Information Processing and Management 56 (2019) 526–540

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

An impact of time and item influencer in collaborative filtering recommendations using graph-based model

Maryam Khanian Najafabadi[a,*], Azlinah Mohamed[a], Choo Wou Onn[b]

[a] Advanced Analytics Engineering Centre, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia
[b] Faculty of Information Technology and Sciences, Inti International University, Malaysia

A R T I C L E I N F O

Keywords:
Hybrid recommender system
Collaborative filtering
Graph modelling
Items influencer
Priorities in user' preferences

A B S T R A C T

Recommender Systems deal with the issue of overloading information by retrieving the most relevant sources in the wide range of web services. They help users by predicting their interests in many domains like e-government, social networks, e-commerce and entertainment. Collaborative Filtering (CF) is the most promising technique used in recommender systems to give suggestions based on liked-mind users' preferences. Despite the widespread use of CF in providing personalized recommendation, this technique has problems including cold start, data sparsity and gray sheep. Eventually, these problems lead to the deterioration of the efficiency of CF. Most existing recommendation methods have been proposed to overcome the problems of CF. However, they fail to suggest the top-n recommendations based on the sequencing of the users' priorities. In this research, to overcome the shortcomings of CF and current recommendation methods in ranking preference dataset, we have used a new graph-based structure to model the users' priorities and capture the association between users and items. Users' profiles are created based on their past and current interest. This is done because their interest can change with time. Our proposed algorithm keeps the preferred items of active user at the beginning of the recommendation list. This means these items come under top-n recommendations, which results in satisfaction among users. The experimental results demonstrate that our algorithm archives the significant improvement in comparison with CF and other proposed recommendation methods in terms of recall, precision, f-measure and MAP metrics using two benchmark datasets including MovieLens and Superstore.

1. Introduction

The emergence of recommender systems have assisted users to deal with the issue of overloading information by retrieving the most relevant sources in the wide range of web services. Online video sharing (e.g., YouTube), E-commerce websites (e.g., Amazon), and Google News are some of the famous websites that use recommender systems to suggest videos, products and news to users in a personalized environment (Alp & Öğüdücü, 2018; Colace, De Santo, Greco, Moscato, & Picariello, 2015; Najafabadi, Mahrin, Chuprat, & Sarkan, 2017). These systems have also been successfully developed for e-government, social networks such as Facebook and e-business applications. The recommender systems present users with the most relevant and most liked items based on their past transactions. In lieu with this, the major task of the recommender systems is to construct a suitable model to calculate the users' interests. In comparing the profile of active users and the profile of other users and in predicting the possible interest of an active user

* Corresponding author.
E-mail address: maryam64266@yahoo.com (M.K. Najafabadi).

https://doi.org/10.1016/j.ipm.2018.12.007
Received 30 April 2018; Received in revised form 5 December 2018; Accepted 18 December 2018
Available online 09 January 2019
0306-4573/

based on the users with similar preferences, Collaborative Filtering (CF) is the most promising technique among the traditional ones to be used. Unlike other recommender techniques (e.g., knowledge-based and content-based techniques) that aggregate the knowledge on users' needs and also item features, CF recommends items based on compatible users' ratings on items (Da Costa & Manzato, 2016; Koohi & Kiani, 2016; Tyagi & Bharadwaj;, 2013). CF technologies are known to have the ability of recommending unanticipated items to users which is not commonly seen or heard before. Hence, these technologies are able to work well in domains where the content or the feature of an item is difficult to analyze. CF technologies obtain users' ratings on items by asking them explicitly or by observing implicitly their interactions with the system. These ratings are stored in a table that is known as the user-item rating matrix. The similarity measurement method such as the Pearson Correlation Coefficient or Cosine Similarity is applied to identify users who have similar preferences on items with active users or to define similar items by calculating the similarity of users' ratings. Once this is done, the prediction based on the ratings of these neighbors will be carried out (Liu et al., 2016; Najafabadi, Mohamed, & Mahrin, 2017; Schall, 2014).

Despite the widespread use of CF in providing personalized recommendation, this technique has its own issues. Among them are (1) cold start problem, (2) data sparsity problem, (3) gray sheep problem and (4) data correlation problem (Xie, Chen, Shang, & Fox, 2014). Hence, this article provides the solutions to the issues mentioned. (1) Cold start problem occurs when an item is newly added in the system or new user has just started to use the system. CF is unable to make good recommendation for the new user because the system has insufficient information on the user (insufficient previous ratings or purchases) to generate recommendation for them. Similarity, in the case of new items, it is unlikely that CF recommend these new items to users because no purchased or ratings expressed by users on these items. (2) Many recommender systems only rate a small subset of available items. Hence, most of the entries in the rating matrix are vacant. As a result of this, identifying similar users or items becomes taxing. Consequently, the similarity between two users or items cannot be calculated and eventually, the accuracy of prediction becomes very minimal. (3) Gray sheep problem leading to poor recommendation for the users whose opinions are not similar with the ones of any group of users. In fact, a user may be taken in the situation of gray sheep, when spend long time in the condition of cold-start problem, because such user has not shown interest on products of system. (4) Furthermore, there is a tendency for active users to consume similar goods, and the ratings for these items will be similar. This indicates that there are strong link among the ratings. The existing similarity measurement method like the Pearson Correlation and Cosine Distance are faced with such issue. Therefore, it can be inferred that similarities cannot be used for rating prediction (Moradi, Ahmadian, & Akhlaghian, 2015; Najafabadi & Mahrin, 2016; Polatidis & Georgiadis, 2016).

