Improving Session-Based Recommendation Adopting Linear Regression-Based Re-ranking

Ramazan Esmeli
School of Computing
University of Portsmouth
Portsmouth, UK
ramazan.esmeli@port.ac.uk

Mohamed Bader-El-Den
School of Computing
University of Portsmouth
Portsmouth, UK
mohamed.bader@port.ac.uk

Hassana Abdullahi
School of Mathematics and Physics
University of Portsmouth
Portsmouth, UK
hassana.abdullahi@port.ac.uk

David Henderson
Fresh Relevance
5 Benham Rd, Chilworth
Southampton, UK
david.henderson@freshrelevance.com

Abstract—Due to the increase in the importance of giving real-time recommendation to e-commerce users, session-based recommender systems become more popular. Session-based recommendation systems have the ability to adapt quickly to respond to changes in user interests and newly added items. The ranking is the core part of recommender systems regardless of recommender system type. Re-ranking is applied to recommender systems to have more personalised recommendations by considering context-awareness. In this paper, we proposed an approach to re-rank recommended items by using a linear regression model. In our approach, we use users' current session features and temporal features of recommended items to measure a user's interest level on a recommended item. We focus on having better recall and precision scores with fewer recommendations to able to prove the success of our re-ranking strategy. We conduct computational experiments on six real-world datasets and show that after applying re-ranking, we can get higher recall and precision scores. These results confirm that taking user interest level on an item in a session into account can improve the chance of getting correct items in top 5 recommendations.

Index Terms—session-based recommendation, re-ranking, linear regression

I. INTRODUCTION

Recommender Systems (RS) is an important component of e-commerce platforms due to the relevance for recommending relevant products based on users' behaviours. It is also an important tool for increasing users' satisfaction by providing personalised recommendations. This has been applied to many domains including movie [1], music [2] and e-commerce [3], [4], and many e-commerce companies such as eBay and Amazon [5] also take advantage of RS.

Recently, Session-Based Recommender Systems (SBRS) [2], [3] gain popularity due to their success in providing real-time recommendation even for unregistered shoppers. In order to improve the accuracy of SBRS, various approaches have been developed, including neural approaches. However, advantage and applicability of neural approaches are still debated [6], [7]. On the other hand, session-based nearest neighbour approaches are less complicated, and in some experiments, have shown better performance than neural-based approaches [6]. Ranking products is the primary purpose of RS to filter most relevant ones within other product options. This process is done in RS by calculating the similarities between the interacted item and other items in a session or calculating latent factors of the sessions and items. However, considering other factors on recommended items can be beneficial to find out the most relevant items for the user, such as re-ranking recommended items by considering context factors [8]. Several works proposed to rank recommended items using context factors [9]–[11]. In these works, experiment results show that context-aware SBRS has better accuracy since they provide increased personalised recommendations [10] as context factors are used to filter and re-rank most relevant items. Context filtering proved that user's context could influence user preferences [9], [10]. Each user session on an e-commerce platform may indicate information about the context, such as time of the day and day of the week that a session started. Also, session activities and item temporal features such as the popularity of an item can be considered as context [10], [11].

Session and item features are used in many works to identify purchase intention prediction and context filtering purposes in a session [10], [12], [13]. The experiment results in these works showed that session features are an excellent indicator to identify users' intention. Also, users leave valuable information while they browse the items. In many domains, these behaviours are mapped to explicit numerical rating by interpreting differently for each action to use in RS algorithms [14], [15]. In this work, we use temporal item features and users' session activities to build a regression model to find out users' intention level on an interacted item. We use users' interacted item and previous session features as implicit feedback and convert them to estimate explicit feedback. The converted explicit feedback is used for the user's interest level

prediction on an item in a session by using the regression model that will help to re-rank the recommended items.

Regression model aids in re-ranking recommended items while recall and precision are mainly used metrics for performance measurement of the RS [16]. Our main purpose is to see improvement on recall and precision metrics $n \in \{1, 3, 5\}$ recommendation since our goal is to predict the right product at the top of the recommendation list. We validate the proposed method using six real-world e-commerce datasets gathered from a UK based personalisation company. The experiment results show that our proposed model improves recall and precision in comparison to baseline SBRS model on recall@$n \in \{1, 3, 5\}$. The proposed method is a general framework that can be applied to other types of SBRS, including deep learning-based SBRS.

The main contribution of this study lies in the following two aspects:

1) We utilise session and product temporary features to build an efficient regression model to identify users' interest level on an interacted item in a session. The regression model takes the current user's previous session details and interacted item features into consideration to identify the users' interest level.

2) We introduce a novel re-ranking method for previously recommended items retrieved from SBRS model. The proposed re-ranking method updates previously produced recommendation score of the items by considering the interest level calculated from the regression model, which helps to improve the accuracy of next item prediction in the top 5 recommended items.

