# An Effective Way for Cross-Market Recommendation with Hybrid Pre-Ranking and Ranking Models

The first-place entry for Cross-market Recommendation at WSDM Cup 2022

Qi Zhang
Interactive Entertainment Group of Netease Inc.
Guangzhou, China
zhangqi21@corp.netease.com

Zijian Yang
Interactive Entertainment Group of Netease Inc.
Guangzhou, China
yangzijian@corp.netease.com

Yilun Huang
Interactive Entertainment Group of Netease Inc.
Guangzhou, China
huangyilun@corp.netease.com

Jiarong He*
Interactive Entertainment Group of Netease Inc.
Guangzhou, China
gzhejiarong@corp.netease.com

Lixiang Wang
Southeast University
Nanjing, China
220191639@seu.edu.cn

## ABSTRACT

The Cross-Market Recommendation task of WSDM CUP 2022 is about finding solutions to improve individual recommendation systems in resource-scarce target markets by leveraging data from similar high-resource source markets. Finally, our team OPDAI won the first place with NDCG@10 score of 0.6773 on the leaderboard. [1] Our solution to this task will be detailed in this paper. To better transform information from source markets to target markets, we adopt two stages of ranking. In pre-ranking stage, we adopt diverse pre-ranking methods or models to do feature generation. After elaborate feature analysis and feature selection, we train LightGBM with 10-fold bagging to do the final ranking.

## KEYWORDS

Cross-Market Recommendation, LightGCN, LightGBM, WSDM Cup

**ACM Reference Format:**
Qi Zhang, Zijian Yang, Yilun Huang, Jiarong He, and Lixiang Wang. 2022. An Effective Way for Cross-Market Recommendation with Hybrid Pre-Ranking and Ranking Models: The first-place entry for Cross-market Recommendation at WSDM Cup 2022. In *Proceedings of Proceedings of the ACM International WSDM Conference (WSDM'22).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

E-commerce companies often operate across markets in different regions or countries around the world. How to leverage data from other markets to optimize the recommender system in a target market, namely Cross-Market Recommendation (CMR), becomes a novel and valuable topic in the industry [1]. In this WSDM Cup challenge, we participants are provided with user purchase and rating data from various markets, with a considerable number of shared item subsets. For online validation, we need to submit a sorted list of 100 candidate items for each user in valid and test sets of 2 target markets. The evaluation metric is weighted NDCG@10 in test sets of the 2 target markets. The rest of the paper is organized as follows. We first analyze the given dataset in Section 2. Section 3 describes the details of our solution for the challenge. Experiments are illustrated in Section 4. Finally, we conclude our work and discuss the future direction in Section 5.

**Table 1: General Description of Data**

| | |
|---|---|
| # markets | 5 |
| # samples | 1147289 |
| # users | 196951 |
| # items | 116207 |
| # unique items | 34740 |

## 2 EXPLORATORY DATA ANALYSIS

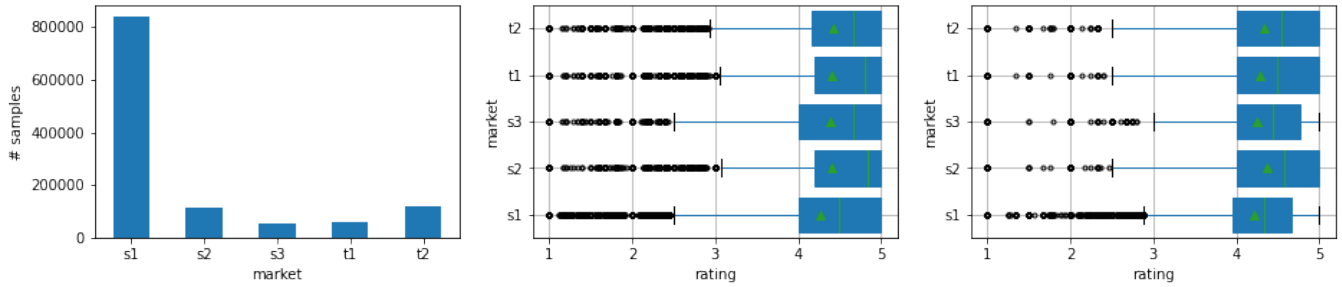
The competition organizer provides user-item-rating dataset from 5 markets, including 3 source markets (*s1*, *s2* and *s3*) and 2 target markets (*t1* and *t2*). Additionally, the dataset in each market consists of several parts which are *train_5core*, valid and test. Basic characteristics of the dataset are shown in Table 1 and Figure 1. Source market *s1* has much more samples than other markets, which may contain abundant information to boost our models. And the distributions of ratings which are mostly between 4 to 5 with average of about 4.6 are quite similar between markets, so that knowledge transfer directly based on ratings is probably reasonable.