To overcome these issue, data mining and machine learning algorithms such as ARM (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017), clustering CF models (Rana & Jain, 2014; Salah, Rogovschi, & Nadif, 2016), graph-based approaches (Li & Chen, 2013; Musto, Basile, Lops, de Gemmis, & Semeraro, 2017; Shams and Haratizadeh, 2017), Bayesian nets (Da Costa & Manzato, 2016; Wei, He, Chen, Zhou, & Tang, 2017) have been applied. These techniques use models in employing pure rating data and provide possible connections between users and items to make recommendations. However, some of these models are highly complex, they require approximation of multiple restrictions, and are sensitive to the statistical properties of data sets. Due to the high cost, many of these theoretical models have not been used in recommender systems. In addition, to overcome sparsity problem in which unrepresentative users or items in the user-item rating matrix are alienated to decrease the dimensionalities, singular value decomposition technique is applied (Polatidis & Georgiadis, 2016; Zhou, He, Huang, & Zhang, 2015). However, in doing this, useful information may be lost and it is hard to factor the matrix because of the high portion of missing values that is caused by its sparseness. The last problem with aforementioned systems is that these works do not take time information such as item launch time or user purchase time into account. Recently there has been strong interest in time exploitation in recommender systems (Shams & Haratizadeh, 2017; Tewari & Barman, 2018; Xu & Yin, 2015). This is due to the reason that time exploitation shows superior recommendation performance. This is because recently seen items reflect users' current interest and this has strong impact in predicting users' preferences.

This paper has proposed a hybrid recommendation model that uses a new approach in incorporating timestamps of selected items by the user to improve recommendation performance. In this paper, users' profiles will be enhanced based on their choices of items by having their purchase time. This model uses the combination of a well-known graph analysis model employed by Google search engine (namely PageRank) and CF technique. This proposed recommendation model addresses limitations of CF. It also reaps the benefit from time information of user activities and item correlations in finding a chain-reaction of items influencer on predicting the users' interests in future. By using the PageRank algorithm, the proposed model determines the possible connections between the items and influencer score of each item in next purchase of user and by using the Cosine similarity measurement used in CF, it defines the degree of similarities between the users. It is worth mentioning that the exact value of the Cosine similarities is not used directly in our algorithm, but rather rank the items for sorting recommendation list by assigning PageRank scores and similarity value obtained of CF to each item. It means our algorithm keeps the preferred items of active user at the beginning of the recommendation list. However, how to exploit user purchase time in a systematic manner in PageRank algorithm remains to be investigated. The contribution of this study will be summarized in Section 1.2.

This article is organized as in the following sections. Section 2 introduces the related works on time-based CF and on graph-based recommendation and provides a detailed description of PageRank algorithm. Section 3 presents the details of proposed algorithm. Section 4 describes our dataset, evaluation metrics, the results of our experiments and comparisons made. Finally, Section 5 provides the conclusion and explanation on future direction of this study.

1.1. Motivation

Because of the importance of recommender systems in information explosion on Internet, many solutions have been proposed to help users and predicting their preferences on the items. However, the drawbacks related to the most effective technique (CF) used in recommender systems such as cold start, data sparsity, gray sheep and data correlation problem are great challenges which prohibit the recommender systems to become rich sources of retrieving the most relevant sources/items for users. Researchers try to improve recommendation techniques to resolve these disadvantages and provide better personalized services. Nonetheless, problems still exist.

First of all, most of the proposed recommender systems have given attention to user-item rating. Therefore, they concentrate only on the rating scores. As a result, rich and vital information such as time information and ranked ordering of items in personalized recommendations can be lost. Consequently, many recommended items that are of interest of the users may come at the very end of the recommendation list. This is not effective as recommender systems must make suggestions which are apt and fresh to users.

Second, most of the graph-based models designed for the recommender systems take action based on finding similarity among users. A few of them consider item correlation or both side of correlations (between user and item) to produce recommendations. Authoritative nodes in graph-based recommendation algorithms influence users' choices and this is tremendously important for recommender systems-related scenarios. Consequently, so far few related works on graph-based recommendation have concentered on finding a chain-reaction of items or sequencing of items in personalized recommendations.

Such challenges are effective motivators to find suitable recommendation model that is capable of keeping user's preferred items at the beginning of the recommendation list, which is vital to the users and results in high accuracy and precision value. Hence, providing the solutions to the mentioned issues is the main goal of this study.

1.2. Contributions

In order to integrate graph-based approach and CF in improving recommendation performance, a hybrid recommendation model is proposed. It is also proposed to overcome the shortcomings of CF and the discussed issues in Section 1.1. Thus, the contributions of this study are described in the following numbered paragraphs.