The paper is organised as follows: In Section 2, we discuss related work on ranking and context awareness in SBRS. Section 3 describes the details of our proposed method. In Section 4, we give experimental results and discuss the performance of the proposed methods. Finally, in Section 5, we conclude the paper and give a new research direction.

II. RELATED WORK

A. Session-Based Recommender Systems (SBRS)

Recently, SBRS has become popular due to the ease of providing real-time recommendations, even for unregistered shoppers. Generally, deep-learning and session-based item-similarity RS types are utilised as SBRS. On the one hand, deep-learning-based SBRS shows good performance in several research [3], [17], while recent works [6] show that session-based item-similarity RS performed better compared to complex deep learning-based SBRS models. [6], [18] conducted experiments on different datasets in order to compare Item-Item similarity-based SBRS and deep learning approaches. They found that most of the results produced in deep-learning approaches could not be reproduced. Also, except for one dataset, simple Item-Item similarity-based SBRS outperformed deep learning-based SBRS. On the other hand, modified session-based item-similarity RS [19] is the most used RS type in different e-commerce domains to recommend items for unregistered users [3], [7] since session-based item-similarity RS can be applied straightforward and training the model needs less resource, data and time. Also, session-based item-similarity RS is a powerful baseline on the comparison of deep learning approaches for SBRS performance [3], [17]. [20] proposed a session-based item-similarity RS model (S-KNN). In their proposed model, the most identical session to the current session is found, and the items are ranked based on their weights which are calculated from the most identical sessions. In addition, a Temporal KNN (T-KNN) model was proposed in [7], and the authors introduced a temporal awareness to the session-based item-similarity RS by changing the last item with previously seen items in the session for the next recommendations.

B. Re-ranking In Recommender Systems

Re-ranking methods are commonly used in RS to improve prediction accuracy by improving personalisation. One of the ways of re-ranking is taking into account the diversity awareness of the recommended items [21]–[23]. [21] designed a diversity aware re-ranking framework for the session-based recommendation, in which the ranking score of recommended items is updated as based on the results of the diversity level between last interacted item of the session and recommended items. They found that when re-ranking is applied, the RS model produced better performance in terms of recall and precision. In another work [22], re-ranking is applied on paper recommendation domain. In their work, they re-ranked recommendations based on the similarity scores of the paper content that the user previously browsed and recommended papers for the user. Their experiment result showed that there is 28 % of improvement in recall@1. Similarly, in [24], image features are used for re-ranking recommendation list. The image features of recommended items are compared to previously seen images and based on the similarity value as a result of the comparison, and the recommendation list is re-ranked. Experiment results demonstrated that re-ranking helped in considerably improving the recommendation performance in terms of recall and precision. In [23] aggregated recommendation diversity was investigated by applying various ranking methods. They found that diversity level increased by a certain level in recommendation list with small accuracy loss after applying re-ranking. Also in work [25], a post-processing method is designed for re-ranking recommendations based on the popularity of the items, the main aim of the designed method was to escalate less common items in the long tail recommendation list. Experiment results indicate that personalised recommendations can be obtained even common items are kept to minimise. Another approach for re-ranking is to consider the users' context [26]–[29]. For instance, the price distribution of previously interacted products, the season of the year and the time of the day could have a substantial effect on the accurate level of recommendations.

Furthermore, in [30], prediction and matrix factorisation models are combined for playlist continuation problem for Spotify music listening platform. They re-ranked the candidate

tracks based on the results of the prediction and recommendation models by calculating latent factors for playlist and tracks using a prediction model after training the model using matrix factorisation.

Our work has a similar aim to the above-reported works in terms of re-ranking the candidate items retrieved from base ranker; however, we use regression model trained using the features of the user's previous sessions and temporary features of the items. Previous works use diversity awareness, and context filtering such as colour, category of the product, time of the day and day of the week. While, we combine item temporal, and users' previously browsed sessions' features in order to build a regression model to find users' interest level on an item in a session.

C. Implicit Feedback to Explicit Rating Conversion

While users are browsing a product, to request a rating for the product is not a reasonable expectation. Firstly, they need to use the product and evaluate its quality, and then they may give their rating on the product. However, while they are browsing the product, they can leave valuable indications about their intentions such as how much time they spend on a product, how many times they have browsed the same product and details of the previous session behaviours. When a user interacts with an item, this can be considered as one class problem (OCCF) [31], [32] since each viewed item shows a positive intention. On the other hand, the intention level could be different for each interaction, depending on users' implicit signals. In order to interpret users' intention level on an item in a session, mapping these signals to explicit numerical values are broadly used in different domains [14], [15], [33].

