Collaborative Filtering via Learning Characteristics of Neighborhood based on Convolutional Neural Networks

Yugang Jia, Xin Wang, Jinting Zhang
Fidelity Investments
{yugang.jia,wangxin8588,jintingzhang1}@gmail.com

ABSTRACT

Collaborative filtering (CF) is an extensively studied topic in Recommender System. Recent approaches use the statistical framework based on local Taylor approximations to unify both user based and item based CF algorithms and improve the performance of estimating unknown ratings. In this paper, we propose a new Machine Learning approach based on Convolutional Neural Networks to exploit complex latent user-item relations, using features extracted from the neighborhood of unknown rating via local approximations. Experimental results on two benchmark data sets demonstrate the effectiveness of the proposed approach via comparing to state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Collaborative filtering**.

KEYWORDS

Collaborative filtering, Local Taylor approximation, Convolutional Neural Networks

1 INTRODUCTION

Moving toward a digital era, the information overload creates a potential challenge for user to have timely access to information of interest. As a result, there has been huge amount of effort spent to develop recommendation system in the recent years [2]. Among the methods, collaborative filtering has become one of the most researched techniques since the term was coined in 1992 [6]. The fundamental assumption of CF is that if users rate a set of items similarly, or have similar behaviors (e.g., buying, watching, listening), they will rate or act on other items similarly [7, 17]. If we put the ratings in the structure of a user-item rating matrix, the task of predicting unknown ratings can be viewed as sparse matrix completion problem.

A realistic assumption for this problem is that the user or item in the matrix is exchangeable, i.e. the user could be located in any row of the matrix without changing the definition of problem. For such exchangeable data set, the ratings can be represented by a convenient latent variable model, where the ratings are modeled as a noisy signal of a "well-behaved" function evaluated over latent variables associated with the rows and columns [3, 9]. One popular way of solving this problem is based on low rank approximation and matrix factorization (e.g. singular value decomposition SVD) [14]. Here the fundamental assumption is that the latent vectors of user and item can be approximately solved for and the underlining function mapping latent vectors of user and item to a rating is to do an inner product of two vectors.

Recently, a blind regression based statistical framework is proposed [4] to address CF problem. The method is based on the local approximation via Taylor expansion and has finite sample error bounds with minimal assumptions. An unknown rating is approximated using ratings in defined neighborhood. This also shed a light on why the traditional collaborative filtering approach works in practice. However, the weights to combine multiple generated approximations to create one final estimate are determined by the similarity measured with Gaussian kernel and thus the learned weights may not fit well on some data sets [4].

In view of this, we propose a Machine Learning approach based on Convolutional Neural Networks (CNN) to improve the performance of unknown rating prediction by learning useful representation of neighborhood rating distributions. The proposed method consumes features extracted from the neighborhood of unknown ratings via local approximations, to learn implicit relations between items and users. Experiment results demonstrate the effectiveness of the proposed algorithm via comparing to state-of-the-art methods.

2 ESTIMATING UNKNOWN RATINGS VIA LOCAL APPROXIMATION

In this section, we introduce how we apply the method proposed in [4] to estimate unknown ratings.

Let $u = 1, 2, .., M$ denote indices of M users and $i = 1, 2, .., N$ denote indices of N items. A rating from user u for item i will be denoted as y_u^i, for example $y_u^i \in \{1, 2, 3, 4, 5\}$. The metadata for users (such as gender, age, etc) will be represented as Ψ and the meta data for items (for example movies are represented by genres, year of production, etc) will be represented as Φ. We also represent a set of observed user-item pairs as Γ, such that $(u, j) \in \Gamma$ if y_u^j is observed. Our goal is to estimate the unobserved rating y_u^i given observed ones and user-item metadata.

To further illustrate the neighborhood matrix, all relevant ratings are put in Fig. 1. There are three types of information represented by different shapes: ratings from same user (circles), ratings for same item (squares) and ratings from other user-item pairs (triangles).

