I-CARS: An Interactive Context-Aware Recommender System

Rosni Lumbantoruan†, Xiangmin Zhou†, Yongli Ren†, Lei Chen‡
† *School of Science, RMIT University,* Melbourne, Australia
‡ *Hong Kong University of Science and Technology,* Hong Kong, China
†{rosni.lumbantoruan, xiangmin.zhou, yongli.ren}@rmit.edu.au, ‡ leichen@cs.ust.hk

***Abstract*—Context-aware recommendation has attracted significant attentions over online sites due to its smart context adaption in improving recommendation quality. However, the user's instant contexts do not follow his/her regular user behaviour patterns, thus have not been well captured for advanced personalization of recommendation generation. In this work, we propose an Interactive Context-Aware Recommender System (I-CARS), which allows users to interact and present their needs, so the system can personalize and refine user preferences. I-CARS iteratively asks a question to a user to trigger feedback in term of her recent contexts and incorporates the response to recommend items most likely satisfying his/her instant interests. Specifically, we first propose a Personalized Weighted Context-Aware Matrix Factorization (PW-CAMF) that enables the personalization of important contexts for each user. Then we propose two question selection strategies that exploit user preferences through feedback. We have conducted comprehensive experiments over two real datasets. The experimental results prove the effectiveness of our I-CARS system compare to existing competitors.**

***Keywords*-interactive, context-aware, recommender system**

I. Introduction

With technology continually evolving, so are the online activities like shopping, which becomes an important aspect of people's daily life. Hostingfacts reported around 1.92 billion people are expected to conduct online purchase in 2019[1] and Amazon Marketplace alone offers more than 353 million products online[2]. The massive number of online items and social users raises some challenges, e.g. users find it challenging to navigate through all the items to look for what they like, while the companies meet the challenge of advertising the right produts to the right users. Context-aware recommender systems (CARS) emerge to tackle these issues. Basically, CARS interpolate contexts behind user transactions to better understand user preferences, then recommend items based on their relevance to the users preferences. Even though there has been significant progress on the quality of the context-aware recommendation, there are cases where the user's behaviour can not be tracked solely on their previous transactions due to the influence of external factors in decision making. For example, Alex, as a food reviewer, only reviews all restaurants but Asian. However, lately most of his subscribers request him to review Asian food, which makes him needs to visit Asian restaurants. Failing to capture this change and generating recommendation based on the previous transactions might lower the recommendation quality. It becomes worthwhile investigating the current users' contexts by allowing them to refine their contexts via the interaction with the system.

In this work, we study interactive context-aware recommender system. Given a target user u and an item set I, interactive recommender system aims to exploit the interaction between u and the system for capturing the dominant contexts and preferences of u, and return a list of items with the best relevance to u in the dominant context subspace. In this paper, we focus on the problem of effective interactive context-aware recommendation for e-business applications. As formulated above, three key issues need to be addressed to tackle the problems. First, we need to construct a model that enables the personalization of user preferences and allow different factors priorities in the final recommendation. For instance, Alex puts the flavour and variety of menu in the order of priority when choosing a restaurant. Failing to capture personalized user preferences and its influence in decision-making, may mislead the recommendation results. Second, we need to design a selective question selection strategy to retrieve user feedback and maximize the effect of the interaction. Each question aims to retrieve user feedback and refine their preferences. Finally, we need to design an advanced learning approach that incorporates user rating feedback into the model to keep the recommendation quality.

There has been considerable research that facilitates interaction in recommendation [1], [2]. Specifically, He et al. give higher weight for the new interaction to refine user preferences [1], and Christakopoulou et al. refine user preferences by incorporating absolute and relative feedback [2]. However, none of them considered the contexts as a part of the user interaction, which negatively affects the personalization of recommendation results. Approaches have been proposed to exploit the social contexts for user interest identification [3], [4]. Zhou et al. [3] and Lumbantoruan et al. [4] identify the different importance of contexts prior to the recommendation generation for a particular dataset and a particular target user respectively for more personalized and effective recommendation. Working on top of these CARS cannot meet the system objective, since the contexts of each user might evolve during the recommendation, but all previous works assume the users' preference does not change.

