CONVERSATIONAL RECOMMENDER SYSTEM IN BANKING USING FAST MATRIX FACTORIZATION

Yonathan Lesmana* and Abba Suganda Girsang

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
agirsang@binus.edu
*Corresponding author: yonathan.lesmana@binus.ac.id

Received September 2021; accepted December 2021

Abstract. *A recommender system with good accuracy can be a competitive advantage for a bank, especially in the digital banking era. This study combines a conversational-based approach with the matrix factorization method, namely Fast eALS which can learn from implicit data effectively. The solution proposed in this study is a conversational recommender system that can interact with users and capture user preferences that change over time. The Fast eALS has proven to be able to perform incremental updates quickly to update recommendations online. The recommendation performance is evaluated by measuring HR and NDCG metrics on top-N recommendations. The measurement results show that Fast eALS has better accuracy when compared to the popularity-based algorithm. With Fast eALS, the incremental update process can be carried out quickly even large embedding dimension, thus demonstrating good scalability.*

Keywords: Conversational recommender system, Matrix factorization, Fast eALS, Banking

1. **Introduction.** The banking industry in Indonesia is still growing, even the Indonesian banking industry is seen as one of the best in the world today. However, the large number of players in the banking and financial services industry in Indonesia requires banks to seek effective marketing and sales strategies to win the competition. Banks have millions of lines of data and information regarding transaction patterns and customer profiles that are reliable but usually not optimally utilized. If this can be achieved, the bank can provide tangible benefits to its customers and prospective customers.

Generally, every bank has and offers a wide variety of products and services for retail customers which include savings, time deposits, loans, credit cards, mutual funds, bancassurance, and much more. This results in customers being flooded with too many choices (over-choice). Recommender Systems (RS) are tools and techniques used to provide recommendations to users through the effective use of available information [1]. Recommender systems are useful for both users and service providers because they reduce the effort needed to find and select items from the many options available [2].

This study focuses on the use of recommender systems to produce recommendations for retail banking products and services using the conversational approach. The solution proposed in this study is a personalized recommendation model that utilizes offline user historical data processing and simulates conversations to explore user preferences online. Using a conversational-based approach naturally eliminates sparsity and cold-start problems which are usually a problem in collaborative filtering. In addition, the conversational-based approach can capture user preferences that change over time.

DOI: 10.24507/icicel.16.10.1119

This paper is the first to implement an interactive conversational recommender system in the banking domain. As part of the experiment, this work built a prototype that illustrates how users and the system interact. We provide prior works that motivated this study in Section 2. This paper outlines the proposed system and methodology in Section 3 and evaluates it in Section 4. Finally, conclusion and future work are summarized in Section 5.

2. Related Works.

Research on recommender systems in the banking domain has lagged behind the progress of recommender system research in other domains. Most of the research on recommender systems in banking is limited to exploring the use of a hybrid approach and collaborative filtering [3-11]. Research on recommender systems in the banking domain is difficult due to its unique characteristics: 1) no explicit feedback provided by users, and 2) cold-start problem [4]. Financial products such as banking are included in the category of items with high complexity, and have a long-term financial impact with high monetary costs [1].

One of the existing techniques used to address cold-start problems is an interview process which is used for preference elicitation and forming a user profile and then making recommendations based on that profile [12]. Conversational-based approach helps users find items interactively by asking users preferences, thus eliminating the problem of sparsity and cold-start problems. Furthermore, the conversational-based approach is able to capture user preferences that change over time. The conversational-based approach was adopted from a knowledge-based recommender system which has long been considered suitable for products with high complexity such as financial services products.

Early research [12-16] argues that it is more effective to combine the matrix factorization model and conversational-based models. In Conversational RS, it is important to learn effectively from as few questions as possible with the user. One way to achieve this is to initialize the learning model using the initial embedding data learned offline so that the system can learn new user preferences more quickly [14].