Based on the assumption of local function approximation (Taylor's expansion), we use the following equation to estimate y_u^i given y_v^j, y_u^j and y_v^i:

$$\hat{y}_u^i(v, j) \approx y_v^i + y_u^j - y_v^j, s.t.(v, j) \in \Gamma_\beta(u, i) \tag{1}$$

DLP-KDD'19, August 5, 2019, Anchorage, AK, USA

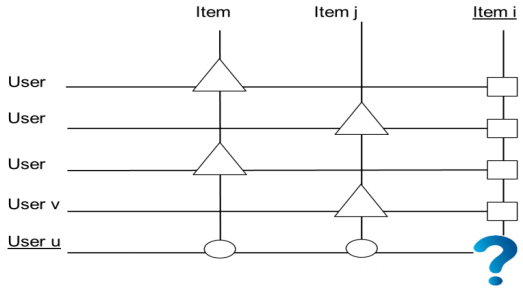


Figure 1: Illustration of neighbor ratings. The question mark refers to the ratings y_u^i to be computed. The circles are ratings from user u and squares are ratings for item i respectively. The triangles are ratings from other users for other items

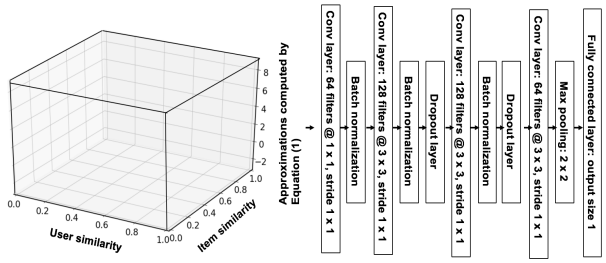


Figure 2: Input data and CNN architecture. We use ratings in the range 1 to 5 for example. Distribution of approximations for an unknown rating is described along three dimensions: user similarity, item similarity and magnitudes, which serves as the 3D input to CNN and approximations in each magnitude are treated as one more channel.

where $\Gamma_\beta(u,i)$ is the set of user-item pairs that has at least β overlapping ratings with user u and item i. We can use a weighted combination of generated estimates in Equation (1) to estimate unknown rating. Intuitively the similarity between users u and v, items i and j can be used to generate the weights.

A Gaussian kernel can be used to do a weighted combination of all estimations yielded by Equation (1) for the neighborhood [4]:

$$\hat{y}_u^i = \sum_{(v,j)\in\Gamma_\beta(u,i)} w_{v,j}\hat{y}_u^i(v,j) \tag{2}$$

with

$$w_{v,j} = \exp\left(-\lambda \min(S_{u,v}, S_{i,j})\right) \tag{3}$$

where λ is a predefined parameter, $S_{u,v}$ and $S_{i,j}$ are a pair-wise similarity measure for user and movie respectively:

$$S_{u,v} = \frac{1}{2L_{u,v}(L_{u,v}-1)} \sum_{i,j} \left((y_u^i - y_v^i) - (y_u^j - y_v^j)\right)^2 \tag{4}$$

$$S_{i,j} = \frac{1}{2L_{i,j}(L_{i,j}-1)} \sum_{u,v} \left((y_u^i - y_u^j) - (y_v^i - y_v^j)\right)^2 \tag{5}$$

where $L_{u,v}$ and $L_{i,j}$ is the number of overlapping ratings for user and movie respectively.

3 PROPOSED CNN MODEL

CNN is a state-of-the-art Machine Learning method which has shown strong capability in image recognition and natural language processing [11, 15]. Very recently CNN has also been applied in building content-based Recommender Systems, where CNN is used to learn the representation of recommended contents such as audio signals, item descriptions or documents and then the interaction of content representation and user profile is further exploited for making recommendations [16, 18, 20]. Different from existing Recommender systems using CNN to learn useful representation of user or item, we propose a method to exploit unknown rating's neighborhood information via local approximations.