[1]https://hostingfacts.com/internet-facts-stats/
[2]https://www.retailtouchpoints.com/resources/type/infographics/how-many-products-does-amazon-carry

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00154

Motivated by the limitations of the current systems, we propose an Interactive Context-Aware Recommender System (I-CARS) that enables the contextual-based interaction with the users. I-CARS interactively triggers users to reveal their recent preferences by asking a question representing their previous transactions and context consideration. First, we propose a personalized context-aware recommendation based on matrix factorization, called $personalized\ weighted - context - aware\ matrix\ factorization$ (PW-CAMF) that learns the individual contexts importance and priority based on users' previous transactions. Then, we propose the interactivity on top of PW-CAMF by proposing some item selection strategies that choose the most relevant question in order to refine user preferences. Finally, we propose the contextual learning to adapt the model with each user response. Our key contributions are summarized as below.

1) We propose a new context-aware recommender system called PW-CAMF that generates the initial recommendation for a target user. PW-CAMF models the importance difference and priority of contexts for each user.
2) We propose some question selection strategies to select an item that will be used a medium to trigger user feedback. The question selection strategies consider user contexts in selecting the most relevant question to ask in each interaction.
3) We propose a learning approach that incorporates the user context-based rating feedback into the system. This feedback mechanism enables the model to capture the evolvement of user contexts and the refinement of user preferences during the interaction.
4) We conduct extensive experiments to verify the effectiveness of the proposed solution on two benchmark datasets.

The rest of the paper is organized as follows: Section II reviews the related work on interactive recommender systems. Section III presents our I-CARS framework and its main components. Experimental results are reported in Section IV. Finally, Section V concludes the whole paper.

II. Related Work

We review the three existing research that closely related to our work, including the context-aware recommender system, the interactive recommender system, and the exploitation and exploratory trade-off.

Context-Aware Recommender System: Content and contextual information have been embedded in various recommender systems. Typical examples include predefined and static attribute sets [5], dynamic context [6]–[9], and dynamic contexts with importance difference [3], [4]. In [5], Baltrunas et al. model the interaction of the contextual factors to the specific situation in which the item will be consumed. CAMF-CC which grouping the items per category shown to have a beneficial effect comparing other alternatives (CAMF-C and CAMF-CI). In [6], Ren et al. use location, queries, and web content for making a contextual recommendation. Zhang et al. [7] identifies features that a user concerns and recommends a product that performs well on those features. Zhou et al. [8] fuse the content and social relevance to identify the relevant video over online sharing communities. Meanwhile, Zhou et al. [9] exploit the contextual information of social users to enhance the video recommendation to multiple users. These approaches preset the contexts used for the final recommendation, which ignores personalization of contexts, thus the contexts used may not be optimal for the real world situations. Recent approaches proposed to support the different importance of contextual information [3], [4]. Zhou et al. [3] identify the optimal features based on the correlation between a feature to a feature set to remove the feature redundancy. An approximately optimal feature set is identified for the content-context interaction graph-based social recommendation. In [4], Lumbantoruan et al. identify the personalized contexts for each user by using topic modeling, UW-NMF, and then generate the recommendation based on the most relevant contexts to the target user. However, none of these approaches can handle the situation where the context interests of a target user change during the recommendation generation.

Interactive Recommender System: Several recommender system approaches have been proposed to utilize the user interaction during the recommendation process. Typical examples include dialog-based recommenders [2], [10] and critique-based recommenders [11], [12]. In [2], Christakopoulou et al. elicit user's preferences by utilizing user responses into the model in order to get a better recommendation result. This model also handles cold-start users by introducing initial embedding that is learned offline by assuming that the new user is similar to these average offline users. This approach provides an outstanding result with their absolute feedback strategy. In [10], Wang et al. interactively refine the user preference by considering the influence of the other users for the final recommendation. For critique-based recommender systems, McCarthy et al. [11] and Salem et al. [12] treat the previously successful critiquing sessions as the additional candidate items proposed during the interaction. The aforementioned approaches do not personalize the contexts for each user. In addition, they are heavily dependent on the rarely available rich-predefined datasets for the modeling of item features prior to recommendation. In this work, we aim to refine the personalized user preferences interactively through feedback during the recommendation process. This enables users to take part in the recommendation process without having to know the details of the underlying algorithm.