Recent research on Matrix Factorization (MF) has begun to focus on implicit feedback. Implicit feedback has a one-class problem due to the absence of negative feedback, which is usually tried to solve by modelling the missing data as negative feedback as used in element-wise Alternating Least Square (eALS). However, this method reduces the efficiency of the implicit MF learning process so that it is impossible to generate recommendations online with dynamic data. Fast eALS learning algorithm [17] was designed for whole-data based learning with non-uniform weighting based on item popularity. The new Fast eALS learning algorithm produces better prediction accuracy at a faster time than Alternating Least Square (ALS), Randomized block Coordinate Descent (RCD) and Bayesian Personalized Ranking (BPR) [17,18]. This paper will explore how the new Fast eALS works on a conversational-based model.

3. Proposed System and Methodology.

The use of Collaborative Filtering (CF) has long been known to have advantages in terms of accuracy, simplicity, justifiability, efficiency and stability but has problems such as sparsity and cold-start problems. There is also a fundamental weakness in CF where user preferences are inferred based solely on the user's behavior history, assuming the user has a fixed or static preference. In fact, user preferences can change over time both by the influence of internal and external factors. Research on recommender system with a conversational approach has emphasized the importance of interactivity in recommender system so that users have a more active role in recommendations. Conversational RS continuously adapts the predicted recommendations throughout the user's interaction with the system using the latest feedback from users. This feedback loop is expected to help the system infer changing user preferences more accurately and increase the model's level of confidence in the recommendations generated.

The conversational-based approach is considered suitable to be applied to recommendations in the banking domain for several reasons. The first reason is the cold-start and sparsity conditions that are common in banking. Second, banking products are not among the items acquired by the customer in a short period of time. The third reason relates to the nature of the banking product itself which has high complexity and high value [1], in this case the long-term financial impact appears.

Conversational RS proposed in this study uses a centralized architectural approach to provide an omnichannel experience, where personalized recommendations are displayed to users through multiple channels (multi-channel). With a centralized approach, there is only a single source of truth for user recommendations across all channels. The proposed Conversational RS is divided into two components: Conversational Agent and Online Recommender.

3.1. **Conversational Agent.** Conversational Agent is a presentation layer that handles user interaction with the system. The Conversational Agent is in charge of managing the presentation of recommendations and questions to users and also receiving feedback. Recommendations, questions and feedback take many forms, especially in multi-channel environments. With so many channels, each of which has different characteristics and users, Conversational Agents must be able to display recommendations and questions and then capture feedback in various formats according to the characteristics of each channel served.

Previous research has shown that there are many ways to ask questions to collect feedback from users [14,15,19]. In the scope of this research, Conversational Agent collects feedback from users by asking absolute questions as a mechanism to explore preferences.

3.2. **Online Recommender.** Online Recommender is a centralized recommendation modeling based on matrix factorization using user-item interaction data. The solution proposed the Fast eALS algorithm. Online Recommenders are trained using historical data (Offline Initialization) and adapt by updating model parameters based on new user-item interactions (Online Updating) forming a continuous recommendation-feedback loop, as shown in Figure 1.

The Offline Initialization module builds an initial prediction using historical data. Users who access the system are identified and the Online Recommender provides an initial ranked list for that user to the Conversational Agent. Conversational Agent will display