The set of users in each market is mutually disjoint. However, items overlapped across markets are predominate in both target markets as shown in Table 2. So building a recommender system on target markets that makes better use of items' information from

*Corresponding author
[1]The code is available at https://github.com/opdai/wsdm2022-xmrec-top1-solution.


*WSDM'22, February, 2022, Phoenix, USA*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn




(a) Distribution of Samples (b) Average Rating of Items (c) Average Rating of Users

**Figure 1: Distributions of samples, items' rating and users' rating per market**

**Table 2: Summary of Overlap Items**

| Market | Total | *s1* | *s2* | *s3* | *t1* | *t2* |
|---|---|---|---|---|---|---|
| *t1* | 3429 | 3412 | 2634 | 2007 | - | 2037 |
| *t2* | 8834 | 8782 | 2733 | 1808 | 2037 | - |

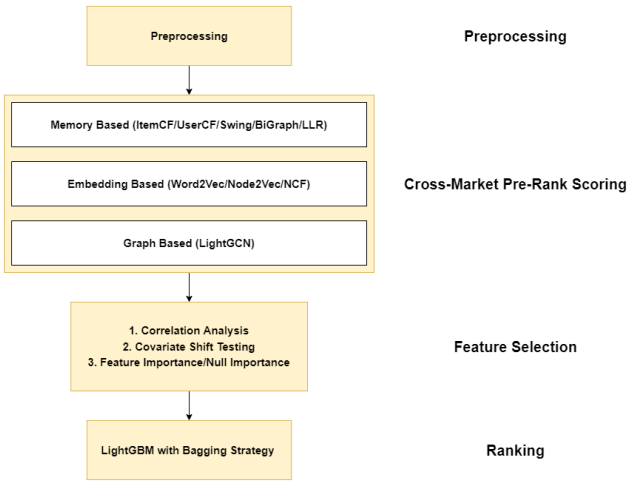



**Figure 2: An overall framework and pipeline of our solution**

other markets is quite important for this task from our perspective. *s1* is a high-resource market and almost contains all items in *t1* and *t2*. Particularly, the number of overlapped items between *t2* and *s1* are the biggest among others, so it's significant to transfer knowledge from *s1* to *t2*. How to better utilize the data and transfer knowledge from source markets is our major focus.

## 3 METHODOLOGY

Our solution for this task mainly consists of 4 steps, preprocessing, pre-rank scoring with cross-market data, feature selection and final ranking. This two stage training pipeline is very effective based on our experiment. The overall framework is shown in Figure 2.

Given that the dataset provided is of high quality, we do not do much preprocessing, only label encoding, dropping duplicates and

**Table 3: Dataset using in different stages**

| Stage | Dataset |
|---|---|
| Pre-Rank Scoring | All datasets in source markets + *train* & *train_5core* & cross-market *valid* in the target market |
| Final Ranking Model | *valid* of each target market |

marking scores in *train_5core* set as 5 for all when models using cosine function to measure the similarities. After preprocessing, we separate model training into two stages, pre-ranking and ranking. Pre-ranking stage is not for candidates generation but for getting diverse similarity scores between users and items which are used as features in the ranking stage afterward. Therefore, pre-rank scoring can be regarded as kind of pretraining as well. However, due to the different degrees of similarities between different markets, some of the features generated through pre-ranking stage could probably be redundant or deficient in some way for one of or both of the target markets. So we do feature selection before final ranking.

As shown in Table 3 below, we use different parts of dataset in different stages. In pre-ranking stage, all source market dataset is used, besides, for target markets, we exclude the *valid_run* set individually when doing pre-rank scoring for the corresponding target market to avoid label leakage in final ranking stage. For example, when doing training for *t1*, we exclude the *valid_run* dataset in *t1*, using all of the *train* and *train_5core* dataset and *valid_run* dataset in *t2*. *t2* training is following the same way.