Exploration and Exploitation Strategy: Recommender Systems predict and recommend the most relevant items to the needs of the target users. It can be done by either exploiting the items that have the highest relevancy prediction to the users or by encouraging them to browse unrated items while attempting to gain new information about user preferences through interactions. In [2], Christakopoulou et al. use Thomson Sampling (TS) and Upper Confidence Bound (UCB) strategies to balance between the exploitative and exploratory strategies. TS is a bandit algorithm balancing both strategies by selecting items using a sampling strategy [13], while UCB selects the items with the highest confidence bound to avoid missing preferences for promising items [14]. Although these methods addressed the balance between exploratory and exploitative strategies, there is still room for the improvement of both strategies by considering the context while selecting the most informative items/questions. In this work, we aim to balance both strategies by considering the personalization of contexts in selecting the question/item to ask.

III. Framework of I-CARS

In this section, we present a novel Interactive Context-Aware Recommender System (I-CARS), including PW-CAMF model, question selection, and context learning.

A. PW-CAMF

It is a fact that users tend to prioritize contexts when they select online items or give feedback during the interaction. Recent work has considered the personalization of contexts for each user in the pre-filtering approach [4]. However, it assumes that once identified, contexts contribute uniformly to the final recommendation. In this paper, we removed this assumption by using contextual modeling approach that uses the contexts inside the recommendation-generating algorithms.

Specifically, on the basis of Matrix Factorization, we define context personalization for each user by introducing different weight parameters for both contextual information and latent user factors. We can define this officially as follows:

$$\hat{r}_{ui,[c_1...c_g]} = \mu + b_i + b_u + (v_u \odot w_u)^T \cdot q_i + (w_u \odot \sum_{l=1}^{|L|} \sum_{j=1}^{z} B^i_{lj}), \tag{1}$$

where $\hat{r}_{ui,[c_1...c_g]}$ denotes the predicted rating of user u on item i in the contexts of $[c_1...c_g]$, e.g. the weather context, and it can have various pre-defined values: windy, raining, and sunny; $z = \sum_i^g z_i$, where g is the number of contextual factors and z_i is the number of possible values (conditions) of the i-th contextual factor c_i. B^i_{lj} denotes the parameters modeling the interaction of item i's l-th category (L denotes the set of item categories) with the j-th contextual value. Note, there is only one parameter for each contextual condition and item's category pair. In practice, if item i and item j have the same category, $B^i_{lj} = B_{ljc_j}$; v_u are user factors, and q_i are item factors; $\odot$ denotes the element-wise multiplication or Hadamard product, and $(v_u \odot w_u)$ indicates the interest of user u on latent factors of item i by considering the weight given by user u to each of the factors; w_u denotes the weights of different contexts for user u; Given $|L|$ denotes the number of different item categories and z denotes the number of contextual conditions, then the total number of parameters is $(|L| \cdot z)$. In this way, we consider each user's priority in terms of contexts while modelling their preferences. Thus, PW-CAMF tries to minimize the following objective function $\mathcal{L}(\theta)$, by minimizing the prediction error as belows:

$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{u,i,c \,\in\, \kappa} (r_{ui,[c_1...c_g]} - \hat{r}_{ui,[c_1...c_g]})^2 + \frac{\lambda}{2}|\theta|_2 \tag{2}$$

where θ is the set of unknown parameters to learn (μ, b_i, b_u, v_u, q_i, w_u, B^i_{lj}), κ is the set of all user, item, rating, and context pairs, while $\lambda = (\lambda_v, \lambda_q, \lambda_w, \lambda_{B^i_{lj}})$ is the set of the regularization parameters. Stochastic Gradient Descent (SGD) was conducted to update the parameters. Like most recommender systems, the proposed PW-CAMF is trained based on users historical transaction data (e.g. ratings) for generating recommendations. Later, it periodically updates the model to incorporate user responses. In the next sections, we present our interaction mechanism that works on top of PW-CAMF.

