International Journal of Innovative Computing, Information and Control
Volume **13**, Number **5**, October **2017**

ICIC International ©2017 ISSN 1349-4198
pp. 1579–1594

REVIEW OF THE TEMPORAL RECOMMENDATION SYSTEM WITH MATRIX FACTORIZATION

Ismail Ahmed Al-Qasem Al-Hadi, Nurfadhlina Mohd Sharef
Md Nasir Sulaiman and Norwati Mustapha

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia
esmail.hadi2009@gmail.com; { nurfadhlina; nasir; norwati }@upm.edu.my

Received August 2016; revised February 2017

Abstract. *The temporal recommendation system (TRS) is designed for providing users with an accurate prediction based on the history of their behaviour during a precise time. Most TRS approaches use matrix factorization and collaborative filtering, which are primarily based on the distribution of the user preferences. Recently, TRS has gained significant attention because it improves the accuracy of prediction. This is because since the temporal drift in the user preferences is observed, users' preferences within the short term and long term can be utilized to predict the best item to be recommended. Several existing review papers have focused on the general problems of the recommendation system (RS) and similarity measures, and they refer to recent improvements based on three recommendation strategies which are userrating, tagging and trust values. However, there is a lack of recent review papers of TRS with rating score strategy, especially in terms of learning factorization features of temporal terms. This paper fills this gap and highlights the issues and challenges for both general and temporal RS techniques. The prediction approaches based on collaborative filtering technique are reviewed depending on the behaviour of users and items. The challenges and approaches of temporal-based RS are discussed. This review includes the matrix factorization approaches that are integrated with such temporal factors as long-term preferences, short-term preferences, decay, and drift. The outcome of this review prioritizes guidance to focus on matrix factorization, temporal terms and drifting of users' preferences.*

Keywords: Recommendation system, Collaborative filtering, Matrix factorization, Temporal

1. **Introduction.** The recommendation system (RS) gathers information on its users' preferences on a set of products or items. Usually, the preferences of the user are tracked implicitly when the user selects the item and explicitly when the user gives a rating score to this item. RS is used for providing users with recommendations of items based on filtering and prediction techniques. There are four types of filters used in RS: content-based filtering (CBF), collaborative filtering (CF), demographic filtering (DF) and hybrid filtering (HF). HF is performed by combining either CF and CBF or CF and DF [1]. However, the CF technique is the most familiar and the most widely implemented for personalized recommendations in RS [2]. The CF technique aims to decide which users are similar to a target user and then recommend similar items to a target user based on the learned items of other users with similar behaviour. The users express their preferences as rating scores for the particular items, but these preferences may not be sufficient for analyzing the behaviour of users and obtaining accurate recommendations.

As a result of lacking users' preferences in the CF technique, there are three issues: data sparsity, scalability and cold start [3,4]. CF also has two types of challenges: handling the large-scale dataset and extracting the rich information from the collected data. The CF technique uses the rating scores of users to provide the personality features and obtain the accurate predictions for the set of items rated by the active user with the lowest cost. However, the CF technique has a weak prediction performance due to the high percentage of missing data solved by the prediction approaches such as factorization approaches. For instance, the singular value decomposition (SVD), baseline and the matrix factorization (MF) are three kinds of factorization approaches which are used to predict the missing scores.

In CBF technique, the recommendations have extracted from the information associated with the items such as films, books and trips ranked by users previously [5]. The main factor of CBF technique is the similarity task among the items [6]. The items of MovieLens dataset have genre values which refer to the kind of these movies such as action, romance, comedy, and war. The genre features are used to get the genre similarity [7] and also the genre features can be added to the latent features of the users' preferences to predict the missing rating scores in the rating matirx [8]. In addition, the content features of users as age, occupation, and geographical location can be used to extract the user's interest for each item [4]. However, the CBF technique analyzes the profiles of items and then computes the similarity between the preferred items which are hard extracting for some data types as audio/video products. Another limitation of the CBF is about the scope of possible interests, where the system can recommend the items which are preferred by the common users and cannot recommend the new items which are out of the scope of the interesting items named cold-start problem [9].

In DF technique, the position of the rating score of the user can be used to classify the kinds of users who are demographically similar to specific users, and generalize from their ratings of predicting items. For instance, the user's features of a restaurant or a cafe are used to recommend specific items to a user who lives in the same demographic location [10].

HF techniques are used for solving the cold start problem such as by predicting a new item for the user using both the explicit features (rating scores) and other features of content or demographics [11]. CBF and DF are unsuitable for huge datasets – e.g., Netflix or MovieLens 10 M – because they are more scalable (costly) and have less accurate predictions than the CF techniques.

MF is typically known as an important factorization approach to achieve high accuracy by learning the important features in the rating matrix. The factorization features help explain the behaviour of users such as latent feedback and baseline features. These features can be integrated with other temporal features. The temporal recommendation system (TRS) is the latest improvement of factorization approaches based on the short term, long term or a combination. Although many review papers have highlighted the issues of RS and their approaches, most of them have not covered the dynamicity aspect. There are several TRS approaches including time weight CF [12], temporal dynamics [13], and dynamic MF [14]. The dynamic approaches are used to track the behaviour of users overtime. This paper aims to explore the issues of TRS, and the temporal factors that can improve the performance of CF technique. The overview paper [15] has covered the factorization approaches based on learning the factorization features and the global effect of temporal. The hypothesis of this review covers the lack of review papers for the TRS approaches and describes the important factors that effect CF performance.

The rest of this review is organized based on five sections: first, the definition of CF and the challenges of this technique, second, the current factorization approaches and their challenges, third, the temporal preferences and the current TRS approaches, fourth, the temporal challenges and their effects on the prediction performance of the CF technique, and finally discussion and conclusion.

