Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/10661
DOI URL: http://dx.doi.org/10.21474/IJAR01/10661

RESEARCH ARTICLE

TIME-AWARE RECOMMENDER SYSTEM FOR E-COMMERCE APPLICATIONS

Ayat Yehia Hassan[1], Dr. Etimad Fadel[2] and Dr. Nadine Akkari[3]

1. Master Student, Department of Computer Science, King Abdul-Aziz University, Jeddah, Saudi Arabia.
2. Associate Professor, Department of Computer Science, King Abdul-Aziz University, Jeddah, Saudi Arabia.
3. Assistant Professor, Department of Computer Science, Lebanese University, Lebanon.

Manuscript Info

Manuscript History
Received: 12 January 2020
Final Accepted: 15 February 2020
Published: March 2020

Keywords:-
Context-Aware Recommender Systems, Time-Aware Recommender Systems, Content-Based Filtering, Collaborative Filtering, Recommender System

Abstract

As e-commerce websites began to develop, users found it difficult to find the most appropriate choice from the immense variety of items.Recommender System (RS) is a subclass of information filtering used to predict a rating that a user would give to an item. Recommender systems have been applied to several domains such as online streaming, Marketing, and e-commerce to assist in decision making.Recently, contextual information has been recognized as a useful factor in improving the quality of recommendations in different fields. However, it is under investigation in the area of online shopping. Among all contextual information, time is considered as one of the most important dimensions. This paper integrates time dynamics with implicit feedback (add to cart and Transactions) in an online shopping recommender system using the Matrix Factorization algorithm (MF). The integration is done using two approaches: The first approach is Bias in which the time is used as the third column in the user rating matrix. The second approach is the Decay function which produces new ratings by aggregating implicit feedback with time dynamics and gives a higher weight to the new items over older ones. Using the "Retailrocket" online shopping dataset, the experimental results demonstrate the effectiveness of the decay function over the traditional context-aware matrix factorization CAMF (bias) in terms of precision, recall, and Mean Average Precision (MAP).

Introduction:-

Recommender Systems (RSs) are software tools and techniques to provide suggestions for items to a user[1]. The main goal for Recommender Systems is to help individuals to choose from a huge number of items that an application may offer. In recent years, Recommender Systems play an important role in highly rated internet sites such as Amazon, YouTube, Netflix, and Yahoo. Besides, many companies are deploying Recommender Systems (RSs) as part of their services[2].

Recommendation researches exploited the context (e.g. location, time, weather, device, etc.) where users expressed their preferences. It has been proven that using the context information has a valuable effect on increasing the performance of the recommendation[3]. Among all contextual information, time is considered as one of the most important dimensions for the following reasons. First, it facilitates tracking the changes in user preferences over time to identify new habits and interests[4].Second, the winning team of the Netflix Prize competition[5] found that

Corresponding Author:- Ayat Yehia Hassan
Address:- Master Student, Department of Computer Science, King Abdul-Aziz University, Jeddah, Saudi Arabia.

using time context can significantly increase the reliability of the recommendations. Third, collecting temporal information is generally easy and doesn't require any additional devices or user efforts. Due to these benefits, the investigation of Time-Aware Recommender System (TARS) becomes very popular in recent years[4]. The main goal of TARS is dealing with changes in user preferences over time. The changes depend on the season (e.g. day of the week, time of day, etc.), or user tests. Also, it could depend on the application domain which is the way to address time in a specific field. The field of e-commerce differs from other fields as it focuses on recommending products that users are interested in purchasing, not only the products that match their preferences. This will help to increase the revenue of the e-storesthat dependmainly on the sold items. Researches that consider purchasing information such as[3, 4, 6] did not address the time dimension.