1 First, the novelty of this study is to integrate time information of selected items in graph-based approach that is called PageRank algorithm. This is done to utilize influencers' items on other users' choices by considering user-item interaction behavior. Then the algorithm ranks items for sorting recommendation list by assigning an important score for each item that has been obtained from followers' count of items and their association. The proposed algorithm dynamically keeps track of users' inclination toward different types of items with respect to time. It can be said that (to the best of the researchers' knowledge), this algorithm is the first graph-based approach that is able to capture the information of users' preference according to time and correlation between items. Our proposed recommender system keeps users' preferred items at the beginning of the recommendation list, which results in satisfaction among users.

2 Another distinctiveness of our algorithm relates to the extraction of list of unseen items by active users. This is done through computing of similarity between two users using the Cosine Distance method. Due to sparsity of data, similarity computation based on Cosine method is not that accurate. Hence, we do not use the exact value of similarity to generate the prediction of the active users on unseen items. Recommendation list are arranged in order that suits active users' interest by assigning similarity value obtained from Cosine and output of PageRank (influencer score of each item in next navigation). This implies that our algorithm keeps the preferred items of active users at the beginning of the recommendation list. It is worth to note that if similarities between active users with other users are not found, our system will make recommendations based on PageRank scores. This is due to items with high score will be recommended to users. In addition, although the data is sparse and few items are selected or rated by each user, only a few ratings or selected items are needed to construct our recommendation model. The details of proposed algorithm will be explained in Section 3.

3 This proposed novel recommendation model has a significant improvement with respect to other models for place recommendation, thus confirming the effectiveness of approaches inspired by PageRank in a wider range of scenarios. Our algorithm has been tested on two datasets, namely, Superstore dataset and MovieLens. The experimental results showed that our proposed algorithm can indeed effectively and accurately cope with the limitations of basic CF. In addition, when our algorithm was compared with basic CF and other extended version of CF, results revealed that our algorithm generated better performance within the perimeters of the metrics of Precision, Recall and MAP. In particular, the proposed recommender system keeps user's preferred items at the beginning of the recommendation list, which results in high MAP value.

2. Related work

Currently, many online companies and commercial systems such as book recommendations in Amazon.com, social network in Twitter, movie recommendations in Netflix.com and Moviefinder, music recommendations in Last.fm or in search engines (e.g., Google) apply recommender systems to generate recommendations to their customers. In general, many recommender systems have been developed to help users to find their desired products, or services (such as books, digital products, movies, music, TV programs, and web sites) by accumulating and examining the behaviors of internet users. In this section, we will review the works related to the main aspects of our proposed algorithm: time-based CF and graph-based recommendation and provide a detailed description of PageRank algorithm.

2.1. Researches on time-based CF

CF is the most applied and successful technique in a recommender system because this technique recommend unanticipated items to users, which are not similar to those they have seen before. CF uses the similarity measurement methods which make recommendations to active user according to the interests of users who have similar behaviors or similar purchase patterns. However, existing similarity measurement methods in CF have reduced the accuracy of recommendations due to problems such as data sparsity, cold start and data correlation. The extraction of users' interests and modeling the user behaviors are the main issues in the recommender systems. Therefore, researchers have proposed the recommendation methods (Li & Chen, 2013; Musto et al., 2017; Rana & Jain, 2014; Salah et al., 2016; Shams & Haratizadeh, 2017) that overcome problems of CF and employ the users profile to understand the users' interests. A user profile may contain demographic data (e.g., age, gender, education), explicit/implicit users feedback (e.g., viewed, bought, rates, star ranking, clicked), usage history, or product' features. The user profile construction or user behavior modeling is an important issue in recommender systems. It is important to know that users' interests are influenced by a set of latent factors such as the comedy movies versus drama movies in a movie recommendation, or the timestamps of selected items. But, focusing on some latent factors such as content based features of selected items is difficult to be extracted or be parsed in the recommendation algorithms. Therefore, in this research, we consider user profile which includes the timestamps of selected items, and propose an evolutionary model of user behavior (based on the user profile) to create a recommender system.

Several research works have incorporated time information in the recommendation algorithms in order to monitor the activities carried out by each user over time (Tong et al., 2017; Tewari & Barman, 2018). Tong et al. (2017) took user purchase time, trust and rating information into consideration to improve the recommendation accuracy of a CF-based recommendation system. Zheng and Li (2011) built an effective recommendation model by investigating the importance and usefulness of item launch time, user purchase time and tag when predicting users' preference. However, existing research works that investigate time information in recommendations are still scarce and they are not that dynamic. The user profile built in our proposed algorithm consists of users' activities over time which can adapt fast to the changes in users' interests. In addition, this user profile shows the correlation between items and the influence of each item purchased in the next navigation step (next user' choices).

2.2. Researches on graph-based recommendation and Pagerank algorithm

Graph-based recommendation algorithms epitomize the associations between users and items. They do this in the form of a bipartite graph in which there is a biased or unbiased link between a user and each item that the user has rated. There are two steps involved in these algorithms. The first step is constructing a graph representing a data and the second step is making recommendations based on the analysis of the graph. These recommendation algorithms would have different types of graphs (Li, Jiang, & Jin, 2018; Shams and Haratizadeh, 2017; Musto et al., 2017). However, the major component of these graphs is the association between users and items that have been rated by them. Therefore, the most common approach is constructing a bipartite graph where the link between one part of the graph (users) and the other part of the graph (items) is observable. The connections are from one part of the graph, users, to the other part, items. Most of these graph-based approaches show the link between a user and each item the user has rated. The association between users and items is not looked at from the point of view of profile matching in these approaches. So, this paper employs a graph-based approach to obtain information on correlation between items along with a probability value indicating how much the user prefers to choice an item in future.

