Session-based Recommendation Using an Ensemble of LSTM-and Matrix Factorization-based Models

Yoshihiro Sakatani
sakatani.yoshihiro@sdtech.co.jp
sdtech Inc.
Minato, Tokyo, Japan

ABSTRACT

In situations where most users of an e-commerce platform are non-repetitive and have a high bounce rate, product recommendation systems cannot utilize the users' metadata or behavioral data of previous sessions. The 2021 SIGIR Coveo Data Challenge is aimed at benchmarking product recommendation models for this session-based problem; the goal is to predict the next products of e-commerce sessions without any user information. In this paper, I describe my approach where I leverage matrix factorization and LSTM to model user-product interactions for session-based recommendation. This approach achieved highly competitive performance placing 2^{nd} on the final private leaderboard of the subsequent product prediction.

CCS CONCEPTS

• **Information systems** → *Recommender systems*; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Recommender Systems, Session-Based Recommendation, Recurrent Neural Network, Matrix Factorization

ACM Reference Format:
Yoshihiro Sakatani. 2021. Session-based Recommendation Using an Ensemble of LSTM-and Matrix Factorization-based Models. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Product recommendation systems have contributed greatly to the exponential growth of e-commerce businesses by improving user experience and increasing revenue. However, in situations where most users are non-repetitive and have a high bounce rate [9], product recommendation systems cannot utilize the users' metadata or behavioral data in previous sessions. In such situations, prediction in a short time using only in-session information has become an important topic for e-commerce merchants[1, 2, 11].

SIGIR 2021 Coveo Data Challenge [10] is aimed at benchmarking recommender systems that perform the task of estimating the user-product interaction based on in-session information, such as previously interacted products, timestamp, and user action. This challenge is based on a real-world dataset of five million e-commerce sessions newly released by Coveo.The data contains detailed user browsing behaviors in each session (user action, interacted products, queries) and product catalog metadata (image, description, and price), but no information on users, such as user attributes or user behaviors in previous sessions

This challenge provides two tasks to the participants: (1) a recommendation task and (2) an intention task. This paper focuses on the former: product recommendation. The recommendation task has two evaluation criteria: predicting the product immediately following a browsing event, and predicting the subsequent 20 products following a browsing event.

The sequential pattern of browsing events becomes especially important when there is no information about the user. My approach focuses on this point and generates sequential features to input them to neural network models with an input layer suitable for them, such as an LSTM layer. In addition, in order to treat tens of thousands of targets products, I also prepared a model that incorporates matrix factorization, which has been shown to be effective for session-based recommendation tasks in recent years.

Finally, my approach achieved a very competitive performance placing 2^{nd} on the final leaderboard result of the subsequent items prediction with an ensemble of predictions of these models. The code is publicly available at https://github.com/sakatani/sigir-2021/.

2 DATA DESCRIPTION

The training dataset contains over five million sessions, consisting of 36 million browsing events and 820,000 search events. The test datasets of Stage 1 and 2 (the challenge has the two stages) contain 140,000 sessions (570,000 events) and 330,000 sessions (1.3 million events), respectively. Out of the 36 million browsing events included in the training data, 26 million are pageview events, and 10 million are product events. The browsing event data includes session ID, event type, hashed product ID, timestamp, and hashed URL. For product events, the product ID and the user action (detail, add, remove, or purchase) are also included in the data.

