A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation

Weijia Zhang*†
zhangweijia@meituan.com
Meituan
Shanghai, China

Jin Zhan*
senkin13@gmail.com
DataRobot
Tokyo, Japan

Zhongshan Huang*
huang4211819@126.com
No affilation
Beijing, China

Lu Wang
wlu@microsoft.com
Microsoft
Beijing, China

Qiang Wang
qiang.wang@meituan.com
Meituan
Shanghai, China

ABSTRACT

To encourage the development of multilingual recommendation systems, Amazon pushlished a multilingual and multi-locale shopping session dataset, and KDD Cup 2023 challenge on Multilingual Session Recommendation Challenge was hosted based on this dataset.

In this paper, we present our solution for this competition. Following a widely-used setting in recommender system, our solution consists of two stages: recalling and ranking. In the first stage, we utilize various recalling strategies to retrieve a set of candidate products, including covisit matrix based collective filtering, graph embedding based I2I searching, text transformer based I2I searching and BPR based U2I searching. In the second stage, we develop a model to predict the probability of each user engaging with the candidate products. This model is an ensemble of two Catboost models, which include various statistical features and embedding similarity features. Finally, we achieved 4th place in Task1 and 3rd place in Task2.

CCS CONCEPTS

• **Information systems** → **Personalization**.

KEYWORDS

recommender systems, sequential recommendation

ACM Reference Format:
Weijia Zhang, Jin Zhan, Zhongshan Huang, Lu Wang, and Qiang Wang. 2018. A Hybrid Approach of Statistics and Embeddings for Multilingual and Multi-Locale Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

*Equivalent contributions.
†Corresponding Author

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

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

1 INTRODUCTION

1.1 Background

Recommender systems, also known as RecSys, play a crucial role in various online services, including e-commerce, social media, news platforms, and online video streaming. Among the different tasks performed by these systems, session-based recommendation, which utilizes customer session data to predict their next interaction, stands out as an essential function.

However, session-based recommendation in the realm of real-world scenarios involving multilingual and imbalanced data has received limited attention in previous studies.

To bridge this research gap, Amazon introduced the "Multilingual Multi-locale Shopping Session Dataset"[3], which comprises millions of user sessions originating from six distinct locales, and hosted KDD Cup 2023 on Multilingual Session Recommendation Challenge based on this dataset.

1.2 Datasets And Tasks

The dataset encompasses products in 6 languages, namely English, German, Japanese, French, Italian, and Spanish. Notably, the dataset exhibits an imbalanced distribution, with a lower number of French, Italian, and Spanish products compared to English, German, and Japanese, as shown in Table 1.

With this dataset, three tasks were introduced:

- *Next Product Recommendation*: Given a session data and the attributes of each product, the goal of this task is to predict the next product that a customer is likely to engage with. The test set for Task 1 comprises data from popular locales, i.e., English, German, and Japanese locales.
- *Next Product Recommendation for Underrepresented Languages and Locales*: The goal of this task is similar to Task1 while the test set is constructed from the underepresented locales, i.e., French, Italian, and Spanish.
- *Next Product Title Generation*: Given a session data and the attributes of each product, the goal of this task is to predict the title of the next product that a customer will engage with. Unlike Tasks 1 and 2, which focus on recommending existing products, predicting new or "cold-start" products presents a unique challenge.

Our primary focus was on Task1 and Task2. Section 2-5 of this paper will provide a detailed description of our solution for Task1

	Locale	# of session	# of items	# of interactions
Task1	English	1,182,181	500,180	4,872,506
	German	1,111,416	518,327	4,836,983
	Japanese	979,119	395,009	4,388,790
Task2	French	117,561	44,577	416,797
	Italian	126,925	50,461	464,851
	Spanish	89,047	42,503	326,256

Table 1: Details of the Dataset

and Task2, encompassing preprocessing methods, recall strategies and ranking models. Furthermore, the experimental results will be presented in Section 6.

2 OVERVIEW

As depicted in Figure 1, our solution is composed of two main stages: recalling and ranking. Although these stages are distinct, they share a significant amount of data. Therefore, we first generate multiple matrices and embeddings, which serve as the foundation for both recalling and ranking.