Recent research works have revealed that the bipartite graph methods have been extended by adding more layers. For example, researchers (Li & Chen, 2013; Zhang, Zhou, & Zhang, 2010) take into account a three-layer graph using tags assigned to items by users to improve recommendation. Other works (Alp & Öğüdücü, 2018; Lee, Park, Kahng, & Lee, 2013) have used different types of nodes in a multi-layer structure to develop context-aware recommendation through a random walk in the graph. Other researchers have (Musto et al., 2017; Shams & Haratizadeh, 2017) revised the structure of the graph. They have also considered a star heterogeneous graph structure, where users and items can be linked by different types of nodes. They have also employed this graph structure to improve the model–based recommendations (Musto et al., 2017; Shams & Haratizadeh, 2017) or to make recommendation through improvement of personalized PageRank algorithms in heterogeneous networks. To the best of the researchers' knowledge, none of these graph algorithms are designed to capture preferences of users along with time information of selected items. In addition, most of these graph algorithms depend on the content based features which are not available to the system in all applications and may be costly to collect them.

In this research, an improved PageRank algorithm has been used to find the correlation between items and influencers' items in predicting users' interest on items. This algorithm is used to select appropriate items and obtain the satisfaction of more users on recommended items. PageRank algorithm is Google's well-known graph algorithm which has a numerical weighting mechanism to each website page hyperlinked in the world-wide web. The output of this algorithm is a probability distribution of website page that shows the likelihood that an internet user randomly clicking on web links to arrive at a particular web page. The equation used in PageRank algorithm is (Eq. (1)) (Shams & Haratizadeh, 2017):

$$\mathrm{PR}\ (\mathrm{A}) = \frac{1-d}{N} + d\left(\frac{\mathrm{PR}\ (\mathrm{B})}{L\ (B)} + \frac{\mathrm{PR}\ (\mathrm{C})}{L\ (C)} + \frac{\mathrm{PR}\ (\mathrm{D})}{L\ (D)} + \ldots\right), \tag{1}$$

PR (A) represents the probability distribution of node (A). PR (B), PR (C) and PR (D) are the probability distribution of input links to node A. The notation d indicates a damping factor to be used in the random surfer model in order to prevent from loop happen

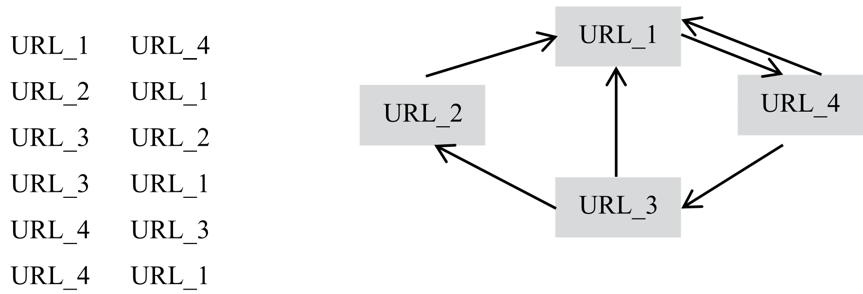


Fig. 1. Sample input data for PageRank algorithm.

between the two nodes. This damping factor holds the randomly clicking on links. Damping factor is in the range 0 and 1 but based on analysis on various studies, it is better to be set around 0.85. N represents the total nodes in graph-based approach. $L(B), L(C), L(D)$, .. are the number of output links from Node B, C and D, respectively. To illustrate, assuming that we run PageRank algorithm with the following input data file as shown in Fig. 1.

After running PageRank algorithm with the above input data file, we will get the following output based on calculation done by Eq. (1) on each node (or URL).

As shown at Table 1, the output clearly shows that URL_1 has the highest page rank and then URL_4 and followed by URL_3 and last URL_2. The algorithm works based on the following link structure:

If a URL (web page) is being referenced or linked by other URLs, then its rank steps up. This is because by being referenced, it means that this page is important. In which, this is the case of URL_1. While if an important URL like URL_1 is being cited by other URLs, this will boost the destination's ranking which is the case of URL_4 that is referenced by URL_1. Now, that is the reason why URL_4 ranking is higher than the other two URLs (URL_2 & URL_3). If we look at the various arrows in Fig. 1, we can see that URL_2 is being referenced the least and this is the reason for having the lowest ranking.

As we have discussed above, users' purchase time and data correlation are two important elements to grasp users' current interests in order to provide effective personalized recommendations. Therefore, it is necessary that our recommendation algorithm can be adapted by having users' preferences over time and correlation between items. This research is the first academic research to design graph-based approach for keeping users' preferred items at the beginning of the recommendation list. It is done by considering the activities carried out by each user over time in a graph-based approach to discover the correlation between items and then rank the ordering of items in giving personalized recommendations. However, the best way to exploit users' purchase time in a systematic manner in PageRank algorithm remains to be investigated.