Every detail event is accompanied by a pageview event with the same URL and timestamp. There are 57,000 products, approximately 34,000 of which are associated with meta-information such as category, price, description vector, and image vector. Some products are linked to only some of these pieces of information. The timestamp is anonymized and only the weekly structure is stored; the month, date, and year are unknown. Moreover, the datasets

SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada

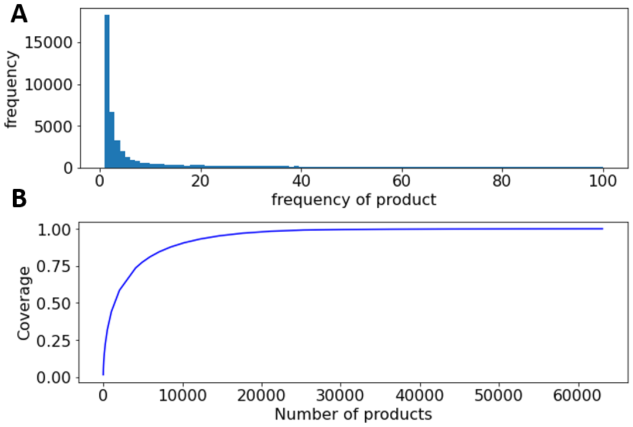

Figure 1: A) Distribution of the product frequencies. There are 13,000 products with frequencies greater than 100, which are not shown in the figure. Over 180,000 products appeared only once in the dataset. B) Coverage rate of the events where the products are sorted by frequency. The most frequent 1500 products account for more than half of all events.

contain no information about the user, such as their attributes or their behavior in previous sessions.

The same URL is often assigned to the same product. There are 57,000 unique product IDs and 93,000 unique URLs in the product events but only 150,000 product-URL combinations.

3 APPROACH

3.1 Data Preparation

In addition to the provided training dataset, the test datasets for the recommendation task and intent task of Stage 1 and 2 were also used to train the models. The enlarged training dataset was divided into five folds grouped by session ID so that each fold contained all events from the same session. In each fold, the data inside the fold was used for validation and the out-of-fold data was used for training.

The browsing events of the datasets were first sorted by session ID and timestamp. The browsing events in the same session with the same timestamp were further sorted by event type and user action in order to make the order of the event types and user actions consistent across all data. Next, the pageview events that occurred concomitantly with a detail event in the same session with the same URL were removed from the datasets. For the events other than the detail and pageview, several event pairs that occurred simultaneously in the same session were found, but these were not removed.

3.2 Feature Engineering

In addition to the 13 raw features in the original training data, additional 20 features were designed and divided into five main groups. I used label encoding for all the categorical features and applied the Yeo-Johnson power transformation [12] to all numeric features to fit these features to neural network training.

Product Features In addition to the raw product data (hashed product ID, hashed category, price, description vector, and image vector), the count of hashed product IDs and hashed categories, the first product of each session, the category of the first product, and whether the metadata (price, image vector, description vector) were null or not were also used as product features.

The frequency of products showed a long tail distribution where some high-frequency products accounted for a large portion of the events (Fig 1). To prevent the model from concentrating too much on low-frequency products, products with a frequency of eight or less were aggregated into a single value. This process significantly reduced the cardinality of the products from over 61,000 to 27,000, but the remaining products still covered 99.4% of the events in the training data. The image and description vectors were subjected to principal component analysis, and the first through fourth principal components were used as input to the models.

Count Features The cumulative summation of each type of event (product event, search event, and pageview event) in each session was counted and used as a count feature. Since the validation dataset was created using k-fold cross validation grouped by session ID, the value of validation data was not reflected in the count features. Counting of pageview and search events between product events and after the last product event of each session were also used as the features.

Length Features The temporal length of the events and the elapsed time from the first event of each session are computed from the timestamp. They were used as length features after converting the units to seconds.

Time Features The hour and day of the week of the events, and whether the events were on the weekend or not were extracted from the timestamp. These three features were label-encoded as categorical features.

URL Features As with the hashed product IDs, the hashed URLs with a frequency lower than eight were replaced with a dummy singular value, resulting in a cardinality reduction from 90,000 to 30,000. I also added the first hashed URL of each session to the features.

3.3 Target

For each event, the product of the next product event was set as the target. When the product of the next product event was a low-frequency product, the product of the product event after the next product event was set as the target.

3.4 Models

The model architecture diagram of my approach is illustrated in Figure 2. Since the available data was very different between sessions with and without product events, different models were applied to each session type: Product event-based model and Pageview event-based model.