In this work, we take the user's session features and item temporal features into consideration to map numerical ratings, which indicate the level of users' interest in the item in a session. However, in our work instead of utilising the calculated numerical values in the RS model directly, we use them in the linear regression model to predict users' interest level on a given item for a session.

III. PROPOSED INTEREST LEVEL BASED RE-RANKING METHOD

The proposed method consists of four steps which are session and feature extraction, interest level mapping, recommendation and re-ranking process, and performance evaluation (Fig 1). In order to implement and evaluate the performance of the proposed approach, we split the dataset as train and test datasets.

A. Session and Feature Extraction

Session specifications can give precise information about users' intention. For example, how many times a user browsed or purchased products in the past sessions. Also, item specifications such as the popularity of an item on different user actions in each session can indicate user interest level on an item. We extract the following features from a session:

1) **Number of previous sessions:** indicates the frequency of visits to a website by a user.
2) **Number of previously viewed product:** this feature shows how many items were browsed in previous sessions.
3) **Number of previous purchases:** this indicates how many times a user has purchased in previous sessions.

For the items, we extract the popularity of the item for each session outcome:

1) **Popularity of an item for view:** indicates the number of views of an item.
2) **Popularity of an item for add to basket:** indicates the number of times an item is added to the basket.
3) **Popularity of the item for purchase:** indicates the number of purchases of an item.

The above-extracted features are used to train the regression model. We divide popularity related features to the total number of the unique items in the dataset to have normalised feature.

B. Interest Level Mapping

Mapping of interest level helps to calculate user interest level for each interacted item in the session. In order to have interest level mapping, firstly, we create a linear model which gives different weights to each factor. The weights are identified based on the indication of users' interest level for each feature. For example, the basket popularity of an item can show more interest level when compared to the viewing popularity of the item. Therefore, we level the weights based on the popularity of users' actions on items. Similarly, each previous session features are weighted differently. For instance, a higher number of page viewed in the previous sessions can show the user is solely exploring products rather than focusing on a particular product, so its weight may be less compared to weights of the number of previous sessions and number of previous purchases.

$$
\begin{aligned}
i_{interest\ rate} = {} & 0.6 * (i_{basket\ popularity})/totalitem + \\
& 0.5 * (i_{browse\ popularity})/totalitem + \\
& 0.9 * (i_{purchase\ popularity})/totalitem + \\
& 0.8 * (u_{number\ of\ previous\ sessions}) + \\
& 0.5 * (u_{number\ of\ pages\ viewed}) + \\
& 0.9 * (u_{number\ of\ previous\ purchases})
\end{aligned}
\tag{1}
$$

The weights for the factors seen in Equation1 are selected using the result of experiments based on regression model performance. It should be noted that we restrict the weights between 0 and 1 and in each search, weights are incremented by 0.1. To determine the interest level, interest rates are calculated (Equation 1) which are converted to interest levels (Equation 4). Firstly, we calculate the average interest rate of all items(Equation 2) to find out the threshold that will help to group the users' interests for each item into three levels. In Equation 2, $interest\_rate_n$ is the calculated interest rate for the nth item, $\#interactions$ is the number of item
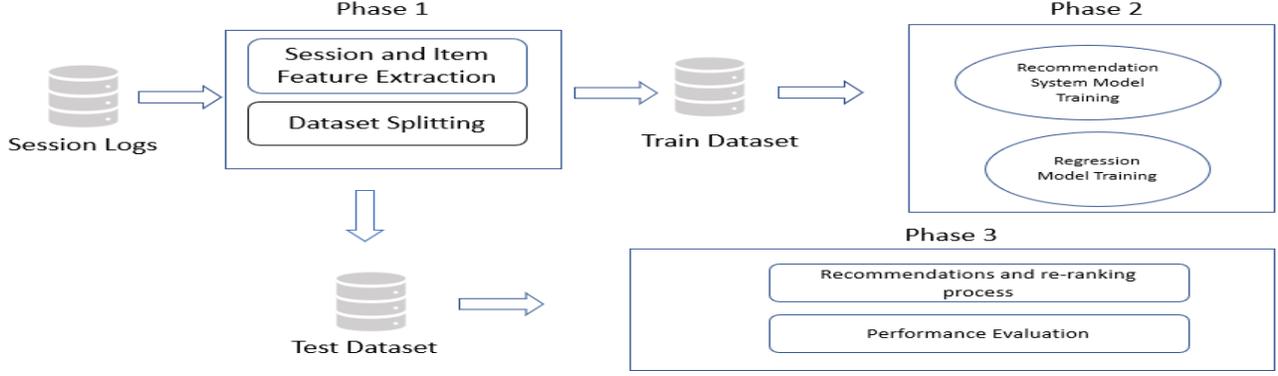
Fig. 1. Proposed re-ranking model for SBRS

interactions in the train dataset and $\overline{x}$ is the calculated average interest rate of whole items in the train dataset.

