JKAU: Comp. IT. Sci., **Vol. 13** No. 1, pp: 46 – 67 (2024 A.D.)
DOI: 10.4197/Comp.13-1.3

Context-Aware Recommendation System Using Matrix Factorization

Dimah Alahamdi[1] and Masarra Talat Aldramley[1]
[1]*Department of Information systems, King Abdulaziz University*
Jeddah, Saudi Arabia dalahmadi@kau.edu.sa, meldremele@stu.kau.edu.sa

Abstract. In a commercial field, millions of new items would be added to the daily sales field. Suggesting proposed items to users for purchasing process is a critical point. Finding the best suggestions based on the user needs and behavior increase the sales productivity. Incorporating context information in recommendations process have been accompanied by many domains and applications. Different methods and strategies have been used to find recommendations. While time is an important factor for continuously updates and changes in the user preferences, incorporating it has been proved its effectiveness to enhance recommending performance. Time-aware recommender systems (TARS) has been used in a wide range of recommendation modeling. In this proposed paper, we focus to deal with three different context-aware algorithms. First, traditional matrix- factorization using explicit ratings. Second, enhanced version after dealing with time-target as basic factors for getting the results. Last, depend on the previous version, we enhance it by shrinking weights using the mathematical decay-function algorithm to improve prediction accuracy. we build our solutions and implement them using a real dataset of commercial website as our empirical case study. From our analysis and experiment, we finally evaluate the proposed model using different metrics on measuring relative performance of enhanced TARS over traditional matrix factorization.
Keywords—Context-aware recommender systems, Time-aware recommender systems, Evaluation metrics, Matrix-Factorization algorithm

I. INTRODUCTION

location, time, device type, mood, etc.) [1]. It is the newer generation of traditional RS by **D**aily, most people make their own decision based on recom- mendations found on social posts or through conversations with friends. Recommender systems (RS) are software tools that pro- vide the users with items suggestions of different types based on their needs such as (e.g., movies, travel trips, songs, music, or books). The main goal of these systems is to choose the suit- able choices to individuals from huge number of options. In last years, these systems help e-commerce companies achieve higher selling income by tracking their customers' purchasing behav- iors. There are different techniques of recommendation systems. Some of them are traditional types, which are exploiting the context with respect of users' changes. In an example of recommending a movie, the

working based on the concept of collaborative filtering and content-based filtering. These techniques depend on drawing the possibilities based on the similarities between item and user profile 'Content-Based fil- tering'. Alternatively, the interactions between users and items 'Collaborative-Based filtering' to have a list of possible recom- mendations. However, the world is wider than simple interactions; it is also about contextual information.

Context-aware RS (CARS) is a recommender system that tracks the differences in user preferences based on the context (tracking context could be 'Location', where the user would prefer to watch a movie at the cinema most likely than at home. The concept of using context information improve the value of the relevance of recommendations in e-commerce domain and it supported by different researchers. Deploying specially the time in context- aware recommendation becomes very popular in the last recent years. In the coming sections, we explored many aspects about the core of the research. First, a background of recommendation systems is described. Then, related works of similar researches are presented. After that, we focus to describe the proposed algorithms and dataset preparations. As last, a brief summary of results findings and future works are written.

II. Background and Related Work

A overview

These (RSs) systems have been available for many different domains, such as e-commerce, engineering, and education. Many popular websites such as Facebook, Amazon, YouTube, Linked In,

Netflix, IMDB have incorporated these systems into their business engines to supply their users with the best possible suggestions based on their preferences and profiles. All websites utilize these recommendation frameworks to achieve their aims and objectives, like obtaining user satisfaction and selling more items.

There are two approaches for recommendation systems; either traditional or contextual. The first approach is that **traditional one**. Many studies have been applied based on traditional approach. This approach is classified into two main algorithms; Content- Based (CB), Collaborative-Based Filtering (CF).

Content-Based Filtering: Content-based filtering is a tradi- tional approach that can be defined as methods or rules applied in the recommendation system where items are selected based on their description. The analysis of the description is performed by these methods, including the items which have been previously rated by the users [7]. There are many issues involved in this approach that only the user's current interests can be used as the basis for suggestions from the model. Stated differently, there is limited room for the model to expand upon the consumers' cur- rent interests. Content-based filtering provides limited novelty because it has to match a user's profile attributes with things that are available. In the case of item-based filtering, only item profiles are built, and consumers are given products based more on how similar they rank or search for than on their past history. There is nothing unexpected or shocking that a flawless content-based filtering system can uncover.

CF is another approach that depends on finding correlations between different recommendation system users. This method tries to suggest potential items that have been rated by other users due to shared interests [9]. Although the success of CF, there are some related problems which are below [9] [12]:

- **Cold-Start issue:** It is not easy to get accurate results if there are not enough users to match.
- **Sparsity:**The records in the database are spare for the user/rat- ing matrix.
- **Popularity bias:** It is hard to find a neighbor user if someone has unique preferences.

Contextual or Multi-dimensional Approaches: Before explain- ing the contextual recommender system, we need to understand the concept of "Context."

The context is any information determined as a factor for classi- fying entities [19]. We mean by an entity is any object in the world or user. The context is an activity and action between the user and the system. For that, determining which context is relevant for each case is a crucial process [15][20].

Context-Aware Recommender Systems: The reason behind using CARS is to find accurate predictions for user preferences by incorporate the traditional approaches with contextual information [23]. CARS are the systems where a recommendation engine's deployment is based on some pre-defined settings including current activities, location, or time interval needed for learning [10].

Time-Aware Recommendation Systems (TARS): Time-aware Recommendation System is a derived type of CARS. The main criteria of this type is the use of Time as a context metric in the prediction process [29]. Recommendations list may be changeable due to the different preferences from user to user depend on the desired time. Time context means 'The measurable time at specific time where an action or process is placed'. Several time formats such as hours, days, years, seasons, etc. have been recorded for developing recommendations. Time of the day; 12:00 am, 1:00 pm, etc., or day of the week; Monday, Tuesday, etc.. As an example, users who prefer to buy clothes in Spring season different from who prefer in Summer. Also, an order is processed at night make a different sense of order done at morning. Time consideration effect on the users' interest and this effect on them prefer actions [30][2].

There is a flexibility to deal with temporal context where does not need to link with other devices or extra efforts.