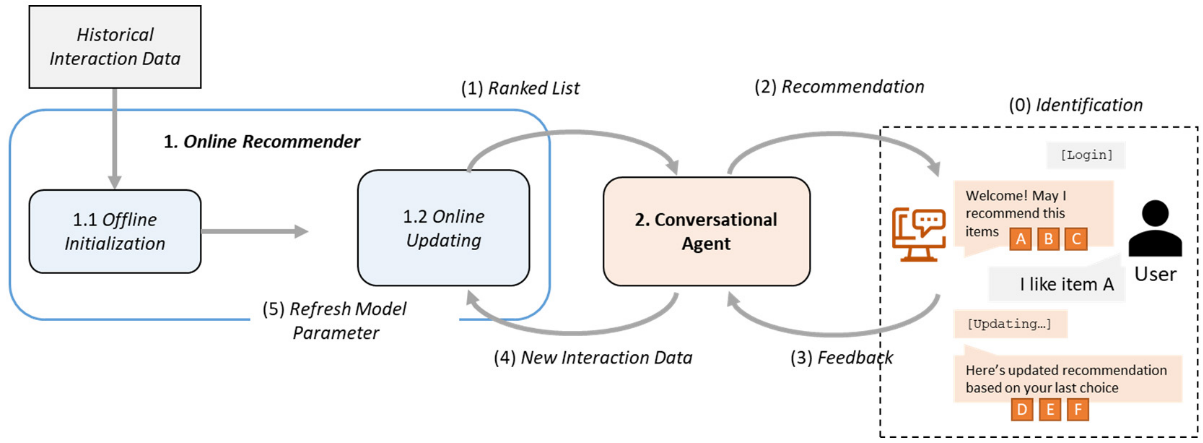


FIGURE 1. Process flow on proposed Conversational RS

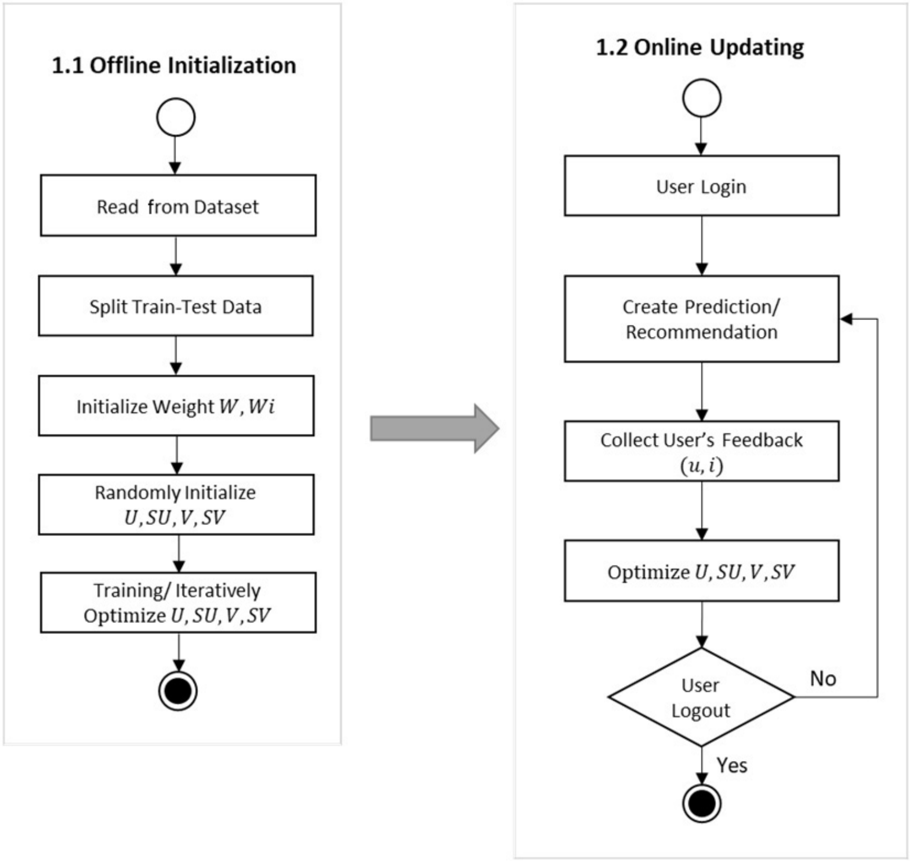

1.1 Offline Initialization
Read from Dataset
Split Train-Test Data
Initialize Weight W, Wi
Randomly Initialize U, SU, V, SV
Training/ Iteratively Optimize U, SU, V, SV
1.2 Online Updating
User Login
Create Prediction/ Recommendation
Collect User's Feedback (u, i)
Optimize U, SU, V, SV
User Logout
No
Yes

Figure 2. Key processes of Offline Initialization and Online Updating

recommendations and questions based on interaction models and techniques that are appropriate to the channel used by the user. Users provide feedback on the system and that feedback is then translated into new interaction data which is then sent to the Online Updating module and used to update and optimize model parameters, and so on to form a loop. The stages of the process carried out in the Offline Initialization and Online Updating are depicted in Figure 2. In this study, Offline Initialization uses 100 iterations, which is enough for Fast eALS to converge based on previous research [17].