$$\overline{x} = \frac{\sum_{n=1}^{\#interactions} interest\_rate_n}{\#interactions} \quad (2)$$

Secondly, to identify the gap between each interest level, we calculate standard deviation of interest rate s using all items' interest rate in train dataset (Equation 3). In the Equation 3, #interactions is total interactions in the dataset, $interest\_rate_n$ is calculated interest rate using Equation 1 for n th item in a session, s is standard deviation of users' interest rate on items.

$$s = \sqrt{\frac{1}{\#interaction - 1} \sum_{n=1}^{\#interaction} (interest\_rate_n - \overline{x})^2} \quad (3)$$

Finally, we create a normalisation function (Equation 4) that will return three values based on the $interest\_rate$, $\overline{x}$ and s. We name this value as $interest\_level$. $interest\_level$ is not known for the interacted items in test dataset. We utilise regression model to predict $interest\_level$ of each interacted item for each session in test dataset.

$$interest\ level = \begin{cases} 1, & \text{if } interest\ rate < \overline{x} \\ 2, & \text{if } \overline{x} < interest\ rate < \overline{x} + s \quad (4) \\ 3, & \text{if } interest\ rate > \overline{x} \end{cases}$$

C. Recommendation and Re-ranking Process

The re-ranking process is a key contribution of this paper. The aim of re-ranking is to predict the next product that user may interact in the top 5 recommendation list. The re-ranking process involves recommender system model training and linear regression model training, which are described below.

1) Recommender System Model Training: We use session-based item-item similarity RS in this work. The model details are explained in [3]. We retrieve 100 recommendations from the trained model for each session in the test dataset.

2) Linear Regression Model Training: A linear regression model is used for the re-ranking purpose. The model uses the features from the session and item. We use default parameters for the linear regression model as shown in the sklearn website[1] [34]. The retrieved 100 recommended products are used as candidate products to be re-ranked. Each product is represented by the item and session features. Session features can be interpreted as session context [10], [35] since each session can have different features specific to the user.

3) Re-ranking: We use the 100 candidate items which are retrieved from the recommendation model to re-rank and update recommendation list by considering users' interest levels which are predicted from the regression model, on candidate items. Predicted value from the regression model is used as an adjustment score of the recommendation score of recommended products (Equation 5). In this Equation, $Recommendation_s$ is the recommendation score of a recommended item and $Regression_s$ is interest level prediction score for a recommended item in the session.

$$new\ score = Recommendation_s * Regression_s \quad (5)$$

Finally, new recommendation list is created by ranking recommended items using $new\ score$(Equation 6).

$$final\ list = Rank(new\ score) \quad (6)$$

An example of the re-ranking process is shown in Figure 2. The recommendation model initially produces ranked recommendation list, later, for the session, and each recommended item, a user's interest score is calculated. Finally, the recommended items' scores are updated, and the recommendation list is re-ranked.

D. Performance Evaluation

To evaluate the performance of the proposed method, we use test dataset consisting of the last three days of a month's interactions. First, we hide the last two interacted items

---

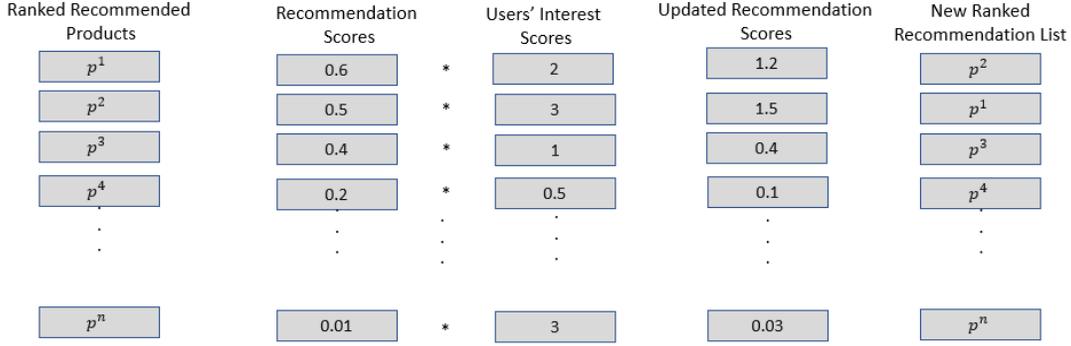[1]https://scikit-learn.org/stable/modules/linear_model.html

Fig. 2. An illustration of re-ranking process

from each test session, then we use the last item of known interactions in the session to predict the hidden items.

