A Learning to Rank method for Cross-market Recommendation

Peng Zhang
zzhangpeng@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Rongxiu Gao
gaorx@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Changyv Li
changyulve@gmail.com
University of Electronic Science and Technology of China
Chengdu, Sichuan, China

Baoyin Liu
liubaoyiny@gmail.com
Northeastern University
Shenyang, Liaoning, China

Sheng Zhou*
zhousheng_zju@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Abstract

The cross-market recommendation aims at recommending relevant products to users in a target market by leveraging data from similar high-resource markets. Although recommendation has been widely studied in the literature, the biases of the individual markets limit the generalization of model learned on individual market. Based on the experimental data provided by the WSDM Cup challenge, we study data on various markets and propose a learning to rank model by combining the individual similarity as well as the structural feature. More specifically, We first benchmark several collaborative filtering methods using a single target market data, including a lot of the classic approaches as well as the advanced approaches. Then we fit some of these approaches to a cross-market use. Finally, we use learn2rank method to fuse all the methods mentioned to boost the performance. We get 0.6746 on the leaderboard which is the third place of the challenge. The source codes have been released at https://github.com/miziha-zp/BiuG-XMRec-WSDMCup22.

CCS Concepts: • **Information systems → Data mining**; • **Computing methodologies → Knowledge representation and reasoning**.

Keywords: Cross-market Recommendation, Collaborative Filter,

*corresponding author

WSDM '22, February 21–25, 2022, Phenix, AZ

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Peng Zhang, Rongxiu Gao, Changyv Li, Baoyin Liu, and Sheng Zhou. 2022. A Learning to Rank method for Cross-market Recommendation. In *Proceedings of he 15th ACM International WSDM Conference (WSDM '22).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The cross-market recommendation aims at recommending relevant products to users in a target market by leveraging data from similar high-resource markets. The WSDM Cup 2022 track 3 Cross Market Recommendation focuses on improving recommendations to users in target markets (e.g., markets with scarce resources) by using data from similar high-resource markets to recommend relevant products, e.g., using data from U.S. markets to improve recommendations in target markets.

Recommendation in individual market has been widely studied in the literature, where methods including collaborative filtering, graph neural network and feature engineering have achieved tremendous success. A naive way for cross-market recommendation is that unify the data from different sources and apply existing success methods. However, the user-product interaction data and the user comment data may convey certain biases in individual markets.

In this paper, we provide our solution to the Cross Market Recommendation challenge. The general idea is to utilize four groups of representative methods, namely item similarity based methods, user similarity based methods, matrix factorization based methods and graph neural network based methods. We consider the characteristics of cross market recommendation and borrow idea from the domain adaptation and meta-learning. More specifically, we made corresponding adaptations for some of methods so that they can well fit the problem setting. Finally, we use a learning to rank framework to make the best of these methods in a unified framework.

2 Pipline

2.1 Dataset

The challenge provides user-item interactions with ratings as training data from three source markets, respectively s1, s2, s3, and two target markets data, respectively t1, t2. One interaction of every user in both source markets and target markets is split for validation, another interaction of every user in target markets is split for the test. A 5-core version dataset is provided to further facilitate the data preprocessing burden, which filtered users and items with less than five interactions.

2.2 Evalution

The goal is to have the best possible recommender system in terms of NDCG@10 on the target markets t1 and t2. The NDCG is defined as:

$$NDCG = \frac{1}{|U|} \sum_{i \in U} \frac{DCG_i}{IDCG_i} \tag{1}$$

where DCG_i is defined as follow and $IDCG_i$ is the ideal value of DCG_i coming from the best ranking:

$$DCG_i = \sum_{j \in \mathrm{Con}(i)} \frac{1}{\log_2 (\mathrm{rank}_{ij} + 1)} \tag{2}$$

Similar to [1], for every user in t1 and t2, score every item from the 100 candidate items, so as to get the best mean NDCG@10 of all users across the two markets. HR@10 is also considered for information purposes. For that, one can use the data on these markets and also get help from the data available from the source markets s1, s2, and s3.

3 Methodology

Despite the good results FOREC[2] achieved, it only studied to adapt NFM-like method[5] to this task. How to apply other CF methods to solve this cross market task has not been investigated. We first benchmark the performance of the classic CF approach in the case of using only the target market. Then we try to apply other CF methods to solve this cross market task. Finally, we used the learn2rank method to combine these methods to get the best results. We introduce our solution using the three sections below.

- We first benchmark collaborative filtering method on a single market, including a lot of the classic approaches as well as the advanced approaches.
- Then we try to fit some of these approaches to a cross-market use.
- For greater performance, we use a learn2rank method to fuse all the scores mentioned above.

3.1 Collaborative Filtering Method

In this section, we select some CF methods that have been widely proven to be effective to benchmark because there is so much research in this area in recent years.