3. The proposed hybrid recommendation model

This section explains our proposed recommendation algorithm as a solution to overcome drawbacks of basic CF and the discussed issues in Section 1.1. It is noted that our algorithm is based on the same logic of CF, as both find the similarity among users based on their overlapping interactions on items. Nonetheless, when comparing our system with CF and the existing recommendation systems; our system has several advantages. The architecture of the proposed algorithm is shown in Fig. 2. This development was divided into three Phases known as: (1) graph-based approach, (2) Collaborative Filtering (CF), (3) hybrid filtering phase. In Phase I, in order to provide the relationship between items based their purchase time, our algorithm incorporated the timestamps of selected items into users' profiles. Then, it determined the authoritative items that influenced other users' choices by assigning score to each item. In other words, our algorithm extracted users' interests based on their activities over time. Then, a sequencing of items in the form of graph structure was identified. This was done to calculate the influence of each item on users' next purchase. In Phase II, the similarity between users was computed by employing the Cosine Distance measure. In this phase, the profiles of active users were compared with other users' profiles and a list of unseen items by active users was extracted. In phase III, in order to predict users' interest on items, our algorithm influenced the similarity value obtained from Phase II with output of Phase I (rank scores obtained of each item).

The goal of our algorithm was to have a list of arranged items according to active users' interest. Hence, the items at the beginning of the recommendation list showed a higher preference among users. It is important to note that when data was sparse or new user just started to use the system, the existing similarity measurement methods such as Cosine measure could not compute the similarities between the users, due to non-existence of co-rated items among users. Therefore, in enhancing the precision and the recall of

Table 1
URL ranks based on PageRank algorithm.

1.	URL_4 has rank:	1.3705281840649928
2.	URL_2 has rank:	0.4613200524321036
3.	URL_3 has rank:	0.7323900229505396
4.	URL_1 has rank:	1.4357617405523626

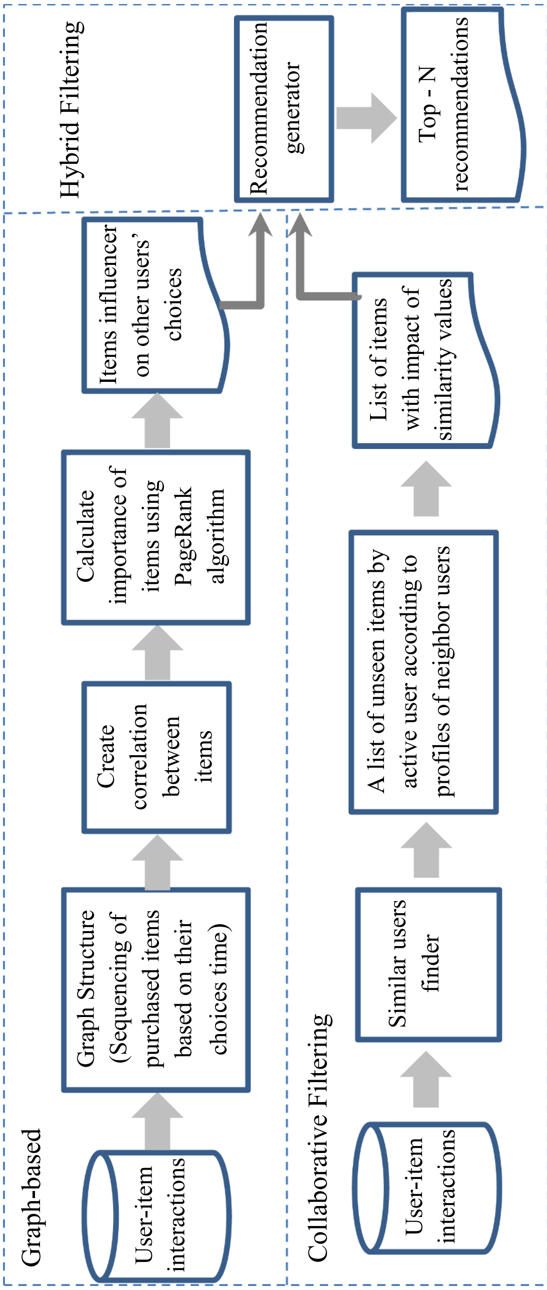


Fig. 2. Diagram of the proposed hybrid recommendation model.

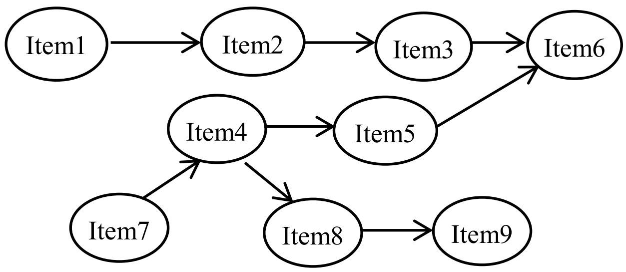


Fig. 3. Sequencing of purchased items based on their choices time.

recommendations made, our system has been established to exploit the output of graph-based approach to suggest a set of items that have more follower count among users. The details of each phase will be described in the following subsections.