From the recommendation model, we retrieve 100 product recommendations using last interacted item. We use these items as candidate items in order to predict users' interest level on these items in the session. After having interest level prediction using the linear regression model, the 100 product recommendation list is updated based on users' interest level prediction on the items. From updated and re-ranked list, we select top-n ($n \in \{1, 3, 5\}$) items since users are mostly interested in recommendations which are in the head of the recommendation list. Selected items are compared with the hidden items in which we refer to as ground truths. We use the $Recall@n$ (Equation 7) and $Precision@n$ (Equation 8) metrics to measure how the models perform on predicting hidden items in the session [36], [37]. $groundtruth$ refers to the hidden items in the session, $recommendations@n$ shows the top n ranked items after re-ranking applied.

$$Recall@n = \frac{|groundtruth \cap recommendations@n|}{|groundtruth|} \quad (7)$$

$$Precision@n = \frac{|groundtruth \cap recommendations@n|}{|recommendations@n|} \quad (8)$$

In order to evaluate how trained regression model performs, we use the root mean square error(RMSE) metric (Equation 9). In Equation 9, e_t is the error between predicted and actual interest level value of the tth item. Lower RMSE score shows better model performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} e_t^2} \quad (9)$$

IV. EXPERIMENTS AND DISCUSSION

In this section, we give the details of the experimental design in order to validate the proposed approach. We run these experiments on six different datasets provided by a UK based personalisation company.[2] The overview of the

[2]https://www.freshrelevance.com/

experimental design is seen in Figure 3. We split dataset as train and test datasets last three days of the sessions are selected for the test dataset. Train dataset is used to train the recommendation and linear regression model. The test dataset is used to get recommendations and to calculate users' interest level score for each recommended item in a test session. Finally, the recommendation list is re-ranked based on the recommendation score and users' interest score on recommended items.
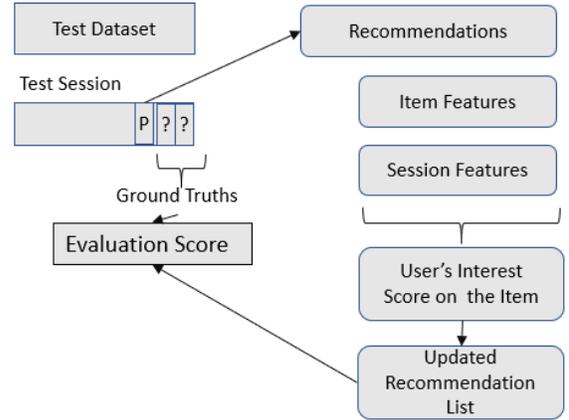

Fig. 3. Experiment design to evaluate the proposed approach(P refers to last interacted product in the session)

A. Dataset Analysis

In this subsection, we give details of the datasets we used in the computational experiments. The dataset statistics about the number of items, the number of users, the number of interactions in the datasets are given in Table I.

We eliminate sessions with less than 5 item interactions to alleviate the drawback of sparsity and cold start problem. New statistics generated after sessions filtering is shown in Table II.

It can be seen from Table II that the difference in the number of sessions between the original dataset and dataset after filtering is significant. This difference gives an indication

TABLE I
DATASET DETAILS BEFORE FILTERING

websiteid	#users	#items	#interactions
dataset 1	148832	3835	551851
dataset 2	361330	33273	987432
dataset 3	478089	20480	765231
dataset 4	301089	9111	876078
dataset 5	215240	17431	519251
dataset 6	266871	34152	981654

TABLE II
DATASET DETAILS AFTER FILTERING SESSIONS WHICH HAVE LESS THAN 5
ITEM INTERACTIONS

websiteid	#users	#items	#interactions
dataset 1	15048	496	161938
dataset 2	34613	28620	294003
dataset 3	25477	12559	225713
dataset 4	37475	7565	402593
dataset 5	14347	12512	147922
dataset 6	44489	25947	428388

of the level of sparsity, and cold start in original dataset since most of the sessions have less than 5 item interactions.

Moreover, we analyse users' previous session details since we use these features in regression model training. Table III shows statistics about the average number of sessions, average number of purchases and average the number of viewed different products in a session for a user.

TABLE III
AVERAGE NUMBER OF PREVIOUS SESSIONS, PREVIOUS PURCHASES AND
PAGE VIEWINGS FOR REGISTERED USERS IN SIX DATASETS

websiteid	#session	#purchases	#views
dataset 1	2.8010	0.3647	8.7956
dataset 2	2.2141	0.0264	6.1945
dataset 3	0.4131	0.0052	4.1581
dataset 4	0.9042	0.0351	5.1308
dataset 5	2.0006	0.1003	8.9196
dataset 6	3.8511	0.8343	9.1571

As shown in Table III, in dataset 3, the interactions are done by new users since the number of the previous session is very small when compared to other datasets. Also, dataset 3 has the lowest number of purchases in the sessions.

