6th Place Solution to Amazon KDD Cup '23 Task2: Next Product Recommendation for Underrepresented Languages

Chiaki Ichimura
NS Solutions Corp.
ichimura.chiaki.3g6@jp.nssol.nipponsteel.com

Daiki Toda
NS Solutions Corp.
toda.daiki.3nh@jp.nssol.nipponsteel.com

ABSTRACT

In this paper, we present our solution, which achieved 6th place in the Amazon KDD Cup Task2 (Next Product Recommendation for Underrepresented Languages) in 2023. Our recommendation pipeline comprises two stages: candidate generation through several recommendation algorithms and ranking using machine learning models. We incorporate predictions from multiple individual recommender models, combining them based on prediction ranks, and achieve a Mean Reciprocal Rank (MRR) of 0.44798 on the competition leaderboard.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, Collaborative Filtering, Content-Based Filtering, Learning to Rank, KDD Cup

ACM Reference format:

Chiaki Ichimura, Daiki Toda. 2023. 6th Place Solution to Amazon KDD Cup '23 Task2: Next Product Recommendation for Underrepresented Languages. In *KDD Cup 2023 Workshop: Multilingual Session Recommendation Challenge, August 9, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 3 pages.

1 Introduction

Modeling customer shopping intentions and providing personalized recommendations are crucial for e-commerce stores to enhance customer experience and engagement. Task2 in the Amazon KDD Cup 2023 competition [1] aim to predict the next engaged product for sessions from French, Italian, and Spanish. This task requires selecting products for recommendation from a large inventory and a significant number of users. From a computational standpoint, it is not feasible to employ machine learning models to predict "Does the user prefer the product?" scores for all possible user-product combinations. Therefore, we adopted a two-stage recommendation approach [2], consisting of candidate generation and ranking stages. In the candidate generation stage, methods such as collaborative filtering and content-based filtering are used to generate several hundred potential candidates per session. In the ranking stage, machine learning models determine the best-recommended products for each user. This approach effectively addresses the challenge of computational complexity while achieving highly accurate recommendations. The remaining sections of this paper primarily focus on the methods used in candidate generation, ranking, and model blending.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
KDDCup '23, August 9, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).

2 Our Approach

2.1 Candidate Generation

In this task, we are provided with data on customer behavior history and product information [3]. Using this data, we experimented with several candidate generation methods, including collaborative filtering-based methods and content-based filtering-based methods. The following list presents the candidate generation methods employed in the best single model:

- **Association rule**. We calculate the co-occurrence between products and select top products with a high co-occurrence score with the products in sessions as candidates.
- **Similar products**. We compute TF-IDF vectors from the text data that combines product information such as title, brand, color, description, etc. Then, we select top products with high cosine similarity to the products in sessions as candidates.
- **Popular products**. We choose top products with the same brand as the products in sessions that have high popularity, indicating they have been frequently accessed in the overall session data.
- **Implicit Matrix Factorization** [4]. We employ Implicit Matrix Factorization (IMF), a type of matrix decomposition method, to assign scores to products in each session. Then, we select top products with high scores as candidates. For implementation, we use AlternatingLeastSquares by implicit [5].

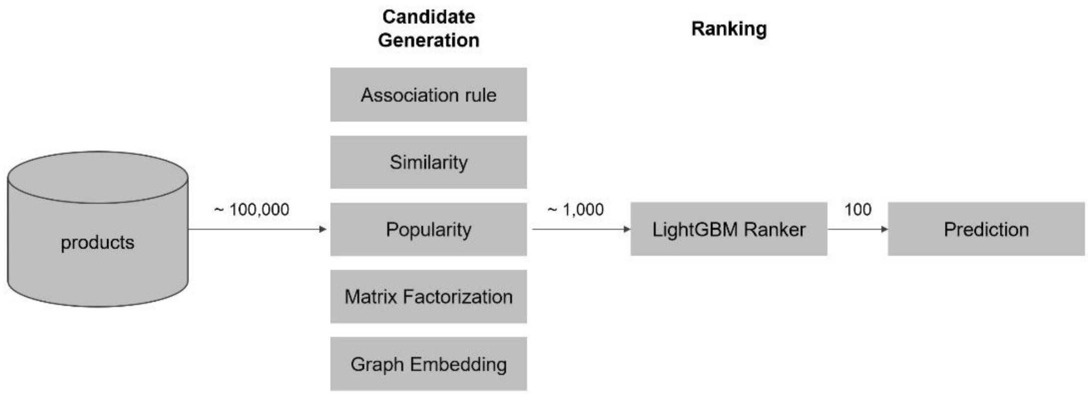

Figure1. Overview of our recommendation pipeline. The numbers above the arrows indicate the number of products at each stage.

- **Bayesian Personalized Ranking** [6]. We adopt Bayesian Personalized Ranking (BPR), another type of matrix decomposition method, to assign scores to products in each session. We select top products with high scores as candidates. For implementation, we use BayesianPersonalizedRanking by implicit [5].
- **Graph embedding**. We consider products in sessions as nodes, and the relationships between products that appear consecutively as edges. Using the session data, we construct graph data. From this graph data, we obtain embedding vectors for each node using proNE [7]. Based on these embedding vectors, we select top products with high cosine similarity to the products in the session as candidates.