3.1. Graph-based approach

The main goal of this phase is to discover significant correlation among items. This is done by considering time information of selected items and providing sequencing of items from more influential items in which users have higher preference. The main challenge in this phase is to create between items a link that is able to find authoritative items in predicting users' interest. This is crucial in creating a model of users' behaviors that can be incrementally adapted with changes in users' interests. The user profile construction is an important matter in a recommender system (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017; Rana & Jain, 2014). This is due to the accuracy of such systems that is highly dependent on its validity in analyzing users' behaviors. Therefore, we enhance users' profiles by having their choices of items based on their purchase time. With the assumption that timestamp of selected items can reflect users' current interests, we exploit the correlation among items based on the analysis of their behaviors whose historical consumptions overlap in time. Graph-based methods are the research fields in machine learning that find the correlation between items or users in a database (Alp & Öğüdücü, 2018). Fig. 3 shows graph-based approach designed (an improved PageRank algorithm) for ranking the items using users' chosen time of purchase on the sample data used in our experiment. It is noted that, nodes in our graph structure were items. For example, suppose User1 had seen/ bought Item1, Item2, Item3 and Item6 respectively. So Item1 influenced in purchasing Item2, Item2 in purchasing Item3 and so on. User 2 has seen Item4, Item5 and Item6 respectively. So Item4 influenced in purchasing Item5 and Item5 influenced in purchasing Item6. We defined a sequence of influencer items using users' chosen time as input for PageRank algorithm. In other words, each item in user' profile was sorted in front of other item based on their chosen time. An advantage of the proposed algorithm was that when data was sparse and a small percentage of items were seen by each user, our algorithm could construct the graph structure to find the link between items as shown at Fig. 3. This was because the correlation between items was discovered based on the interaction among all users on items instead of only focused on interaction between two users.

After preparing a sequence of purchased items based on their chosen time, our proposed algorithm calculated the importance of each node using PageRank algorithm. The importance of each node (item) was identified by score calculated from Eq. (1) in Section 2.2. This score value indicated how much the users preferred to select that item. In other words, the score value assigned for each item represented a probability distribution over interests of users. A model of the users' behaviors in item selection was adopted as a link between items. Therefore, our model was not just a model of the users' behaviors, but also showed the link between items. Indeed, the analysis of correlation between items from the interactions of users on items to find the authoritative nodes was very important, since these nodes influenced other users' choices and this was crucial in recommender systems. However, the exploitation of timestamp on users' activities in order to find the correlation between items in graph-based structure was a challenging issue. In previous research works, there was no graph-based model designed to find the link between items by considering users' purchase time. To the best of our knowledge, the current research is the first one in which the users' profiles capture the preference information provided by explicit or implicit feedbacks with focus on time of users' choice on items in a graph structure. These users' profiles are utilized for input data in PageRank algorithm to infer the ranking of users over unseen items and grab the chain-reaction of items influencer on predicting the users' interests in future.

3.2. Collaborative filtering (CF) phase

This phase of our algorithm extracted a list of unseen items by active user by means of similarity computation based on Cosine Distance method. To achieve this capability, the matching degree of active users' profile and the profiles of other users was measured by Eq. (2). The user profile represented a user's preferences on items. Then a list of unseen items by active user that were available in profiles of neighbor users was provided.

$$W_{Xy} = \frac{x\ .\ y}{|x| \times |y|} = \frac{\sum_{i=1}^{n} (r_{Xi})(r_{yi})}{\sqrt{\sum_{i=1}^{n} (r_{xi})^2}\sqrt{\sum_{i=1}^{n} (r_{yi})^2}} \tag{2}$$

where r_{Xi} and r_{yi} were the rating vectors of users x and y on item i, and $|x|$ and $|y|$ showed the length of the vectors x and y.

	i_1	i_2	i_3	i_4	i_5	i_6	…	i_n
u_1		5						
u_2	3		2				3	
u_3			4	3			5	1
u_4						1		
u_5			3		4		5	
u_6				2		3		3

Similarities between user u_3 with other users in database.

	u_1	u_2	u_4	u_5	u_6
u_3	0	0.72	0	0.82	0.54

Fig. 4. Fragment of a sparse user-item matrix and similarity calculation.

$\sum_{i=1}^{n}(r_{xi})^2$ and $\sum_{i=1}^{n}(r_{yi})^2$ indicated the sum of ratings on items by user x and y, respectively (Najafabadi & Mahrin, 2016; Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017).

Since computing the similarities between the users by employing the existing similarity computation methods was significant, our algorithm employed them in predicting the interests of active users. First, the similarity between active user and other users by Eq. (2) was calculated. Then, a matrix of unseen items by active user that were available in profiles of neighbor users along with influence of calculated similarity value was generated. Note that this similarity value has the influence on unrated entry of active user on items in matrix. When there were no explicit ratings in users' profiles, we used implicit data like history of purchased products instead of the user ratings on items (such as the number of common purchases between two users) to calculate the similarity between users. Only less number of users' evaluations on items was needed for our algorithm to find similar users. This was because this phase of our algorithm was aimed to construct a list of unseen items by active user based on matching his/her profile with profiles of neighbor users. An illustration of a sparse rating matrix with the 1–5 scale is shown in Fig. 4. As shown in Fig. 4, the most of the entries in the rating matrix are empty. We wanted to provide a list of unseen items by user $u_3$by comparing his/her profile with other users' profiles.