B. Question Selection for Interactions

Here, we focus on how to select suitable items to form a question to ask users for their feedback on them. With these direct interactions to users, I-CARS aims to capture the recent users' contextual preferences by asking question that will reduce the uncertainty of user preferences. Given a set of items of size n, we aim to select an optimal item subset of fixed size that will minimize the user preference uncertainty. Via the feedback from a user to the formed question about an item, we gain some information to further understand the user's preference. Here, we propose two strategies, $MaxIUWF$ and $MaxWeight$, to find the items to ask users: Given a set of I of all candidate items, $MaxIUWF$ and $MaxWeight$ perform the selection over I by two steps: (1) compute the item value of each $i_j \in I_l$, with respect to the remaining of the set $I_l \backslash \{i_j\}$; (2) compare weight of two candidates and remove the minimum weight value. The two steps are conducted recursively until one item with the maximum information gain found. Both $MaxIUWF$ and $MaxWeight$ are respectively described as below.

MaxIUWF Selection Strategy: Maximum Item-User-Weight Factor triggers user feedback by selecting an item $i*$ that falls into the most important context and latent factors for user. Note, since we update w_u during the interaction, we also facilitates context in this parameter. MaxIUWF tries to balance the importance of context with user and item vectors and is calculated as follows:

$$i^* = \arg\max_i \sum_{h=1}^{d} \left((v_{u_h} \odot w_{u_h})^T \cdot q_{i_h}\right) \tag{3}$$

where v_{u_h}, q_{j_h} and w_h denote the feature vector for user, item and context weight for each latent factor h. MaxIUWF orders the question to ask user from the highest to the lowest contexts and factors importance from the user previous transaction. By prioritizing all these factors in each interaction with the users, I-CARS will learn their current preferences more effectively.

MaxWeight Selection Strategy: Maximum-Context-Weight question selection strategy relies only on the priority parameter w to select the item $i*$ to trigger user response. We aim to measure the importance of this parameter in learning user preferences. MaxWeight selection strategy is defined as below:

$$i^* = \arg\max_i \sum_{h=1}^{d} w_{ui_h} \tag{4}$$

which is the sum of weights over all user's latent factors. This strategy gives the opportunity to learn and prioritize corresponding user and item latent factors based on user's

input:θ = (μ, v_u, q_i, w_u, B^i_{lj}), new feedback=(u,i^*, c, $\hat{\hat{r}}$)
output: refreshed parameters $\hat{\theta}$ = ($\hat{\mu}$, $\hat{v_u}$, $\hat{q_i}$, $\hat{w_u}$, $\hat{B}_{ijc_j}$), recommendation list
1. **for each user** u interaction **do**
2. $\hat{\mu} \leftarrow \mu$; $\hat{v_u} \leftarrow v_u$; $\hat{q_i} \leftarrow q_i$; $\hat{w_u} \leftarrow w_u$, $\hat{B}_{ijc_j} \leftarrow B^i_{lj}$
3. *incorporate feedback* (u,i^*, c, $\hat{\hat{r}}$) to the training set
4. *update* $\mu = \frac{1}{n}\sum_{i=1}^{n}(r_{ui,[c_1 \ldots c_g]})$
5. *refine* v_u, q_i, w_u, B^i_{lj} with SGD
6. **return** $\hat{\theta}$ and recommend top-ranked items to user u

Fig. 1: Context Learning

response. Recalling that our question selection strategies are incorporating contexts during the question selection, it is possible for $MaxWeight$ or $MaxIUWF$ to ask the same question to a user but in a different context. When we ask an item, we assume that the user will provide feedback to the system, then we thus utilize this to model their preferences.