B Predictions Evaluation Metrics

1. Mean Absolute Error (MAE):

This metric determines the conflict between the value of pre- dicted rating and the actual ones [36] using the following formula:

$$MAE = 1/n \sum |y - \bar{y}| \quad (1)$$

2. Root Mean Squared Error (RMSE):

Second metric is similar to the previous one but there are some modifications in the mathematical formula [35]. This helps to find more accurate result when the error is high. The formula is written below:

$$RMSE = \sqrt{\sum (\bar{y} - y)^2 / N} \quad (2)$$

3. Precision Metric (or true positive rate):

Precision p at k items means an accuracy metric which mea- sures the percentage of relevant items k out of all retrieved recommended items [37][66]. Precision expresses the exact- ness. Example, the percentage of clothes that are really liked for user. If we suppose that the value of precision is 60% of top-20 recommendations that means, there is 60% of recom- mendation are actually relevant to the user. The formula of precision is described mathematically as below:

P = (no of relevant items at k) / (no of all retrieved items at k).

4. Recall Metric:

Recall r at k items means an accuracy metric which measures the percentage of relevant items k out of all retrieved relevant items [38][37]. Recall expresses the totality. Suppose that we have 60% recall value at top-20 items which means 60% of all retrieved relevant items are the top-20 items. The formula of recall is described mathematically as below:

R = (no of recommended that are relevant) / (no of all relevant retrieved items at k).

III. ALGORITHMS

Despite the CARS has been proved its positively impacts in many studies and domains, there is a scarcity in e-commerce do- main. There is a limitation in developing real-world experiments using purchasing activities. In this thesis we address the problem by proposing three model traditional MF, time-target aware MF, and decay time-aware MF and incorporated them in ecommerce dataset.

A Matrix Factorization

Matrix Factorization is a powerful tool used to solve database sparsity and scalability. Factorization techniques have become very popular in recent recommendation studies

[32] [33]. These algorithms are used to decompose the user-item matrix into two-dimensional matrices -the primary form composed of both user and item vectors in a rating pattern form [33]. One dimension is placed for a user vector, and the second dimension is for an item vector. A recommendation is inferred from the high ratings between the interaction of user and item. One advantage of using MF is to solve the missing data in a dataset where the explicit feedback is not available rich for all users. Matrix factorization uses implicit feedback to indirectly fill the matrix by observing the user's behavior, such as purchasing history, search history [5].

C Context-Aware Matrix Factorization (CAMF)

Traditional matrix factorization (MF) recommendation systems that disregard context take advantage of the rating matrix R as men- tioned in equation [3.6] to represent known ratings. The cardinality of the user set U and the item set I are two dimensions of the rating matrix.

D Exponential Decay Function

The user interests' changes align with the changes in time, for that the CF-technique based models does not provide an accurate ratings. Therefore, to overcome the problem, there is a need to incorporate a time-decay function with ratings process. This decay-time function numeric values to shrink the rate related to their value. It gives the proper higher weights to the recently added items.

E Incorporating with Half-life Calculations

Half-life simply is the time period required to decrease the importance of an item to half as shown in figure [7]. It is an ex- ponential decay function after editing function parameters which will be donated with the letter h which indicates how many days must pass for a particular rating to lose half of its initial relevance during the ratings process. The experiment's current time is de- noted by the symbol tn, whereas the day before the current day is represented by tn-1, and so on until we reach tn-h, which is the day after the half-life time has passed. At that point, the rating will be

$$\hat{r}_{ui} = p_u q_i^T$$

B Biased Matrix Factorization (Biased-MF)

(3)

reduced to the half of its original value.

Table I. Caption

Some bias terms are used to reduce the expected error between the predicted and the actual value [40]. There are three parameters to be determined for each user and item pair (u,i). μ which is the average ratings between

Actual Amount Is the actual preference of the all items, bi and this term is the average rating of one item i minus μ and bu that is the average rating given by a user u minus μ. The resulting equation is below:

$$r_u i = q_i^T p_u + \mu + b_i + b_u \qquad (4)$$

user of any item.

Predicted Amount Is the new reduced value that indicated the updated value after implementing decay-function. Elapsed time Is the difference in days be- tween the user-item interac- tion date and the total dataset

Algorithm 1 : Model Biased Matrix Factorization BMF 1: Input:
Initialize List of Q_u latent vector
Half-life value
Initialize List of P_i latent vector
Indicator I and regulation coefficient λ
μ the average ratings.
2:**While not** convergence do
3: Update Q according to Equation [] 4: Update P according to Equation [] time period.
Is the time required must consumed to get the half

5:**EndWhile**
6: Return Q and P
7: Calculate the model-bias B according to Equation [1.1] 8: Predict the list of items according to Equation [1.2]

value of the initial amount.

$$R = (r_u i)(u \in U, i \in I) \in R`(mn) m = |U| n = |I| \quad (5)$$

IV. DATASET

The quality of datasets plays a critical role in the result of any empirical research [51]. As the first step in our proposed system is to understand what datasets are intended to use. For that, we do efforts in processing our dataset. The dataset has been collected from one of an online real-world e-commerce studying materials store in Saudi Arabia, Prinlty . This website offers three studying materials (books, lecture slides, and test-banks) for universities students. A student can add materials to his Wishlist and then purchase them. Depending on their behaviors through the system, and the importance of each type of material, we need to recommend a list of materials for each customer (student in our case). These behaviors were selected over a period of time of three months (in our case one Studying Semester duration). The system contains about 30,000 records, where around 30% of them contain contextual information which we will focus on. This dataset is extracted from August 2022 to November 2022.

Table II. Caption

Dataset		#of users	#of interactions	#of items	Per user	Per item	#of purch
Printly		385	9697	4880	25.18	1.98	0.0052%

These processes allow us to prepare our data and prevent any expected errors [43]. Every learning system has its needed require- ments where data must be accurate to satisfy them. According to the report [25], published by Liu 2018, preparing data takes up 70% of a data scientist's time. Data profiling, cleaning, integra- tion, and transformation are the primary activities (or tasks) that make up the entire preparation process. This section introduces three phases on dataset preprocessing; data cleansing, Arabic text processing, and data transformation.

Figure 1 Data Preparation Model

A Data Cleaning

The first phase is about data cleansing. This level deals with the whole dataset. In other words, it is the process that is done in the data analysis phase. It is defined as the pro- cedure of identifying and correcting (or eliminating) any cor- rupt or inconsistent values of the dataset. In our experi- ment, we implements the data cleansing as four-stage process:

Figure 2 Data Cleaning Phase

1. Stage 1: 'Exclude Outdated Data values/ NULL Dates'

Researchers prefer to use datasets without missing data, or inconsistent values, however, this is not possible when data is collected using real-time applications [44]. If it is observed that several tuples lack of accurate values for a number of at- tributes, the missing values or outdated date can be corrected with random values or ignored. According to a survey of 85 prospective studies, 35% of researchers decided to leave out the missing data from their dataset. There are several meth- ods to treat with such situation, however we decided to apply the below: The used method is that 'Listwise deletion': it is called 'Complete-case deletion' a method of treating missing values. It eliminates all tuples for any case if it has one or more missing values. The dataset is proposed to treat the time period of only one academic semester which is Three-months. The duration between 28th of August to 24th of November is considered to be our legible case study. For that, we exclude any interaction done before or after that date or has NULL date value.