2. **Collaborative Filtering.** CF is the most popular technique in RS because it requires the fewest computational resources. CF uses the rating scores of users to computes the similarity between any two users based on similarity metrics [5] to predict the users' interest in items. Several approaches are used to predict the missing rating scores and these approaches are divided into two categories: the first is learning approaches, which can be further categorized into supervised and unsupervised learning approaches. Supervised learning approaches include the fuzzy-genetic [2] which is used to learn the users features. Second, unsupervised learning approaches attempt to reduce the dimensionality of a rating matrix based on clustering strategies [16]. The challenges of CF can be summarized into data sparsity, cold start and data scalability. As a result of streaming the rating scores into the rating matrix, other challenges emerge including stability, outlier preferences and drift. These challenges are described as follows.

a) Data Sparsity. RS based on CF uses the rating scores of the users for items arranged in the rating matrix. Usually, the user tends to rate a small number of items, which leads to a high percentage of missing scores in the rating matrix. The CF extracts the behaviour of common users, which are shared in some items with the active user for predicting accurate items for the active user. Similarity evaluation between the common users and the active user will be impossible or unreliable if there is a high percentage of missing scores [17]. Several approaches are used to solve data sparsity [18-21], most of which attempt to propose new similarity metrics [22] and use the approach of k-neighbours [23] to predict the accurate items. The k-neighbours are the most similar users to the active user. However, most of these approaches do not exploit the features of temporality to improve the CF performance.

b) Cold Start. The cold start problem happens when it is impossible to create dependable recommendations because of a lack of initial ratings. There are three types of cold start: new users, new items and new community [3]. The new user is the most significant in RS, in which a user has a very small number of rating scores [5,24]. The cold start is addressed by several approaches such as the functional factorization [25], the content-based hybrid [6], and the probabilistic [24].

c) Scalability. The scalability is a result of information overload of users and items which increase the calculations complexity. Several approaches have attempted to improve the scalability such as the clustering [16], and distributed approaches [26].

d) Stability. The stability of the results for each prediction approach is very sensitive and several factors are produced overtime such as drift, and information overload [27].

e) Deviation. As a result of streaming the rating scores into the rating matrix in the memory, some rating scores deviate from accurate results [28], which leads to inaccurate predictions because of the low effects of these scores on the neighbours. When increasing the deviation rating scores (noise), some predictive approaches such as SVD cannot yield accurate predictions because the relationship between users and items is weak owing to the high percentage of unknown rating scores. This challenge is solved by the ensemble approach, which learns the accurate latent feedback of rating scores [29].

f) Drift. This challenge is a result of several factors that prompt the user to change his mood during the long term, such as new announcements for movies or products. Machine learning can extract the drift factor by training the set of weights, which helps extract the user behaviour and his interests during the suitable time. Several papers have attempted to solve the previous challenges, which are summarized in Table 1.

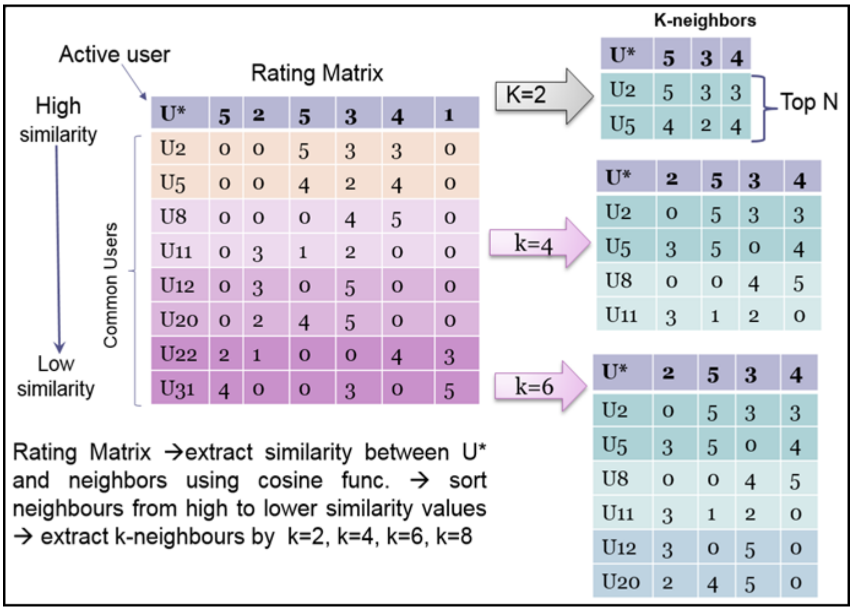
Table 1. Summary of the collaborative filtering approaches

Paper	Problems			Approaches					Features				
	Data sparsity	Scalability	Cold start	Similarity	Clustering	SVD	k-neighbours	Top-N	Weights	Users' similarity	Items' similarity	Latent feedback	Contents
Sarwar et al. [19]	√					√		√				√	
Sarwar et al. [20]	√	√		√			√	√		√	√		
Melville et al. [30]	√		√				√		√	√			√
Roh et al. [16]	√	√			√		√		√	√		√	√
Yu et al. [24]			√					√	√	√		√	√
Han et al. [26]		√					√		√	√			
Al-Shamri and Bharadwaj [2]	√	√					√		√	√		√	√
Huang and Gong [31]	√						√			√	√	√	
Ahn [32]			√	√						√	√		
Wang et al. [33]	√	√					√		√	√			√
Bobadilla et al. [34]	√			√			√			√			
Verbert et al. [21]	√			√					√	√		√	
Bobadilla et al. [35]	√						√		√	√			
Ning and Karypis [36]	√					√		√				√	
Mackey et al. [37]	√	√				√						√	
Bobadilla et al. [4]	√			√			√	√	√	√			
Bobadilla et al. [38]		√		√			√			√	√		
Tyagi and Bharadwaj [39]	√	√			√				√	√			
Ren et al. [23]	√	√		√	√		√	√	√	√			√
Ranjbar et al. [40]	√					√						√	
Li et al. [41]			√			√						√	√
Al-Hadi et al. [29]	√				√	√				√	√	√	
Total	17	9	4	7	4	6	12	6	11	17	5	11	7

Table 1 compared 22 papers through the problems of CF technique, the approaches, and the features that are used to learn the user's behaviours and item's behaviours. A check symbol (denoted by √) indicates that approaches or features are addressed in the paper stated in the table's row. Referring to Table 1, there are 17 approaches that have addressed the data sparsity problem, 9 approaches that have addressed data scalability and 4 approaches that have addressed the cold start problem. Most solutions for data sparsity focus on improving the similarity metrics [21,34,38] or divide the features of users based on the clustering approach [16,23,39] or the k-nearest neighbours [2].

A distributed hash table based on CF [26] is used to generate a scalable distributed RS with accurate prediction by mapping keys to profiles. The imputation-based MF method is used to mitigate the sparsity problem [40] by learning the accurate latent feedback of a user's preferences and minimizing the overfitting via two methods of convergence. Recently, the method of ensemble divide and conquer [29] solved the data sparsity problem by learning the accurate latent feedback of users and items according to the SVD algorithm. However, the imputation-based MF method and ensemble divide and conquer have weaknesses in terms of the behaviour drift of the users and the popularity decay of items over time. In addition, the factorization features of a user's preferences and the textual information of items are used to solve the cold start problem [41].

The approach of k-nearest neighbours (KNN) is the most popular for avoiding the data sparsity. Figure 1 shows an example of extracting the KNN from a rating matrix based on the sparsity issue. The shortcoming of KNN approach is the low coverage for the users which share the items with active user, and there are even lower numbers of ratings available for the neighbours of the active user [23]. The neighbors provide a weak rating prediction when the rating matrix is sparse. However, the neighbors can be integrated with other features (e.g., baseline, latent, temporal) to predict the sparse rating scores [42-45].