For each unknown rating's neighborhood $\Gamma_\beta(u,i)$, we get the value of $\hat{y}_u^i(v,j)$ calculated by Equation (1). We extract features to capture the distributions of generated approximations. Each neighborhood is used as one data instance in input for training and testing. Firstly we calculate user similarity between u and v and item similarity between i and j by using the meta data. The similarity is computed based on a Gaussian Kernel. Then we split the range of similarity [0, 1] into 11 groups. Each group contains $\hat{y}_u^i(v,j)$ in one of ranges [0, 0.1), [0.1, 0.2), $\cdots$, [0.9, 1) or 1. For each group of similarity we get the raw counts of $\hat{y}_u^i(v,j)$ at different magnitudes. The format of input data is shown in Fig. 2 where we use the rating in a range from 1 to 5 for example.

We use CNN to capture complex latent relationships between users and items in the global neighborhood defined by $\Gamma_\beta(u,i)$ and local neighborhoods within $\Gamma_\beta(u,i)$. Our goal is to learn a highly nonlinear mapping from local approximations of a rating to a true rating. Fig. 2 shows detailed information of each layer. We use rectified linear unit (ReLU) function as activation function for each convolutional layer. To output a estimate of rating, the last layer uses linear function to fit target ratings and we minimize the mean squared error (MSE) as training objective. In another situation, we can also use *Sigmoid* activation and cross entropy loss to train a binary classification task, for example we predict if it is a like against dislike. We use Adam algorithm [12] to optimize for parameters.

Table 1: Details of MovieLens and Epinions data sets

Data set	# users	# items	# ratings
MovieLens 1M	6,040	3,706	1,000,209
Epinions	49,290	139,738	664,824

4 EXPERIMENT

The proposed method was tested on two benchmark data sets, MovieLens 1M [1] and Epinions [5] data sets. Statistics of these two data sets are shown in Table 1. The Epinions data was more sparse. We compared to four published methods, blind regression with Gaussian Kernel (BR+GK) [4], SVD++ [10, 13], neural collaborative filtering (NCF) [8] and graph convolutional matrix completion (GCMC) [19]. Our model is referred as CF_CNN (Collaborative Filtering with CNN).

All models were trained and tested with 5-fold cross validation (CV). For the proposed approach, we experimented with different

Table 2: Comparison of the performance on estimating ratings in terms of RMSE.

Methods	MovieLens 1M	Epinions
SVD++	0.863±0.001	1.063±0.002
BR+GK	0.871±0.001	1.185±0.002
NCF	–	–
GCMC	0.855±0.002	1.180±0.002
CF_CNN	**0.853±0.001**	**1.049±0.002**

hyper-parameters and network structures. The parameters such as number of filters among {32, 64, 128} for each Conv layer, dropout rate among {0.1, 0.2} and regularization parameter for each layer among {0.05, 0.1, 0.2, 0.4} were tuned using a hold-out set in the training data (10% of training data) in the first CV fold. We observed that the architecture shown in Fig. 2 achieved the best performance in general on both MovieLens and Epinions data sets. We trained our model by 50 epochs in each CV. For the parameters in compared methods, we set the parameter β to be 5 and 1, the number of overlapping ratings L to be 8 and 2 on MovieLens 1M and Epinions respectively for BR+GK. We set number of factors to be 20 and also use other default parameters for SVD++ on both data sets. We set number of factors to be 10 and use default network setups for both Matrix Factorization and Multilayer perceptron modules in NCF. We also use default hyper-parameters for GCMC and train the model with 500 epochs on both two tested data sets.

We compared the averaged root mean square error (RMSE) on test set for estimating ratings. A lower RMSE indicates a better performance. The results are summarized in Table 2, where the neighborhood was defined by β being 5 and 1 for MovieLens and Epinions respectively. The proposed method outperformed the other compared algorithms on both two data sets. GCMC achieved a better performance on MovieLens 1M than SVD++ and BR+GK, while SVD++ outweighed GCMC on Epinions. Since NCF model was derived to predict if it is a hit for a user-item pair, we did not apply the model in this experiment. The result shows that our model facilitates capturing complex collaborative information more accurately than using a Gaussian Kernel based weighted combination of approximations.