2. Stage 2: 'Fill NULL of Main Attribute with Mean value'

There is another method to deal with NULL values where the missing values are replaced with the sample mean of the data. This method is called Single Mean Imputation. In our case, explicit rating has a value range from 1 to 5 where 1 is the minimum and 5 is the maximum. As the Mean value is 3, we replaced the NULL values in this column to be filled with 3.

3. Stage 3: 'Handling Outlier Values'

We plotted the data values of the number of items per each university in the system. We have four universities which are (King Abdul-Aziz university, University of Jeddah, Univer- sity of Um-AlQuraa, Effat University). After we plotted the data, we noticed that 'Effat University' is the lower outlier where it has only 58 items conversely there are more than 3000 items per each of the other universities. For that, we decided to ignore the outlier and deal with only the three common universities. Below algorithm illustrates the steps of applying the method for detecting the outlier.

4. Stage 4: 'Removing Unwanted Values'

Uniqueness in our dataset is required to improve our data quality for that we handle duplication of two factors item university, and user item.

5. Stage 5: 'Handling with Duplicates Values'

In our system, we need to narrow down the number of records to focus on dealing with the actual procurement actions. Each placed order in the system has status called order status and is labeled with a numeric value to indicates its status. After we analyzed the workflow of each status, we noticed that the only statuses indicating the completion of purchasing process are 2 and 5. For that, we filtered the dataset depend on these statuses and removed any other not related record.

B Arabic Text Preprocessing

While we prepare our dataset for processing, we noticed that two needed entities, university and target type, which are filled manually by the editor using Arabic Language. Entering the data manually retrieved different text results and caused difficulties in analyzing them. The methods are text cleaning, text normalization, stop word removal, text stemming, and text mapping.

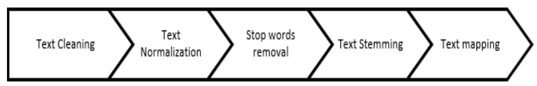

Figure 3 Arabic Text Preprocessing Method

Text Cleaning: In this step, URL links, special characters, white spaces, punctuation marks, non-Arabic letters, and numerals have to be removed.

Text Normalization: It is the step where reduces Arabic letters variations and diacritic marks and convert them to the base form. For example, ‘Alef-1’ in Arabic language is written in four different forms, Hamza top of letter ‘1’, Hamza below letter ‘1’, Madda top of letter ‘1’, or without Hamza ‘1’.

Stop words Removal: All stop words are eliminated in this step since they are less significant than the original terms. For example,

Arabic propositions such as ‘ <_,1 ; u_,t"", ‘u_"’, ‘ <_,1 ; u_ J1’, Arabic

pronouns such as ‘ L'·’1’, ‘u· >-"·’. Stop words list is

called in Arabic ‘

Table 2 Examples of Root Words

Term	Root	Term	Root
جامعه	ج م ع	تلخيصات	ل خ ص
		ملخصات	
		ملخص	
ملك	م ل ك	تجميعات	ج م ع
مالك		تجميع	
		تجاميع	

Text Mapping: The benefit of this step is to unify the fi- nal text label of each group of initial text terms. Based on the previous step and using a simple Python lookup method, we grouped the relative words and mapped them to the relevant En- glish term. Examples of text mapping showing in the table below:

Table 3 Examples of Text Mapping

Group Root	Mapped English Term
ملك عبد عزز	KAU
أمم قري	UOQ
جدد	UJ

V. PROPOSED SOLUTION

According to the key literature outlined in

[illegible]" which is found available in GitHub as an

open-source list contains around 750 words.

Text Stemming: It is a method of natural language processing, reduces word inflection to its root forms by removing affixes (or words branches). ISRI stemmer, proposed by Information Science Research Institute (ISRI), is an algorithm provides functions for rooting Arabic words [47]. Python built-in libraries provides ISRI stemmer which is included in Natural Language Toolkit

this paper, we pro- posed three layers of solutions that evaluate the use of MF with varying parameters in the domain of recommendation systems.

(NLTK) which is used in this research. ISRI tool output shown in the table.

They are biased MF with explicit ratings, CAMF, and Decay CAMF.

A Phase1: Biased MF Model

The first level is to apply the biased matrix-factorization to our dataset in its initial form using explicit user-item ratings in the system which is illustrated in below figure.

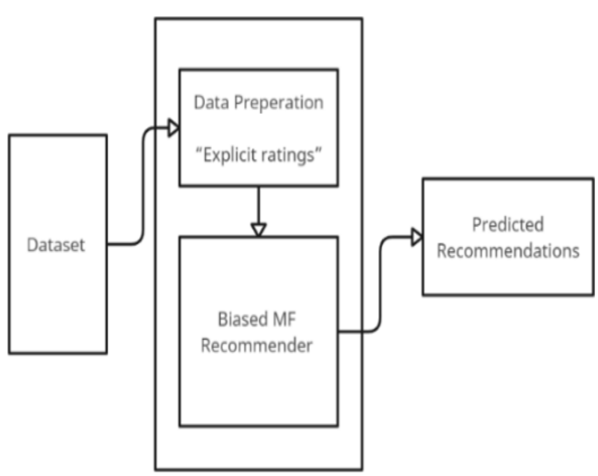

Figure 4 Biased MF model

Table 5 Data Distribution over the 3 Months

Month#	Week#				Description
1	1	2	3	4	First month of the semester
2	1	2	3	4	Second month of the semester
3	1	2	3	4	Last month of the semester

Biased MF model generates users' rating activity by filling up the user-item matrix using user-vector and item-vector and considering k- features which relate them. In our system, the website provides each user the ability to give explicit feedback by rating any item using a five-stars scale. This interaction between the user and the item is stored in the system as numerical values between 1 to 5. In this research, the value of 5 represents the maximum preferences that a user could interact with an item. While value 1 represents the least preferences. The model uses this information to fill up the matrix and then generate a list of recommended items.