C. Context Learning through Interaction

Context learning aims to facilitate the continuous learning of user preferences by incorporating the user response into the recommender. Christakopoulou et al. [2] incorporates the new observation for user u in either (u, i^*, 1) or (u, i^*, 0), which respectively represent whether the user likes or dislikes the item. This approach does not incorporate contexts in both the interaction and the model. Recalling the importance of contexts in recommendation, we also incorporate context learning after each interaction. As the system incorporate user's feedback, the more compact the system is in understanding the recent user's preferences.

Specifically, contexts are considered when selecting a question to ask user. Given user u response (u, i^*, c, $\hat{\hat{r}}$), this new observation will be incorporated into the model and the parameters θ of the active user will be relearned. User u's response does not directly alienate user u's attitude towards contexts or item but corresponds to many unknown factors that contribute to the item selection. Unlike the existing literature, where the feedback rating is 0 or 1, we define the feedback rating, $\hat{\hat{r}}$, for either positive or negative feedback as below:

$$\hat{\hat{r}} = \frac{1}{n}\sum_{i=1}^{n}\left(r_{ui,[c_1 \ldots c_g]}\right), \tag{5}$$

and by following [10], we also define a rate higher than 3 as a positive rating. Given the formula in Eq. (5), we ensure the new observation will always reflect the tendencies of users/items in giving/receiving higher (lower) rates than others. These different tendencies have been facilitated as described in Eq. (1). The algorithm for context learning is depicted in Fig. 1.

Specifically, Fig. 1 shows how to refine the modelling of user preferences given the new feedback from each user interaction. Specifically, the user interaction feedback are adapted back to the model by updating corresponding user u parameters (μ, v_u, q_i, w_u, B^i_{lj}). As the system is continuously relearning the corresponding parameters, the new belief about the user, item, and context is changing as well. This changes will influence the question that will be selected in the next interaction. Once the interactions are completed, the top-ranked items are generated based on the most recent model.

TABLE I: Dataset

Data	User	Item	Category	Context	Avg.Rating
STS Travel	317	248	30	27	39
InCarMusic	42	139	10	8	96

IV. Experimental Evaluation

In this section, we examine the effectiveness of our interactive recommender system by conducting comprehensive experiments on two real public datasets.

A. Experimental Set-up

Following existing works [15]–[17], we conduct experiments over two real benchmark contextual datasets, STS travel and InCarMusic dataset. STS (South Tyrol Suggest) is a travel rating data derived by STS app and have been used by [15] and [16]. InCarMusic is a dataset that is collected to assess the effect of context on music preferences in the car and has been used by [17]. The details of these datasets are described in Table I. We divide each dataset respectively into training, feedback, and test dataset. The training dataset is used by the model to learn the users' preferences based on their historical data. The feedback dataset is treated as the interactions from the user to the system, while the test dataset is used to evaluate the model performance. Ground truth is retrieved from the test dataset where the rating is equal to or larger than 4 or 1 for the binary dataset, we also ensure that at least one positive rating exists for each user.

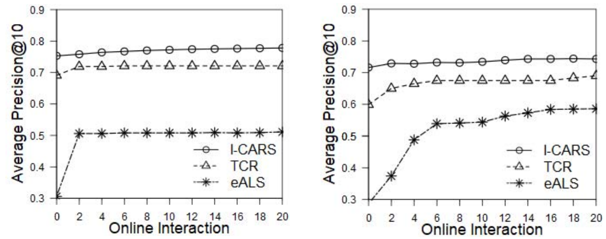
B. Evaluation Methodology

We compare the proposed methods with 8 state-of-the-art baselines, which can be categorized into three groups: i) 4 traditional collaborative filtering including deep learning recommendations (ItemKNN [18], SVD++ [19], AutoRec [20], Weighted-SVD [1]); ii) 2 content/contextual recommendations (CAMF [5], EFM [7]), and (iii) 2 interactive/conversational recommendations (TCR [2], eALS [21]). LibRec Package [22] is used to run the experiment for SVD++, ItemKNN, and EFM. For EFM, we treat the contexts as positive/negative sentiment based on the existence of the contexts in rating. We apply 5-fold cross validation in this study, and all the baselines are well tuned.