3.1.1 Item Similarity Based Method.

Inspired by [10][1], We score candidates using statistical information related to their similarities to items that users have historically interacted. For instance, $N(u) = \{i_1, i_2, ...i_n\}$ is the items set that user u interacted. $S(u, i) = \{s_{i,1}, s_{i,2}...s_{i,n}\}$ is the similarities set, where $s_{i,k}$ is the similarity between item i and item $N(u)_k$. We used some statistical information from $S(u, i)$ as the final scoring. For the similarity calculation between the two items, different ways are adapted by us. We are surprised to find that the performance of these very simple methods can exceed those of the so-called state-of-the-art methods.

- IOU,intersection of union between the users set of two items.
- Cosine Multi-hot,cosine similarity between multi-hot representation of the users collections of two items.
- Cosine Item2vec, cosine similarity between the items vectors, which is gotten from the [1]

For the statistics for this similarity sequence, max, mean, std, median, length, 5%, 95% are adapted. We empirically find that IOU with mean may be the best choice.

3.1.2 User Similarity based methods.

Similar to these item similarity based methods, we use the similarity between the users who have historically purchased the candidate and the users we want to serve to score. These methods are slightly inferior to those item similarity based method.

3.1.3 Matrix Factorization(MF) based methods.

Rendle et al.[14] systematically and fairly tested those embedding based methods[14][7][5]. They show that with a proper hyper-parameter selection, a simple dot product substantially outperforms those MLP based methods. They provide invaluable experience in parameter tuning, for instance, a larger embedding size often achieves better results with proper regularization and learning rate. So we set embedding size as 2048, learning rate as 0.001, and weight decay as 0.005, for both t1 and t2. Different from [14], we adapt a BPR loss[13] instead of Binary Cross Entropy loss, set batch size as 8192 instead of 1 and a standard dot product MF is used instead of a MF with user and item bias term[12]. We empirically find that such a simple method MF achieves better performance than those similarity based methods and those so-called state-of-the-art methods.

3.1.4 Graph Neural Network based methods.

Some GNN based methods including LightGCN[6], UltraGCN[11] and gf-cf[15] are tested.

LightGCN simply removes the feature transformation and nonlinear activation in NGCF[16] which is empirically proved unnecessary. We carefully tune the parameters of LightGCN including embedding size, the layer of graph convolutional network (GCN), learning rate, weight decay, and batch size. Similar to MF, a larger embedding size can achieve

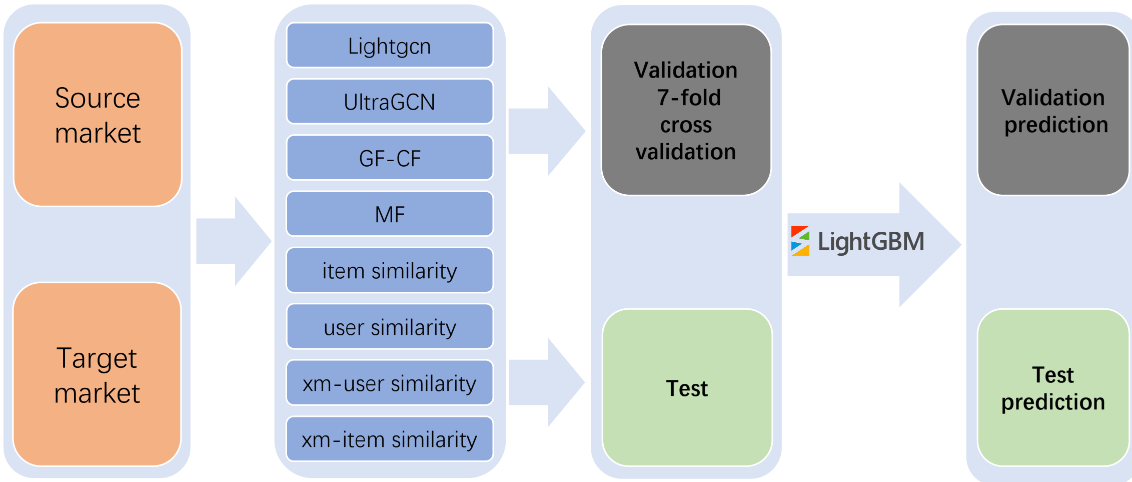

Figure 1. The overall framework of the proposed solution.

greater performance significantly. Finally, we set the parameters as MF and ensemble three models with different layers(respectively 3,4,5). LightGCN is the best CF method in our benchmark, which gets validation score 0.698 for t1 and 0.607 for t2.

UltraGCN[11] is an ultra-simplified formulation of GCNs, which skips infinite layers of message passing for an efficient recommendation. The UltraGCN has too many hyper-parameters. We only tune the hyper-parameter contained in MF and find that it is hard to get a better result than MF.

GF-CF[15] is a simple and computationally efficient method. GF-CF develops a general graph filter-based framework for CF, built upon the closed-form solution. We run the code from the author and the result is worse than MF.

3.1.5 Other Method.

We also run other four methods including EASE-r, ItemKNN, SLIM, Item2vec from the collection Open-Match-Benchmark[9], which has collected many classical as well as advanced CF methods. We did not find an impressive approach in these benchmarks.

3.2 Cross Market Recommendation

[2] investigate the problem of cross market recommendation, and proposed a NMF based methods named FOREC, which took advantage of the thoughts of domain adaptation and meta-learning. But other CF methods to solve this cross market task have not been investigated. In the section, we describe some of the efforts we have made. Some benchmarking results are shown in the Table 1.