B Phase2: Biased CAMF

In order to implement CAMF model for this system, we need to clarify the needed contexts and explain how to model the ratings (weight) of each user-item interaction de- pends on context scenario. The table [23] below shows in- formation about the contexts used in the proposed CAMF.

The proposed time period of our system is one academic semester which means Three months. Each target type has dif- ferent importance weight depend on the time of purchasing; first month, second month, or third month. The distribution of the three months is shown in table.

The order of the month or the week for each target category affects the importance (weight) of the context as below:

Table 4 Contextual Dimensions

Contextual Dimensions		
Context *c*	Contextual Information	#of conditions
Target Type	(Books, Lecture Slides, Test Bank)	3
Time of Purchase	Date of procurement process	Within 22 August to 27 November

1. Target Type 1: 'Books –'

For Students, books in general has higher importance at the beginning of any academic semester and it decreases when we close to the end. The distribution of impor- tance as numerical terms are shown in the table below [25]:

Table 6 The Distribution of Importance

Purchased Period	Week #	Importance (Weight)
Month [1]	1 to 4	5
Month [2]	1 to 4	4
During first three weeks of month [3]	1 to 3	3
During last week of month [3]	4	2
Not purchased but added Wish-list Months [1-2-3]	-	1
Other case Months [1-2-3]	-	0

We can notice that we give the importance number '3' during the three months where not all students buy lecture slides at the same time periods.

2. Target Type 2: 'Lecture Slides –

According to the previous, the second target type is "Lectures slides", which unlike the 'Books", the students need them throughout the whole semester with equal importance. For that, the numerical labels of this type are shown below

Table 7 Importance Distribution for Lecture Slides

Purchased Period	Week #	Importance (Weight)
Month [1]	1 to 4	3
Month [2]	1 to 4	3
During first three weeks of month [3]	1 to 3	3
During last week of month [3]	4	2
Not purchased but added Wish-list Months [1-2-3]	-	1
Other case Months [1-2-3]	-	0

We can notice that we give the importance number ‘3’ during the three months where not all students buy lecture slides at the same time periods.

3. Target Type 3: ‘Test Bank –

In contrast with the ‘Books’ type, the demand for getting a list of test banks of any study subject in- creases while we are moving closer to the final exam dates at the end of the semester. Based on that, we labeled the importance as shown in the table below:

Table 8 Importance Distribution for Test Banks

Purchased Period	Week #	Importance (Weight)
Month [1]	1 to 4	2
Month [2]	1 to 4	3
During first three weeks of month [3]	1 to 3	4
During last week of month [3]	4	5
Not purchased but added Wish-list Months [1-2-3]	-	1
Other case Months [1-2-3]	-	0

The model above helped to fill the importance (weight) of the observed records in our user-item matrix. After that, the matrix will be filled by CAMF engine in order to get the final recommendations list.

C Phase 3: Decay CAMF

Decay CAMF is an enhancement of CAMF model. The half-life decay function is inserted in the prediction learning process in order to update the rating of user-item matrix.

Table 9 Data Sample with Decay

User Id	Item Id	Importance	Decayed Importance
5117	639	5	-
1118	8760	?	-
2392	2779	5	-
4439	219	4	-
5911	3444	?	-
6956	5794	5	-

In our case, the dataset considers (90 days), and we assume half-life =45 days. That means 45 days must pass to give the user's rate half of its value. Additionally, importance of purchased items should have higher value for the recent more than the old ones. Importance of CAMF model is updated using the decay function. After that, the matrix will be filled by Decay CAMF engine in order to get the final recommendations list.

VI. IMPLEMENTATION AND PERFORMANCE EVALUATION

A Implementation Settings

In the previous section, we prepared our dataset for the three level MF model (BMF,

C CARSKIT Java Library

To model the three-level algorithms and retrieve recommen- dation list. In addition, to measure the evaluation results using precision, recall, and MAP metrics. Java version 1.7 or higher is required.

CARSKIT Library: CARSKIT is an open-

CAMF, decay CAMF). In this step, we uploaded three CSV file for each algorithm to be implemented and return a list of recommended items per each user. There are necessary implementation settings should be prepared. MF algorithm has three main parameters; number of factors, number of iterations, and regulation factor to avoid overfitting if the number of latent factors increased [49].

B Software Tools

The implementation of the data analysis and algorithms needed different software tools.

1. Python language (PyCharm Environment)

To preprocessing the dataset and implement the importance of CAMF.

source java-based library specialized in providing context-aware engine with all re- quired functions. All functions are available for user, modifica- tions, and distributions under the rules of GNU (General Public License) [50].

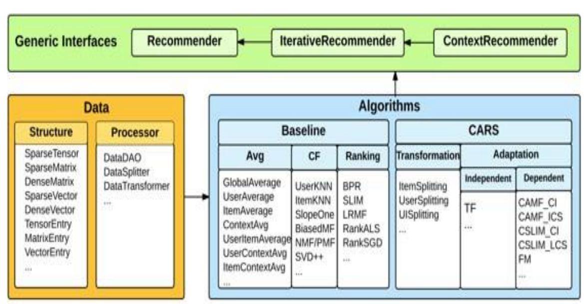

Figure 5 CARSKIT Design

Input of CARSKIT: We uploaded three dataset files one for each algorithm (BMF, CAMF, Decay CAMF) as CAV files.

Table 10 Dataset Files

BMF algorithm	Dataset with actual user ratings (Without contexts).
CAMF algorithm	Dataset with initial importance based on time-target context.
Decay CAMF algorithm	Dataset with modified importance using half-life decay function

Output of CARSKIT: The output of each iteration is a CSV file contains a list of recommendation items per user. Additionally, a performance summary of evaluation information measurement for every metric is provided. In the case of item-predictions, precision, recall, and MAP metrics are preferred to use.

D Evaluation

This part of the study expresses the results of comparing the accuracy between the three implemented algorithms. We imple- mented models ranging from the baseline predicator to the more accurate MF model with half-life decay function; Biased Ma- trix Factorization (BMF), Context Aware Recommendation Sys- tem (CAMF) in its biased form, and in its decay form (Decay CAMF).

To determine which parameter combination works best with all matrix factorization techniques, varies parameter combinations were explored. As mentioned in section [5.1], the number of iterations is set to the maximum = 100 and the number of latent factors = 10. In addition, we train the data four times with different

number of recommended items N=5, N=10, N=15, and N=20. As for the default learning regulation settings were tuned with γ will be set to 0.2, λ will be 0.0001, and α= 2 for all iterations. Also, μ=0 and σ =0.1 are set to generate a normal data distribution. These values seem to be the best for all cases depend on the studies [11] [18]. The user-items vectors q_i and p_u will be set their random values.