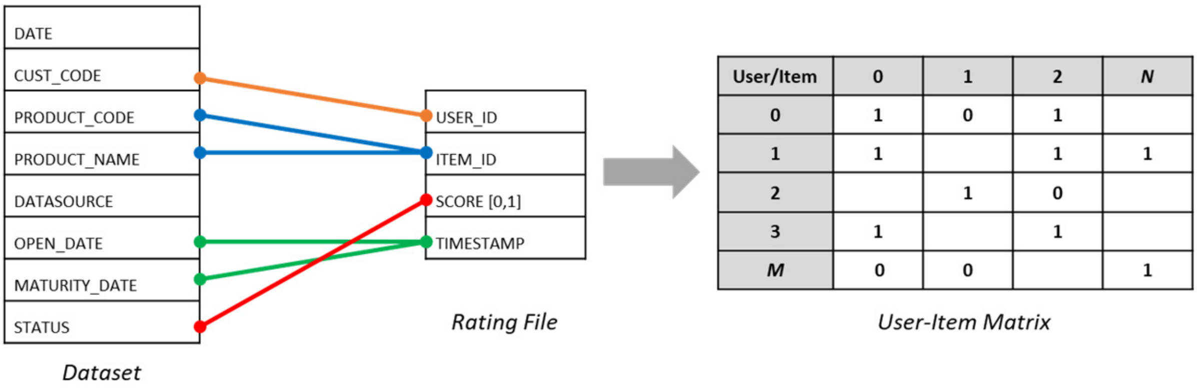
Fast eALS can learn model parameters efficiently and use an incremental update strategy so that it can perform the real-time online learning inside the Online Updating module. If U and V are model parameters learned from offline training and (u, i) are new interactions, Fast eALS performs optimization process for p_u and q_i only. This is based on the assumption that this new interaction should not change U and V too much from a global perspective but change the local features for u and i significantly. Based on empirical studies [17], one iteration of the online update process is sufficient to get good results. Since new interactions better reflect recent user interests, these new data are usually given a higher weight $w_{new} = 1$ [17,18].

In implicit data conditions, the value $r_{ui} = 0$ indicates that there is no interaction or unobserved, which does not necessarily mean that u does not like i (one-class problem). It can have two meanings: 1) the user does not know that the item exists, or 2) the user really does not like the item. So, the one-class problem is formulated as a problem to estimate the value of unobserved entries in R which is used to rank items. The strategies commonly used are sample-based learning [20] and whole-data-based learning [21]. A previous study [17,18] used whole-data based learning by developing an algorithm based on Fast eALS to solve the inefficiency problem. The Fast eALS algorithm has been adapted for implicit data conditions by using variable weighting on missing entries based on item popularity

[17,18]. $\propto$ is used to determine the significance level of popular items for unpopular items. Popular items are more likely to be known by the general user. Therefore, we have a higher belief that the lack of interaction on popular items is due to the user not liking the item or the item is irrelevant to the user.

3.3. **Dataset.** The biggest challenge in conducting research in the financial services industry is due to concerns about data confidentiality and the unavailability of public datasets. This study uses real customer portfolio data from one of the major banks in Indonesia that contains information about product ownership by customers. We translate the data into anonymous datasets by replacing sensitive information. The dataset consists of data on funding and lending products from the retail banking segment and history of product ownership by customers. Due to the large number of data and the high sparsity in the dataset received from the bank, we follow the general practice by filtering data that has less than five interactions.

For the model to process implicit data to predict user preferences from product ownership history data, it is necessary to carry out a translation process to convert the status to a binary value $[0, 1]$. Then the product ownership data by the customer is converted into a user-item interaction matrix $R \in \mathbb{R}^{M \times N}$ with the user in the row and the item in the column, as shown in Figure 3. If the customer does not have a particular product, then that entry is defined as a missing entry.