The first stage is candidate item recalling, which involves retrieving a set of potential products that each user may be interested in from the entire pool of products. This is achieved by utilizing the preprocessed matrices and embeddings. Approximately 200 items are retrieved for each user.

The second stage is candidate item ranking, where we predict the probability of a user engaging with each retrieved item. The items are then sorted based on these probabilities, and the top 100 items with the highest probability are recommended.

3 PREPROCESS

To achieve effective recall and ranking, we utilize a variety of features and models in both processes. In this section, we will elaborate on the generation of these commonly used data, which form the fundamental basis for our recall and ranking methods.

3.1 Covisit Matrix

One of the key features we utilize is the fact that items visited within the same user session often exhibit similarities. To capture this, we constructed a co-visit matrix where each entry represents the number of times item i and item j were visited together within the same user session.

Furthermore, we devised three types of weights to address biases that may arise from different user behaviors. The formula for calculating each entry of the co-visit matrix is as follows:

$$e(i,j) = \sum_{u \in U} \sum_{(i,j) \in u} \frac{1}{W_p(P_{i,j}) + W_c(C_i) + W_c(C_j) + W_l(L_s)} \quad (1)$$

$$W(x) = x^{\alpha} \quad (2)$$

- i and j represent item i and item j, respectively, while s and S denote user session u and the full session set, respectively.
- The closer the items are in the session, the higher their similarity. We use $P_{i,j}$ to denote the positional difference between item i and item j and we use different α values in $W_p(\cdot)$ to promote diversity, like 0.7, 1.0 and so on;
- Popular items, which are visited more frequently, have greater opportunities to calculate similarity with other items. To address this, we introduce a weight factor. C_i and C_j denote the total visited count of item i and item j in the training data, and we use different α values in $W_c(\cdot)$ to promote diversity, like 0.4, 1.0 and so on;
- Covisit in the long sessions also need to be penalized. To address this, we introduce another weight factor. L_s denotes the length of $session_s$ and we use different α values in $W_l(\cdot)$ to promote diversity, like 0.7, 1.0 and so on.

3.2 Text Transformer

In this dataset, there are several text features associated with items, such as title, description, color, material, and more. To leverage these textual information, we employed transformers using two approaches:

- Pretrained sentence transformer[6]. We use a pretrained sentence transformer for multilingual text [1] to encode each attribute of an item into a vector.
- Fine-tuned transformer with contrastive loss. To improve generalization in the competition dataset, we fine-tuned the transformer by referring to the method described in [4]. First, we formulated an item as a "sentence" by flattening its key-value attributes described by text. Then, we used a transformer to encode this "sentence" into a vector, denoted as h_s. Finally, a contrastive loss is utilized to fine-tune the transformer:

$$L = -log \frac{e^{sim(h_s, h_i^+)/\tau}}{\sum_{i \in \beta} e^{sim(h_s, h_i)}} \quad (3)$$

where sim is the cosine similarity function; h_i^+ is the embedding of the next item to $item_s$ in the user session; β is all items in the batch and τ is a temperature parameter.

3.3 Graph Embedding

Because of the ability to utilize high order connection, graph neural network has been widely used in the recommender system[2, 5, 8]. In our approach, we first build a large item graph, where the nodes represent items and an edge consisting of <item i, item j> denotes a successive visit of item i and item j. We then use ProNE[9] to extract a 2000-dimensional graph embedding for each item. The larger the embedding dimension, the better the performance, but it also slows down the process. We chose ProNE for several reasons:

- ProNE uses Sparse Randomized tSVD for fast embedding, which makes the process faster.
- ProNE uses Spectral Propagation for embedding enhancement, which makes the embeddings more accurate.

3.4 Item2vec

We treat all the Items in the same session as a sentence, and then train a word2vec skip-gram model to extract item embeddings.