U*	5	2	5	3	4	1
U2	0	0	5	3	3	0
U5	0	0	4	2	4	0
U8	0	0	0	4	5	0
U11	0	3	1	2	0	0
U12	0	3	0	5	0	0
U20	0	2	4	5	0	0
U22	2	1	0	0	4	3
U31	4	0	0	3	0	5

Rating Matrix →extract similarity between U* and neighbors using cosine func. → sort neighbours from high to lower similarity values → extract k-neighbours by k=2, k=4, k=6, k=8

U*	5	3	4
U2	5	3	3
U5	4	2	4

U*	2	5	3	4
U2	0	5	3	3
U5	3	5	0	4
U8	0	0	4	5
U11	3	1	2	0

U*	2	5	3	4
U2	0	5	3	3
U5	3	5	0	4
U8	0	0	4	5
U11	3	1	2	0
U12	3	0	5	0
U20	2	4	5	0

FIGURE 1. Example of k-nearest neighbours based on sparsity issue

3. **Matrix Factorization.** Recently, MF has become a popular approach for CF [8,46,47] because it is one of the successful approaches that address the data sparsity and cold start problems [25]. MF integrates three approaches: SVD, baseline and latent factors. MF is typically used to uncover latent semantic factors and is able to handle the scalability issue quite well. The MF will be processed according to a series of steps involving SVD factors, baseline factors and latent factors. MF simplifies the rating scores by characterizing the features of both users and items to extract latent factors from a matrix of rating scores. SVD acts a simple MF approach that extracts the latent feedback between users and items as shown in Equation (1),

$$\widehat{r}_{ui} = p_u q_i^T, \tag{1}$$

where $\widehat{r}_{ui}$ represents the prediction value of the missing preference, and p_u and q_i^T represent the latent feedback of users and the transpose latent feedback of items, respectively. The second approach of MF is the baseline, which represents the baseline features of users and items. Equation (2) represents the prediction approach using the variants of the baseline approach,

$$\widehat{r}_{ui} \leftarrow b_{ui} = \mu + b_u + b_i, \tag{2}$$

where μ represents the global average of rating scores, and b_u and b_i represent the observed deviations of user u and item i, respectively. The MF approach is a combination of the baseline features of users and items and the interactive feedback of SVD as shown in Equation (3), which produces the predicted rating scores in the rating matrix.

$$\widehat{r}_{ui} = \mu + b_u + b_i + p_u q_i^T. \tag{3}$$

The factorization approaches are integrated with several approaches such as the neighbourhood. For example, the neighbour-based approach [43] integrates the factors of the baseline with the distance between the rating scores and the baseline features of the neighbours who provide the rating scores for each item as shown in Equation (4),

$$\widehat{r}_{ui} = b_{ui} + \left(\sum\nolimits_{x \in N_i} sim_x (r_{xi} - b_{xi}) \Big/ \sum\nolimits_{x \in N_i} sim_x\right), \tag{4}$$

where N is the set of users who provide item i by rating scores, and sim_x is the similarity value between user x and the active user. However, most factorization approaches suffer from overfitting in the predicted rating scores. The overfitting in the predicted rating scores means a few of the predicted values are bigger than the range scale of the rating scores [29], e.g., the range scale of MovieLens dataset [0-5] when the predicted rating scores are such as $\{5.3; 6.2; 5.5\}$. The main challenge of prediction based on MF approaches is minimizing overfitting in the latent space by reducing the error. Therefore, several factorization approaches use the stochastic gradient descent algorithm [43,48,49] for the observed ratings such as in Equation (5),

$$\min \sum_{(u,i) \in k} \left(r_{ui} - p_u q_i^T\right) + \lambda \left(\|p_u\|^2 + \left\|q_i^T\right\|^2\right), \tag{5}$$

where k denotes the training set of users and items chosen for training the global rating matrix. The output of Equation (5) represents the regularized squared error of all known rating scores r_{ui} to optimize the latent feedback of users and items based on two factors p_u and q_i^T. The factors of both $\|p_u\|$, $\left\|q_i^T\right\|$ indicate the standard Euclidean norm of the interacting factors of a user u, $\|p_u\|^2 = \sum\limits_{u=1}^{m} (p_u)^2$, and an item i, $\left\|q_i^T\right\|^2 = \sum\limits_{i=1}^{n} \left(q_i^T\right)^2$, where m and n are the dimensions of the global rating matrix based on the training set. Several factorization approaches use a control parameter λ for normalizing the overfitting in the predicted rating scores based on the regularized squared error [13,43,50]. Clearly, Equation (5) explores three latent factors: the norm of latent feedback of users, the norm of latent feedback of items and the interaction between users and items. The factorization approach [13] uses the baseline features of users and items in Equation (6) for learning the accurate factorization features and λ for minimizing the regularized squared error of the known rating scores,

$$\min \sum_{(u,i) \in k} \left(r_{ui} - \mu - b_u - b_i - p_u q_i^T\right) + \lambda \left(b_u^2 + b_i^2 + \|p_u\|^2 + \left\|q_i^T\right\|^2\right). \tag{6}$$

The parameters of the approach described in Equations (5) and (6) contain two parts. The first part is used to find the parameters that fit the given rating scores, and the second part acts as the regularization term for avoiding overfitting by correcting the scales of the parameters. The MF factors in Equations (1), (2), (3) and (4) are used as prediction approaches for the missing rating scores, but those in Equations (5) and (6) are used to compute the regularized squared error to determine the difference between the observed rating scores and the predicted rating scores.

The neighbourhood based on factorization approach [44] is used for exploring the past transactions and improving the prediction accuracy of the CF technique. This approach explored the explicit rating scores $R(u)$ and latent rating scores $N(u)$ within the same set of neighbors who provide each item by rating score as shown in Equation (7),

$$\widehat{r}_{ui} = b_u + b_i + \left(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) \, w_j + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) q_i^T, \tag{7}$$

where q_i^T is the transpose latent factor of item i, y_j is the other latent factor of each item rated by a user u within the set of neighbours $N(u)$ and w_j is the constant number. The neighborhood based factorization approach [44] has improved by integrating the clustering and asymmetric factors to the factorization and neighborhood factors [47]. The clustering based factorization approach [47] is used for classifying the users and items and for tracking how users in each category rate the other categories containing the items. These approaches refer to the ability of MF with several approaches for extracting the features of users. The factorization approaches [43,44,50] combine between the latent feedback of MF and the features of the neighbourhood. The clustering based factorization approach [47] has used the content information to categorize the explicit feedback which is highly scalable for the huge datasets such as MovieLens 10MB and Netflix Prize.