The effectiveness of the above evaluation is measured by using the most common error-based metrics, RMSE and MAE, as used in [3], [5]. *MAE* measures the absolute difference between the set of predicted ratings and the corresponding ground truth in the test set. Meanwhile *RMSE* measures the root square of prediction error on all instances of the test data-set. We also evaluate the proposed method with the existing interactive/conversational approaches, the effect the question selection strategies, and the effect of model learning in incorporating user response after each interaction. We evaluate the effectiveness in terms of *Hit Ratio* (HR) and *Average Precision* at k ($AP@k$). For *HR*, a hit occurs when

TABLE II: Effectiveness of context personalization

	RMSE		MAE	
	STSTravel	InCarMusic	STSTravel	InCarMusic
SVD++	1.174	1.376	0.949	1.151
ItemKNN	1.023	1.519	0.799	1.289
AutoRec	1.145	1.860	0.928	1.480
Weighted-SVD	1.040	1.130	0.834	0.894
CAMF	1.159	1.132	0.919	0.887
EFM	1.304	1.448	0.982	1.245
PW-CAMF	**0.927**	**1.092**	**0.740**	**0.869**



(a) $STS\ Travel$ (b) $InCarMusic$

Fig. 2: I-CARS vs. existing competitors

the recommended item is in a user's ground truth while no-hit for another case [23]. Later, *HR* was calculated by dividing the number of hits to the size of ground truth in test set. $AP@k$ is a widely used precision-based metric [24] to represent the precision of the top-k (we set k = 10) ranked recommendation list compared to the user's ground truth. *AP@k* is defined as the average of precision computed at each position when hit occurs in the top-k recommendation list as defined in [2]. We also conduct the paired sample t-test to evaluate whether the improvement is statistically significant.

C. Experimental Results

1) Comparing with Non-Interactive Algorithms: We compare the effectiveness of our proposed PW-CAMF with other recommended systems, SVD++, ItemKNN, CAMF, Weighted-SVD, AutoRec, CAMF, and EFM. To get a fair comparison, we test and set the optimal parameter for each algorithm and test their performance over all datasets. Table II shows the effectiveness comparison of PW-CAMF over the two datasets in terms of *RMSE* and *MAE*.

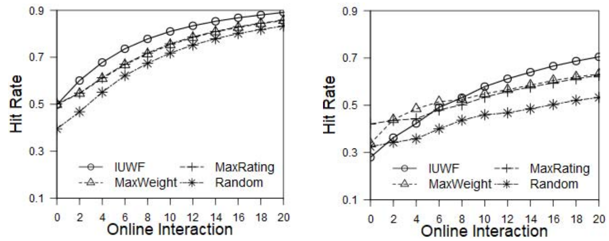
As we can see, our PW-CAMF achieves better performance over all non-contextual recommenders by showing the lowest *RMSE* and *MAE* over the two datasets. This is because the PW-CAMF facilitates contexts and its personalization which has been proved, can increase the recommendation accuracy. AutoRec performs worst for both small STSTravel and InCarMusic dataset. The main reason is that deep learning models are data hungry [25], but both STSTravel and InCarMusic are very sparse. Comparing with content/contextual recommenders (CAMF and EFM), our PW-CAMF obtains much higher effectiveness because of enabling the personalization of contexts for each user. The high efficacy of our PW-CAMF has proved its superiority as the initial method for building I-CARS over other competitors.

2) Comparing with Interactive Algorithms: We compare our I-CARS with the existing competitors in interactive systems (eALS and TCR) in term of recommendation accuracy. We set the interactions as 20 and recommend top 10 for each interaction. Figure 2 shows the comparison of I-CARS and two existing competitors, TCR and eALS in terms of AP@10. Clearly, I-CARS significantly outperforms TCR and eALS over all datasets. I-CARS can surpass both competitors even before the online interaction due to three reasons. Firstly, our I-CARS, by using PW-CAMF as the offline model, is able to learn each user's preferences more precisely by treating different contextual importance for each user. Secondly, I-CARS with *MaxIUWF* item selection strategy can select the proper question to ask during the interaction for optimising the refinement of user's preferences. Thirdly, the user's contextual preference is captured during the interactions and used to update the model to refine the recommendation list. The paired samples t-test results show that there is a significantly different performance ($p <$ 0.005) between I-CARS and both competing methods (eALS and TCR) for all datasets. TCR by actively selecting the best question to ask in each interaction can steadily increase the performance. Similarly, eALS has the advantage of converging at an early stage by incorporating item popularity on its models. However, both TCR and eALS lose the advantage of learning personal user context that is rich in these datasets. The results prove the superiority of our I-CARS over other competitors.

