A Practical Two-stage Ranking Framework for Cross-market Recommendation

Zeyuan Chen[1], He Wang[2], Xiangyu Zhu[3], Haiyan Wu[1], Congcong Gu[4], Shumeng Liu[2], Jinchao Huang[2], Wei Zhang[1*]
[1]East China Normal University, [2]Xiaomi Inc., [3]JD.com, [4]Pingan Inc.
chenzyfm@outlook.com,{wanghe11,liushumeng,huangjinchao1}@xiaomi.com,zhuxiangyu3@jd.com
gucongcong169@pingan.com.cn,hywuu@outlook.com,zhangwei.thu2011@gmail.com

ABSTRACT

Cross-market recommendation aims to recommend products to users in a resource-scarce target market by leveraging user behaviors from similar rich-resource markets, which is crucial for E-commerce companies but receives less research attention. In this paper, we present our detailed solution adopted in the cross-market recommendation contest, i.e., WSDM CUP 2022[1]. To better utilize collaborative signals and similarities between target and source markets, we carefully consider multiple features as well as stacking learning models consisting of deep graph recommendation models (Graph Neural Network, DeepWalk, etc.) and traditional recommendation models (ItemCF, UserCF, Swing, etc.). Furthermore, We adopt tree-based ensembling methods, e.g., LightGBM, which show superior performance in prediction task to generate final results. We conduct comprehensive experiments on the XMRec dataset, verifying the effectiveness of our model. The proposed solution of our team *WSDM_Coggle_* is selected as the second place submission[2].

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Personalization.*

KEYWORDS

cross-market recommendation, user behavior analysis, feature engineering

ACM Reference Format:
Zeyuan Chen[1], He Wang[2], Xiangyu Zhu[3], Haiyan Wu[1], Congcong Gu[4], Shumeng Liu[2], Jinchao Huang[2], Wei Zhang[1*]. 2022. A Practical Two-stage Ranking Framework for Cross-market Recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Phoenix, AZ, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

[1]https://xmrec.github.io/wsdmcup/
[2]The source code is available at https://github.com/loserChen/WSDM_CUP_Rec_2022

*All the corresponding to Wei Zhang.

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM '22, February 21–25, 2022, Phoenix, AZ, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recommender systems (RS) are ubiquitous in online platforms and mobile applications, such as e-commerce. Cross-domain recommendation is one kind of RS that leverages the interactions of overlapping items in source domains to benefit the recommendations in a target domain. There has been a number of studies with various domain definitions and recommendation scenarios [7, 15]. However, few studies have been conducted for cross-market recommendation.

The recommendation scenario of cross-market recommendation [2] is that the model learns from interactions of overlapping items in different markets to improve recommendation performance in a target domain, hoping to utilize information from rich source markets. Although it is crucial for E-commerce companies to combine different markets of various countries to solve the cold-start and data sparsity problems [18] occurred in a resource-scarce target market, little progress is made partly due to the lack of publicly available experimental data.

Thanks to the cross-market recommendation contest in WSDM CUP 2022, it provides the XMRec dataset which contains abundant user-item interaction records in different markets for further research purpose. To address this challenge, we introduce a practical two-stage ranking framework which contains hybrid model ranking and GBDT learning [13].

The rest of the paper is organized as follows. We first provide a brief data analysis in Section 2. Section 3 is about hybrid models ranking. GBDT learning is in Section 4, which includes feature engineering, models building, and ensemble modeling. Experimental results are illustrated in Section 5. Finally, we make a conclusion in Section 6. The overall framework of our approach is shown in Figure 1.

2 DATASET

The dataset[3] provided by the sponsor contains five folders: s1, s2, s3, t1, and t2. The folders s1, s2, and s3 contain the data of the source markets (train.tsv, train_5core.tsv, valid_qrel.tsv, and valid_run.tsv) for training and validating. The other folders t1 and t2 involve the data of the target market. Inside each, there are the training set (train.tsv and train_5core.tsv) and the public/private test set (valid_qrel.tsv and valid_ run.tsv/test_run.tsv), respectively. For ease of use, we simply concatenate train.tsv and train_5core.tsv by deleting the repeated samples and generate train_merge.tsv for each folder. Concretely, train_merge.tsv contains the training data with the fields of userId, itemId, and rating. valid_qrel.tsv is the validation positive samples, with a data structure similar to train.tsv.