Recently, temporal RS based on MF and latent feedback has become a promising direction to extract the features of users during a period of time divided into two terms: long term and short term. Consequently, the latent factors of MF based on temporal terms are very important factors to extract the latent feedback of users and items during the suitable time. For instance, some users like to watch television series that extends for a long time, e.g., four months. As an additional example of long term, some users – specifically, young users – like movies based on action, war or sports for a long time more than other genres of movies.

The long term changes the baseline features of users and items as described in the next section. The short-term is more important than long-term because there is a specific time for watching the specific movies such as the matches time in FIFA World Cup. The short-term and long-term preferences are extracted as temporal weights. The temporal weights are integrated with other factorization factors to predict the sparse rating scores based on the user's interest over time which improves the prediction accuracy of the recommendations. The researchers modelled the long-term and short-term for improving the quality of global RS based on MF and other approaches which are described in the next sections.

4. **Temporal Recommendation System.**

The prediction approaches summarized in Table 1 did not focus on drifting of the popularity of items and the tastes of users over time. Temporal RS is designed to recommend items to users at a suitable time; time is an essential factor in making the final decision and is used in many approaches to obtain accurate predictions. Consequently, the temporal is a promising direction to improve the quality of RS by tracking the interests of users over time. There are several approaches available for temporal RS such as the time weight CF approach [12], which improves the quality of RS based on increasing the weights of recent rating scores. Currently, factorization approaches are the identical successful approaches for RS. The temporal period effects on the preference of users are long-term preferences and short-term preferences.

Table 2 shows the temporal approaches which integrate the temporal and the factorization factors [45,46] to improve the performance of RS. Most of these approaches use the interest of users as an effective factor to change the predicted values over time. Therefore, several temporal factors can affect the quality of products over the long and short terms, which affects the popularity of products, such as the seasons of the year and the expiration

Table 2. Summary of the temporal RS approaches

Papers	Approaches					Temporal factors					Users factors				
	Clustering	k-neighbours	SVD	Baseline	Latent Factor	Time weight	Drift	Decay	Short-term	Long-term	User interest	Satisfaction	Uses location	User Features	Tracking user
Ding and Li [12]	√					√			√		√			√	
Blanco-Fernández et al. [59]		√									√				
Lathia et al. [51]	√	√	√	√					√		√				
Koren [13]			√	√	√		√	√		√	√				
Božidar et al. [52]									√			√	√	√	
Yin et al. [53]				√		√	√		√	√	√	√			√
Saha and Mahanti [60]						√			√		√			√	
Yang et al. [45]			√	√	√				√		√				
Adibi and Ladani [55]	√					√			√	√	√				
Yuan et al. [56]						√			√		√		√	√	
Campos et al. [54]		√									√	√	√		√
Ye and Eskenazi [46]		√	√	√	√	√	√		√	√	√				
Lin et al. [57]			√		√					√					
Total	3	4	5	5	4	6	3	1	9	5	11	3	3	4	2

date of the products. The temporal diversity of recommendations is an important factor [51] that affects the quality of RS and gives the feedback of the diverse tastes of users over time. Furthermore, time decay is an important factor for tracking the popularity of items or products. The temporal dynamics [13] used the drift and the decay factors for tracking the changes of the global behaviour of users and items but did not use the personal behaviour. The satisfaction factor integrated with the short-term factor to track the interest of users [52,53]. The interest of users is also analyzed during temporal periods via several rules [54]. The MF approaches use clustering and the weights of time [55] to improve the performance of a rating matrix. The point-of-interest approach [56] covers the activities of users and the location influences over the short term. The online learning approach [57] is used to effectively update the user's interest online by the sequence of rating scores and considers the latent factors of the low rank matrix. This approach attempts to reduce the time complexity and learn the accurate latent factors. However, the online learning filtering approach does not cover the drift and the time decay factors, which lower the quality of predictions.

Table 2 shows the k-neighbours as an important approach which is used in a few of temporal approaches to learn the short-term preferences [46,58] but with limited coverage of these features and high cost. Table 2 is focused on the interest factor which related to other temporal factors which are summarized in Table 3.

Table 3 shows the temporal factors related to the user's interest such as short-term, time weight and long-term as three significant factors used to learn the user's interest over time. In addition, Table 3 shows the significant temporal factors which are explored by several temporal approaches and shows the gap of these approaches. Most approaches have focused on three factors which are time weight, short-term and long-term, but few of approaches have focused on the factors of drift and decay which may lower the prediction

Table 3. The relation of users' interest with other temporal factors

Temporal factor	Frequency	Interests relations-based researches
Time weight	6	[12,46,53,55,56,60]
Drift	3	[13,46,53]
Decay	1	[13]
Short-term	8	[12,45,46,51,53,55,56,60]
Long-term	4	[13,46,53,55]

accuracy of CF. Therefore, this research focuses on exploring the drift and the decay factors and their influences on the quality of temporal preferences and then on the prediction performance of the CF technique. The temporal approaches [13,46,53,55] have covered the long-term as a global factor, but they have not covered the personality influences of long-term preferences which maybe lower the prediction performance of the CF technique. Furthermore, there are 8 temporal approaches in Table 3 which use the short-term preferences for learning the user's interest but just 3 of these approaches explore the drift factor and 5 approaches did not learn the drift factor which can be considered as gap of the 5 approaches. These relations and gaps can be covered in this research to provide the accurate predictions using the CF technique. In the next subsections, several details about the important temporal approaches which have improved the quality of RS.

4.1. **Long-term preferences.** The long-term preference feature has a global temporal effect on all features of both users and items, whereas the short term gives a local temporal effect. The long term is defined by the user baseline based on Equation (8) [46],

$$b^t_{ui} = \mu + b_u \frac{e - t_{ui}}{e - s} + b_u \frac{t_{ui} - s}{e - s} + b_i. \tag{8}$$

The time vectors in Equation (8) use the first time s, the last time e and the current time t_{ui} in which item i is rated by user u. The temporal features in this approach will be extracted according to the taste of the user during the long term because the user determines the taste of each item, but the item cannot determine the interest of the user.

4.2. **Short-term preferences.** The conventional approach of CF did not include temporal features to perform recommendations. Session-based CF is introduced to overcome the incorporation of context with CF, in which the temporal context is used with CF by using session profiles for producing recommendations [61]. Indeed, the session has been commonly used and discovered in Web mining research, e.g., for analysis of movie preferences that are captured over time. There are temporal properties for any session that could be useful for modelling latent feedback such as user mood and drift of users over time. There are several standard variables for session-based CF – namely, time of day, day of the week, day of the month, month of the year and movie diversity; all are introduced to monitor the temporal aspects in RS.