### 3.1 Pre-rank Scoring

In pre-ranking stage, we use several different methods or models to get user/item representations and user-to-item similarity scores which represent that how much the user is interested in the given item. Based on pre-rank scoring, hundreds of diverse features are generated for the next stage of model training, which boost our final model significantly.

To be specific, the pre-ranking models can be classified into 3 categories, which are memory-based, embedding-based and graph-based models. For memory-based models, we use some traditional collaborative filtering (CF) models like ItemCF [7], UserCF [9], Swing [11], Loglikelihood Ratio (LLR) [2], Bi-Graph [12] to get

user-to-item similarity scores. These memory-based models above score 0.59-0.62 (NDCG@10, the same below) on the leaderboard. Especially for ItemCF, which takes the least time to train and performs the best among all of the memory-based models above.

As for embedding-based models, Word2Vec, Node2Vec [3] (both in DFS and BFS ways) and NCF [5] are used to generate user/item embeddings and user-to-item similarity scores. NCF perform the best among embedding-based models with the score of 0.61-0.62 on the leaderboard. Scores of the rest models are ranging from 0.35 to 0.46, which do not seem good enough comparing others. These models are probably not strong solo players, but also contributing to the final ranking models in some degree based on our experiment.

Node2Vec is a simple but efficient embedding-based model. Unlike Word2Vec and DeepWalk, Node2Vec uses a biased random walk procedure to efficiently explore diverse neighborhoods in DFS or BFS ways, and thus generate richer representations.

Graph-based models are really prevalent these years, and have become new state-of-the-art for collaborative filtering. To better represent users and items, we also adopt LightGCN [4] to do pre-rank scoring. Specifically, LightGCN learns user and item embeddings by linearly propagating them on the user-item interaction graph. For better performance, we use 4-layer LightGCN with the embedding dimension of 2048, nodes dropout rate=0.4 and learning rate=0.001.

Moreover, considering the cross-market differentiation for user/item representations and user-to-item interest quantification, we calculate the scores based on different market combinations. For example, to generate pre-ranking scores for *t1*, we can use market combinations of *s1-t1*, *s1-s2-s3-t1*, *s1-s2-t1*, *s1-s3-t1*, *s1-s2-s3-t1-t2*, etc. as pretraining corpus, likewise for *t2*. However, LightGCN training is relatively time-consuming, so we only conduct some of the combinations for *t2* later to get the final boosting. We use ItemCF as the example to show the results of this cross-market combination strategy in Table 4. All scores are offline NDCG@10 calculated with valid set. We can see that 10 features are generated through this process for each pre-ranking model. In most cases, our models perform better with more data. However, we can find that although the data in *s1* is richer than that in *s3*, *s3* still contributes more to *t1* than *s1* does with the scores 0.6805 versus 0.6781. In addition, it's obvious that different source markets have different contributions to the two target markets. *s3* contributes the most to *t1*, then *s2* and *s1* come after. As for *t2*, *s1* dominates the main contributions, thus *s3* and *s2* provide limit gains. The two target markets share different characteristics. By calculating pearson correlation coefficients between the 10 pre-rank scoring features generated through different market combinations, it's found that the pearson correlation coefficients in *t2* are much higher and of less difference between each other than that in *t1*, as shown in Figure 3. Obviously, *t2* market suffers serious multicollinearity problem when training the model with these features. Feature refining should be adopted to get a more precise and generalized model.