1. Precision: The precision results of implementing Top-N model using BMF, CAMF, and Decay CAMF are illustrated in Table [11] and Figure [5].

Table 11 Precision Results

	Precision of Top-N Recommendations			
Algorithm	Top-5	Top-10	Top-15	Top-20
BMF	0.4025	0.3950	0.4103	0.4108
CAMF	0.5302	0.5307	0.5554	0.5588
Decay CAMF	0.5704	0.5832	0.5885	0.5801

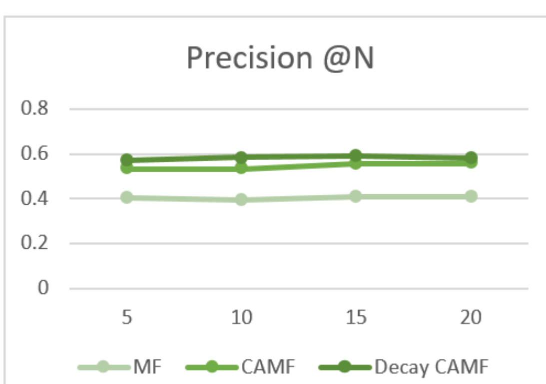

Figure 6 Precision Results

If we notice that as we increase in the number of required list of recommendation, the precision value increase as well. Where the performance in finding 10-20 results better than 5-10 results.
As shown in the above table [12], the average Precision results for three algorithms of BMF, CAMF, Decay CAMF models are 40.53%, 53.5%, and 58.55% respectively. The findings of the experiment show that Decay CAMF outperforms the performance better than others.

Table 12 Average Precision Results

	Average Precision @N	
Algorithm	AVG Top-N	%
BMF	0.4053	40.53%
CAMF	0.5350	53.5 %
Decay CAMF	0.58055	58.55 %

2. Recall :

The recall results of implementing Top-N model using BMF, CAMF, and Decay CAMF are illustrated in Table [12] .

Table 12 Recall Results

	Recall of Top-N Recommendations			
Algorithm	Top-5	Top-10	Top-15	Top-20
BMF	0.210	0.251	0.253	0.255
CAMF	0.381	0.370	0.379	0.382
Decay CAMF	0.392	0.420	0.455	0.485

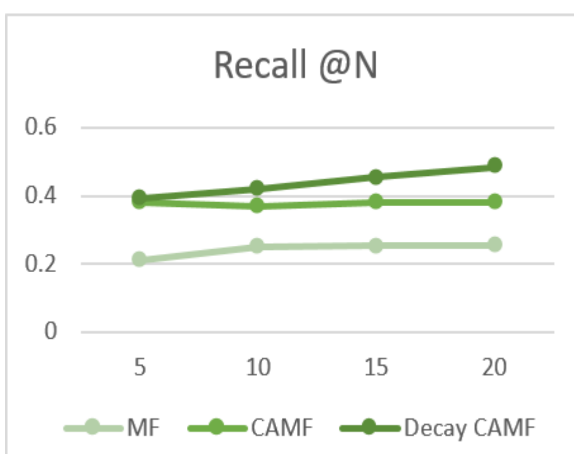

Figure 7 Recall Results

The BMF is the least algorithm with recall value where the incorporating time into CAMF and Decay CAMF improves the optimization of recall. As shown in the above table [38], the average recall results for three algorithms of BMF, CAMF, Decay CAMF models are 24.2%, 35.3%, and 43.8% respectively. The experiment shows that demonstrating Decay func- tion into CAMF algorithm achieves a high recall rate comparing with other algorithms. The recommended items have a larger ratio of coverage for the entire user vector and the recommended group is stronger.

3. MAP:

The MAP results of implementing Top-N model using BMF, CAMF, and Decay CAMF are illustrated in Table [13].

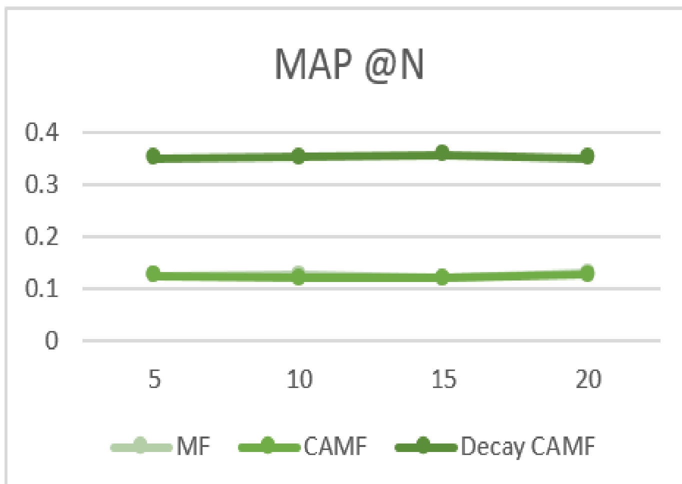

Figure 8 MAP Results

Table 14 Average MAP Results

	Average MAP @N	
Algorithm	AVG Top-N	%
BMF	0.1249	12.27%
CAMF	0.1227	12.49%
Decay CAMF	0.3526	35.26%

Table 13 MAP Results

	MAP of Top-N Recommendations			
Algorithm	Top-5	Top-10	Top-15	Top-20
BMF	0.1253	0.1255	0.12	0.1289
CAMF	0.1250	0.1201	0.1205	0.1255
Decay CAMF	0.3520	0.3522	0.3553	0.3510

The average values of MAP metric is slightly the same for CAMF and BMF however the Decay CAMF outperforms the ac- curacy by 35.26% contrast with 12% for both BMF and CAMF.

We observed from our experiments that, CAMF shows better in performance and prediction accuracy than Biased MF and Decay CAMF has better results than both of them. The average improve- ment of MAE values; BMF is 0.627, and 0.533 for CAMF, and

0.478. Where they registered values of 0.945, 0.764, and 0.680 respectively for BMF, CAMF, and Decay CAMF. Another impor- tant notice that when we increase the number of latent factors the error ratings decrease which enhance the overall performance.

VII. DISCUSSION AND FINDINGS

1. We tested three-levels of Matrix Factorization with different parameters in its biased traditional, Context and Decay forms. The CAMF model outperform over BMF while Decay CAMF

is significantly most effective model. The model based on CAMF run better than other methods due to the applicability of model implicit feedbacks and track the changes in user behaviors over contexts.

2. Using Bias parameters for MF increases the significant results in each experiment and that can be noticed in RMSE results which approaches from 0.956 to 0.930 and in MAE results which decreased from 0.633 to 0.622.