On MovieLens 1M data set, we also investigated how the neighborhood size impacts the compared algorithms' performance. We examined the numbers of elements in all extracted neighborhoods in test set of the first CV fold. As shown in Fig.3 with the red line, around 70% of test cases had more than 14000 elements in its neighborhood. We also compared the performance of SVD++, BR+GK and CF_CNN on neighborhoods with different sizes. It obviously showed that the performance became better when a neighborhood contained more neighbor ratings and thus there existed more local approximations. The performance gain achieved by SVD++ and CF_CNN compared to BR+GK became larger when there are more elements in a neighborhood. In experiments, it is also worth noting that the RMSE changed little after elements reached a certain volume, therefore it is possible to use a sufficient sample set of neighbors to make estimations good enough in practice for reducing computational costs.

Besides having our model estimate true ratings, we also tested the model's performance on predicting responses such as like or dislike. On both used data sets, we transformed ratings {4, 5} to 1 for a like and {1, 2, 3} to 0 for a dislike. For SVD++, BR+GK and GCMC, we did not retrain the model with binary labels. We used

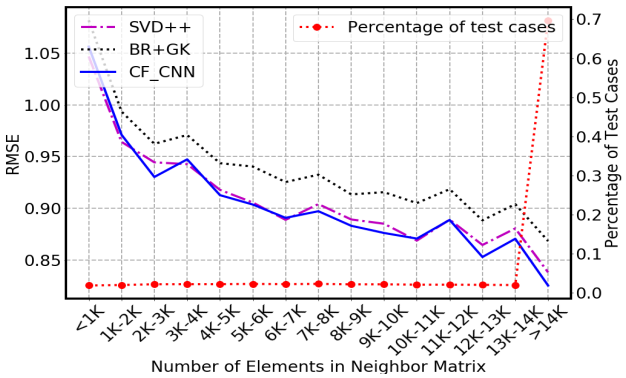
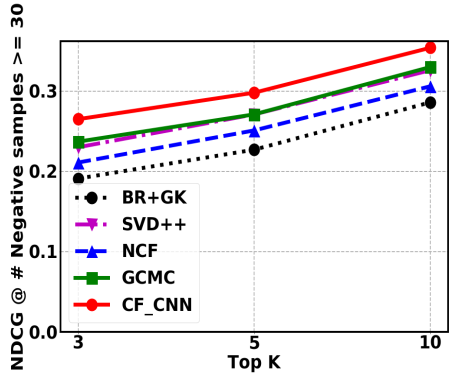
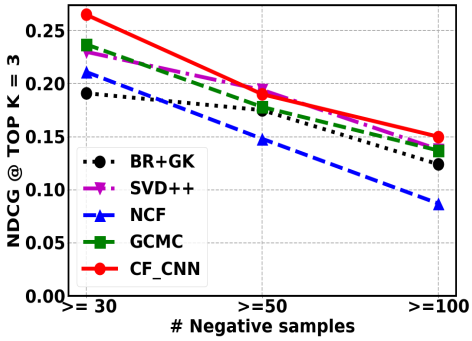


Figure 3: Impact of neighborhood size on RMSE for MovieLens 1M. The performance of compared methods improves as the size of neighborhood increases.