During the analysis of the product information data, we observed instances where certain products share the same ID but differ in locales. Due to Task2 having a relatively smaller training dataset compared to Task1, we utilized association rules by incorporating both Task1 and Task2 data. This approach allowed us to leverage session data from other significant locales and enhance the overall performance of Task2.

2.2 Ranking

In the ranking stage, we employ ranker models that utilize the generated candidates and session data to predict the products with the highest probability of being engaged next.

2.2.1 Preprocessing. The training data exhibits an imbalance issue, with a significantly larger proportion of negative samples. This is due to the presence of only one positive label representing the next product, while several hundred candidates are generated per session. To address this, we performed downsampling by utilizing only 20% of the negative samples.

2.2.2 Model. For ranking the candidates, we used the LightGBM ranker. Table 1 presents the hyperparameter settings used for the LightGBM model in the best single model configuration.

Table1: Hyperparameters setting of LightGBM Ranker.

Hyperparameter	**Value**
objective	lambdarank
boosting	gdbt
num_leaves	50
max_depth	10
learning_rate	0.05
feature_fraction	0.9
bagging_fraction	0.9
bagging_freq	5
min_data_in_leaf	30

2.2.3 Features. We constructed approximately 100 features for the ranker model. These features are categorized as follows: features based on candidate strategy, session, product, and session-product. The list below outlines the features incorporated in the best single model:

- **Features based on candidate strategy**
  - Association rule scores and ranks of the candidate in relation to the last products of the session
  - TF-IDF vector similarity scores and ranks of the candidate in relation to the last products of the session
  - Graph embedding vector similarity ranks of the candidate in relation to the last products of the session
  - IMF ranks of the candidate in relation to the session
  - BPR ranks of the candidate in relation to the session
  - Popularity rank of the candidate in relation to the last products of the session
- **Features based on session**
  - Locale
  - Session length
  - Number of unique products in the session
  - Ratio of the number of unique products to the session length

- Number of unique brands in the session
- Ratio of the number of unique brands to the session length
- Mean price of products in the session
- Maximum price of products in the session
- Minimum price of products in the session
- Standard deviation of prices of products in the session
- Total price of products in the session

- **Features based on product**
  - Price
  - Total action count (global)
  - Total action count (locale)
  - Total action count multiplied by price (global)
  - Total action count multiplied by price (locale)
  - Total action count of the product's brand (global)
  - Total action count of the product's brand (locale)
  - Mean price of the product's brand
  - Maximum price of the product's brand
  - Minimum price of the product's brand
  - Standard deviation of prices of the product's brand
  - Price difference compared to the mean brand price
  - Number of locales that carry the product
  - Is the {column} null or not (column=[color, size, model, material, author])
  - Action count ratio to the total action count
  - Action count multiplied by price ratio to the total action count multiplied by price
  - Action count ratio to the total action count of the product's brand
  - Action count multiplied by price ratio to the total action count multiplied by price of the product's brand
- **Features based on session – product**
  - Price difference of the product compared to the mean price of products in the session
  - Price difference of the product compared to the minimum price of products in the session
  - Price difference of the product compared to the maximum price of products in the session
  - Price difference of the product compared to the last price of products in the session
  - Price difference of the mean brand price compared to the mean price of products in the session
  - Price difference of the mean brand price compared to the minimum price of products in the session
  - Price difference of the mean brand price compared to the maximum price of products in the session
  - Price difference of the mean brand price compared to the last price of products in the session
  - Is the {column} of the product the same as the last products of the session (column=[brand, color, size, model, material, author])
  - Word2Vec vector similarity of the products to the last products of the session
  - Graph embedding vector similarity of the products to the last products of the session
  - IMF score from the session to the product
  - BPR score from the session to the product

2.3 Model Blending

We used 19 LightGBM models, each trained with different amounts of training data, varying numbers of candidates, candidate methods, and LightGBM hyperparameters. The predictions from these models were blended based on ranks. The blending process improved the MRR score of the single model from 0.44714 to 0.44798 on the competition leaderboard.

3 Conclusion

In this paper, we presented our solution to Task2 of the Amazon KDD Cup 2023 competition. Our approach utilized a two-stage recommendation process, aiming to provide highly accurate recommendations while addressing the challenge of computational complexity. By generating candidates through various recommendation algorithms and ranking them using machine learning models with carefully crafted features, we achieved an MRR score of 0.44798, which placed us in 6th position on the competition leaderboard

REFERENCES

[1] Amazon KDD Cup '23 Task 2: Next Product Recommendation for Underrepresented Languages. https://www.aicrowd.com/challenges/amazon-kdd-cup-23-multilingual-recommendation-challenge/problems/task-2-next-product-recommendation-for-underrepresented-languages
[2] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H. Chi. 2020. Off-policy Learning in Two-stage Recommender Systems. In Proceedings of The Web Conference 2020 (WWW '20). Association for Computing Machinery, New York, NY, USA, 463–473.
[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, & Xianfeng Tang (2023). Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation
[4] Y. Hu, Y. Koren and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 263-272, doi: 10.1109/ICDM.2008.22.
[5] Implicit. https://github.com/benfred/implicit
[6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09). AUAI Press, Arlington, Virginia, USA, 452–461.
[7] Zhang, J., Dong, Y., Wang, Y., Tang, J., & Ding, M. (2019). ProNE: Fast and Scalable Network Representation Learning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19 (pp. 4278–4284). International Joint Conferences on Artificial Intelligence Organization