speaking, we loop over the entire review document, and calculate the activation value of each n -gram in the text. The activation value is assumed to be equally distributed for each word in the n -gram. After removing the stop words, we sum up the activation values that have been assigned to each word, and rank the vocabulary according to the activation scores. We can then generate descriptive interpretations of each latent factor by examining the top k words.

EVALUATION

We evaluate our model on the Yelp 2013 dataset², dividing the dataset into training/validation/test set with a 70/10/20 split.

²<https://www.yelp.com/dataset/challenge>

Figure 2. Examples for Latent Feature Interpretation.

In our experiments, we set the number of latent factors to be 32. The width of the convolutional filters are set to 5 to extract 5-gram features from reviews.

In order to quantitatively evaluate the recommendations given by our approach, we examine the Root Mean Squared Error (RMSE) of the predictions on the test set. In summary, we achieve an RMSE of 0.987, which is slightly better than the probabilistic matrix factorization model, whose RMSE is 0.993.

We qualitatively analyze the recommendation explanations given by our approach using the explanation extraction method described above. The top 10 words for three latent factors are displayed in Figure 2. As we can see, the word clusters given by the model are indeed descriptive of the restaurants. We can therefore utilize them to provide recommendation explanations. Take “taste” as an example, if the latent feature vector of a restaurant has a high value in the dimension corresponding to “taste”, we can presume that the restaurant is great at providing delicious cuisine, and vice versa. On the other hand, if the vector representation of a specific user is high in that dimension, we may guess that, among other factors, “taste” is an important factor that influence the ratings he or she gives to the restaurant.

CONCLUSION

We demonstrate how a convolutional neural network can be integrated into matrix factorization so as to produce interpretable recommendations. Our approach has the capacity of improving the transparency of CF algorithms, and has the potential to generate recommendation explanations.

REFERENCES

1. Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.
2. Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.
3. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2921–2929.