[3]https://github.com/hamedrab/wsdm22_cup_xmrec

Table 1: Statistics of the datasets.

Dataset	# Users	# Items	# Interactions
s1	77,776	11,807	793,300
s2	20,311	3,408	794,477
s3	8,568	2,332	379,092
t1	9,955	3,559	599,600
t2	18,504	8,941	1,216,378

valid_run.tsv is the validation samples wherein each row has 99 negative samples and 1 positive sample for each unique userId. test_run.tsv is the test candidate samples with the same positive or negative sample ratio as valid_run.tsv. The basic statistics of the datasets are summarized in Table 1.

3 HYBRID MODEL RANKING

In order to improve the diversity of the model, we choose a few deep graph recommendation models and traditional recommendation models to implement result prediction. Since we do not significantly change these models, we just simply introduce them used in the contest. In what follows, we mainly introduce our ranking-augmented graph neural network based on practical findings during this contest.

3.1 Ranking-augmented Graph Neural Network

Before delving into the computational formulas of this module, we need to firstly clarify how to use the dataset for result prediction. We use the training set to train with sampling 99 negative items for each positive interaction, public test set to validate model performance and test candidate samples to predict without using any data from source markets, due to no distinct performance improvement by trying cross-market information. In order to distinguish the degree to which a user prefers an item, we build user-item bipartite graph $\mathcal{G}$ based on explicit feedback, i.e., rating. As such, we form the following formula, i.e., $((rating)/10 + 0.5) * 0.95 + 0.05/2$, which decides the fine-grained edge weights in graph $\mathcal{G}$. Assume there are M users and N items occurring in the target markets, then we have a node feature matrix $\boldsymbol{X} \in \mathbb{R}^{(M+N)\times d}$ and an adjacency matrix $\boldsymbol{A}^{(M+N)\times(M+N)}$ for the graph.

By convention, GNN performs representation along with the edges of graph $\mathcal{G}$, which is defined as follows:

$$\boldsymbol{X}_{l+1} = \hat{\boldsymbol{A}}\boldsymbol{X}_l + \boldsymbol{X}_l \odot \hat{\boldsymbol{A}}\boldsymbol{X}_l\,, \tag{1}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-0.5}\boldsymbol{A}\boldsymbol{D}^{-0.5}$ denotes the normalized adjacency matrix without self loops and l is the index of the propagation layer. We let $\boldsymbol{X}_0 = \boldsymbol{X}$, consisting of input user and item representations. Distinct from conventional graph convolution networks which do not consider information transfer between the adjacent layer, we additionally encode the similarity between $\boldsymbol{X}_l$ and $\boldsymbol{X}_{l+1}$ so that more messages from similar nodes can be passed [17]. By doing this, it can boost the ranking performance based on our extensive experiments.

After propagation of L layer, we obtain multiple layer-wise representations, namely $\{\boldsymbol{X}_0; \boldsymbol{X}_1; \ldots; \boldsymbol{X}_L\}$. Since the representations from different layers emphasize different semantics, we simply perform average-pooling on them to constitute embedding $\bar{\boldsymbol{X}}$. In order to mitigate the popularity bias in recommendation, we adopt a simple but effective solution, i.e., the L_2 based normalization operation, on $\bar{\boldsymbol{X}}$ so as to obtain final embedding $\tilde{\boldsymbol{X}}$.