Due to the lack of datasets that use context information in the field of e-commerce, the impact of using the time context in the e-Commerce domain has not been explored in the literature. Many of the TARS research results can't be applied in the e-commerce field. Because this field not only concerned with the user's preferences, but it also concerned with the act of buying that leads to the main goal of increasing the store's revenues.For this purpose, this work applies TARS in an online shopping domain. The main contribution of the work is to integrate time with MF using two different approaches for recommending online shopping products. The first approach is Bias in which the time is used as the third column in the user rating matrix. The second approach is the Decay function which produces new ratings by aggregating implicit feedback with time dynamics and gives a higher weight to the new items over older ones.

The rest of the paper is organized as follows. Section 2 &3 are dedicated to introducing background information and to provide a brief description of related works, respectively. Section 4 presents an approach for applying time dynamics using the decay function in the online shopping domain. Section 5 contains the experiment and performance analysis. Results and discussion are explained in section 6. Finally, conclusions are reported in section 7.

Background:

This section provides background information about the Matrix Factorization algorithm (MF), Time-Aware Recommender System (TARS), and evaluation matrices for Recommender systems (RSs).

Matrix Factorization:

Matrix factorization (MF) is a type of collaborative filtering (CF) algorithms used in RSs [3]. Matrix factorization depends on approximating the item rate matrix R by finding the product of two smaller rectangular matrices[3]as shown in the example in fig1. MF algorithm starts with initializing two matrices randomly as shown in fig1. The first one will be u x f matrix X, while the second will be a f x s matrix Y. When these two matrices multiply with each other, they result in u x s matrix, which is exactly the size of our Rating matrix R in which we are trying to predict. Dimension f represents the number of latent factors we're using to estimate the rating matrix. Generally, f is between 10–250. if we take a number of latent factors =1, it means we are choosing the most popular items that have the most interactions without considering any personalization. Increasing the number of latent factors (The number of rows or columns related to a specific item or user) will improve the personalization, hence improving the quality of the RS. but at some point, when the number of latent factors becomes too high, it will cause model overfitting problem and that will decrease the quality of the recommendation. Thus, we need a regularization term to avoid overfitting. As shown in the equation below, the algorithm tries to get the matrices P and Q filled by predicting the ratings in the training set using the following equation:

$$\min \sum_{\langle u_a, i_x \rangle \in \tau} (r_{u_a,i_x} - P_{u_a,i_x})^2 = \min_{p_*, q_*} \sum_{\langle u_a, i_x \rangle \in \tau} \left(r_{u_a,i_x} - q_{i_x}^T p_{u_a}\right)^2 + \lambda \left(\|q_{i_x}\|^2 + \|p_{u_a}\|^2 \right) \quad (1)$$

Where p, q are the two factors matrices, τ is the set of ratings in the training set and $\lambda(\|q_{i_x}\|^2 + \|p_{u_a}\|^2)$ is a regularization term used to prevent overfitting.

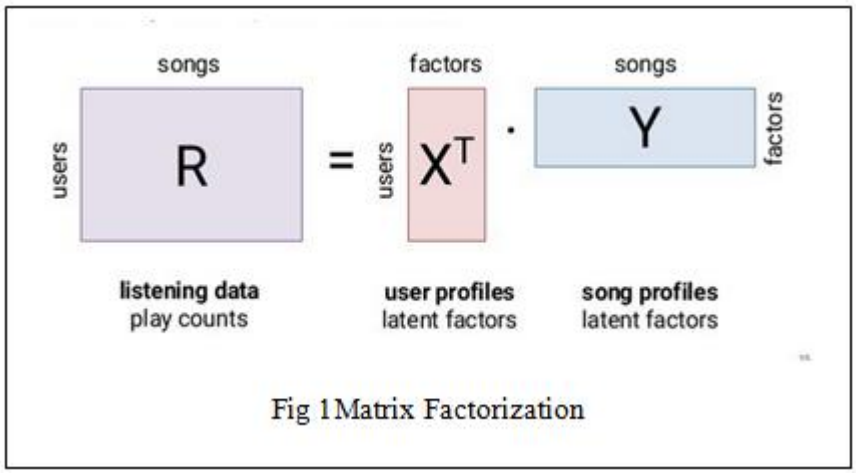

Fig 1Matrix Factorization

In this research, the Latent Factor Algorithm with matrix factorization is chosen as it is the commonly used approach and can be adapted to include contextual information which is the temporal data that can be inferred from the e-commerce site analytics logs.