B. Experiment Results and Discussion

In order to perform experiments, we build a session-based item-item similarity RS model. The details of the parameters used in the session-based item-item similarity RS model explained in Section 2. Secondly, in order to implement re-ranking of recommended items, we build the regression model, in which extracted features from item popularity and session specifications are used to train the regression model. Recommendation list from recommender model is used as candidate items that will be ranked based on the *interest level* prediction of the regression model.

Test sessions are selected from test dataset with ten cross-validation method to have more reliable results. Each fold has 500 test sessions, last two interacted items in the test session are hidden then we compare the performance of the baseline method and proposed re-ranked approach on predicting these hidden items. We put the performance comparison of the baseline and proposed approach for each dataset in Table IV, and the results shows the average scores of ten cross-validations. In the result table, $Baseline$ illustrates to the recommendation model without applying proposed re-ranking method. $Re-Ranked\ All$ refers to re-ranked applied Recommendation model, and for training the regression model, whole train dataset is used. $Re-Ranked\ 0.8$ indicates re-ranked applied Recommendation model, and for training the regression model, 80 % of train dataset is used. $Re-Ranked\ 0.5$ refers to re-ranked applied Recommendation model, and for training the regression model, 50 % of train dataset is used.

As seen in Table IV, computational experiments show that there are considerable improvements in terms of recall and precision scores for all datasets. These results show that when re-ranking is applied, the RS model can have a better prediction performance for users' next item interaction. Also, the table shows that it is possible to capture users' interest level using users' session activities and items' temporal features. Moreover, results prove that integrating users' interest level on recommended items in a session could be utilised to re-rank the recommendation list to improve personalisation by recommending the right product on top of the recommendation list.

We can see the highest performance improvements in dataset 3 in all n recommendation cases since there were not any matched recommended items on top top-n($n \in \{1, 3, 5\}$) recommendation. On the other hand, the lowest improvement is found in dataset 6 with 62%. We also analysed the performance of regression models(Table IV). The best-performed regression model was found in the trained on dataset 3 with 0.6887 RMSE. On the contrary, the regression model trained on dataset 1 gave the worst performance with 1.5766 RMSE. We also investigated the performance change of the recommender system when different dataset size is used to train the regression model. In the first, second, and last experiments, we used 50%, 80%, and 100% of the dataset, respectively. However, based on the performance improvements on the recommender model, we could not find a positive correlation between the performance of trained recommender and regression models (Table V). For example, the performance results of the recommendation model in dataset 2 did not show any difference when we trained the regression model in different training dataset size.

Also, we measure the regression model performance for various sampling dataset. It should be noted that we did not apply sampling on the test dataset. In other means, we keep test dataset as same for each sampling size in order to have a fair comparison regardless of whether sampling is applied or not.

Moreover, we investigated the performance of the regression

TABLE IV
PERFORMANCE COMPARISION OF RE-RANKED AND BASELINE RECOMMENDATION LIST USING RECALL(R) AND PRECISION(P) METRICS

Dataset	Method	P@1	R@1	P@3	R@3	P@5	R@5
dataset 1	Baseline	0.006	0.003	0.005	0.007	0.007	0.018
	Re-Ranked All	0.016	0.008	0.026	0.039	0.021	**0.053**
	Re-Ranked 0.8	0.016	0.008	0.025	0.038	0.021	**0.053**
	Re-Ranked 0.5	**0.018**	**0.009**	**0.029**	**0.044**	0.020	0.050
	Improvement(%)	66.667	66.667	84.138	84.091	64.000	64.000
dataset 2	Baseline	0.000	0.000	0.000	0.000	0.000	0.000
	Re-Ranked All	**0.004**	**0.002**	**0.007**	**0.007**	**0.010**	**0.024**
	Re-Ranked 0.8	**0.004**	**0.002**	**0.007**	**0.007**	**0.010**	**0.024**
	Re-Ranked 0.5	**0.004**	**0.002**	**0.007**	**0.007**	**0.010**	**0.024**
	Improvement(%)	100	100	100	100	100	100
dataset 3	No Re-Ranked	0.008	0.004	0.004	0.006	0.004	0.009
	Re-Ranked All	0.016	0.008	**0.015**	**0.022**	**0.014**	**0.034**
	Re-Ranked 0.8	0.016	0.008	0.014	0.021	0.012	0.030
	Re-Ranked 0.5	**0.018**	**0.009**	0.013	0.020	0.012	0.030
	Improvement(%)	55.556	55.556	72.727	72.727	73.529	73.529
dataset 4	Baseline	0.000	0.000	0.001	0.001	0.001	0.002
	Re-Ranked All	**0.006**	**0.003**	**0.004**	**0.006**	0.005	0.013
	Re-Ranked 0.8	**0.006**	**0.003**	**0.004**	**0.006**	**0.006**	**0.014**
	Re-Ranked 0.5	**0.006**	**0.003**	**0.004**	**0.006**	**0.006**	**0.014**
	Improvement(%)	*100.000*	*100.000*	*83.333*	*83.333*	*85.714*	*85.714*
dataset 5	No Re-Ranked	0.002	0.001	0.001	0.001	0.000	0.001
	Re-Ranked All	**0.014**	**0.007**	**0.009**	**0.014**	**0.013**	**0.033**
	Re-Ranked 0.8	0.012	0.006	**0.009**	0.013	**0.013**	**0.033**
	Re-Ranked 0.5	0.012	0.006	**0.009**	**0.014**	**0.013**	0.032
	Improvement(%)	85.714	85.714	92.857	92.857	96.970	96.970
dataset 6	Baseline	0.004	0.002	0.001	0.002	0.002	0.004
	Re-Ranked All	0.018	0.009	**0.017**	**0.026**	**0.014**	**0.035**
	Re-Ranked 0.8	0.018	0.009	**0.017**	**0.026**	**0.014**	**0.035**
	Re-Ranked 0.5	**0.024**	**0.012**	**0.017**	**0.026**	**0.014**	0.034
	Improvement(%)	83.333	83.333	92.308	92.308	88.235	88.571