Based on the final embedding $\tilde{\boldsymbol{X}}$, we could have two representations for user u and item i, i.e., $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$. To predict the potential interaction probability for the considered user-item pair, we proceed some changes to fuse GMF and MLP referred to classical neural collaborative filtering method NeuMF [6] and improve the ability of model prediction, which can be defined as follows:

$$\hat{y}_{ui} = \sigma(z_{GMF} + z_{MLP}) = \sigma(\boldsymbol{h}^T(\boldsymbol{e}_u \odot \boldsymbol{e}_i) + \text{MLPs}(\boldsymbol{e}_u \oplus \boldsymbol{e}_i))\,, \tag{2}$$

where $\boldsymbol{h} \in \mathbb{R}^{d\times 1}$ is a trainable weight vector. ReLU is adopted as the middle-layered activation function in MLPs. $\odot$ denotes the element-wise product of vectors and $\oplus$ means the concatenation operation. σ is the sigmoid function.

For training this module, we employ the cross-entropy loss, which is given by:

$$\mathcal{L} = -\big(y_{ui}\log\hat{y}_{ui} + (1-y_{ui})\log(1-\hat{y}_{ui})\big)\,. \tag{3}$$

To avoid overfitting user preferences, the label smoothing trick [16] is adopted based on corresponding ratings. Finally, we keep the prediction score of public/private test set from this module for the follow-up experiments.

3.2 Other Used Recommendation Models

Unlike the way of using data mentioned above, the recommendation models adopt another way to utilize the datasets. Here we use the training set to build the corresponding features and give predictions on the public/private test set. Similarly, we keep the corresponding score for further research. It is worth noting that the training set used here includes all the data from source markets and target markets so as to learn cross-market information and improve model performance. Unless otherwise specified, the data usage setting is used. The used recommendation models will be introduced simply. Generally, these models are usually applied in the recall stage in industrial scenarios.

3.2.1 ItemCF. The core idea behind ItemCF [12] is to recommend items that is similar to items of interest to the user in the past. ItemCF focuses on maintaining users' historical interests to make recommendations more personalized and reflects users' own interest inheritance.

3.2.2 UserCF. Similarly, UserCF tends to recommend items that are liked by other users with similar interests. And UserCF focuses on the hot spots of small groups similar to users' interests. The recommendation results are more social and reflect the popularity of items in users' interest groups.

3.2.3 Swing. Swing considers local graph structure relations such as user-item-user. For users who click on items i and j together, the fewer items they click on, the more similar items i and j are.

3.2.4 PersonalRank. PersonalRank [14] is a graph-based algorithm which utilizes random walk iteratively and the access probability of nodes gradually converges.