3. Incorporating time-target contexts in recommendation sys- tems CAMF increased the effectiveness of all recommen- dations predications over the traditional biased MF. As we showed in the experiment results, MF with average precision, recall, and MAP of 40.53%, 24.2%, and 12.27% respectively. Where the CAMF including contexts in its model with average precision, recall, and MAP of 53.5%, 35.%, and 12.49%.

4. When we incorporated the half-life decaying mechanism into our prediction process, both of our models' recommendation accuracy considerably improved when compared to the more basic models.

5. In a general thought, as shown by the findings above, the model's accuracy increases each time a particular impact (number of latent factors) increases which let us capture user- item interactions in more depth.

6. In running our experiments, we observed that, while the num- ber of iterations is directly proportional to the effectiveness of our models, giving it a better MAE and RMSE scores, the number of iterations is also directly proportional to the time it takes to learn and generate our latent factors.

VIII. CONCLISION

Enhance the method of providing customers with recommen- dations, which meet their preferences, is a great need due to the vast amount of information produced by the internet. This pa- per uses matrix factorization algorithm as main function to create online e-commerce recommendation system (RS). MF has a va- riety of versions and Context-Aware Recommendation Function (CAMF) is the most commonly one in RS. CAMF is selected to be evaluated in our proposed approach. CAMF deals with contextual information to retrieve the predications for users.

In our case study, we apply three levels of MF; Biased MF, Context-Aware MF, and Decay Context-Aware MF. Explicit rating for user-item pairs is selected to be the main metric for BMF model. Where Time of purchasing and type of target materials are selected to be the contextual information in CAMF and Decay CAMF. The experiment incorporates an exponential Half-life decay function into CAMF with a half-life value of 45 days.

To implement the overall experiment, we used an open-source Java library 'CARSKIT', and also python libraries to execute all needed functionalities. After, we got a list of results, we run precision, recall, MAP, MAE, and RMSE as accuracy metrics
to evaluate the performance of our
experiment under the three algorithms MF, CAMF, and Decay CAMF.

Using our dataset, experimental results show that both CAMF versions (Biased and Decay) yield large enhancements in predica- tion accuracy than the traditional BiasedMF. However, accuracy was improved using half-life with least prediction errors based on the results of 0.478, 0.680 of MAE and RMSE.

IX. FUTURE WORKS

As for future works, we aim to update the model of three-level MF and incorporate more valuable contexts such as name of author. Additionally, we are planning to investigate the implementation of other versions of MF algorithms such as SVD+, and NMF and then incorporate contexts and decay function to these models to track how they will be affected. As another plan, we could study to apply the model in a larger dataset and using all combinations of hyper-parameters with no limits to generalize the results.

X. REFERENCES

1. Adomavicius, G., & Mobasher, B. (2011). Context-aware Recommender Systems. Artificial Intelligence Magazine, 32(3), 67-80.
2. Campos, P.G.; D´ıez, F.; Cantador, I. Time-aware recom- mender systems: A comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adapt. In- teract. 2014, 24, 67–119.
3. A. Hassan, E. Fadel, and N. Akkari, Time-Aware Recom- mender System for E-Commerce Applications, 2020. p. 534- 542.
4. Koren, Y., Bell, R. and Volinsky, C, Matrix factorization techniques for recommender systems. Computer, 2009: p. pp.30-37
5. Rendle, Steffen, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Fast context-aware recommendations with factorization machines." In Proceedings of the 34th in- ternational ACM SIGIR conference on Research and devel- opment in Information Retrieval, pp. 635-644. ACM, 2011.
6. Hornung, T., Ziegler, C. N., Franz, S., Przyjaciel-Zablocki, M., Scha¨tzle, A., & Lausen, G. (2013). Evaluating hy- brid music recommender systems. Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intel- ligence, WI 2013, 1, 57-64. https://doi.org/10.1109/WI-IAT.2013.
7. Basilico, J.; Hofmann, T. Unifying collaborative and content- based filtering. In Proceedings of the TwentyFirst Interna- tional Conference on Machine Learning, Alberta, Canada, July 4-8, 2004.
8. 8. hmadian, S., Meghdadi, M., Afsharchi, M.: Incorporat- ing reliable virtual ratings into

social recommendation sys- tems. Appl. Intell. 48, 4448–4469 (2018).G. F. S. M. Asfiya Shireen Shaikh Mukhtar, “An Introduction of Advanced En- cryption Algorithm: A Preview,” Int. J. Sci. Res., 2014.

9. Li, Jiyun, Pengcheng Feng, and Juntao Lv. ”ICAMF: im- proved context-aware matrix factorization for collaborative filtering.” In Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on, pp. 63-70. IEEE, 2013.

10. Howe, A.; Forbes, R. Re-considering neighborhood-based collaborative filtering parameters in the context of new data. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA, October 26-30, 2008; pp. 1481–1482.

11. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. Adv. Artif. Intell. 2009, 2009, 421425.

12. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Pro- ceedings of the 10th International Conference on World Wide Web, Hong Kong, May 1-5, 2001; pp. 285–295.

13. Tahmasebi, F., Meghdadi, M., Ahmadian, S., Valiallahi, K.: A hybrid recommendation system based on profle expansion technique to alleviate cold start problem. Multimed. Tools Appl. 80, 2339–2354 (2021).

14. Chen, W., et al., A hybrid recommendation algorithm adapted in elearning environments. World Wide Web, 2014. 17(2): p. 271-284.

15. Manolis Vozalis and Konstantinos G Margaritis. On the en- hancement of collaborative filtering by demographic data. Web Intelligence and Agent Systems: An International Jour- nal, 4 (2):117–138, 2006.

16. Gupta J, Gadge J (2015) Performance analysis of recommen- dation system based on collaborative fltering and demograph- ics. In: 2015 international conference on Communication, Information & Computing Technology (ICCICT’15).

17. Wang, Y., Chan, S. C. F., & Ngai, G. (2012). Applicabil- ity of demographic recommender system to tourist attrac- tions: A case study on TripAdvisor. Proceedings of the 2012 IEEE/WIC/ACM International Conference on Web Intelli- gence and Intelligent Agent Technology Workshops, WI-IAT 2012, 97-101.https://doi.org/10.1109/WI-IAT.2012.133.

18. 18. Zheng, Yong, Bamshad Mobasher, and Robin Burke. ”Incorporating context correlation into context-aware ma- trix factorization.” In Proceedings of the 2015 International Conference on Constraints and Preferences for Configuration and Recommendation and Intelligent Techniques for Web Personalization-Volume 1440, pp. 21-27. CEUR-WS. org, 2015.