TABLE V
COMPARISON OF REGRESSION MODEL PERFORMANCE IN DATASETS
WITH DIFFERENT DATA SAMPLING

Dataset	RMSE		
	1	0.8	0.5
dataset 1	1.5766	1.5769	1.5777
dataset 2	0.7890	0.7889	0.7897
dataset 3	0.6887	0.6887	0.6894
dataset 4	0.7836	0.7836	0.7834
dataset 5	1.1195	1.1207	1.1217
dataset 6	0.9956	0.9995	1.0052

model trained on different dataset samples(Table V). The computational experiments showed that the regression model trained on more data performs better in terms of RMSE metric. Since having more data can help to learn the patterns in user behaviours better. The experiments show that the best-performing regression model found in dataset 3 with 0.6887 when the whole train dataset is used.

V. CONCLUSION AND FUTURE DIRECTION

In this work, we applied a re-ranking approach using a linear regression model on the recommendation list for next item prediction. Re-ranking approaches have been applied in different works [21]–[23], [25], [27], [38] where context-awareness, diversity and popularity based re-ranking options have been used. In this work, we applied a linear regression model to re-rank given the candidate recommendation list, and predict an interest level for recommended items based on multiple factors such as users' behaviours in users' previous sessions and recommended items' features. Computational experiments showed that applying re-ranking on recommendations based on the interest level improved recommendation accuracy. The improvement of correct next item prediction recommendation can have a significant contribution for user experience since generally, in e-commerce platforms, in the first page, five recommended items are listed. Similar to [22], our work confirms the importance of context-awareness in a session-based recommendation to improve the recommendation model performance by having a better recall and precision metrics.

For future direction, a combination of classification and regression models can be useful when re-ranking the recommended items. Moreover, the proposed method can be adapted to deep learning RS models, since our method is independent of the recommendation model. In this way, our proposed method can be considered as re-ranking the recommended items after the recommender system has built. Also, we a future research we will apply sampling methods before training the regression model for class imbalance problem [39]–[41]. This could be interesting since, in the datasets, most of the sessions are ending without transaction that can lead to imbalanced results in the regression model.

REFERENCES

[1] F. Pecune, S. Murali, V. Tsai, Y. Matsuyama, and J. Cassell, "A model of social explanations for a conversational movie recommendation system," in *Proceedings of the 7th International Conference on Human-Agent Interaction*. ACM, 2019, pp. 135–143.

[2] M. Ludewig, I. Kamehkhosh, N. Landia, and D. Jannach, "Effective nearest-neighbor music recommendations," in *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 2018, p. 3.

[3] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[4] C. Pei, Y. Zhang, Y. Zhang, F. Sun, X. Lin, H. Sun, J. Wu, P. Jiang, J. Ge, W. Ou *et al.*, "Personalized re-ranking for recommendation," in *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 2019, pp. 3–11.

[5] B. Smith and G. Linden, "Two decades of recommender systems at amazon. com," *Ieee internet computing*, vol. 21, no. 3, pp. 12–18, 2017.

[6] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Performance comparison of neural and non-neural approaches to session-based recommendation," in *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 2019, pp. 462–466.

[7] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, no. 4-5, pp. 331–390, 2018.