3) Comparing Different Question Selection Strategies: We compare our question selection strategies ($MaxWeight$ and $MaxIUWF$) with two baselines ($Random$ and $MaxRating$). We compare their efficiency in learning the user's preferences based on the question they select in each interaction. HR is used to check whether the user responds positively/negatively to the question. Figure 3 depicts the comparison result of our question selection strategies with the baselines in term of HR. The better the question selection strategy, the faster it is learning the user's preferences.

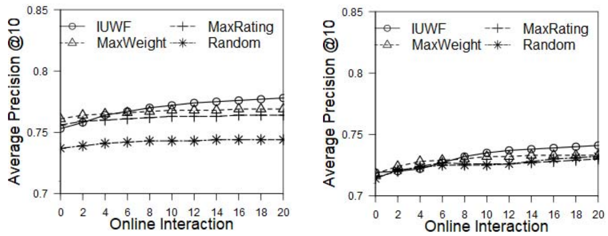
Clearly, our proposed question selection strategies ($MaxWeight$ and $MaxIUWF$) show better performance compared to the baselines. Paired samples t-test amongst the strategies also shows that there is a significant difference (with $p <$ 0.005) between the ($MaxWeight$ and $MaxIUWF$) with the ($MaxRating$ and $Random$) for both datasets. $MaxWeight$ also shows significant differences compared to $Random$ but performs quite similar with $MaxRating$. In addition, $MaxIUWF$ differs significantly to $MaxWeight$, where $MaxIUWF$ is better than $MaxWeight$. Even $MaxWeight$ that only relies on contextual weight can slightly outperform $MaxRating$ strategy that only relies on the highest predicted rating. We believe, both $MaxRating$ and $Random$ can steadily increase its performance due to the initial model, PW-CAMF, which has considered personalized contextual information for each user. Based on this result, we use $MaxIUWF$ as the question selection strategy for I-CARS due to its efficiency in learning user's preferences.

4) Effect of Different Question Selection Strategies in I-CARS Learning: I-CARS continuously learns the user's preferences by asking question to the user and incorporates the feedback to the model. In this experiment, we test the influence



(a) $STS\ Travel$ (b) $InCarMusic$

Fig. 3: Comparison of different question selection strategies



(a) $STS\ Travel$ (b) $InCarMusic$

Fig. 4: Effect of iteratively incorporating user feedback

of using different question selection strategies in I-CARS in term of returning the most relevant items at the top rank recommendation. We set 20 as the number of interactions. For each interaction, we recommend the top 10 items to the active user. Figure 4 compares the performance of I-CARS in term of $AP@10$ when using different question selection strategies.

Clearly, I-CARS using $MaxIUWF$ can gradually increase its performance in predicting most relevant items compare to I-CARS with other question selection strategies. For InCarMusic and STSTravel dataset, I-CARS using $MaxIUWF$ shows similar performance at the early phase of the interaction. However, later it overpasses the other alternatives and proves that it is better in learning the user's preferences after incorporating user response. A paired samples t-test results show that there is a significant difference ($p < 0.005$) between $MaxIUWF$ and $MaxWeight$ with all other strategies for all datasets, except for comparison of our alternative approach $MaxWeight$ to $MaxRating$ for InCarMusic dataset that shows a quite similar performance ($p < 0.07$). Our insight is $MaxWeight$, which heavily depends on the presence of the contexts, does not have enough information to learn the preferences from only 8 available contexts in InCarMusic. It can be seen that by incorporating personalized user context and user/item factors as in $MaxIUWF$, I-CARS is able to refine the user preferences correctly. By considering this result, we use I-CARS with $MaxIUWF$ in comparison with existing interactive/conversational recommender systems.