Baltrunas, Linas, Bernd Ludwig, and Francesco Ricci. ”Ma- trix factorization techniques for context aware recommenda- tion.” In Proceedings of the fifth ACM conference on Recom- mender systems, pp. 301-304. ACM, 2011.

19. Dourish, Paul. ”What we talk about when we talk about context.” Personal and ubiquitous computing 8.1 (2004): 19-30.

20. Liu, N. N., Cao, B., Zhao, M., & Yang, Q. (2010). Adapt- ing neighborhood and Matrix Factorization models for con- text aware recommendation. In CAMRa2010 (pp. 7–13). https://doi.org/10.1145/1869652.1869653.

21. Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse recommendation: n- dimensional Tensor Factorization for Context-aware Col- laborative Filtering. Proceedings of the Fourth ACM Conference on Recommender Systems - RecSys ’10, 79. https://doi.org/10.1145/1864708.1864727.

22. Ricci, G., De Gemmis, M., & Semeraro, G. (2012). Ma- trix and Tensor Factorization Techniques applied to Recom- mender Systems: a Survey. Matrix and Tensor Factorization Techniques Applied to Recommender Systems: A Survey, 1(1), 94-98.

23. Lu, J.; Indeche, M.A. Multi-Context-Aware Location Recom- mendation Using Tensor Decomposition. IEEE Access 2020, 8, 61327–61339.

24. Liu, C.L.; Chen, Y.C. Background music recommendation based on latent factors and moods. Knowl.-Based Syst. 2018, 159, 158–170.

25. Said, A., De Luca, E. W., & Albayrak, S. (2011). Infer- ring contextual user profiles improving recommender perfor- mance. CEUR Workshop Proceedings, 791.

26. adian, S., Meghdadi, M., Afsharchi, M.: A social recom- mendation method based on an

31. Rostami, M., Oussalah, M., Farrahi, V.: A novel time-aware food recommender-system based on deep learning and graph clustering. IEEE Access 10, 52508–52524 (2022).

32. Baltrunas, L.; Amatriain, X. Towards time-dependant recom- mendation based on implicit feedback. In Proceedings of the Workshop on Context-Aware Recommender Systems, New York, NY, USA, October 25, 2009.

33. Qi, L., Wang, R., Hu, C.; Li, S.; He, Q.; Xu, X. Time-aware distributed service recommendation with privacy- preservation. Inf. Sci. 2019, 480, 354–364.

34. Hassan, A. Y., Fadel, E., & Akkari, N. (2022). Exponential Decay Function-Based Time-Aware Recommender System for e-Commerce Applications. International Journal of Ad- vanced Computer Science and Applications, 13(10).

35. Sa´nchez-Moreno, D., Zheng, Y., & Moreno-Garc´ıa, M. N. (2020). Time-aware music recommender systems: Modeling the evolution of implicit user preferences and

adaptive neighbor selection mechanism. Inf. Process. Manag. 54, 707–725 (2018).

27. G. Adomavicius, L. Baltrunas, E. W. De Luca, T. Hussein, and A. Tuzhilin. 4th workshop on context-aware recommender systems (cars 2012). In RecSys, pages 349–350, 2012.

28. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. Adv. Artif. Intell. 2009, 2009, 421425.

29. Ahmadian, S., Joorabloo, N., Jalili, M., Ahmadian, M.: Al- leviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. Expert Syst. Appl. 187, 115849 (2022).

30. Ngaffo, A.N., El Ayeb, W., Choukair, Z.: A time-aware ser- vice recommendation based on implicit trust relationships and enhanced user similarities. J. Ambient. Intell. Humaniz. Comput. 12, 3017–3035 (2021).

user listening habits in a collaborative filtering approach. Applied Sciences, 10(15), 5324.

36. Akhmatnurov, M., & Ignatov, D. I. (2015). Context-aware recommender system based on Boolean matrix factorisation. In CEUR Workshop Proceedings (Vol. 1466, pp. 99-110).

37. Baltrunas, L., & Ricci, F. (2014). Experimental evaluation of context-dependent Collaborative Filtering using item split- ting. User Modeling and User-Adapted Interaction, 24(1-2), 7-34. https://doi.org/10.1007/s11257-012-9137-9.

38. H ermann, C. Time-Based Recommendations for Lecture Ma- terials. In Proceedings of the ED-MEDIA, Toronto, ON, Canada, 2010; pp. 1028–1033.

39. Campos, P.; Bellogin, A.; Diez, F.; Chavarriaga, J. Simple time-biased KNN-based recommendations. In Proceedings of the Workshop on Context-Aware Movie Recommendation, Barcelona, Spain, September 30, 2010; pp. 20–23.

40. Adomavicius, Gediminas, Ramesh

Sankaranarayanan, Sha- hana Sen, and Alexander Tuzhilin. ”Incorporating contextual information in recommender systems using a multidimen- sional approach.” ACM Transactions on Information Systems (TOIS) 23, no. 1 (2005): 103-145.

41. Bostro¨m, Fredrik. ”Andromedia-towards a context-aware mobile music recommender.” (2008).

42. F. Maxwell Harper and Joseph A. Konstan. 2016. The movie- lens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4 (2016), 19.

43. Peeters M, Zondervan-Zwijnenburg M, Vink G, van de Schoot R. How to handle missing data: a comparison of different approaches. European Journal of Developmen- tal Psychology. 2015;12(4): 377–394. doi:10.1080/1740 5629.2015.1049526.

44. Zhang Z. Missing data imputation: focusing on single im- putation. Annals of Translational Medicine. 2016;4(1): 9. doi:10.3978/j.issn.2305-5839. 2015.12.38.

45. TamraparniDasu and Theodore Johnson. Exploratory Data Mining and Data Cleaning. Wiley, 2003.

46. Yusuf, N., Mohd Yunus, M. A., and Wahid, N. (2019). Ara- bic text stemming using query expansion method. Interna- tional Conference of Reliable Information and Communica- tion Technology, 3–11. https://doi.org/10.1007/978-3-030- 33582-3 1

47. Vasiliev, Y. (2020). Natural language processing with python and SpaCy: A practical introduction. No Starch Press.

48. Li, Fangfang, Guandong Xu, and Longbing Cao. ”Cou- pled item-based matrix factorization.” In International Con- ference on Web Information Systems Engineering, pp.

56. L.O. Colombo-Mendoza, R. Valencia-Garc´ıa, A. Rodr´ıguez- Gonza´lez, G. Alor-Herna´ndez, J.J. Samper-Zapater, Recom- metz: a context-aware knowledge-based mobile recom- mender system for movie

1-14. Springer, Cham, 2014.