Time-Aware Recommender System (TARS):

Time-Aware Recommender System (TARS)is a specific type of Context-Aware Recommender System (CARS). But time context has an advantage among all context dimensions which is its ability to be captured easily because almost all devices have a clock that can determine the time when any interaction between the user and the system occurs. Besides, according to[7], the use of time can improve the quality and accuracy of the recommendation process. There are two types of time variables: continuous and categorical. Continuous variables mean the exact time at which the user interacts with the item such as (Sep 1st, 2019 at 18:15:47)[7]. Categorical variables can be calculated according to specified periods of interest such as (morning, evening, afternoon) [7].

Researchers in[8] introduced seven categories of how the time can be used in RSs:

1. Restriction: In this type, RS matches the user's available time with the time when he is going to use the item. For example, if the user is going to have dinner, the system will recommend only restaurants that are open during the night.
2. Micro-profile: For every user, the system stores different profiles for each time. For example, the user has two profiles, one for weekends and another profile for weekdays. So, the system will recommend a product to the user at Friday morning based only on the products he consumed in past Fridays.
3. Bias: The system stores the time when the user rates an item. So, time will be used as the third dimension of a User's Item matrix. This data is used by collaborative filtering to compare users and to find k-neighbors then to predict user's rating to a non-rated item.
4. Decay: In this type, the system gives more weight to the new items compared to the items that have older interaction.
5. Time Rating: Time is used to give implicit feedback to infer user preferences, for example, the longer a user stays on a product, the greater the user's preference for that product.
6. Novelty: The recommendation system sets a limit date and all products that are older than this date will not be recommended, for example in a news website RS recommended news of, at least, one day ago.
7. Sequence: The RS monitors items that are consumed in a specific sequence or one after another. For example, when a user buys a mobile phone, he is likely to buy the accessories needed for the same phone, so the system will suggest the rest of the products that are usually purchased with the consumed product.

In this paper we will test incorporating the Matrix Factorization Algorithm (MF) with the time dynamics using the decay approach wish is not widely used in the e-commerce domain. Also, we will compare it with the traditional bias approach.

Evaluation Matrices:

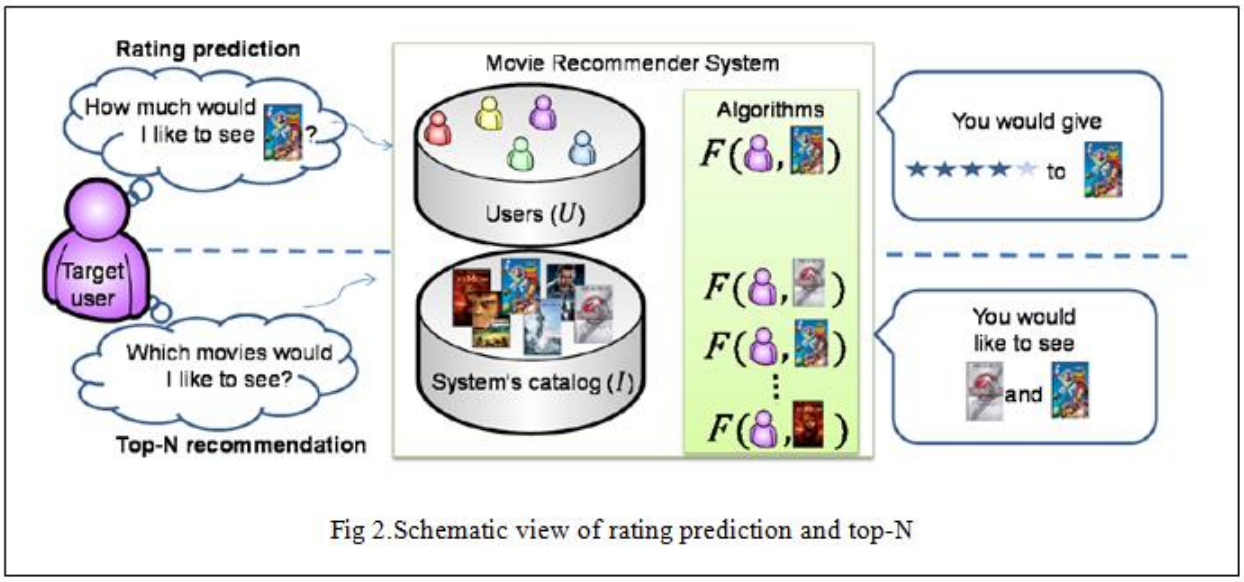

Fig 2.Schematic view of rating prediction and top-N

There are two types of recommenders: rating prediction and top-N item recommendation as shown in Fig.2. The rating prediction can be evaluated by different prediction errors, such as mean absolute error (MAE), root means square error (RMSE) and means prediction error (MPE). The item recommendation can be evaluated through relevance metrics, such as precision and recall, and ranking metrics, such as mean average precision (MAP)[9].

The root-mean-square error (RSME) and Mean Absolute Error (MAE) are used to evaluate the rating predictions but cannot be used for evaluating the top-N recommendations. Therefore, in this research, we used precision and recall. Precision is a fraction of items that are relevant in a retrieved recommendation list as shown in equation (2). The recall is the fraction of the relevant items that are shown in the recommendations lists as shown in equation (3) [10]. One of the shortcomings of the precision metric is that it does not give weight to the location in the list where relevant items are being recommended. For example, if two relevant items appeared in the first two positions in the top-N recommendation list, they will have the same precision value even if the two items appeared in the last two positions in the list. Thus, we need another evaluation metric to take into account the position of the relevant items. Mean average precision (MAP) often used to compare top-N recommender systems [10]. Its value will be higher if the relevant items appear in the earliest positions.

Recall and Precision:

Precision and recall are binary metrics used to evaluate models with binary output. Thus, we need a way to translate our numerical problem (ratings usually from 1 to 5) into a binary problem (relevant and not relevant items). Recall and precision help users to select items that are more similar among a set of items. The metrics consider the prediction procedure as a binary operation that allows the user to specify the good items from other items that are not related or not good. Recall **r** and Precision **p** can be calculated as follow:

$$p = \frac{\text{no. of relevant recommendations}}{\text{no. of recommended items}} \quad (2)$$

$$r = \frac{\text{no. of relevant recommendations}}{\text{no. of all the possible relevant items}} \quad (3)$$

Mean Average Precision (MAP):

The Mean Average Precision (MAP) is similar to precision with a weight for the position of the item in the recommendation lists is considered. It is calculated using the following equation:

$$MAP = \sum_{i=1}^{N} \frac{P@i}{N} r_i \quad (4)$$

In this equation, N is the length of the recommendations list, P@i is the precision at item i. ri = 1 if the item i in the recommendations list is relevant and ri = 0 otherwise [10]. As for precision, the maximum value for the MAP is 1 and increased effectiveness is indicated by higher values.

Related Work: -

This section presents previous works that use TARS in different domains. We classified them according to the domain and the seven categories of how the time can be used in RSs as mentioned in the previous section.