According to Cosine similarity measurement, the similarities between user $u_3$with the other users was calculated as shown in Fig. 4. The list of unseen items by user $u_3$according to profiles of neighbor users (U_2, U_5 and U_6) was item i_1, item i_5, and item i_6. The items in list were sorted based on the similarity value calculated from Cosine Distance as follows:(item i_5, item i_1 and item i_6)

1.	item i_5:	0.82
2.	item i_1:	0.72
3.	item i_6:	0.54

Noted that when the item *i* in list was repeated based on comparing the profile of active user with other users' profiles, this item *i* grabbed the incremental similarities. This was because the item *i* was available in two or more neighbors' profiles of active users and hence, suggesting such items could maximize the active users' satisfaction.

3.3. Hybrid filtering phase

Another uniqueness of our algorithm relates to the predicting unrated/unseen entry of an active user on items. In this phase, the results which are obtained from Graph-based approach and CF phase are going to be combined to finally generate a list of ranked items in our recommender system. Items with higher preference level value based on top-n recommendations will be suggested to user. The aim of our system is to keep the preferred items of active user at the top of the recommendation list and items bottom in the list are less likely to be used by user. This is because users in recommender systems have the tendency to receive a ranking list of new items based on their priorities at the top of the recommendation list (like search results at google that the web pages at top show more relevant web pages for user). We have a list of unseen items by active users in which at least one of the similar users has contributed to or showed interest in them (as shown in Fig. 2 at Section 3). If we want to predict the active user's preferences on this list, the score value calculated from graph-based approach and cosine similarity value obtained from CF will be combined and the first K ranked items are recommended to users as the favorite and related items. In other words, the total ranking of active user over unseen items are sorted according to influence of similarity value between two users and score value of our graph on the item *i* to produce an accurate prediction on ranking of items. As result, we improve the accuracy of our recommender system in terms of both ranking and evaluation of items (ratings/ history of purchased products).

It is worth noting that in our algorithm, when little information is available about the interaction of users on items, the lowest number of similar users can be used to construct a list of unseen items by active users. In addition, when there are no overlapping interactions between profile of active users with other users' profiles, the graph structure of our algorithm can help to generate top-N recommendations in which items with high score value will be recommended. Nonetheless, our proposed algorithm overcomes problem related to CF in sparse situations in which there is little information about users (or there is no information about them due to cold start issue).

4. Experimental results

The purpose of this section is to report the experiments done to verify the accuracy and efficiency of the proposed hybrid algorithm. First, a detailed description of the datasets used for the evaluation and evaluation metrics is presented. Then, the recommendation performance of the proposed algorithm gained by comparing the quality of top-k suggestions is analyzed. Finally, the experiment results gained from other proposed algorithms which act as the benchmark recommendation algorithms for making comparison with our algorithm are presented.

4.1. Dataset and experimental setup

In this research, we selected two datasets which were Superstore[1] and MovieLens to evaluate the proposed algorithm. All experiments conducted on these two datasets inclusive of numbers of purchases made by users, rating scores and time information. This was due to the design of the proposed algorithm which employed users' times of activities in extracting their preferences. MovieLens dataset consisted of 100,000 ratings of 1682 movies which were made by 943 users. Superstore dataset composed of data on purchases made by users on various products which was taken from e-commerce websites. There were 1000 implicit interaction records with products in this dataset. There were no explicit ratings in this dataset, so we used number of purchases for each user instead of users' ratings on items to calculate the similarity between users. The proposed recommender system was implemented in Java runs on a machine with 4 GB of RAM and 3.1 GHz CPU.

4.2. Evaluation metrics

The main idea of recommender systems is to maximize users' satisfaction from recommended items. Hence, in lieu of this, the proposed algorithm suggested a list of items to users in respective of their priorities. The proposed system was evaluated based on the position of every item suggested in the ranked list of the recommender. The aim of proposed system was to keep the preferred items of active users at the top of the recommendation, whereas items in the bottom of the list were less favored by users. This was because users in recommender systems had the tendency to receive a ranking list of new items based on their priorities at the top of the recommendation list.

Hence, three evaluation metrics from information retrieval researches were utilized to evaluate the proposed system. These metrics included Mean Average Precision (MAP@K), Precision and Recall. MAP@K calculated the precision of the first K recommended ranked items. Every ranked item contributed to the MAP@K measure of the recommendation, whereby it was proportion to its position of which was indeed accessed by the user (Musto et al., 2017). The Precision metric in recommender systems indicated the fraction of relevant items recommended to the total number of items recommended (either relevant or irrelevant) to users, in contrast to the Recall metric which indicated the fraction of relevant items recommended to the total number of relevant items (either recommended or not recommended) to users (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017; Tewari & Barman., 2018). Precision and Recall metrics are computed by Eqs. (3) and (4) as in the following formulae:

$$\text{Precision} = \frac{\text{Relevant recommendations generated}}{\text{Total number of recommendations generated}} \tag{3}$$

$$\text{Recall} = \frac{\text{Relevant recommendations generated}}{\text{Total number of relevant recommendations}} \tag{4}$$

The higher values of MAP, Precision and Recall measures show the higher accuracy of recommendations. It should be noted that a bigger value of N recommendations usually results in the increase of the Recall value and the decrease of the Precision value. Therefore, F-measure metric is another metric which is used for finding an optimal trade-off between Recall and Precision values. This metric is computed by Eq. (5):

$$\text{F} - \text{measure} = \frac{2^{*}\ \text{Precision}^{*}\text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