[1] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

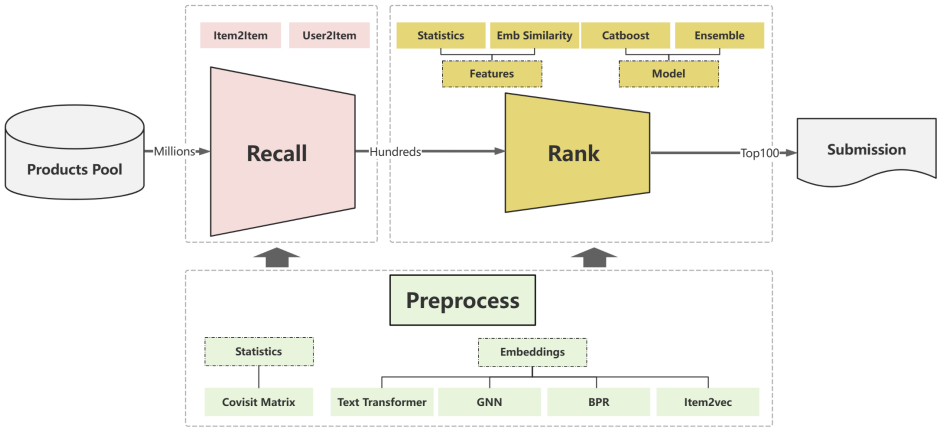

Figure 1: Architecture of our solution.

3.5 Bayesian Personalized Ranking

Bayesian Personalized Ranking[7], a.k.a BPR, is a commonly used matrix factorization method for learning personalized rankings from implicit feedback. It uses stochastic gradient descent (SGD) to optimize a pairwise ranking objective function, as shown in the Equation 4:

$$argmax_\theta l(\theta) = \sum ln\sigma(x_{ui} - x_{uj}) + \lambda||\theta||^2 \quad (4)$$

where i is the postive item and j is the random sampled negative item; θ is the embedding table of users and items and x_{ui} is the similarity of user embedding and item embedding; σ is the sigmoid function and λ is the regularization parameter.

4 RECALLING

In this section, we explain the process of recalling candidate products, which is based on the co-visit matrix and various embeddings described in Section 3.

4.1 Covisit Matrix Based Collaborative Filtering

The top N items that are most similar to the ones that the user has interacted with will be recalled, where the covisit matrix based similarity between users and items is as follows:

$$Sim(u, i) = \sum_{j \in u} W_p(p_j)e(i, j) \quad (5)$$

where u and i represent user u and item i, respectively; $W_p(\cdot)$ is defined in the Equation 2, with a α value of 1.0; p_j represents the position of item j in the user session, with larger values indicating later positions; $e(i, j)$ is defined in the Equation 2.

4.2 Item Embedding Based I2I searching

The recall strategy employed here is similar to collaborative filtering based on covisit matrices. The only difference lies in the calculation of similarity, which is based on the cosine similarity between the embeddings of item i and item j. These embeddings are generated using techniques decribed in the previous sections, such as text transformer embeddings, graph embeddings, and item2vec embeddings.

4.3 BPR Embedding Based U2I searching

This recall strategy involves retrieving items whose embeddings, generated using BPR, are similar to the user's embedding, also generated using BPR.

5 RANKING

We use a popular GBDT model, Catboost [1], to rank the recalled candidate items. In this section, we will provide details about the ranking model.

5.1 Statistical Features

We create statistical features in three aspects: item, user, and user crossing item.

- Item Statistics: Count of interactions in each locale, target locale and all the locales; price in each locale and target locale.
- User Statistic: Nunique and count of interacted items and their brand, color, size, model, and author; mean, median, max, and min price of interacted items.
- User Item Cross Statistics:
  - Count of candidate item's brand, color, size, model and author in user-interacted items.
  - Max, mean, min, sum, last and position-weighted sum of covisit score of session's items and candidate item.
  - Min and last of character-level ASID difference of user-interacted items and candidate item.
  - Max, mean, min, sum, last and position-weighted sum of character-level LCS (Longest contiguous matching subsequence) distance of user-interacted items and candidate item.
  - The rank number of the candidate item under each recalling strategies.