Many researchers investigated the use of decay function in RSs [11-16]. In [17]authors produced a simulated rating by aggregating implicit feedback dataset with time dynamics using the decay function similar. The experimental results demonstrate the effectiveness of their proposed approach. for both rating prediction and top-N-recommendation in the media domain. Another research in[18]set out to increase the performance of neighborhood-based recommender systems by including temporal information with the decay function in addition to ratings. Researchers in[8] categorized the types of how time can be used in the RSs. and provided different scenarios for decay, restriction, and novelty to be applied in the learning domain. Similar to[19] which also used the sequence category with the decay. [8] used the time context in Pre- Filtering and Post-Filtering with content-based and collaborative filtering algorithms. But they did not provide any experimental test on a dataset to evaluate their proposed models and scenarios.[12] proposed an approach that predicts user preferences with consideration of preference changes by learning the order of purchase history in a recommender system. Their approach uses a Kalman filter to predict user preference vectors from user features using the micro profile app. [20] used the time context to improve the user-based collaborative filtering algorithm. they proposed a weight function to consider the changes in the group user's preferences over time. their approach increases the prediction accuracy of the collaborative filtering (CF) prediction algorithm on a movie's dataset. [21] defines a proactive Context-Aware RS that recommends learning objects to teachers and scientists that will produce learning resources the students will consume. Time in [21]is used like Restriction; the RS tries to find learning materials that match the actual user's time.

From the literature, we observe that time is mostly used in the media domain [11, 13-18]and learning domain [21-24][8, 19]. But it is rarely used in the e-commerce domain. The research[12] explored the use of time in the e-commerce domain but only using the micro profile category. Also, we can observe that the change in the user's preferences depends on the season and application domain. Thus, the way to address time in a specific field may not be extended to others. In this research, we adopted the idea of the decay function to be applied to the purchasing data (add to cart & transactions) in an online shopping system.

Incorporating Time Dynamic with MF:

Our approach includes threemain phases as shown in fig3. First,the data pre-processing will be explained in the performance analysis sections. Second the half-life decaying function. Third, the contextual modeling and generating the top-N recommended items.

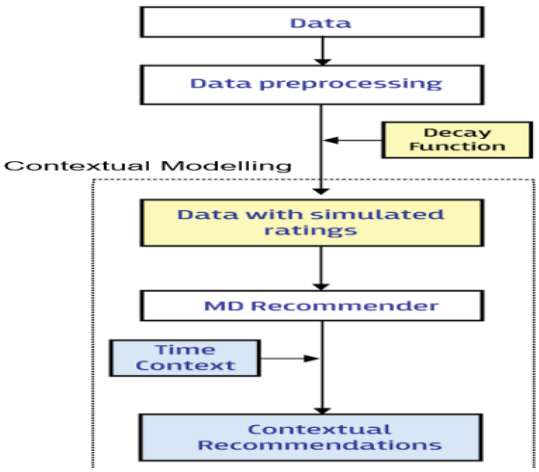

Fig 1:- Experiment Approach.

Decay function can be applied to a quantity that decreases at a rate proportional to its value. Half-life is the period anitem takes to decrease by half. Half-life is very often used to describe quantities having exponential decay. It is a constant over the whole life. In our case, to track the user's changes in preferences over time we need to give a higher weight to newly purchased items over the old ones. Thus, decay could be applied. The formulas below are used in the calculations that involve exponential decay [25].

$$\text{PredictedAmount} = \frac{\text{ActualAmount}}{2^{n}} \quad (5)$$

Where:

$$n = (\text{ElapsedTime})/(\text{Half\_Life}) \quad (6)$$

The half-life is the number of days that have to pass to decrease the weight of a user rate by half. for example, if my dataset considers (365 days), and we assume half-life =133 days. that means 133 days must pass to give the user's rate half of its value (if it was 4 it will become 2 after 133 days). The actual amount is the actual value of user preference. The predicted amount is the new value of user preference after reducing its value using the decay function.After calculating the predicted rate using the decay function. we model the time dimension to be used during the recommendation process in the Matrix Factorization. Then the algorithm generates the top-N recommended items for the user.