V. Conclusions

In this paper, we propose an interactive context-aware recommender system that iteratively learns user preferences through feedback that they provide during the interaction with the system. First, we propose a personalized contextual recommender system that is used as a learning method. Then, on top of this method, we support the interactions of users to the system through feedback by proposing two question selection strategies and incorporate the feedback into the model. Finally, we evaluate the effectiveness of the method in various metrics before and after each interaction. The experimental results have proved that our proposed approach outperforms the state-of-the-art methods in terms of effectiveness.

VI. Acknowledgment

The research presented in this paper has been supported by the Indonesia Endowment Fund for Education (LPDP).

References

[1] H.-H. Chen, "Weighted-svd: Matrix factorization with weights on the latent factors," *arXiv:1710.00482*, 2017.

[2] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," ser. KDD, 2016, pp. 815–824.

[3] X. Zhou, D. Qin, L. Chen, and Y. Zhang, "Real-time context-aware social media recommendation," *The VLDB Journal*, 2018.

[4] R. Lumbantoruan, X. Zhou, Y. Ren, and Z. Bao, "D-cars: A declarative context-aware recommender system," in *ICDM*, 2018, pp. 1152–1157.

[5] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context-aware recommendation," ser. RecSys, 2011, pp. 301–304.

[6] Y. Ren, M. Tomko, F. D. Salim, J. Chan, C. L. Clarke, and M. Sanderson, "A location-query-browse graph for contextual recommendation," *TKDE*, vol. 30, no. 2, pp. 204–218, 2018.

[7] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *SIGIR*, 2014, pp. 83–92.

[8] X. Zhou, L. Chen, Y. Zhang, L. Cao, G. Huang, and C. Wang, "Online video recommendation in sharing community," in *SIGMOD*, 2015, pp. 1645–1656.

[9] X. Zhou, L. Chen, Y. Zhang, D. Qin, L. Cao, G. Huang, and C. Wang, "Enhancing online video recommendation using social user interactions," *The VLDB Journal*, pp. 637–656, 2017.

[10] X. Wang, S. C. Hoi, C. Liu, and M. Ester, "Interactive social recommendation," ser. CIKM '17, 2017, pp. 357–366.

[11] K. McCarthy, Y. Salem, and B. Smyth, "Experience-based critiquing: Reusing critiquing experiences to improve conversational recommendation," ser. ICCBR, 2010, pp. 480–494.

[12] Y. Salem, J. Hong, and W. Liu, "History-guided conversational recommendation," ser. WWW, 2014, pp. 999–1004.

[13] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," in *NIPS*, 2011, pp. 2249–2257.

[14] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *JMLR*, vol. 3, pp. 397–422, 2003.

[15] M. Elahi, M. Braunhofer, F. Ricci, and M. Tkalcic, "Personality-based active learning for collaborative filtering recommender systems," in *AI*IA*, 2013, pp. 360–371.

[16] M. Braunhofer, M. Elahi, F. Ricci, and T. Schievenin, "Context-aware points of interest suggestion with dynamic weather data management," 2013, pp. 87–100.

[17] L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger, "Incarmusic: Context-aware music recommendations in a car," in *EC-Web*, 2011, pp. 89–100.

[18] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *TOIS*, vol. 22, no. 1, pp. 143–177, 2004.

[19] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," ser. KDD, 2008, pp. 426–434.

[20] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," ser. WWW, 2015, pp. 111–112.

[21] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," ser. SIGIR, 2016, pp. 549–558.

[22] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems," in *UMAP Workshops*, 2015.

[23] E. Christakopoulou and G. Karypis, "Local item-item models for top-n recommendation." ACM, pp. 67–74.

[24] H. LIa, "A short introduction to learning to rank," 2011.

[25] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in *ICCV WKSH*, 2015, pp. 142–150.