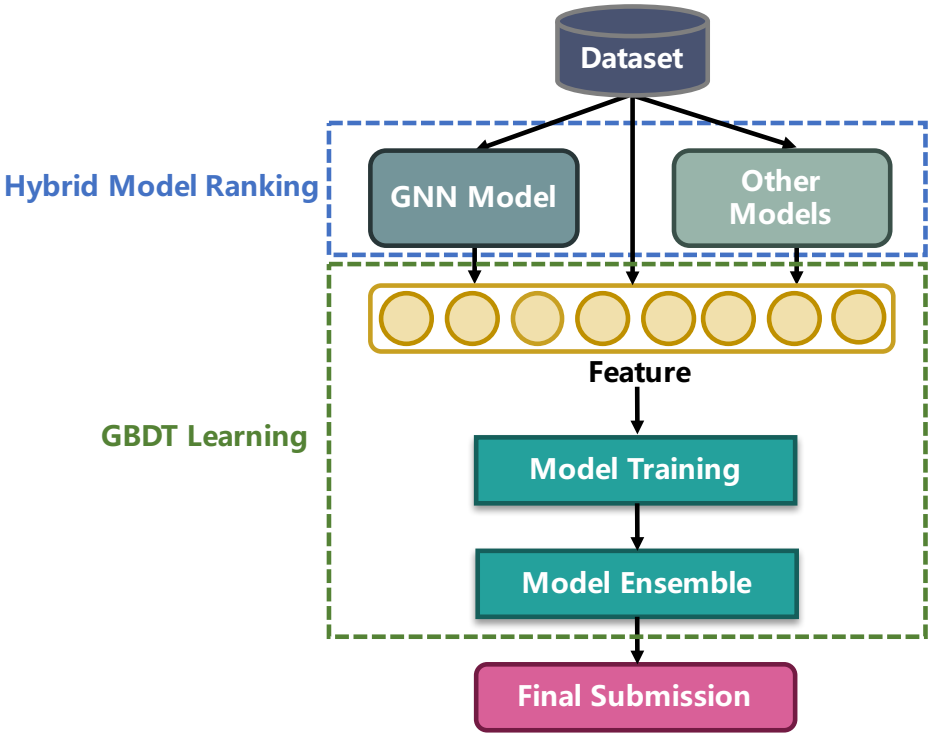

Figure 1: The whole framework of our model to address the challenge of cross-market recommendation .

4 GBDT LEARNING

Thanks to high efficiency and the superior ability of GBDT, we choose GBDT as our base model which is a popular machine learning algorithm, and has quite a few effective implementations such as XGBoost [3], LightGBM [10], and CatBoost [4]. By convention, feature engineering is a pivotal process especially for these tree-based ensembling methods. In order to improve the fitting ability of the whole framework, we investigate novel learning strategies to get different models. Finally, ensembling modeling is adopted which always works very well in different contests.

4.1 Feature Engineering

Except for regarding the prediction results generated by the above recommendation models as ranking features, we also construct statistical features, embedding features, and distance features. These features could capture useful information across different markets, so as to benefit the contest.

4.1.1 Statistical Features. Taking user statistical features as an example, we generate the corresponding features based on the userId, itemId and rating fields such as the count of purchasing, the number of types of items purchased as well as computing minimum, median, maximum, mean and standard deviation of ratings given by each user. And it is analogous to item statistical features.

4.1.2 Embedding Features. We build user and item embedding features respectively and the following introduces the embeddings we used in this contest:

- TF-IDF [9] is always applied to NLP and information retrieval fields, which is a statistical quantity for measuring the importance of a word with respect to a document. As for recommendation, we can think of users and items as words to achieve a similar goal. To avoid dimension explosion, the SVD method [1] is used to remove unimportant components.
- Word2Vec [5] is a neural network model to generate vector representations of words, which also can be used for generating user and item representations.
- DeepWalk [11] could learn network embedding by proceeding truncated random walk based on a user-item interaction graph.

4.1.3 Distance Features. Based on the embedding features we have constructed, we compute cosine distance, Manhattan distance, Jaccard coefficient, euclidean distance and Pearson correlation coefficient to measure the correlation between the two embeddings of the same type from users and items, and the correlation values are used as our distance features.

4.2 Model Training

Here we introduce learning strategies in detail. Based on features introduced above, we use multi-fold cross validation against the public test set of target markets to finish predicting test candidate samples.

In essence, recommendation is the task of learning to rank, which mainly includes three types of learning strategies, i.e., pointwise, pairwise and listwise. The pointwise learning regards the ranking problem as a classification or regression problem. However, the

Table 2: Main results w.r.t. NDCG@10 and HR@10 for cross-market recommendation on target market datasets. The best and second-best performed methods in each metric are highlighted in "bold" and underline, respectively.

Method	t1		t2	
	NDCG@10	HR@10	NDCG@10	HR@10
GNN	0.7131	0.7968	0.6226	0.7255
ItemCF	0.5955	0.6782	0.5235	0.6136
UserCF	0.6136	0.6948	0.5472	0.6547
Swing	0.6049	0.6834	0.5489	0.6350
PersonalRank	0.5922	0.6756	0.5582	0.6618
LightGBM*	0.7197	0.8224	0.6286	0.7421
LGBMRanker*	0.7039	0.7801	0.6381	0.7506
CATBoost	0.7310	0.8246	0.6371	0.7474
XGBoost	0.7354	0.8283	0.6391	0.7512
LightGBM	0.7362	0.8324	<u>0.6399</u>	0.7534
LGBMRanker	<u>0.7388</u>	<u>0.8356</u>	0.6388	<u>0.7555</u>
Model Ensemble	**0.7393**	**0.8369**	**0.6457**	**0.7610**

pairwise learning does not care about the specific value but only considers the relative order. As for the listwise learning, it tackles the ranking problem directly by optimizing the defined loss function on a list of items.