Experiment and Performance Analysis:

Our experiment is designed to evaluate three approaches for TARS. The traditional time un-aware RS (MF), context-aware matrix factorization (CAMF) (Bias approach) and the decay function context-aware matrix factorization (DCAMF). We evaluate the performance of the algorithms using the train-validate-test approach (splitting up the data into a training set, a validating set, and a testing set). Finally, we compare the three approaches in terms of Persian, Recall, and MAP.

Dataset and preprocessing:

"Retailrocket" dataset has been collected from a real-world e-commerce website. Table 1 shows the dataset sample. A visitor can make three types of events, namely "view", "add to cart" or "transaction". For each interaction, the system stores the VisitorID, ItemID, type of the event, the TransactionID for the purchasing events, and the time when the interaction occurred.

The first step for constructing our dataset is pre-processing by filtering the events. The data set has three different events. Views, add to cart, and transactions, However, considering all the events is difficult as it requires huge time for running the experiment. Therefore, we decide to include only the purchase data (buy & add to cart). The number of "buy" events is about 8k and the number of "add to cart" events is about 28k (total size of dataset ≅ 36k).The user-item matrix density has been calculated as follow:$\frac{\#\text{ interactions}}{\#\text{users} \times \#\text{items}} \times 100\%$. The Dataset density= 0.0124%.

Table 1:- Dataset Sample.

Timestamp (Standard Unix timestamp)	**VisitorID**	**Event ("View", "Add to cart", "Transaction")**	**ItemID**	**TransactionID**
1433222531378	57036	view	334662	
1433174518180	189384	transaction	299044	7244
1433223236124	287857	addtocart	5206	
1433224244282	1370216	view	176721	
1433221078505	158090	addtocart	10572	

The visitor id, item id, and the timestamp used to determine the day of the week and the hour of the day where the event occurred. We convert the timestamp into categories as follow:

- Day of the week (Sun, Mon, Tue, Wed, Thru, Fri, Sat).
- Hour of the Day (1 to 24).

Implementation:
In the implementation stage, we assume a value for each of the user's events. For example, 5 for the transaction event and 4 for add to cart event. After that, for each event, we calculate the elapsed time.Then, we used it to find the new predicted rate by giving less weight to older events. For the MF algorithm, we modified the CARSKit library[10] to be able to calculate the evaluation matrices (Precision -Recall-MAP) for N =1 to N=20.

In table 2, we presented the parameters used in the MF &CAMF algorithms. MF has three main parameters: number of factors, number of iterations in the cross-validation and the regularization factor to avoid the overfitting while increasing the number of latent factors.The number of iterations We assume the following values in our experiment based on the appropriate values for our dataset and the CARSKIT default values. we select the number of factors =10 to ensure that we had enough factors to sufficiently describe the variability in the data but not so many that may over-fit the training data. The number of iterations =100.Using the "Retailrocket" dataset, we assume the half-life value is equal to 43 days (one month and a half) since our dataset is between May 2015 to August 2015.

Table 2:- Experiment Assumptions.

Algorithm	Hyper-parameter	Values
MF	# of factors	10
	# of iterations	100
	Regularization factor	default setting
CAMF	# of factors	10
	# of iterations	100
	Regularization factor	default setting

Results:-

In this section, we compared the two time-aware recommender system approaches (Bias and Decay) using three evaluation metrics: Precision, Recall, and MAP. We computed the three matrices for N=1 to N=20 where N is the number of recommended items. Then, we calculated the average. The results of generating the top N recommended items using three algorithms time un-aware matrix factorization (MF), Context-aware matrix factorization (CAMF Bias), and Decay Context-aware matrix factorization (DecayCAMF) are shown in Fig 3,4, and 5. In fig3 the plot represents the persion@N for the three algorithms. We can notice that Decay CAMF increases the effectiveness of the recommendation process on average.Fig4, 5 shows the recall@N and MAP@N respectively for the three approaches. Table 3 presents the results from applying the three different approaches of TARS. The average results indicate that the decay approach performed better than the MFCA and MF with precision, recall, and MAP of 0.05%, 0.45%, and 0.16% respectively.