User/Item	0	1	2	*N*
0	1	0	1	
1	1		1	1
2		1	0	
3	1		1	
M	0	0		1

FIGURE 3. Forming a user-item interaction matrix from the dataset

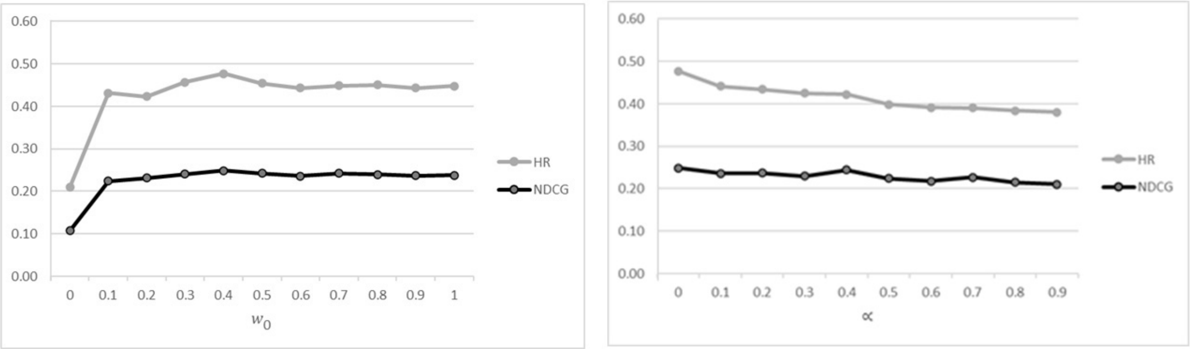
3.4. **Evaluation.** The evaluation in this study uses offline experiments with previously collected datasets (ground-truth), meaning that there is no interaction with users. Performance is measured by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) in top-10 items recommendation with two evaluation protocols: offline protocol and online protocol.

The offline protocol only uses the Offline Initialization module. This protocol evaluates the algorithm's ability to generate one-shot recommendations for users based on static historical data, without steaming new interaction. This approach does not describe the real scenario of a conversational recommender system and is used merely to determine the optimal model parameters. For this offline protocol, the datasets are separated using a leave-one-out, where every last interaction of each user is stored to evaluate the predicted results of the model (test ratings) while the remaining data is used for model training (train matrix).

The online protocol simulates user interaction with the conversational recommender system using both the Offline Initialization module and the Online Updating module. System is not only initialized offline using the train matrix, but also updated online with new interactions from test ratings. This online protocol evaluates the algorithm's ability to accept new data and perform incremental updates. For online protocols, the data are sorted chronologically and then separated 90-10, where the first 90% of interactions become training data (train matrix) and the last 10% becomes test data (test ratings).

All variables that are not tested will be fixed. For regularization, this study sets $\lambda = 0.01$ which is usually used as the default value for L2 regularization. Based on previous research [17,18,22], the weights to be observed are set uniformly as 1.

4. **Result and Discussion.** Evaluation with offline protocol is used to determine the optimal model parameters. The first series of tests is carried out to determine the value of w_0 and $\propto$ with $K = 64$. The results are shown in Figure 4. For the dataset used in this study, $w_0 = 0$ resulted in the worst HR and NDCG values significantly. The best results were achieved at $w_0 = 0.4$ although the difference was not too significant. In Figure 4(b), HR and NDCG values continue to decrease as the value of $\propto$ increases. At the value of $\propto = 0.1$ the HR value decreased by 8.12% and NDCG decreased by 5.33% when compared to $\propto = 0$. This shows that the dataset used is more suitable by using uniform weighting, compared to variable weighting based on item popularity.



(a)

(b)

FIGURE 4. HR and NDCG (a) vs. w_0 with $\propto = 0$; (b) vs. $\propto$ with $w_0 = 0.4$