The short-session information is captured by extracting relationships of the neighbourhood with other users [46]. The shortcoming of this approach is that it overlooked the other latent factors such as the user mood factor. The time interval can be defined as a session using a specific period of time, e.g., month, week, or day [45,46]. The short-term approach shows that the performance of RS based on short-term preferences is better than the performance of long-term preferences [45]. The short-term approach [45] uses the neighbours who provide the rating scores for items during the session or the short term to predict the missing rating scores as shown in Equation (9),

$$\widehat{r}_{ui} = b_u + b_i + \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j + |N(u,t)|^{-\frac{1}{2}} \sum_{j \in N(u,t)} \alpha_j \right) q_i^T, \tag{9}$$

where $N(u)$ and $N(u,t)$ are the sets of neighbours who rate each item during the whole time and the short term t, respectively; y_j is the latent feedback of the set $N(u)$; and α_j is the latent feedback of the set $N(u,t)$. However, the overfitting in the predicted scores is the limitation of several factorization approaches, which must minimize these values by a controlling weight [14,25]. The short-term approach [46] is a good example of short term that uses the baseline variants of $N(u,t)$ instead of the latent feedback as shown in Equation (10),

$$\widehat{r}_{ui}(t) = b_{ui} + |N(u,t)|^{-\frac{1}{2}} \sum_{j \in N(u,t)} [(r_{uj} - b_{uj}) w_{ij}] + p_u q_i^T. \tag{10}$$

4.3. **Integration of long- and short-term preferences.** The relation between the members of RS – either users or items – increases over a long term, and some of these relations are more suitable for a specific period or short term. The relation among RS members can be improved based on integrating between short-term and long-term using Equation (11) [46],

$$\widehat{r}_{ui}(t) = b_{ui}^t + |N(u,t)|^{-\frac{1}{2}} \sum_{j \in N(u,t)} [(r_{uj} - b_{uj}) w_{ij}] + p_u q_i^T, \tag{11}$$

where b_{ui}^t is the long-term features learned by Equation (8). The item taxonomy and the temporal dynamics are integrated to solve the problems of recommendation with stage, stage identification, and item recommendation in e-commerce [62]. Consequently, this approach formalizes the long-term behaviour of users based on the item taxonomy and then specifies the accurate phase of the user. This approach uses CF to provide a personalized item list of the active user through other similar users during the same stage. However, the temporal dynamics approach [62] solves the problem of recommendation with a stage in e-commerce based on taxonomic knowledge, which is suitable for private datasets but unsuitable for others.

Most temporal RS approaches in this review are designed based on a trade-off between long- and short-term interests. To illustrate this direction, the approach of temporal RS over the tweet stream [63] allows users to post tweets (a message sent using Twitter), and the users can receive recommendations of topics (hashtags) based on their real-time interests. This approach uses real time to extract the interest of users based on the latest preferences.

5. **Challenges of the Temporal Recommender System.** There are several challenges for RS based on the type of recommendations. The main types of recommendations are general recommendations and personal recommendations. Several researchers focus on general recommendations based on the rating scores and have used several approaches to solve the main challenges of general recommendations such as missing rating scores in the rating matrix including data sparsity and cold start. The general rating matrix suffers from the missing rating scores, which are solved by using one of the famous prediction approaches such as the factorization approaches or the k-nearest neighbours approach.

These approaches extract the behaviour of users and items based on several complexity approaches that contain several factors that compute based on static functions. The second type is personalized recommendations by the CF technique, and this type has been affected by the problems of data sparsity and cold start. The factorization approaches have been used for predicting the sparse rating scores in the rating matrix [13,43,44,46]. There are a few temporal-based factorization approaches [13,46,62] that have solved the data sparsity problem and other subproblems such as drift in user interest or time decay in item popularity. Table 4 has described the important differences among these approaches based on the definition of the temporal periods in each approach and the differences among these periods.

Table 4. Comparison between the temporal features

Paper	Long-term	Short-term	Disadvantage
Koren [13]	Designate a set of time points for controlling and compute the distance between last time and the time of rate to effect on user base vectors.	Divide the period of time into static periods of time for effecting on the baseline vector of items.	It is difficult to use static time interval because the time is increasing continuously.
Hong et al. [62]	The whole time is used as long-term profile to identify the current recommendation stage.	The whole time is divided into static periods of time as short-term.	
Yang et al. [45]		The factorization features of neighbours that provide rating scores by users for each item during time interval are integrated with the other factorization features to learn the short-term preferences.	The short-term preferences based on neighbours have low efficiency in the prediction because the neighbors feedback is the small values due to a high percentage of missing data in the rating matrix.
Ye and Eskenazi [46]	First and last time are used to compute 2 time vectors as global effected vectors on user base.		

Table 4 shows the important existing works in this review based on temporal preferences. The first temporal preferences are defined by dividing the timeline into a static number of slices [13,64], but the time is changed over time which lowers the quality of this temporal slices. The second temporal preferences use the end time and the current time vectors to define the long-term preferences in the temporal dynamics approach [13]. The long-term preferences are defined also by [46] who used three temporal vectors which are the start, the end time of the whole items in the rating matrix, and the current time for each item i rated by user u. The short-term preferences are defined by the latent feedback of neighbors during a session [45] and also by the baseline values of neighbors during a session [46]. However, both short-term preferences by latent [45] or baseline values [46] have weakness because of the high percentage of sparsity in the rating matrix. This is because sparse matrix lowers the ability of utilizing quality feedback of neighboring users when predicting the suitable rating for the active user.

The temporal based preferences such as the long-term and short-term preferences [46] are used to extract the temporal influences on the user's interest and the item's popularity over time. However, both the long-term and the short-term preferences [13,45,46] have limitation due to the issue of drifting users interest and the decaying items popularity which reduce the performance of temporal preferences in improving the accuracy prediction of the CF technique. This research focuses on finding the accurate solutions for these gaps. The decay and the drift are the significant gaps in temporal based recommendation system which are described in the following two subsections.

5.1. **Time decay.** Time is an important factor that affects the predicted values, and the time decay vector is used for tracking the popularity of items. The decay vector is extracted by using an exponential decay rate. Equation (12) [65] is used for measuring the weight of each observed rating score r_{ui},

$$f_{ui}^{\beta}(t) = e^{-\beta(t-t_{ui})}, \tag{12}$$

where t is the current time, t_{ui} is the time that user u provides item i by a rating score, and β is a static number that takes either 0 or 1 to control the decay rate for any prediction. The online learning approach [65] uses the temporal information and the time decay rate for adjusting the cosine similarity function between any two items and obtain the predicted values based on the nearest neighbours of each item. The same vector rate is used in the temporal dynamics approach [13] within the neighbours of the active user, which rate some items during a specific time to affect users' base factor or prediction values.