Given that the density of the used data set which is = 0.0124%. The results show that adding the time as a context using the decay function to generate a new user rate is improving the system prediction for the users purchasing actions in terms of precision, recall, and MAP.

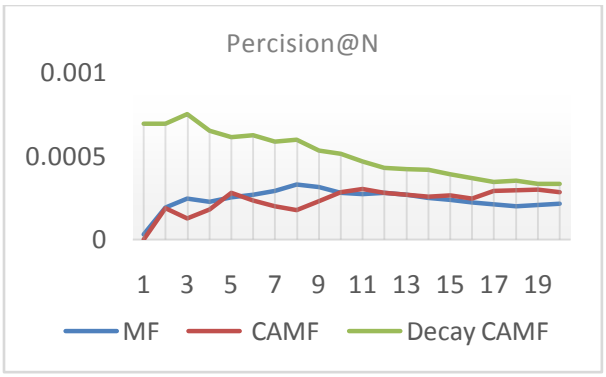

Fig 2:- Percision@N(MF,CAMF,DecayCAMF).

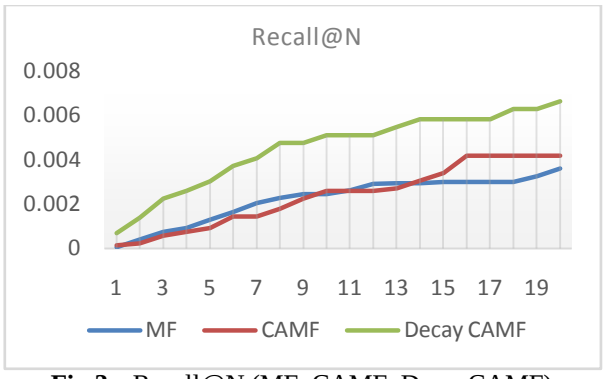

Fig 3:- Recall@N (MF, CAMF, DecayCAMF).

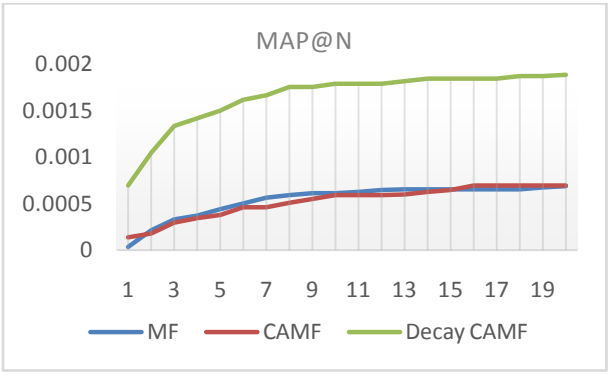

Fig 4:- MAP@N(MF,CAMF,DecayCAMF).

Table 3:- The average of the evaluation matrices (precision, Recall, MAP) for the three algorithms (MF, CAMF, DecayCAMF).

	AVG Precision	**AVG Recall@N**	**AVG MAP@N**
MF	0.02%	0.22%	0.05%
CAMF	0.02%	0.24%	0.05%
Decay & CAMF	0.05%	0.45%	0.16%

Conclusion:-

In this paper, the accuracy of two methods for dealing with the change in user preferences (bias and decay) in an online shopping system is evaluated. We incorporated the exponential decay function into a matrix factorization model. We evaluated its quality in terms of precision, recall, and MAP. We compared the decay method with the bias method using K-fold cross-validation with 80% as training. We focused on the user purchasing actions (add to cart and transaction). Experimental results show that the decay function outperformed the traditional CAMF (bias) in terms of precision, recall, and MAP. In future work, this study could be applied to a larger dataset to generalize the results. The work could also be extended by evaluating the use of the decay function method with other states of art algorithms.

