Context-Regularized Neural Collaborative Filtering for Game App Recommendation

Shonosuke Harada*
Kyoto University
Kyoto, Japan
sh1108@ml.ist.i.kyoto-u.ac.jp

Kazuki Taniguchi
CyberAgent, Inc.
Tokyo, Japan
taniguchi_kazuki@cyberagent.co.jp

Makoto Yamada
Kyoto University/RIKEN AIP
Kyoto, Japan
myamada@i.kyoto-u.ac.jp

Hisashi Kashima
Kyoto University/RIKEN AIP
Kyoto, Japan
kashima@i.kyoto-u.ac.jp

*The work was performed during an internship at CyberAgent.

ABSTRACT

People spend a substantial amount of time playing games on their smartphones. Owing to growth in the number of newly released games, it is getting more difficult for people to identify which of the broad selection of games they want to play. In this paper, we introduce context-aware recommendation for game apps that combines neural collaborative filtering and item embedding. We find that some contexts special to games are effective in representing item embeddings in implicit feedback situations. Experimental results show that our proposed method outperforms conventional methods.

KEYWORDS

Recommender systems, Neural collaborative filtering, game app

INTRODUCTION

In recent years of information overload, recommender systems which offer appropriate items to each user in a personalized way are widely used and play a significant role [2, 6, 8]. Recommender systems

ACM RecSys 2019 Late-breaking Results, 16th-20th September 2019, Copenhagen, Denmark

are popular and successfully developed particularly in E-commerce services including YouTube [4], Netflix [1], and Amazon [11], to name a few.

Recommendation systems basically focus on predicting each user's preference for each kind of item. Collaborative filtering [13] is a widely used personalized recommendation method that recommends a new item using past user–item interactions. A typical collaborative filtering algorithm would be based on matrix completion [8], which decomposes a user–item matrix into user latent features and item latent features. For a long time, matrix completion algorithms based on factorization algorithms have been the first choice in recommender systems [8].

Recently, deep learning approaches [15–17] have gathered appreciable attention in the recommender systems community. However, a collaborative denoising autoencoder (CDAE) [17] could not improve its performance even if they use non-linear activation function and deeper models. One of the reason would be that CDAE equals to SVD++ when the identity function is used as an activation function and applies a linear kernel to model user-item interactions. [5].

To handle this issue, the neural collaborative filtering (NCF) [5] has been proposed, which was the first successful deep-learning-based collaborative filtering algorithm. NCF employs a simple neural network architecture consisting of only multi-layer perceptrons and a generalized matrix factorization (GMF). Thanks to its simplicity, it can train deep learning models without overfitting and, surprisingly, outperforms state-of-the-art collaborative filtering using only user–item information.

Another successful collaborative filtering algorithm is based on word embedding [10]. More specifically, pointwise mutual information (PMI) [3], which is computed from the item–user matrix, is used as a regularizer in addition to the matrix completion loss function. Thanks to PMI regularization, we can embed a similar item pair into a similar location at a latent space; this helps significantly to train deep learning models efficiently. This approach is promising. However, to the best of our knowledge, no deep-learning-based approaches have been put forward.

In this paper, we propose the context-regularized neural collaborative filtering (CNCF), which enjoys the representation power of deep learning and can be efficiently trained thanks to PMI-based regularization. Specifically, we naturally combine NCF and PMI regularization [10, 14], in which item latent vectors are shared in both NCF and PMI-based embedding. Thanks to its simplicity, CNCF can be efficiently trained using a standard deep learning package. Through experiments on real-world game app recommendation tasks, the proposed method significantly outperforms the vanilla NCF, which is a state-of-the-art recommender algorithm.

PROBLEM FORMULATION

Let $Y \in \mathbb{R}^{M \times N}$ be the user-item (game app) matrix whose elements are $y_{ij} = 1$ if the user i installed game app j and 0 otherwise. Let M and N be the number of users and the number of items (game apps), respectively. This is a standard implicit feedback setting. If $y_{ij} = 1$, it means that user i installed

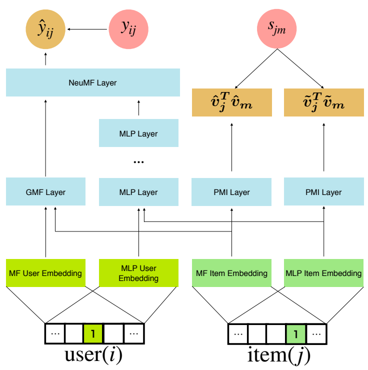

Figure 1: Overview of CNCF.

item j. However, in implicit feedback setting, the existence of interaction does not always mean that user i likes item j and unobserved interactions might assume that user i does not recognize item i.