49. Y. Zheng, B. Mobasher, and R. Burke. Carskit: A java- based context-aware recommendation engine. In Proceedings of the 15th IEEE International Conference on Data Mining Workshops. IEEE, 2015.

50. G. Adomavicius and A. Tuzhilin. Context-aware recom- mender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, Recommender Systems Handbook, pages 217 – 250. Springer, 2011.

51. L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context rel- evance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing, pages 1–20, 2011.

52. Li, Fangfang, Guandong Xu, and Longbing Cao. ”Coupled matrix factorization within non-iid context.” In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 707-719. Springer International Publishing, 2015.

53. Pagano, Roberto, Paolo Cremonesi, Martha Larson, Bala´zs Hidasi, Domonkos Tikk, Alexandros Karatzoglou, and Mas- simo Quadrana. ”The Contextual Turn: from Context-Aware to Context-Driven Recommender Systems.” In RecSys, pp. 249-252. 2016.

54. Isinkaye, Folajimi, and Ojokoh. ”Recommendation systems: Principles, methods and evaluation.” Egyptian Informatics Journal 16, no. 3 (2015): 261-273.

55. J.-W. Son, A. Kim, S.-B. Park, et al., A location-based news article recommendation with explicit localized semantic anal- ysis, in: Proc. 36th ACM SIGIR Int. Conf. on Research and Development in Information Retrieval, ACM, 2013, pp. 293–302.

showtimes, Expert Syst. Appl. 42 (3) (2015) 1202–1222.

57. C. Biancalana, F. Gasparetti, A. Micarelli, G. Sansonetti, An approach to social recommendation for context-aware mobile

services, ACM Trans. Intell. Syst. Technol. 4 (1) (2013) 10.

58. S. Seifu, S. Mogalla, A comprehensive literature survey of context-aware recommender systems, Int. J. Adv. Res. Com- put. Sci. Softw. Eng. 3 (4) (2016) 40–46.

59. A. Hawalah, M. Fasli, Utilizing contextual ontological user profiles for personalized recommendations, Expert Syst. Appl. 41 (10) (2014) 4777–4797.

60. A.M. Otebolaku, M.T. Andrade, Context-aware media rec- ommendations for smart devices, J. Ambient Intell. Hum. Comput. 6 (1) (2015) 13–36.

61. C. Musto, G. Semeraro, P. Lops, M. de Gemmis, Con- textual eVSM: a content-based context-aware recommen- dation framework based on distributional semantics, in: E-Commerce and Web Technologies, Springer, 2013, pp. 125–136.

62. Wang, C.; Zhang, M.; Ma, W.; Liu, Y.; Ma, S. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, May 13- 17, 2019; pp. 1977–1987.

63. Hariri, N.; Mobasher, B.; Burke, R. Context-aware Music Recommendation Based on Latent Topic Sequential Patterns. In Proceedings of the Sixth ACM Conference on Recom- mender Systems, Dublin, Ireland, September 9-13, 2012; pp. 131–138.

64. Wang, D.; Deng, S.; Xu, G. Sequence-based context-aware music recommendation. Inf. Retr. 2018, 21, 230– 252.

65. Sa´nchez-Moreno, D.; Zheng, Y.; Moreno-Garc´ıa, M.N. In- corporating Time Dynamics and Implicit Feedback into Mu- sic Recommender Systems. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intel- ligence (WI), Santiago, Chile, 3–6 December 2018; pp. 580–585.

66. Chen, J.; Wang, C.; Wang, J.; Yu, P.S. Recommendation for Repeat Consumption from User Implicit Feedback. IEEE Trans. Knowl. Data Eng. 2016, 28, 3083–3097.

67. Tuan, C.-C.; Hung, C.-F.; Wu, Z-H. Collaborative location recommendations with dynamic time Periods. Pervasive Mob. Comput. 2017, 35, 1–14.

68. Koren, Y. Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Trans. Knowl. Discov. Data 2010, 4, 1.

69. Li, X.; Xu, G.; Chen, E.; Zong, Y. Learning recency based comparative choice towards point-of-interest recommenda- tion. Expert Syst. Appl. 2015, 42, 4274–4283.

نظام توصيات مدرك للسياق باستخدام تحليل المصفوفة

ديمه الاحمدي[١]، مسره طلعت الدريملي[٢]

[١] قسم نظم المعلومات، كلية الحاسبات وتقنية المعلومات

جامعة الملك عبد العزيز، جدة ، المملكة العربية السعودية

, meldremele@stu.kau.edu.sadalahmadi@kau.edu.sa

مستخلص. في المجال التجاري، سيتم إضافة ملايين العناصر الجديدة إلى المبيعات اليومية. يعد اقتراح العناصر المقترحة للمستخدمين لعملية الشراء نقطة حاسمة. إن العثور على أفضل الاقتراحات بناء على احتياجات المستخدم وسلوكه يزيد من إنتاجية المبيعات. لقد رافق دمج معلومات السياق في عملية التوصيات العديد من المجالات والتطبيقات. تم استخدام أساليب واستراتيجيات مختلفة للعثور على التوصيات. في حين أن الوقت عامل مهم للتحديثات والتغييرات المستمرة في تفضيلات المستخدم، فقد ثبت أن دمجه فعال في تعزيز أداء التوصية. تم استخدام أنظمة التوصية المدركة للوقت (TARS) في مجموعة واسعة من نماذج التوصيات. في هذه الورقة المقترحة، نركز على التعامل مع ثلاث خوارزميات مختلفة واعية بالسياق. أولاً، تحليل المصفوفات التقليدي باستخدام التصنيفات الصريحة. ثانياً، النسخة المحسنة بعد التعامل مع الهدف الزمني كعوامل أساسية للحصول على النتائج. أخيراً، اعتماد على الإصدار السابق، قمنا بتعزيزه عن طريق تقليص الأوزان باستخدام خوارزمية دالة الاضمحلال الرياضية لتحسين دقة التنبؤ. نحن نبني حلولنا وننفذها باستخدام مجموعة بيانات حقيقة لموقع تجاري كدراسة حالة تجريبية لدينا. من تحليلنا، قمنا أخيراً بتقييم النموذج المقترح باستخدام مقاييس مختلفة لقياس الأداء النسبي لـ TARS المحسن على MF التقليدي.

الكلمات المفتاحية. أنظمة التوصية المدركة للسياق، أنظمة التوصية المدركة للوقت، مقاييس التقييم، خوارزمية تحليل المصفوفة.