3.4.1 Product event-based model. For the sessions with product events, the final predictions were generated by an ensemble of

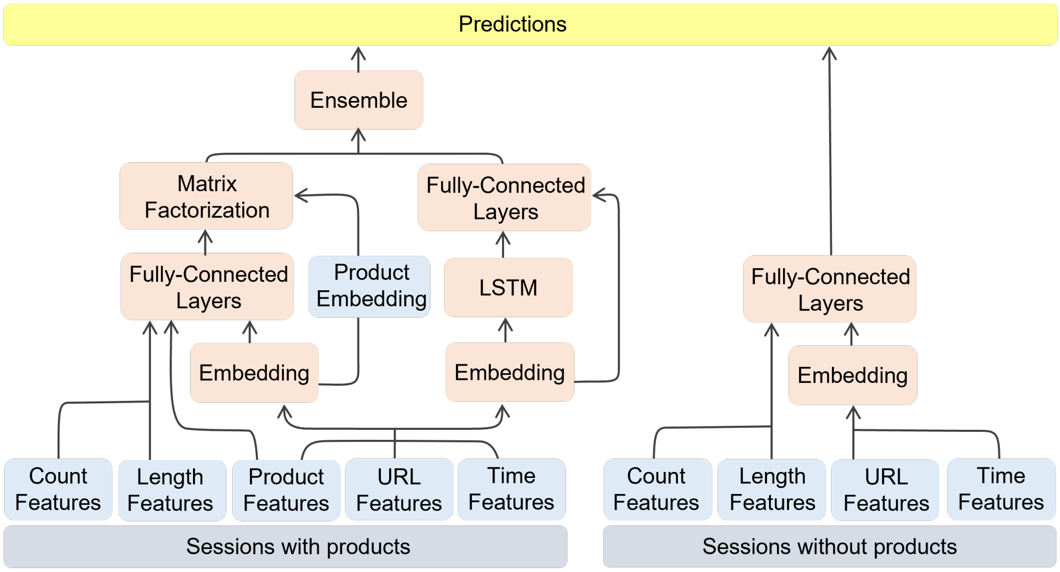

Figure 2: Model Architechture. An ensemble of the predictions of two neural network models was used for the sessions with product events. Predictions for the sessions without product events were generated by a single neural network model.

predictions from two neural networks: the matrix factorization-based model (MF-based model) and the LSTM-based model.

The use of matrix factorization in session-based recommendation is similar in concept to the session-based matrix factorization (SMF) proposed in a previous study [6]. The SMF is inspired by the traditional matrix factorization-based methods[5, 7], which model user-item interaction, and is designed to make the matrix factorization applicable to session-based recommendation. In the SMF, latent user vector of the user-item matrix factorization is replaced with a session preference vector. Thus, the SMF learns latent factors for sessions and products by performing a dot product operation of their embeddings and generates the logits for products. I adopted the SMF because the previous study has shown that the SMF marked competitive scores in MRR@20 for the next item prediction [6], which is the evaluation metric for this challenge, with RSC15, which is similar to the datasets for this challenge. The winning team at the WSDM2021 Challenge [3], where the task was a session-based recommendation just like this challenge, also adopted session-based matrix factorization and incorporated it with a multi-layered perceptron model [8]. Given the above, incorporating matrix factorization into neural networks was expected to be promising for this challenge.

The LSTM-based model is similar to GRU4Rec [4], which also has shown high MRR@20 scores on several datasets including RSC15 [6]. It was pointed out that GRU4Rec focused too much on the next item and I changed GRU to LSTM in order to account for sequences of longer length.

In addition to all features of each event, product IDs, hashed URLs, and categories of the last 20 events, and image and description vectors of the last two events were input to these models in order to account for sequential structures. For the MF-based model, the embeddings of categorical and numeric variables were combined by summation and concatenated with the featue embeddings of the past events. The concatenated vector was input to the fully-connected layers. For the model with LSTM, only the embeddings of the category variables were input as in GRU4Rec. All the embeddings were concatenated and input to the fully-connected layers.