3.2.1 Item Similarity Based Method for Cross Market.

As mentioned above, the similarity between two items can be the IOU of the collections of users who have bought these items. When it comes to the cross market setting, users in the source markets are considered to add to the collections. We simply take all the users in both the source markets and target markets into the collections, and then the score for t1 is improved from 0.676 to 0.686, and for t2 is improved from 0.556 to 0.566.

3.2.2 User Similarity Based Method for Cross Market.

Similar to the above section, we take all the users in both the source markets and target markets into the collections. We have empirically found that this can also improve the performance of those user similarity based methods.

3.2.3 Other Attempts.

We also benchmark the MF++ and NFM++ mentioned in [2], which take the source market data into the training process to improve the performance. But We empirically found that when the embedding size is large enough, the data from source market can hardly improve the performance and even disrupt its performance.

3.3 Learning to Rank

Using the single market based and cross market CF methods above, we can get the scores for every user-item pair of both validation data and test data, then we train a ranking model using validation data. LightGBM, short for Light Gradient Boosting Machine, is a free and open source distributed gradient boosting framework[17] for machine learning originally developed by Microsoft[8]. We adapt a lambdarank loss[3] for the ranking, which significantly improved performance by 0.005 compared to a logloss. Follow [4], 7-fold cross-validation is applied to get offline score. The leaderboard result is the average of the 7-fold predictions. To alleviate the problem of data scarcity and to improve the generalization ability of the model, we train a single LightGBM ranker using both market t1 and t2 data. The overall framework of our solution is shown in Figure 1.

In order to get a higher score further, we took some reasonable tricks: First, we just concatenate train and train5core

Table 1. Some benchmarking results of individual recommendation methods.

Market	T1		T2	
Method	NDCG@10	HR@10	NDCG@10	HR@10
LightGCN	0.698	0.806	0.607	0.721
MF	0.690	0.785	0.597	0.701
UltraGCN	0.681	0.780	0.577	0.677
GF-CF	0.675	0.761	0.556	0.650
XM-Itemcf-iou-mean	0.686	0.782	0.556	0.650
Itemcf-IOU-mean	0.677	0.761	0.566	0.663
LightGBM	**0.725**	**0.826**	**0.632**	**0.749**

data and remove the duplicate sections as our train data. Second, when scoring for test data, we add validation ground truth data to train.

4 Conclusion & Future Work

In this paper, we have introduced an empirical method for the Cross-market Recommendation of the WSDM Cup 2022. Our team *BiuG* was ranked third place on the final leaderboard. We tested several methods of success in the individual market recommendation, then adapt some of them to the cross market setup. Finally, we use a ranking model to synthesize all of the above methods. Despite the good results that have been achieved, we believe that a more simple and universal approach to adapting those popular methods can also be proposed in the future.

References

[1] Oren Barkan and Noam Koenigstein. 2016. ITEM2VEC: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. 1–6. https://doi.org/10.1109/MLSP.2016.7738886

[2] Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. *Cross-Market Product Recommendation.* Association for Computing Machinery, New York, NY, USA, 110–119. https://doi.org/10.1145/3459637.3482493

[3] Christopher Burges, Robert Ragno, and Quoc Le. 2007. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman (Eds.), Vol. 19. MIT Press. https://proceedings.neurips.cc/paper/2006/file/af44c4c56f385c43f2529f9b1b018f6a-Paper.pdf

[4] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, Article 27 (may 2011), 27 pages. https://doi.org/10.1145/1961189.1961199

[5] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 355–364. https://doi.org/10.1145/3077136.3080777

[6] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. *LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation.* Association for Computing Machinery, New York, NY, USA, 639–648. https://doi.org/10.1145/3397271.3401063

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE. https://doi.org/10.1145/3038912.3052569

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[9] Jan Lee. 1981. Transcript of question and answer session. In *History of programming languages I (incoll)*, Richard L. Wexelblat (Ed.). ACM, New York, NY, USA, 68–71. https://doi.org/10.1145/800025.1198348

[10] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80. https://doi.org/10.1109/MIC.2003.1167344

[11] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. *UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation.* Association for Computing Machinery, New York, NY, USA, 1253–1262. https://doi.org/10.1145/3459637.3482291

[12] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[14] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. *Neural Collaborative Filtering vs. Matrix Factorization Revisited.* Association for Computing Machinery, New York, NY, USA, 240–248. https://doi.org/10.1145/3383313.3412488

[15] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B. Khaled Letaief, and Dongsheng Li. 2021. *How Powerful is Graph Convolution for Recommendation?* Association for Computing Machinery, New York, NY, USA, 1619–1629. https://doi.org/10.1145/3459637.3482264

[16] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 165–174. https://doi.org/10.1145/3331184.3331267

[17] Jerry Ye, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng. 2009. Stochastic Gradient Boosted Distributed Decision Trees. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (Hong Kong, China) *(CIKM '09)*. Association for Computing Machinery, New York, NY, USA, 2061–2064. https://doi.org/10.1145/1645953.1646301