5.2. **Interest drifting.** Rating scores are good features that help understand the user mood, but rating scores of users on items tend to drift over time [8] for several reasons that prompt the user to change his mood during the long term such as the new announcements for movies or products. Figure 2 shows an example of the drift behaviours where this figure summarizes the activities of users on one day about the rating profile of MovieLens dataset.

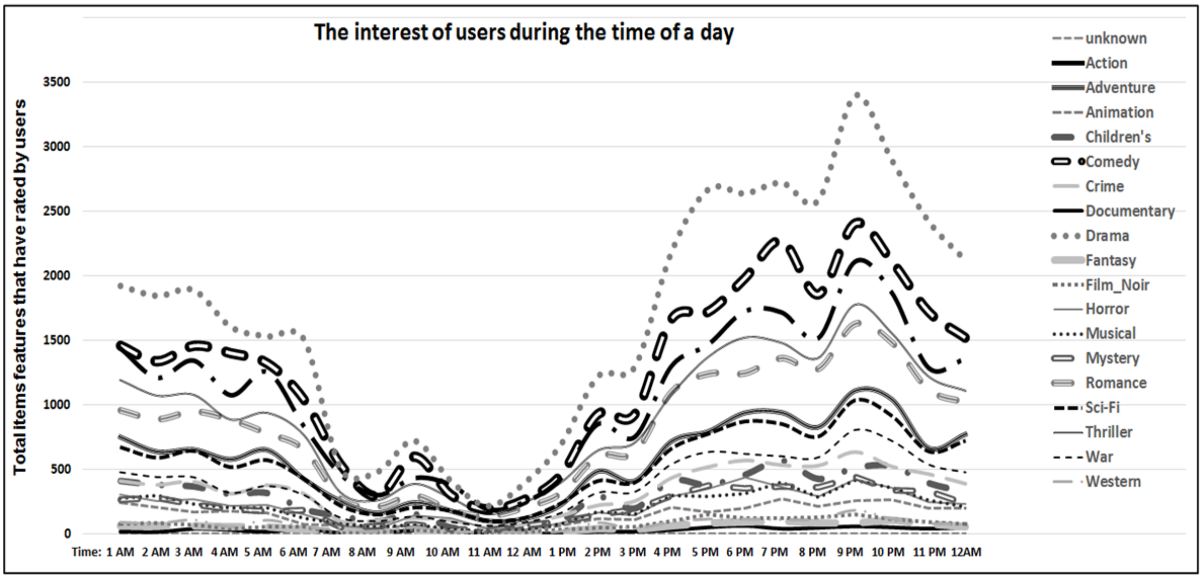


Figure 2. The users interest in the movies based on genre during a day

In Figure 2, the behaviour of users drifts from hour to the next hour for each genre of item/movie. For instance, the drama is more popular in the beginning of the day followed by comedy and action later that day. Some preferences of some genres are low during the morning and it increases during a day and others are high in the morning and it decreases during a day. The drifting parameter represents an important vector for getting the accurate predictions using each prediction approach. For an instance, the recommendation based on interest-drifting approach [9] is proposed as a solution for addressing the accuracy and decay limitation. This approach integrates the content-based with CF for extracting the time decay and the drift vectors and merge both them within the prediction function.

A few RS approaches have attempted to track global drifting through the whole dataset [53]. The drifting of user behaviour is a global vector that acts a new challenge for personal RS [46], and it is determined by the temporal drift vector in different directions. The global drifting vector of users is used in the temporal dynamics [13] based on a smooth function for the user base, which meshes well with a gradual concept drift. Furthermore, the time of item preferences in the temporal dynamics [13] is divided into a fixed number of segments to extract the taste for the items. However, the taste for the items will be specified by the taste of the user, and we must learn how we can track the user's interests during a specific time such as summer and extract the drift of user interest during this time.

In social network websites such as MovieLens and Netflix, the number of users and items increases every day along with the rating scores of users. Several approaches of RS focus on normalizing the sparse values and use k-nearest neighbour to achieve better accuracy, whereas a few approaches attempt to track user interest and the drift vector over time. Moreover, the dynamic approaches learn latent feedback of the rating matrix using several factorization approaches. There is a relationship between the rating scores of the first session and the rating scores of the second session in the rating matrix based on the user's mood and the item's popularity. Figure 3 summarizes the significant factors that affect the prediction accuracy of the CF. The factorization approaches and temporal-based factorization approaches use statistical calculations [13] such as global and local averages and also the vectors of the long term and short term [45,46]. There is a relationship between the factors in Figure 3, such as the user mood and cyclical factors, in which the mood of users is affected according to the nature of each season in the year and other factors such as the item's popularity.

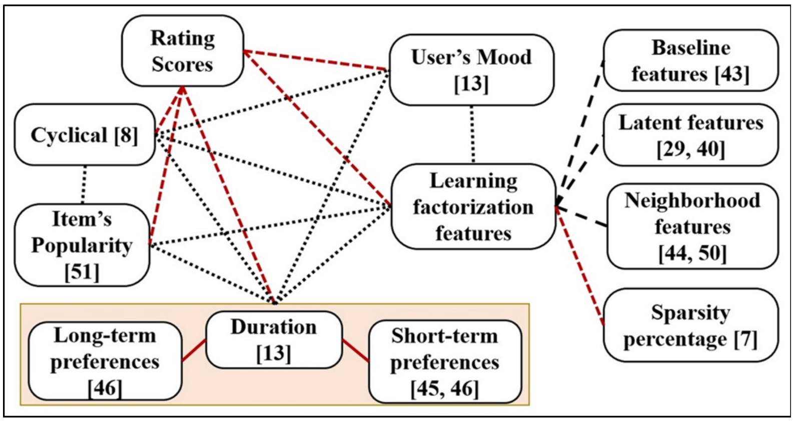


Figure 3. Factors that may affect the prediction performance of the CF

6. **Discussion and Conclusion.** The previous sections have described the existing issues, problems and approaches of both RS and TRS. In summary, the achievements of TRS-based MF are not satisfactory, specifically for the personal interaction between long-term and short-term preferences. The clustering is an important approach for RS that reduces the dimensionality of the rating matrix. However, limited works are available

on clustering approaches that use a rating matrix based on CF. MF is a very important approach for developing the performance of RS. Several approaches have focused on extracting the features of MF based on latent factors without looking at user behaviour, which drifts over time.