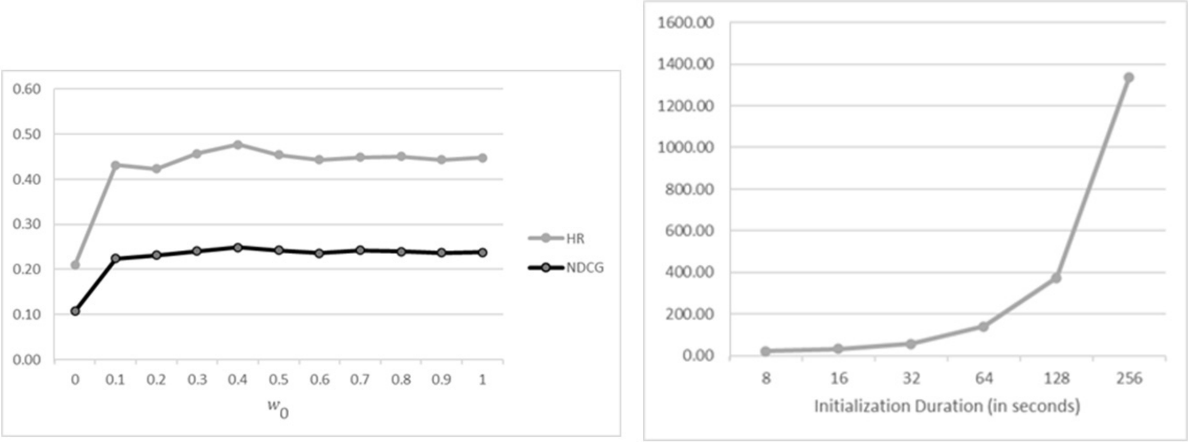
Using $K = 64$, $w_0 = 0.4$ and $\propto = 0$, the next experiment measured the accuracy of Fast eALS in various N on the Top-N recommendation. Due to the unavailability of benchmarks especially in banking domain, this study compares Fast eALS algorithm with simple popularity-based algorithm on various number of recommendation items (N) and displays the results in Table 1. Fast eALS consistently produces better NDCG when compared to popularity-based algorithms. The biggest difference is in the Top-10 recommendation where Fast eALS produces 73.44% higher NDCG than popularity-based algorithms.

The size of the latent factor or embedding dimension K can also affect accuracy, with trade-offs in computational complexity. A larger K will increase the accuracy of the model,

TABLE 1. HR and NDCG from Fast eALS vs. Popularity-Based on various Top-N values

N	HR			NDCG		
	Fast eALS	**Pop**	$\frac{Fast\ eALS-Pop}{Pop} \times 100\%$	**Fast eALS**	**Pop**	$\frac{Fast\ eALS-Pop}{Pop} \times 100\%$
10	0.4769	0.3299	44.56%	0.2491	0.1436	73.47%
25	0.6125	0.6433	−4.79%	0.2812	0.2206	27.47%
50	0.6782	0.8357	−18.85%	0.2950	0.2572	14.70%
100	0.7735	0.9162	−15.58%	0.3142	0.2702	16.28%

especially on large data, but will increase the processing time. Figure 5 shows the change in HR and NDCG values and the increase in initialization time for various K values. The significant increase in HR and NDCG occurred at $K \leq 32$. On the other hand, the time required for the offline initialization process increases exponentially as the K value increases. Therefore, this study uses $K = 64$ which is considered large enough to maintain accuracy with acceptable processing times.



(a) (b)

Figure 5. (a) HR and NDCG; (b) training duration at various K

In the evaluation with the online protocol, measurements were made to find out how the impact of the weighting of the new interactions on the Online Updating algorithm. Figure 6 shows that the best result achieved on $w_{new} = 1$.

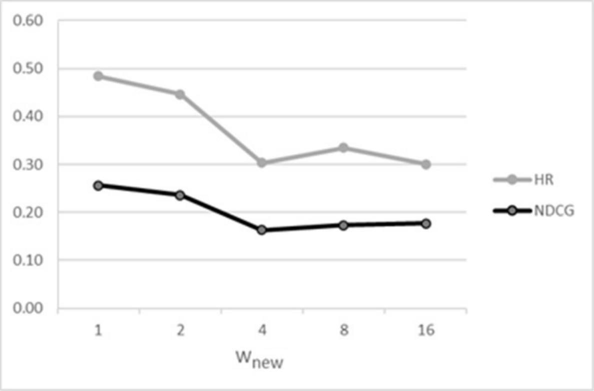
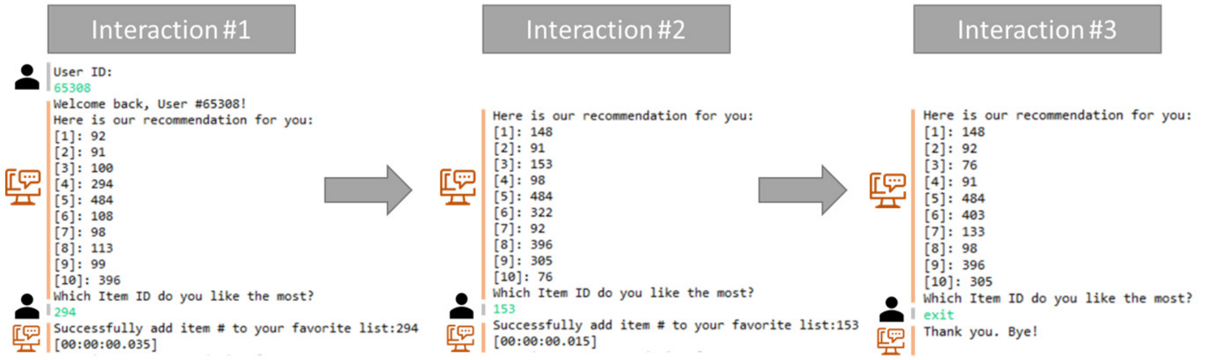