[8] L. Zhang, P. Liu, and J. A. Gulla, "A deep joint network for session-based news recommendations with contextual augmentation," in *Proceedings of the 29th on Hypertext and Social Media*. ACM, 2018, pp. 201–209.

[9] I. Kamehkhosh, G. Bonnin, and D. Jannach, "Effects of recommendations on the playlist creation behavior of users," *User Modeling and User-Adapted Interaction*, pp. 1–38, 2019.

[10] D. Jannach, M. Ludewig, and L. Lerche, "Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts," *User Modeling and User-Adapted Interaction*, vol. 27, no. 3-5, pp. 351–392, 2017.

[11] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017, pp. 306–310.

[12] R. Esmeli, M. Bader-El-Den, and A. Mohasseb, "Context and short term user intention aware hybrid session based recommendation system," in *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2019, pp. 1–6.

[13] A. Martínez, C. Schmuck, S. Pereverzyev Jr, C. Pirker, and M. Haltmeier, "A machine learning framework for customer purchase prediction in the non-contractual setting," *European Journal of Operational Research*, vol. 281, no. 3, pp. 588–596, 2020.

[14] M. Reusens, W. Lemahieu, B. Baesens, and L. Sels, "A note on explicit versus implicit information for job recommendation," *Decision Support Systems*, vol. 98, pp. 26–35, 2017.

[15] G. Li and Q. Chen, "Exploiting explicit and implicit feedback for personalized ranking," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[16] M. H. Aghdam, "Context-aware recommender systems using hierarchical hidden markov model," *Physica A: Statistical Mechanics and its Applications*, vol. 518, pp. 89–98, 2019.

[17] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 241–248.

[18] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 101–109.

[19] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, "Item-based collaborative filtering recommendation algorithms." *Www*, vol. 1, pp. 285–295, 2001.

[20] S. E. Park, S. Lee, and S.-g. Lee, "Session-based collaborative filtering for predicting the next song," in *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*. IEEE, 2011, pp. 353–358.

[21] R. Esmeli, M. Bader-El-Den, and H. Abdullahi, "Improving session based recommendation by diversity awareness," in *UK Workshop on Computational Intelligence*. Springer, 2019, pp. 319–330.

[22] X. Li, Y. Chen, B. Pettit, and M. D. Rijke, "Personalised reranking of paper recommendations using paper content and user behavior," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, p. 31, 2019.

[23] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2011.

[24] A. Bhardwaj, W. Di, M. R. Hamid, R. Piramuthu, and N. Sundaresan, "Extraction of image feature data from images," Aug. 5 2014, uS Patent 8,798,363.

[25] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," in *The Thirty-Second International Flairs Conference*, 2019.

[26] M. Shokouhi and Q. Guo, "From queries to cards: Re-ranking proactive card recommendations based on reactive search history," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 695–704.

[27] D. L. Nguyen and T. M. Le, "Recommendation system for facebook public events based on probabilistic classification and re-ranking," in *2016 Eighth International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2016, pp. 133–138.

[28] D. Skoutas, M. Alrifai, and W. Nejdl, "Re-ranking web service search results under diverse user preferences," in *VLDB, Workshop on Personalized Access, Profile Management, and Context Awareness in Databases*, 2010, pp. 898–909.

[29] W. Rong, B. Peng, Y. Ouyang, K. Liu, and Z. Xiong, "Collaborative personal profiling for web service ranking and recommendation," *Information Systems Frontiers*, vol. 17, no. 6, pp. 1265–1282, 2015.

[30] M. Volkovs, H. Rai, Z. Cheng, G. Wu, Y. Lu, and S. Sanner, "Two-stage model for automatic playlist continuation at scale," in *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 2018, p. 9.

[31] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008, pp. 263–272.

[32] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 273–282.

[33] H. Chen, D. Niu, K. Lai, Y. Xu, and M. Ardakani, "Separating-plane factorization models: Scalable recommendation from one-class implicit feedback," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 669–678.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[35] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 66, 2018.

[36] T. Pradhan and S. Pal, "A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity," *Future Generation Computer Systems*, 2019.

[37] M. Kunaver and T. Požrl, "Diversity in recommender systems–a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017.

[38] D. Véras, R. Prudêncio, and C. Ferraz, "Cd-cars: Cross-domain context-aware recommender systems," *Expert Systems with Applications*, vol. 135, pp. 388–409, 2019.

[39] M. Bader-El-Den, E. Teitei, and T. Perry, "Biased random forest for dealing with the class imbalance problem," *IEEE transactions on neural networks and learning systems*, 2018.

[40] M. Bader-El-Den, "Self-adaptive heterogeneous random forest," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2014, pp. 640–646.

[41] T. Perry, M. Bader-El-Den, and S. Cooper, "Imbalanced classification using genetically optimized cost sensitive classifiers," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 680–687.