The temporal dynamics [13] improved the MF approach, which reflects the modifications of the users' features and the items' features over time. The temporal approaches use several temporal vectors manually to learn the drift in the user's behaviour, which increases the time complexity and lowers the quality of prediction when other reasons are in effect. The drift in the user's preferences and the time decay of the item's popularity are the main challenges in the temporal-based factorization approaches, whose performance in automatic learning by a learning algorithm such as the genetic algorithm or the particle swarm algorithm is poor.

The rapid growth of information on RS has caused many challenges related to the information overload issue owing to the highly sparse matrix. Therefore, a suitable recommendation based on CF can be provided by focusing on the important factors related to the temporal preferences, e.g., mood drifting and time decay. The preferences of both short term and long term can be integrated using the factorization approaches. This review paper attempts to cover the important temporal issues based on MF and latent feedback of a user rating matrix. In our future work, we will focus on learning the factors of drift, decay and overfitting to improve the prediction quality of the CF technique.

REFERENCES

[1] F. Ortega, J. L. Sánchez, J. Bobadilla and A. Gutiérrez, Improving collaborative filtering-based recommender systems results using Pareto dominance, *Information Sciences*, vol.239, no.4, pp.50-61, 2013.

[2] M. Y. H. Al-Shamri and K. K. Bharadwaj, Fuzzy-genetic approach to recommender systems based on a novel hybrid user model, *Expert Syst. Appl.*, vol.35, pp.1386-1399, 2008.

[3] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, Recommender systems survey, *Knowledge-Based Syst.*, vol.46, pp.109-132, 2013.

[4] J. Bobadilla, A. Hernando, F. Ortega and A. Gutiérrez, Collaborative filtering based on significances, *Information Sciences*, vol.185, no.1, pp.1-17, 2012.

[5] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowledge & Data Engineering*, vol.17, no.6, pp.734-749, 2005.

[6] S.-S. Weng and C.-H. Lee, Integration of content-based approach and hybrid collaborative filtering for movie recommendation, *Bus. Inf.*, pp.571-588, 2013.

[7] T. Ma, X. Suo, J. Zhou, M. Tang, D. Guan, Y. Tian, A. Al-Dhelaan and M. Al-Rodhaan, Augmenting matrix factorization technique with the combination of tags and genres, *Phys. A Stat. Mech. Its Appl.*, vol.461, pp.101-116, 2016.

[8] N. Koenigstein, G. Dror and Y. Koren, Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy, *Proc. of the 5th ACM Conf. Recomm. Syst.*, pp.165-172, 2011.

[9] S. Ma, X. Li, Y. Ding and M. E. Orlowska, A recommender system with interest-drifting, *International Conference on Web Information Systems Engineering*, pp.633-642, 2007.

[10] H. W. Tung and V. W. Soo, A personalized restaurant recommender agent for mobile E-service, *IEEE International Conference on E-Technology*, pp.259-262, 2004.

[11] B. Lika, K. Kolomvatsos and S. Hadjiefthymiades, Facing the cold start problem in recommender systems, *Expert Syst. Appl.*, vol.41, no.4, pp.2065-2073, 2014.

[12] Y. Ding and X. Li, Time weight collaborative filtering, *Proc. of the 14th ACM Int. Conf. Inf. Knowl. Manag.*, pp.485-492, 2005.

[13] Y. Koren, Collaborative filtering with temporal dynamics, *Commun. ACM*, vol.53, no.4, pp.89-97, 2010.

[14] J. Z. Sun, K. R. Varshney and K. Subbian, Dynamic matrix factorization: A state space approach, *IEEE International Conference on Acoustics*, vol.22, no.10, pp.1897-1900, 2012.

[15] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer (Long. Beach. Calif.)*, vol.42, no.8, pp.42-49, 2009.

[16] T. H. Roh, K. J. Oh and I. Han, The collaborative filtering recommendation based on SOM cluster-indexing CBR, *Expert Syst. Appl.*, vol.25, no.3, pp.413-423, 2003.

[17] L. Sharma and A. Gera, A survey of recommendation system: Research, *Int. J. Eng. Trends Technol.*, vol.4, pp.1989-1992, 2013.

[18] J. Bobadilla, A. Hernando, F. Ortega and J. Bernal, A framework for collaborative filtering recommender systems, *Expert Systems with Applications: An International Journal*, vol.38, no.12, pp.14609-14623, 2011.

[19] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, *TR 00-043 Application of Dimensionality Reduction in Recommender System – A Case Study*, no.TR-00-043, 2000.

[20] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation, *Proc. of the 10th Int. Conf. World Wide Web*, pp.285-295, 2001.

[21] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, S. Birlinghoven, S. Augustin and E. Duval, Dataset-driven research for improving recommender systems for learning, *Proc. of the 1st Int. Conf. Learn. Anal. Knowl.*, pp.44-53, 2011.

[22] B. K. Patra, R. Launonen, V. Ollikainen and S. Nandi, A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, *Knowledge-Based Syst.*, vol.82, pp.163-177, 2015.

[23] Y. Ren, G. Li, J. Zhang and W. Zhou, Lazy collaborative filtering for data sets with missing values, *IEEE Trans. Cybern.*, vol.43, no.6, pp.1822-1834, 2013.

[24] K. Yu, A. Schwaighofer, V. Tresp, X. Xu and H. Kriegel, Probabilistic memory-based collaborative filtering, *IEEE Trans. Knowl. Data Eng.*, vol.16, no.1, pp.56-69, 2004.

[25] K. Zhou, S. Yang and H. Zha, Functional matrix factorizations for cold-start recommendation, *Proc. of the 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp.315-324, 2011.

[26] P. Han, B. Xie, F. Yang and R. Shen, A scalable P2P recommender system based on distributed collaborative filtering, *Expert Syst. Appl.*, vol.27, no.2, pp.203-210, 2004.

[27] J. Vinagre, Time-aware collaborative fitering: A review, *Doctoral Symposium in Informatics Engineering*, p.43, 2012.

[28] H. Cui, G. Ruan, J. Xue, R. Xie, L. Wang and X. Feng, A collaborative divide-and-conquer K-means clustering algorithm for processing large data, *Proc. of the 11th ACM Conf. Comput. Front.*, p.20, 2014.

[29] I. A. A. Al-Hadi, N. M. Sharef, N. Sulaiman and N. Mustapha, Ensemble divide and conquer approach to solve the rating scores' deviation in recommendation system, *J. Comput. Sci. Sci. Publ.*, 2016.

[30] P. Melville, R. J. Mooney and R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, *AAAI/IAAI*, pp.187-192, 2002.