Figure 6. HR and NDCG vs. w_{new}

The best parameter settings from the previous tests are applied as parameters to the prototype. In this prototype, we can simulate who the user is currently accessing the system by entering the User ID on the input screen. In real world cases, this process is generally replaced with a user logged into the system where the user identification process occurs together with user authentication and authorization. The system then manages the user session as part of the Conversational Agent. In the prototype made in this study, the Conversational Agent module is set to display recommendations for top-10 items and allows users to freely choose the preferred item. Every choice made by the user on the system will be processed to update the model parameters and directly change the next recommendation for that user. Figure 7 shows that the system can dynamically update the recommendations that are displayed to the user at each subsequent interaction. Measured



```
User ID:
65308
Welcome back, User #65308!
Here is our recommendation for you:
[1]: 92
[2]: 91
[3]: 100
[4]: 294
[5]: 484
[6]: 108
[7]: 98
[8]: 113
[9]: 99
[10]: 396
Which Item ID do you like the most?
294
Successfully add item # to your favorite list:294
[00:00:00.035]
```

```
Here is our recommendation for you:
[1]: 148
[2]: 91
[3]: 153
[4]: 98
[5]: 484
[6]: 322
[7]: 92
[8]: 396
[9]: 305
[10]: 76
Which Item ID do you like the most?
153
Successfully add item # to your favorite list:153
[00:00:00.015]
```

```
Here is our recommendation for you:
[1]: 148
[2]: 92
[3]: 76
[4]: 91
[5]: 484
[6]: 403
[7]: 133
[8]: 98
[9]: 396
[10]: 305
Which Item ID do you like the most?
exit
Thank you. Bye!
```

FIGURE 7. An example of the system's ability to update recommendations based on feedback from user

using a notebook (Intel Core i7 10750H CPU and 16 GB RAM), the incremental update process takes less than 40 ms to digest the new interaction data given.

5. **Conclusion and Future Works.** An interactive recommender system must be able to run the process of updating model parameters online in a short time. With the Fast eALS algorithm, this incremental update process can be carried out quickly even for the size of the embedding dimension which is large enough so that this algorithm is suitable for use in conversational recommender systems with large data. From the measurements that have been carried out, it was found that Fast eALS has better accuracy than the popularity-based algorithm in the top-N recommendation. In future research, this Fast eALS algorithm can be further developed to capture negative interactions so that they can better reflect real world scenarios where users are not interested in the recommended product.

REFERENCES

[1] F. Ricci, L. Rokach and B. Shapira, *Recommender Systems Handbook*, Springer US, Boston, MA, DOI: 10.1007/978-1-4899-7637-6, 2015.

[2] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh, Recommendation systems: Principles, methods and evaluation, *Egyptian Informatics Journal*, vol.16, no.3, pp.261-273, DOI: 10.1016/j.eij.2015.06.005, 2015.

[3] L. Aerts, A. Claes, D. de Leeuw and E. de Leeuw, Implicit hybrid recommender system for informative articles, *Seminar Case Studies in Business Analytics and Quantitative Marketing (FEM21001)*, pp.1-52, 2019.

[4] O. Oyebode and R. Orji, A hybrid recommender system for product sales in a banking environment, *Journal of Banking and Financial Technology*, vol.4, no.1, pp.15-25, DOI: 10.1007/s42786-019-0001 4-w, 2020.