Not all of the training data was used for these models; the sessions with less than two product events and the events where the target was a low-frequency product were removed before training.

3.4.2 Pageview event-based model (Product event-free model). In the sessions without product events, only data of pageview events and search events were contained and few categorical features were available. Hence, both the MF-based model, which utilized a product embedding, and the LSTM-based model, which used only categorical variables, were not used for this type of session; a simple multi-layered perceptron model without MF or LSTM was used.

I did not use all of the training data; the sessions with only one event and the events after the first product event of each session were removed before training. The search event data was also not used because less than 9% of the sessions in the test data contained search events. The model was completely dependent on pageview events.

In addition to all features of the last event, hashed URLs of the last nine events were input to this model.

4 EXPERIMENTS

All experiments were performed on Google Colaboratory, and it was made sure that Intel(R) Xeon(R) CPU @ 2.20 GHz, 26 GB RAM, and NVIDIA Tesla V100 PCIe GPU were allocated.

4.1 MF-based model

The matrix factorization-based model consisted of one embedding layer, three fully-connected layers, and one matrix factorization

Table 1: Model performances on the validation and test dataset. The MRR@20 scores are for the next product prediction and the F1@20 scores are for the subsequent products prediction.

	Cross validation		Leaderborad			
Model	MRR@20	F1@20	MRR@20	F1@20	Coverage@20	Popularity bias@20
Most frequent 20	0.0018	0.00268	0.0015	0.00260	0.0003	1.198×10^{-3}
Pageview event-based only	0.0809	0.02257	0.0445	0.02148	0.2958	6.876×10^{-4}
Product event-based only						
MF	0.1897	0.06185	0.1853	0.05980	0.4081	4.444×10^{-4}
LSTM	0.1869	0.06086	0.1810	0.05989	0.4087	4.514×10^{-4}
Ensemble (MF + LSTM)	0.1900	0.06210	0.1850	0.06020	0.4095	4.480×10^{-4}
Pageview event-based + Product event-based						
MF	0.2209	0.07298	0.2152	0.07090	0.4000	1.414×10^{-4}
LSTM	0.2159	0.07319	0.2109	0.07100	0.3991	1.484×10^{-4}
LSTM with MF	0.2164	0.07290	0.2119	0.07088	0.4012	1.475×10^{-4}
Ensemble (MF + LSTM)	0.2206	0.07351	0.2149	0.07132	0.4006	1.449×10^{-4}

head. The embedding dimension of 64 was used for all the categorical features. The output sizes of the fully-connected layers were 1024, 1024, and 64. The model was trained for ten epochs with a categorical cross entropy loss, the Adam optimizer, a one-cycle scheduler, and a batch size of 1024. The models with the best MRR@20 score for the next product prediction on the validation data were used for prediction. The low-frequency products were ignored when calculating categorical cross-entropy loss and did not contribute to the input gradient.

4.2 LSTM-based model

The LSTM-based model consisted of one embedding layer, one LSTM layer, and three fully-connected layers. The output sizes of the fully-connected layers were 512, 512, and the number of unique products. The model was trained for four epochs with a categorical cross entropy loss, the Adam optimizer, and a batch size of 512. As in the MF-based model, the models with the best MRR@20 score for the next product prediction on the validation data were used for prediction. The learning rate was reduced every epoch by a factor of ten from the initial value of 10^{-3}.

4.3 Pageview event-based model

The Pageview event-based model was identical to the MF-based model except for the missing matrix factorization head. The model was trained under the same condition of the MF-based model (ten epochs with a categorical cross entropy loss, the Adam optimizer, a one-cycle scheduler, and a batch size of 1024.) The model selection for prediction and the treatment of low-frequency products were the same as that of the MF-based model.