Thanks to the effective implementation of GBDT's pointwise and pairwise learning versions. We use LightGBM, XGBoost, CatBoost to train by pointwise learning and LGBMRanker [8] to train by pairwise learning. Thus, we could obtain four final models.

4.3 Ensemble Modeling

In the model ensemble stage, we simply adopt the weighted average operation to fuse these four models to get the final results. In fact, we have stacked predictions from multiple models to build these tree-based ensembling models, which also belongs to the model ensemble operation.

5 EXPERIMENTS

We evaluate our method on the XMRec dataset provided by the contest. Table 2 presents the overall performance of our model and all the adopted baselines, from which we have the following key observations:

- In the first part of the table, GNN achieves the best performance on both datasets compared to other traditional models. It may be attributed to the superior power of GNN and our effective module design.
- LightGBM* and LGBMRanker* are the models without using ranking features generated by the hybrid ranking models. By fine-grained feature engineering, they could also generate comparable performance.
- As for the third part of the table, it proves that the GBDT models could obtain major improvements by stacking learning models and multiple models ensemble.

6 CONCLUSION

In this paper, we have introduced our practical two-stage ranking framework for the cross-market recommendation competition of the WSDM Cup 2022. Our team ranks the second place on the final leaderboard[4] with an excellent performance very close to the first place. In our solution, we first conduct various models to give predictions. After that, we train 4 GBDT models by utilizing different learning strategies and implementations of GBDT. Finally, we ensemble these 4 models by weighted average operation. The comprehensive experiment results demonstrate the superiority and effectiveness of our model.

ACKNOWLEDGMENTS

We thank everyone associated with organizing and sponsoring the WSDM Cup 2022.

REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing* 54, 11 (2006), 4311–4322.
[2] Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-Market Product Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 110–119.
[3] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*. 785–794.
[4] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).
[5] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.
[7] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 667–676.
[8] Paweł Jankiewicz, Liudmyla Kyrashchuk, Paweł Sienkowski, and Magdalena Wójcik. 2019. Boosting algorithms for a session-based, context-aware recommender system in an online travel domain. In *Proceedings of the Workshop on ACM Recommender Systems Challenge*. 1–5.
[9] Li-Ping Jing, Hou-Kuan Huang, and Hong-Bo Shi. 2002. Improved feature selection approach TFIDF in text mining. In *Proceedings. International Conference on Machine Learning and Cybernetics*, Vol. 2. IEEE, 944–946.
[10] 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. *Advances in neural information processing systems* 30 (2017).
[11] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 701–710.
[12] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.
[13] Wen Wang and Wei Zhang. 2017. Combining multiple features for image popularity prediction in social media. In *Proceedings of the 25th ACM International Conference on Multimedia*. 1901–1905.
[14] Chen Yang, Tingting Liu, Lei Liu, and Xiaohong Chen. 2018. A nearest neighbor based personal rank algorithm for collaborator recommendation. In *Proceedings of the 15th International Conference on Service Systems and Service Management*. IEEE, 1–5.
[15] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. *arXiv preprint arXiv:1905.10760* (2019).
[16] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3903–3911.
[17] Wei Zhang, Zeyuan Chen, Hongyuan Zha, and Jianyong Wang. 2021. Learning from substitutable and complementary relations for graph-based sequential product recommendation. *ACM Transactions on Information Systems (TOIS)* 40, 2 (2021), 1–28.
[18] Wei Zhang and Jianyong Wang. 2015. A Collective Bayesian Poisson Factorization Model for Cold-start Local Event Recommendation. In *SIGKDD*. 1455–1464.

[4] https://competitions.codalab.org/competitions/36050#results