[5] H. Abdollahpouri and A. Abdollahpouri, An approach for personalization of banking services in multi-channel environment using memory-based collaborative filtering, *2013 5th Conference on Information and Knowledge Technology (IKT2013)*, pp.208-213, DOI: 110.1109/IKT.2013.6620066, 2013.

[6] A. Gigli, F. Lillo and D. Regoli, Recommender systems for banking and financial services, *RecSys Posters*, 2017.

[7] E. S. Gol, A. Ahmadi and A. Mohebi, Intelligent approach for attracting churning customers in banking industry based on collaborative filtering, *Journal of Industerial and Systems Engineering*, vol.9, no.4, pp.9-25, 2016.

[8] D. Gallego and G. Huecas, An empirical case of a context-aware mobile recommender system in a banking environment, *2012 3rd FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing*, pp.13-20, DOI: 10.1109/MUSIC.2012.11, 2012.

[9] D. V. Gallego and G. Huecas, Generating context-aware recommendations using banking data in a mobile recommender system, *The 6th International Conference on Digital Society Generating (ICDS2012)*, pp.73-78, 2012.

[10] A. Felfernig and A. Kiener, Knowledge-based interactive selling of financial services with FSAdvisor, *Proc. of the National Conference on Artificial Intelligence*, pp.1475-1482, 2005.

[11] A. Felfernig, K. Isak, K. Szabo and P. Zachar, The VITA financial services sales support environment, *Proc. of the National Conference on Artificial Intelligence*, pp.1692-1699, 2007.

[12] K. Zhou, S.-H. Yang and H. Zha, Functional matrix factorizations for cold-start recommendation, *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information (SIGIR'11)*, pp.315-324, DOI: 10.1145/2009916.2009961, 2011.

[13] X. Zhao, W. Zhang and J. Wang, Interactive collaborative filtering, *Proc. of the 22nd ACM International Conference on Information & Knowledge Management (CIKM'13)*, pp.1411-1420, DOI: 10.1145/2505515.2505690, 2013.

[14] K. Christakopoulou, F. Radlinski and K. Hofmann, Towards conversational recommender systems, *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.815-824, DOI: 10.1145/2939672.2939746, 2016.

[15] J. Zou, Y. Chen and E. Kanoulas, Towards question-based recommender systems, *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.881-890, DOI: 10.1145/3397271.3401180, 2020.

[16] A. M. A. Al-Sabaawi, H. Karacan and Y. E. Yenice, SVD++ and clustering approaches to alleviating the cold-start problem for recommendation systems, *International Journal of Innovative Computing, Information and Control*, vol.17, no.2, pp.383-396, DOI: 10.24507/ijicic.17.02.383, 2021.

[17] X. He, H. Zhang, M.-Y. Kan and T.-S. Chua, Fast matrix factorization for online recommendation with implicit feedback, *Proc. of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.549-558, DOI: 10.1145/2911451.2911489, 2016.

[18] X. He, J. Tang, X. Du, R. Hong, T. Ren and T.-S. Chua, Fast matrix factorization with nonuniform weights on missing data, *IEEE Trans. Neural Networks and Learning Systems*, vol.31, no.8, pp.2791-2804, DOI: 10.1109/TNNLS.2018.2890117, 2020.

[19] Y. Jin, W. Cai, L. Chen, N. N. Htun and K. Verbert, MusicBot: Evaluating critiquing-based music recommenders with conversational interaction, *Proc. of the 28th ACM International Conference on Information and Knowledge Management*, pp.951-960, DOI: 10.1145/3357384.3357923, 2019.

[20] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz and Q. Yang, One-class collaborative filtering, *2008 8th IEEE International Conference on Data Mining*, pp.502-511, DOI: 10.1109/ICDM.2008.16, 2008.

[21] H. Steck, Training and testing of recommender systems on data missing not at random, *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, DOI: 10.1145/1835804.1835895, 2010.

[22] R. Devooght, N. Kourtellis and A. Mantrach, Dynamic matrix factorization with priors on unknown values, *Proc. of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.189-198, DOI: 10.1145/2783258.2783346, 2015.