Reference:-

1. F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," Recommender systems handbook, pp. 1-34: Springer, 2015.
2. F. Ricci, Rokach, L. and Shapira, B, “Introduction to recommender systems handbook. In Recommender systems handbook,” Springer Boston, MA. , pp. (pp. 1-35), 2011.
3. J. B. Schafer, J. A. Konstan, and J. Riedl, “E-commerce recommendation applications,” Data mining and knowledge discovery, vol. 5, no. 1-2, pp. 115-153, 2001.
4. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce." pp. 158-167.

5. Y. Koren, Bell, R. and Volinsky, C, “Matrix factorization techniques for recommender systems,” Computer, pp. pp.30-37, 2009.
6. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering." pp. 291-324.
7. P. G. Campos, F. Díez, and I. Cantador, “Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols,” User Modeling and User-Adapted Interaction, vol. 24, no. 1-2, pp. 67-119, 2014.
8. E. J. de Borba, I. Gasparini, and D. Lichtnow, "The Use of Time Dimension in Recommender Systems for Learning." pp. 600-609.
9. G. Adomavicius, and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Transactions on Knowledge & Data Engineering, no. 6, pp. 734-749, 2005.
10. D. Guibing Guo; Jie Zhang; Zhu Sun; and Neil Yorke-Smith. In Posters, “ Librec: A java library for recommender systems.,” in Late-breaking Results and Workshop Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization, 2015.
11. J. Chen, L. Wei, U. Liji, and F. Hao, “A Temporal Recommendation Mechanism Based on Signed Network of User Interest Changes,” IEEE Systems Journal, 2019.
12. K. Inuzuka, T. Hayashi, and T. Takagi, "Recommendation system based on prediction of user preference changes." pp. 192-199.
13. F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, “TCARS: Time-and community-aware recommendation system,” Future Generation Computer Systems, vol. 78, pp. 419-429, 2018.
14. H. Feng, J. Tian, H. J. Wang, and M. Li, “Personalized recommendations based on time-weighted overlapping community detection,” Information & Management, vol. 52, no. 7, pp. 789-800, 2015.
15. M. Gueye, T. Abdessalem, and H. Naacke, "Dynamic recommender system: using cluster-based biases to improve the accuracy of the predictions," Advances in Knowledge Discovery and Management, pp. 79-104: Springer, 2016.
16. C. Luo, X. Cai, and N. Chowdhury, "Self-training temporal dynamic collaborative filtering." pp. 461-472.
17. D. Sánchez-Moreno, Y. Zheng, and M. N. Moreno-García, "Incorporating time dynamics and implicit feedback into music recommender systems." pp. 580-585.
18. T. de Zwart, “Time-Aware Neighbourhood-Based Collaborative Filtering,” 2018.
19. J. Luo, F. Dong, J. Cao, and A. Song, “A context-aware personalized resource recommendation for pervasive learning,” Cluster Computing, vol. 13, no. 2, pp. 213-239, 2010.
20. B. Karahodza, H. Supic, and D. Donko, "An Approach to design of time-aware recommender system based on changes in group user's preferences." pp. 1-4.
21. D. Gallego, E. Barra, S. Aguirre, and G. Huecas, "A model for generating proactive context-aware recommendations in e-learning systems." pp. 1-6.
22. R. M. A. T. A. Arora, “Temporal Recommendations for Discovering Author Interests,” in 2019 Twelfth International Conference on Contemporary Computing (IC3) Noida, India, 2019, pp. pp. 1-6.
23. R. Benlamri, and X. Zhang, “Context-aware recommender for mobile learners,” Human-centric Computing and Information Sciences, vol. 4, no. 1, pp. 12, 2014.
24. W. Chen, Z. Niu, X. Zhao, and Y. Li, “A hybrid recommendation algorithm adapted in e-learning environments,” World Wide Web, vol. 17, no. 2, pp. 271-284, 2014.
25. Ludwig–Maximilians, “Demonstration of the exponential decay law using beer froth,” EUROPEAN JOURNAL OF PHYSICS, pp. 21-26, 2001.