4.4 Ablation Studies and Model Ensemble

The performance of my best performing model and ablations on the validation dataset are shown in Table 1. The Pageview event-based model alone showed a rather low score, but when combined with the Product-event based model, it improved MRR@20 by around 0.3 and F1@20 by around 0.01 from the score of the Product-event based model alone.

For all cases, the MF-based model was better for the next product prediction and popularity bias@20, and the LSTM model was better for the subsequent items prediction and coverage@20. The ensemble improved the subsequent product prediction and coverage scores, but not the next product prediction and popularity bias scores. These result indicate that the LSTM-based model focused on sequential structures of products more than the next product. They also indicate that matrix factorization may reduce the popularity bias. To check this hypothesis, I connected a matrix factorization head to the LSTM-based model. The results showed lower popularity bias (LSTM with MF in Table 1) than the LSTM-based model.

Sixty-seven percent of the sessions in the test data contained product events and could be inferred with the Product event-based model. The most of remaining sessions were predicted by the Pageview event-based model. Because 0.172% of the sessions in the test data consisted of search events only, neither the Product event-based nor Pageview event-based models could be applied to them. For these data, the top 20 search query information were used as prediction results, and those with less than 20 search queries were filled in with the most frequent products whereas the most frequent products did not lead to good prediction results (Table 1). The method that uses the top 20 search query information as predictions showed 0.1562 of the MRR@20 score for the next product prediction on the validation data that included search events.

The predictions submitted to the final result leaderboard showed 0.2149 of the subsequent item F1@20 score (2^{nd}) and 0.07132 of the next item MRR@20 score (6^{th}).

5 CONCLUSION

In this paper, I describe my approach to the SIGIR Challenge 2021 workshop. The approach leveraged matrix factorization and LSTM to model user-product interactions for the session-based recommendation and achieved highly competitive performance. Future work involves an investigation on how product embeddings can improve the Pageview-based model by combining them.

REFERENCES

[1] Federico Bianchi, Jacopo Tagliabue, Bingqing Yu, Luca Bigon, and Ciro Greco. 2020. Fantastic Embeddings and How to Align Them: Zero-Shot Inference in a Multi-Shop Scenario. arXiv:2007.14906 [cs.IR]

[2] Federico Bianchi, Bingqing Yu, and Jacopo Tagliabue. 2020. Bert goes shopping: Comparing distributional models for product representations. *arXiv preprint arXiv:2012.09807* (2020).

[3] Dmitri Goldenberg, Kostia Kofman, Pavel Levin, Sarai Mizrachi, Maayan Kafry, and Guy Nadav. 2021. Booking.com WSDM WebTour 2021 Challenge. In *ACM WSDM Workshop on Web Tourism (WSDM WebTour'21)*. https://www.bookingchallenge.com/

[4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[5] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[6] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 28, 4-5 (2018), 331–390.

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

[8] Benedikt Schifferer, Chris Deotte, Jean-François Puget, Gabriel de Souza Pereira Moreira, Gilberto Titericz, Jiwei Liu, and Ronay Ak. 2021. Using Deep Learning to Win the Booking. com WSDM WebTour21 Challenge on Sequential Recommendations.. In *WebTour@ WSDM*. 22–28.

[9] SimilarWeb. 2019. Top sites ranking for E-commerce And Shopping in the world. https://www.similarweb.com/top-websites/category/e-commerce-and-shopping.

[10] Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Bingqing Yu, Patrick John Chia, Federico Bianchi, and Giovanni Cassani. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge. *arXiv preprint arXiv:2104.09423* (2021).

[11] Jacopo Tagliabue, Bingqing Yu, and Marie Beaulieu. 2020. How to Grow a (Product) Tree: Personalized Category Suggestions for eCommerce Type-Ahead. arXiv:2005.12781 [cs.LG]

[12] In-Kwon Yeo and Richard A Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika* 87, 4 (2000), 954–959.