**Table 4: Result of ItemCF model with cross-market combination (all scores are offline NDCG@10 in valid set)**

| ID | *t1* Market Combinations (Num. of Rows) | *t2* Market Combinations (Num. of Rows) | *t1* score | *t2* score | *t1-t2* score |
|---|---|---|---|---|---|
| *s1-s2-s3-t1-t2* | *s1-s2-s3-t1-t2* (1182345) | *s1-s2-s3-t1-t2* (1179560) | 0.6843 | 0.5797 | 0.6142 |
| *s1-s2-s3* | *s1-s2-s3-t1* (1056685) | *s1-s2-s3-t2* (1116463) | 0.6850 | 0.5795 | 0.6143 |
| s0 | *t1* (60400) | *t2* (120178) | 0.6776 | 0.5589 | 0.5980 |
| *t1-t2* | *t1-t2* (186060) | *t1-t2* (183275) | 0.6789 | 0.5596 | 0.5989 |
| *s1-s3* | *s1-s3-t1* (926608) | *s1-s3-t2* (986386) | 0.6839 | 0.5793 | 0.6138 |
| *s1-s2* | *s1-s2-t1* (999572) | *s1-s2-t2* (1059350) | 0.6786 | 0.5793 | 0.6121 |
| *s2-s3* | *s2-s3-t1* (247590) | *s2-s3-t2* (307368) | 0.6847 | 0.5604 | 0.6014 |
| *s1* | *s1-t1* (869495) | *s1-t2* (929273) | 0.6781 | 0.5783 | 0.6112 |
| *s2* | *s2-t1* (190477) | *s2-t2* (250255) | 0.6789 | 0.5601 | 0.5992 |
| *s3* | *s3-t1* (117513) | *s3-t2* (177291) | 0.6805 | 0.5606 | 0.6002 |

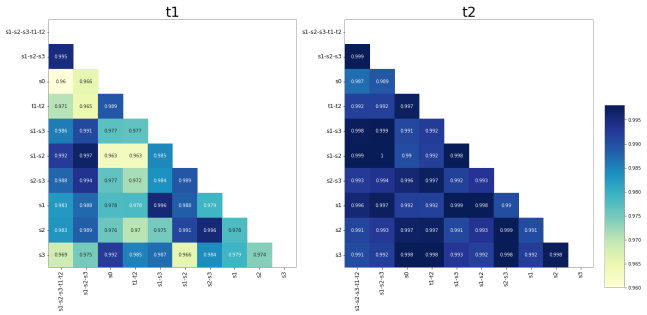



**Figure 3: Pearson correlation coefficients between the 10 pre-rank scoring features generated through different cross-market combinations in *t1* and *t2***

## 3.2 Feature Selection

Hundreds of features are generated from pre-ranking stage, final ranking models are trained based on these features. However, many features are probably suffering distribution shifted issue between training and test set or redundant with severe multicollinearity problems as the analysis shown above. To make the final model more robust, we do further feature analysis and selection before the final model training. Our feature selection mainly based on 3 factors, which are covariate shift test [10], offline cross-validation scores and feature importance analysis. At first, we conduct covariate shift test to exclude distribution-shift features. One of the basic and common assumptions for machine learning tasks is that training and test dataset are independent and identically distributed. Covariate shift is a form of distribution shift between training and test set which is a significant obstacle in developing robust machine learning models. So we need to exclude features whose distributions are significantly shifted between training and test set. Specifically, we

refer to the method mentioned in this article [10] to do the covariate shift test. Secondly, we do heuristic feature selection based on offline k-fold cross validation scores. Under the condition of fixed seeds, folds and model hyperparameters, we eliminate a feature or a group of similar features every time to do k-fold cross validation iteratively in heuristic way. At last, we do further feature analysis based on feature importance and null importance. Null importance [8] is a very prevalent feature selection method in Kaggle. Firstly, Null importances distributions are created by fitting the models over several runs on a shuffled version of the target. And then, feature selection is adopted by comparing the null importances distributions with the actual importances gathered by fitting the models on the original target. The distance between null importances distributions and actual importances should be in a big gap and the variance of the null importances should be high if the given feature is important. After feature selection, we only keep 206 features for target market *t1*, and 147 for *t2*. Especially the feature selection in *t2* market enables our model to get a significant boost on the leaderboard, which secure our top place on the leaderboard.

### 3.3 Final Ranking

Based on the features selected after pre-ranking stage, combining with some global statistic features, similarities calculated with pretrained Word2Vec embeddings, we build two LightGBM [6] classifiers to get the final ranking scores for *t1* and *t2* separately.

We do not do much tuning in this stage except for some searching for model parameters *num_leaves* and *learning_rate*, which are proven to be important for the final results according to our experiments. As for model ensemble, we simply adopt bagging training with 10-fold cross validation to get a more robust model for each target market.