In addition to the user-item matrix, for game app recommendation tasks, we have the first login timestamp, the last login timestamp, and the paid flag, respectively. To use this information for recommendations, we generate a time-dependent matrix $\boldsymbol{T} \in \mathbb{R}^{M\times N}$, where $\tilde{t}_{ij}$ is the difference between the first login timestamp and the last login timestamp and each $\tilde{t}_{ij}$ is later transformed into *normalized dwell time* [18]. Moreover, for the paid flag information, we extract the paid matrix $\boldsymbol{P} \in \mathbb{R}^{M\times N}$, where r_{ij} is 1 if the user u pays money for item i and 0 otherwise.

The final goal of this paper is to build a recommendation model for user-item matrix $\boldsymbol{Y}$ using the user-item matrices $\boldsymbol{Y}$, $\boldsymbol{T}$, and $\boldsymbol{P}$.

PROPOSED METHOD (CONTEXTUAL NCF)

In this section, we propose the contextual neural collaborative filtering (CNCF), which is an extension of the widely used NCF algorithm [5].

Model: The following model with one perceptron layer is used:

$$\hat{y}_{ij} = \sigma(\mathbf{h}^\top(\hat{\boldsymbol{u}}_i \otimes \hat{\boldsymbol{v}}_j \oplus a(W\left(\tilde{\boldsymbol{u}}_i \oplus \tilde{\boldsymbol{v}}_j\right)))),$$

where $\otimes$, $\oplus$, and $\mathbf{h}$, a indicate the element-wise product and the concatenation of the two embeddings, edge weights of the output layer as well as an activation function like *Relu*, respectively. Figure 1 shows the model architecture of CNCF. GMF layer indicates the element-wise product of two embeddings and, in the PMI layer, we compute the inner product of both MF and the MLP j-th item embedding along with the MF and MLP all item embedding. $\hat{\boldsymbol{u}}$,$\tilde{\boldsymbol{u}}$,$\hat{\boldsymbol{v}}$,$\tilde{\boldsymbol{v}}$ denote MF and MLP user embedding and MF and MLP item embedding, respectively. CNCF consists of generalized matrix factorization and multilayer perceptrons.

Using context information: As contextual features, we use time-dependent features $\boldsymbol{T}$ and a paid flag approach $\boldsymbol{P}$ as

$$\mathcal{L}_{\text{context}} = \begin{cases} \sum_{(i,j)\in\mathcal{Y}\cup\mathcal{Y}^-}(1+\alpha t_{ij})y_{ij}\log\hat{y}_{ij}+(1-y_{ij})\log\hat{y}_{ij}, (\text{time}-\text{NCF}) \\ \sum_{(i,j)\in\mathcal{Y}\cup\mathcal{Y}^-}(1+\beta r_{ij})y_{ij}\log\hat{y}_{ij} + (1-y_{ij})\log\hat{y}_{ij}, (\text{paid}-\text{NCF}) \end{cases},$$

where $\alpha \geq 0$ and $\beta \geq 0$ are tuning parameters and $\mathcal{Y}$ is the set of indices of non-zero elements in $\boldsymbol{Y}$ and $\mathcal{Y}^-$ is the set of indices of zero-elements in $\boldsymbol{Y}$. In implicit feedback settings, to address the problem of lacking negative data, treating all unobserved data as negative feedback [6] or negative sampling from unobserved data [5] are popular strategies. Note that we sampled user-item pairs as negative interactions from the unobserved interaction set.