[31] C. Huang and S. Gong, Employing rough set theory to alleviate the sparsity issue in recommender system, *International Conference on Machine Learning & Cybernetics*, vol.3, pp.1610-1614, 2008.

[32] H. J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, *Information Sciences*, vol.178, no.1, pp.37-51, 2008.

[33] Q. Wang, X. Yuan and M. Sun, Collaborative filtering recommendation algorithm based on hybrid user model, *The 7th Int. Conf. Fuzzy Syst. Knowl. Discov.*, vol.4, pp.1985-1990, 2010.

[34] J. Bobadilla, F. Serradilla and J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, *Knowledge-Based Syst.*, vol.23, no.6, pp.520-528, 2010.

[35] J. Bobadilla, F. Ortega, A. Hernando and J. Alcalá, Improving collaborative filtering recommender system results and performance using genetic algorithms, *Knowledge-Based Syst.*, vol.24, no.8, pp.1310-1316, 2011.

[36] X. Ning and G. Karypis, SLIM: Sparse linear methods for top-N recommender systems, *The 11th IEEE International Conference on Data Mining*, Vancouver, BC, Canada, pp.497-506, 2011.

[37] L. Mackey, A. Talwalkar and M. Jordan, Divide-and-conquer matrix factorization, *Adv. Neural Inf. Process. Syst.*, pp.1134-1142, 2011.

[38] J. Bobadilla, F. Ortega, A. Hernando and Á. Arroyo, A balanced memory-based collaborative filtering similarity measure, *Int. J. Intell. Syst.*, vol.27, no.10, pp.939-946, 2012.

[39] S. Tyagi and K. K. Bharadwaj, A hybrid recommender system using rule-based and case-based reasoning, *Int. J. Inf. Electron. Eng.*, vol.2, no.4, pp.586-590, 2012.

[40] M. Ranjbar, P. Moradi, M. Azami and M. Jalili, An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems, *Eng. Appl. Artif. Intell.*, vol.46, pp.58-66, 2015.

[41] F. Li, G. Xu and L. Cao, Two-level matrix factorization for recommender systems, *Neural Comput. Appl.*, pp.1-12, 2015.

[42] N. Lathia, S. Hailes and L. Capra, Temporal collaborative filtering with adaptive neighbourhoods, *Proc. of the 32nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, p.796, 2009.

[43] R. M. Bell and Y. Koren, Lessons from the Netflix prize challenge, *ACM SIGKDD Explor. Newsl.*, vol.9, no.2, pp.75-79, 2007.

[44] Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, *ACM Trans. Knowledge Discovery from Data (TKDD)*, vol.4, no.1, pp.1-24, 2010.

[45] D. Yang, T. Chen, W. Zhang and Y. Yu, Collaborative filtering with short term preferences mining, *Proc. of the 35th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, no.2, p.1043, 2012.

[46] F. Ye and J. Eskenazi, Feature-based matrix factorization via long- and short-term interaction, *Knowl. Eng. Manag.*, pp.473-484, 2014.

[47] N. Mirbakhsh and C. X. Ling, Clustering-based factorized collaborative filtering, *Proc. of the 7th ACM Conf. Recomm. Syst.*, pp.315-318, 2013.

[48] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, *Proc. of KDD Cup Work.*, vol.2007, pp.39-42, 2007.

[49] G. Takács, I. Pilászy, B. Németh and D. Tikk, Major components of the gravity recommendation system, *ACM SIGKDD Explor. Newsl.*, vol.9, no.2, p.80, 2007.

[50] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, *Proc. of the 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp.426-434, 2008.

[51] N. Lathia, S. Hailes, L. Capra and X. Amatriain, Temporal diversity in recommender systems, *Proc. of the 33rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp.210-217, 2010.

[52] K. Božidar, O. Dijana and B. Nina, Temporal recommender systems, *Proc. of the 10th WSEAS Int. Conf. Appl. Comput. Appl. Comput. Sci.*, pp.248-253, 2011.

[53] D. Yin, L. Hong, Z. Xue and B. D. Davison, Temporal dynamics of user interests in tagging systems, *Proc. of the 25th AAAI Conf. Artif. Intell.*, pp.1279-1285, 2011.

[54] 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, nos.1-2, pp.67-119, 2014.

[55] P. Adibi and B. T. Ladani, A collaborative filtering recommender system based on user's time pattern activity, *The 5th Conf. Inf. Knowl. Technol.*, pp.252-257, 2013.

[56] Q. Yuan, G. Cong, Z. Ma, A. Sun and N. M. Thalmann, Time-aware point-of-interest recommendation, *Proc. of the 36th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp.363-372, 2013.

[57] F. Lin, X. Zhou, W. H. Zeng, F. Lin, X. Zhou and W. Zeng, Sparse online learning for collaborative filtering, *Int. J. Comput. Commun. Control Issn.*, vol.11, no.2, pp.248-258, 2016.

[58] H. Yang, I. King and M. R. Lyu, Online learning for collaborative filtering, *International Joint Conference on Neural Networks*, pp.1-8, 2012.

[59] Y. Blanco-Fernández, M. López-Nores, J. J. Pazos-Arias, A. Gil-Solla and M. Ramos-Cabrer, Personalizing e-commerce by semantics-enhanced strategies and time-aware recommendations, *The 3rd International Workshop on Semantic Media Adaptation & Personalization*, pp.193-198, 2008.

[60] S. Saha and A. Mahanti, Collaborative preference elicitation based on dynamic peer recommendations, *The 2nd Int. Conf. Adv. Collab. Networks, Syst. Appl. Collab.*, pp.88-94, 2012.

[61] S. E. Park, S. Lee and S. G. Lee, Session-based collaborative filtering for predicting the next song, *Proc. of the 1st ACIS/JNU Int. Conf. Comput. Networks, Syst. Ind. Eng.*, pp.353-358, 2011.

[62] W. Hong, L. Li and T. Li, Product recommendation with temporal dynamics, *Expert Syst. Appl.*, vol.39, no.16, pp.12398-12406, 2012.

[63] C. Chen, H. Yin, J. Yao and B. Cui, TeRec: A temporal recommender system over tweet stream, *Proc. of VLDB Endow.*, vol.6, no.12, pp.1254-1257, 2013.

[64] L. Zheng, L. Li, W. Hong and T. Li, PENETRATE: Personalized news recommendation using ensemble hierarchical clustering, *Expert Syst. Appl.*, vol.40, no.6, pp.2127-2136, 2013.

[65] N. N. Liu, M. Zhao, E. Xiang and Q. Yang, Online evolutionary collaborative filtering, *Proc. of the 4th ACM Conf. Recomm. Syst.*, pp.95-102, 2010.