## 4 EXPERIMENT

With our training strategy mentioned above, we can get NDCG@10 score of 0.6737 on the leaderboard without elaborate feature selection, and we achieve our *t1* sota with the score of 0.7384.

As shown in Figure 3 above, *t2* suffers from a more serious multicollinearity problem than *t1* does. Meanwhile, *t2* market weights more in the final score. So we did extra exploration for *t2*. According to our experiment, our offline cross validation score is reliable and almost aligns with the leaderboard score, so we can use limited online submitting opportunities more efficiently by validating our ideas based on enough offline experiments. To be specific, we dropped some redundant features and optimized LightGCN for *t2* with cross-market combinations like *s1-t2*, *s1-s2-t2*, *s1-s3-t2*, etc., and this helps us get the final boosting from 0.6737 to 0.6773 on the leaderboard. Some related results are shown in Table 5.

## 5 CONCLUSION & FUTURE WORK

In this paper, we propose an effective method to boost the cross-market recommendation. To better transfer information from source markets to the target markets and avoiding biases introduced, we separate our training pipeline into two stages of ranking scoring. In pre-ranking stage, we employ diverse methods or models to do feature generation by getting pre-ranking scores. After elaborate feature analysis and feature selection, we train LightGBM with 10-fold bagging to do the final ranking individually for each target market. Finally, our team OPDAI is ranked first on the final leaderboard with the *t1-t2* NDCG@10 score of 0.6773.

**Table 5: The results of final feature optimization for *t2***

| *t2* features | *t2* offline score | *t2* online score | *t1-t2* final score |
|---|---|---|---|
| Same as *t1* | 0.6347 | 0.6422 | 0.6737 |
| - Bi-Graph | 0.6348 | - | - |
| + Node2Vec | 0.6357 | - | - |
| + Swing | 0.6360 | - | - |
| - User Embeddings | 0.6366 | - | - |
| + Optimized LightGCN | 0.6378 | 0.6472 | 0.6773 |

Although our approach is effective to help us get the first place in this competition, end-to-end neural networks are much more concise and flexible to solve this task in a more elegant way. We'll leave it for future work.

## ACKNOWLEDGMENTS

XMRec organizing team paid a lot of efforts during the whole process of this competition, we really appreciate it for hosting this fantastic competition. And we would like to thank everyone associated with organizing and sponsoring the WSDM Cup 2022.

## REFERENCES

[1] Hamed R. Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-Market Product Recommendation. *CoRR* abs/2109.05929 (2021). arXiv:2109.05929 https://arxiv.org/abs/2109.05929

[2] Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Comput. Linguist.* 19, 1 (mar 1993), 61–74.

[3] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. *CoRR* abs/1607.00653 (2016). arXiv:1607.00653 http://arxiv.org/abs/1607.00653

[4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *CoRR* abs/2002.02126 (2020). arXiv:2002.02126 https://arxiv.org/abs/2002.02126

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. *CoRR* abs/1708.05031 (2017). arXiv:1708.05031 http://arxiv.org/abs/1708.05031

[6] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3149–3157.

[7] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Comput.* 7 (2003), 76–80.

[8] OLIVIER. 2018. Feature Selection with Null Importances. https://www.kaggle.com/ogrellier/feature-selection-with-null-importances.

[9] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (Chapel Hill, North Carolina, USA) *(CSCW '94)*. Association for Computing Machinery, New York, NY, USA, 175–186. https://doi.org/10.1145/192844.192905

[10] Shubham.jain. 2017. Covariate Shift – Unearthing hidden problems in Real World Data Science. https://www.analyticsvidhya.com/blog/2017/07/covariate-shift-the-hidden-problem-of-real-world-data-science/.

[11] Xiaoyong Yang, Yadong Zhu, Yi Zhang, Xiaobo Wang, and Quan Yuan. 2020. Large Scale Product Graph Construction for Recommendation in E-commerce. *CoRR* abs/2010.05525 (2020). arXiv:2010.05525 https://arxiv.org/abs/2010.05525

[12] Tao Zhou, Jie Ren, Matus Medo, and Yi-Cheng Zhang. 2007. Bipartite network projection and personal recommendation. *Physical review. E, Statistical, nonlinear, and soft matter physics* 76 (11 2007), 046115. https://doi.org/10.1103/PhysRevE.76.046115