5.2 Embedding Similarity Features

The similarity between the user and candidate item, based on the embeddings described in Section 3, show a very impressive effect.

- I2I embedding similarity. This refers to the similarity between the candidate item and the user interacted items. Equation 6 represents the process of calculating I2I similarity.

$$Sim(u, i) = Aggr(cos\_sim(e_i, e_j)|j \in u) \quad (6)$$

where $Aggr$ is the aggregation function, such as max, mean, min, and sum. e_i and e_j are the embeddings of item i and item j, which can be graph embeddings, text transformer embeddings, and item2vec embeddings.
- U2I embedding similarity. As we factorize user and item with BPR, it's natural to use the similarity of user embedding and item embedding as a feature.

5.3 Model

The ranking model in our final submission is an ensemble of two models from different teammates. One model is from Jin Zhan and Zhongshan Huang, and the other is from Weijia Zhang. To ensure diversity, we did not collaborate much until the end of the competition. Below are the details of our individual models and the ensemble method.

5.3.1 Single Model. The first aspect of the model is the training samples. Recalled candidate items that the user did not interact with are labeled with 0, while those that the user interacted with are labeled with 1. Due to significant class imbalance, downsampling is applied. The negative samples are kept at a quantity of n times the length of the positive samples. In Jin Zhan and Zhongshan Huang's model, n is 10, while in Weijia Zhang's model, n is 20. As the data for Task2 is much smaller than that of Task1, we found that training Task2's model with all the samples and training Task1's model with only Task1's data gave the best score.

5.3.2 Ensemble. For more diversity, our models have different features and make predictions on candidate items from different recall strategies. Regarding the ensemble method, we first obtain the top 100 items of each user from the prediction of each model. Then, we apply a weighted blending on the scores, where the weight of Jin Zhan and Zhongshan Huang's model is 7 and that of Weijia Zhang's model is 3.

6 EXPERIMENTS

6.1 Overall Performance

The performance of our models is listed in Table 2, where model A is trained by Jin Zhan and Zhongshan Huang, and model B is trained by Weijia Zhang. Model A also includes some of Weijia Zhang's features. In task 1, model A scores 0.4030, model B scores 0.3968, and the merged model scores 0.4047. In task 2, model A scores 0.4580, model B scores 0.4468, and the merged model scores 0.4601.

Models	Task1	Task2
A	0.4030	0.4580
B	0.3968	0.4468
Ensemble	0.4047	0.4601

Table 2: Overall Performance

6.2 Feature Importance

Model A is trained with all of Jin Zhan, Zhongshan Huang, and Weijia Zhang's features. Therefore, we present the importance of the top 30 features of model A in Figure 2.

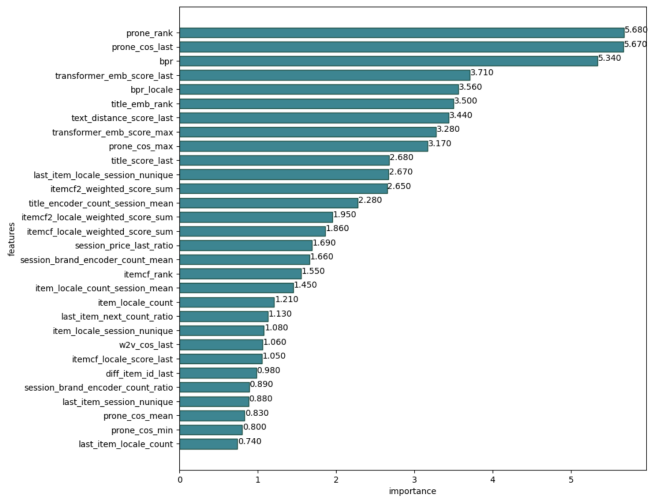

Figure 2: Top 30 Features' Importance.

REFERENCES

[1] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

[2] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[3] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. 2023. Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation. (2023).

[4] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. arXiv:2305.13731 [cs.IR]

[5] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[6] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[7] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*.

[8] 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*. 165–174.

[9] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. Prone: Fast and scalable network representation learning.. In *IJCAI*, Vol. 19. 4278–4284.