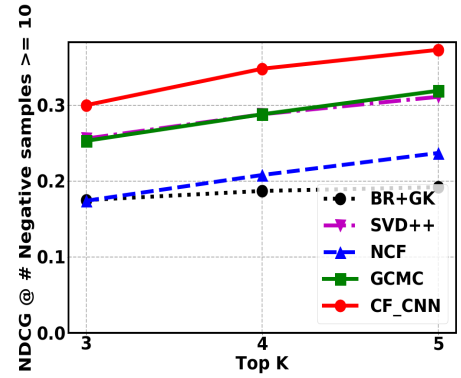


(a) NDCG with varying K

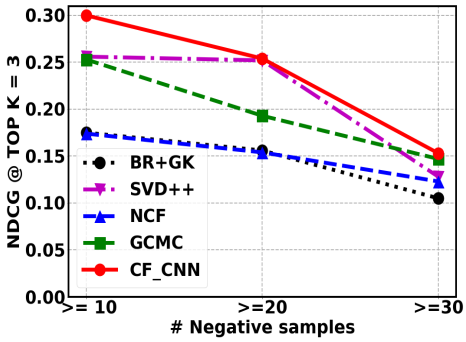


(b) NDCG with varying number of negative samples

Figure 4: Performance of top K recommendations on MovieLens 1M data set



(a) NDCG with varying K



(b) NDCG with varying number of negative samples

Figure 5: Performance of top K recommendations on Epinions data set

the output scores to rank candidate items for each user to test the performance on top K recommendations. We trained our proposed model and NCF model with binary labels and used probabilities of being a like to rank items for top K recommendations. To create the set of candidate items for one individual user, we randomly selected one positive sample and all negative samples from original test set in CV fold for a user, then we formed a new test set with selected (user, candidate items) pairs. We calculated normalized discounted cumulative gain (NDCG) for each user and reported averaged NDCG over all selected users. Results are shown in Fig. 4 and 5. On each data set we firstly defined a minimum of negative samples in candidates and experimented with varying K for Top K recommendations, then we fixed K and experimented with increased number of negative samples. From the results we observed that the proposed CNN based model achieved a better performance in general.

5 CONCLUSIONS

We investigate a novel Machine Learning algorithm based on CNN techniques to estimate unknown user ratings in Recommender System. Our method uses features extracted from the neighborhood of unknown ratings via local approximations to capture complex latent user-item interactions. Experimental results on two benchmark data sets demonstrate that our proposed method achieved a better performance compared to the state-of-the-arts. Based on our observations on the relationship between neighborhood size and performance, our future interests may include exploring a strategy to reduce the computation complexity by dynamically adjusting neighborhood size with slight decrease in recommendation performance.

REFERENCES

[1] MovieLens 1M. [n.d.]. https://grouplens.org/datasets/movielens/1m/.
[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* (2005).
[3] D.J. Aldous. 1981. Representations for partially exchangeable arrays of random variables. *J.Multivariate Anal.* 50, 1 (Jan. 1981), 581–598.
[4] Devavrat Shah Dogyoon Song Christina E. Lee, Yihua Li. 2016. Blind Regression: Nonparametric Regression for Latent Variable Models via Collaborative Filtering. In *Advances in Neural Information Processing Systems*.
[5] Epinions. [n.d.]. http://www.trustlet.org/epinions.html.
[6] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM* 35, 12 (Dec. 1992), 61–70.
[7] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Inf. Retr.* 4, 2 (July 2001), 133–151.
[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. 173–182.
[9] D. N. Hoover. 1981. Row-column exchangeability and a generalized model for probability. In *Exchangeability Probability and Statistics*. Rome, 281–291.
[10] Nicolas Hug. 2017. *Surprise, a Python library for recommender systems.*
[11] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. 1746–1751.
[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization.. In *International Conference on Learning Representations (ICLR)*.
[13] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, USA, 426–434.
[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42 (8 2009), 42–49.
[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105.
[16] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 297–305.
[17] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* (2009).
[18] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D Convolutional Networks for Session-based Recommendation with Content Features. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 138–146.
[19] R. van den Berg, T. N. Kipf, and M. Welling. 2017. Graph Convolutional Matrix Completion. *arXiv e-prints* (June 2017). arXiv:1706.02263
[20] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2643–2651.