Regularization based on Pointwise Mutual Information (PMI) In this paper, in addition to contextual information, we introduce an embedding structure to NCF since it helps to improve prediction accuracy [10, 14]. In particular, we employ the GloVe-based embedding approach [12]. The loss function of a GloVe can be written as

$$J = \sum_{j,m} \left(s_{jm} - \boldsymbol{v}_j^\top \boldsymbol{v}_m \right)^2,$$

where s_{jm} is some similarity measure between item j and item m. In this study, we employ positive pointwise mutual information (PPMI) [9] as a similarity measure:

$$\mathrm{PPMI}(x, y) = \max(\mathrm{PMI}(x, y), 0), \ \mathrm{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)},$$

where $p(x)$ denotes the probability of users installing a game app x and $p(x, y)$ denotes the probability that users install a game app x and y. Finally, the loss function of the context-regularized NCF is given as

$$\mathcal{L} = \mathcal{L}_{\text{context}} + \lambda \left(\sum_{s_{jm}>0} (s_{jm} - \hat{\boldsymbol{v}}_j^\top \hat{\boldsymbol{v}}_m)^2 + \sum_{s_{jm}>0} (s_{jm} - \tilde{\boldsymbol{v}}_j^\top \tilde{\boldsymbol{v}}_m)^2 \right), \quad (1)$$

where $\lambda \geq 0$ is the regularization parameter.

EXPERIMENTS

We gathered game app click information from a commercial game app company. Figure 2 shows some examples of game apps. Then, we used 100,000 users who had installed over 20 game apps and played one of their games within last two years. The number of games was 725 (i.e., $\boldsymbol{Y} \in \mathbb{R}^{100000,725}$), and the number of non-zero entries was 2,854,328.

We implemented all methods using Pytorch and ran experiments using a Tesla P100. We set the learning rate as 0.001 and the batch size as 1024. Then, we used Adam [7] as the optimizer. For the regularization parameters of CNCF, we used α=0.01, β=0.1, and $\lambda = 1$. For all experiments, we set the number of multi-layer perceptrons as four and the number of latent feature representations as 64. The initial model parameters were randomly initialized. We set the negative sampling ratio as 2 that means we sample 2 unobserved interactions as negative samples per one observed interactions for every user.

To evaluate the performance of the item recommendation, we used the *leave-one-out* scheme, which has been widely used in the relevant literature [5]. As evaluation metrics, we adopted *HitRatio* (HR) and *normalized discounted cumulative gain* (nDCG), which are also popular in recommendation tasks [5].

Figures 3 to 6, we show the results of the proposed and the existing methods. As can be seen, the proposed contextual NCF compares favorably with the existing state-of-the-art algorithms.

Figure 2: Examples of Game Apps.

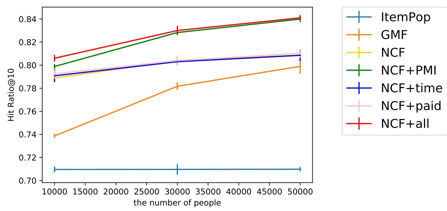

Figure 3: HitRatio@10 for every method.

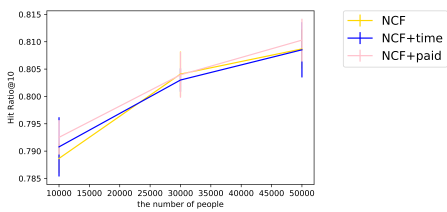

Figure 4: HitRatio@10 for methods related to NCF.

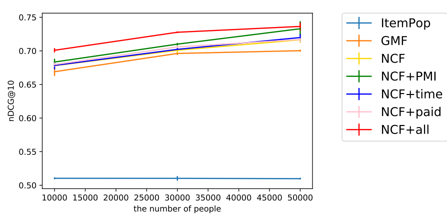

Figure 5: nDCG@10 for every method.

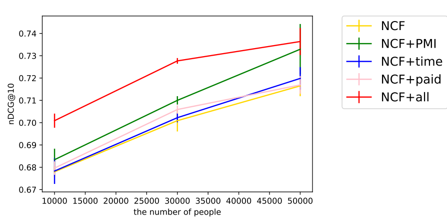

Figure 6: nDCG@10 for methods related to NCF.

REFERENCES

[1] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *KDD cup and workshop*.
[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *the 1st Workshop on Deep Learning for Recommender Systems*.
[3] Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics* 16, 1 (1990), 22–29.
[4] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *RecSys*.
[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*.
[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[8] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*.
[9] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *NIPS*.
[10] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *RecSys*.
[11] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 1 (2003), 76–80.
[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
[13] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*.
[14] Thanh Tran, Kyumin Lee, Yiming Liao, and Dongwon Lee. 2018. Regularizing Matrix Factorization with User and Item Embeddings for Recommendation. In *CIKM*.
[15] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.
[16] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *KDD*.
[17] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*.
[18] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *RecSys*.