4.3. Experimental methodology

In our experiments, we followed a pre-processing technique that has been widely used in the area of evaluating the recommender systems (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017; Shams & Haratizadeh, 2017; Tewari & Barman, 2018). In this technique, user profiles in the form of (user, movie (or product), and ratings (or number of purchases) were randomly split into train and test set as 70% of users with their data was placed in the training set and 30% of them in test set. We ran our algorithm on training set in order to discover meaningful patterns and rules from users' behaviors and utilized these patterns in

[1] https://community.tableau.com/docs/DOC-1236.

predicting the user's unknown preferences on items in testing set. The testing set was used to analyze the effectiveness of our algorithm based on the relevant recommendable items and their priority in user' profile using evaluation metrics presented in the previous section.

Our experiments in this research were aimed to evaluate the quality of top-N recommendations generated by the proposed algorithm. Therefore, we discarded the authoritative items that influenced the next purchases of user from each test user' profile and retained the rest of the user purchase records to evaluate the accuracy of our system at the position of every relevant recommendation generated. In addition, the performance of our algorithm was evaluated under different number of recommendations. A survey among 200 people was conducted for choosing the appropriate value of N. The outcome of survey showed that users generally preferred to grab less than 25 recommendations at a time. So the performance of our algorithm was evaluated under 5, 10, 15, 20 and 25 recommendations. Besides, we have conducted the evaluation experiments under different level of sparsity in users' profiles to show the impact of sparsity levels on performance of our algorithm and basic CF. Based on number of users' ratings (or users' purchases records), users' profiles were randomly sampled and placed in datasets to satisfy sparsity levels between "0.3–0.5", "0.5–0.7" and "0.7–0.9". Noted that "0.3–0.5" was synonymous with lower sparsity level and "0.7–0.9" was synonymous with higher sparsity level (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017, Najafabadi, Mohamed & Mahrin, 2017). In other words, regarding the Eq. (6), users with less number of ratings were placed in higher sparsity level and users with more number of ratings were placed in lower sparsity level.

$$\text{sparsity measure} = 1 - \frac{n\text{R}}{n\text{Users}^{*}\ \ n\text{Items}} \tag{6}$$

Where nR is total number of existing users' ratings, nUsers and nItems are the total number of users and items respectively in the User_item matrix.

We have compared the performance of our proposed algorithm with other recommendation methods in order to evaluate its performance:

- Basic CF: This is a well-known recommendation method in which the similarity between users is calculated using the Cosine Distance measurement.
- GRank (Shams et al., 2017): the authors have proposed a novel graph-based for collaborative ranking domain in which the users' priorities are modeled in a new tripartite graph structure. It is similar to our algorithm to analyze the users' priorities in inferring a recommendation list.
- Graph-based (Musto et al., 2017): the authors have proposed a linked data in graph-based to be automatically feed with features obtained from the LOD cloud.
- User-rank (Gao, Wu, & Jiang, 2011): the authors have proposed a user ranking method in which the weight of a user on items is incorporated into the computation of item similarities and differentials.

4.3. Results and discussion

4.3.1. Comparison with CF based on influence of sparsity measures

This section shows the results of the experiments performed on two state-of-the-art datasets, namely Superstore and MovieLens datasets. Fig. 5 depicts that recommendations generated by our algorithm have higher Precision values compared to the basic CF among all sparsity measures. This is because less number of the irrelevant recommendations to users is generated by the proposed algorithm unlike basic CF that shows more inaccurate recommendations in higher sparsity level. On the average, the precision value using MovieLens and Superstore dataset has improved by 0.18.

In the next step, the performance of proposed recommendation algorithm was compared with basic CF regarding the Recall values. Fig. 6 shows that the differences in recall values between the proposed algorithm and basic CF have increased among three sparsity levels as these have reached their peak when sparsity of data is very high (0.7– 0.9). Based on the information in Fig. 6, it can be concluded that the Recall values of proposed algorithm using MovieLens and Superstore dataset is better than CF under different situations of sparsity. It is because our algorithm is able to satisfy more users from recommended items by generating the relevant recommendations to them.

As a result, when sparsity of data has increased, precision and recall values associated to experimental done by CF have dramatically decreased, whereas our technique manages the recommendation accuracy in different sparsity measures and slight changes have been made among sparsity levels (refer to Figs. 5–6). This was expected, since higher sparsity led to inaccurate neighborhood formation in CF. Accordingly, we have an inaccurate representation of users' preferences. In contrast, our technique can effectively benefit from finding the relationship between users and items rather than only limited to finding the relationship between users.

4.3.2. Comparison with CF based on influence of the number of recommendations

The comparison graphs of the results obtained using proposed recommendation algorithm and basic CF in terms of precision and recall values are shown in Figs. 7 and 8. The results in Fig. 7 show that the performances of our algorithm in terms of precision gradually decrease when the number of recommendations provided for the users increases, whereas in terms of recall, this gradually increases to where over top-20 and top-25 has the peak of recall values as shown in Fig. 8. This is due to a set of the irrelevant recommendations made to users which has been generated when number of recommendation has risen. Consequently, the precision values have decreased. In contrast, the number of relevant items that have not been suggested decreases and the recall values

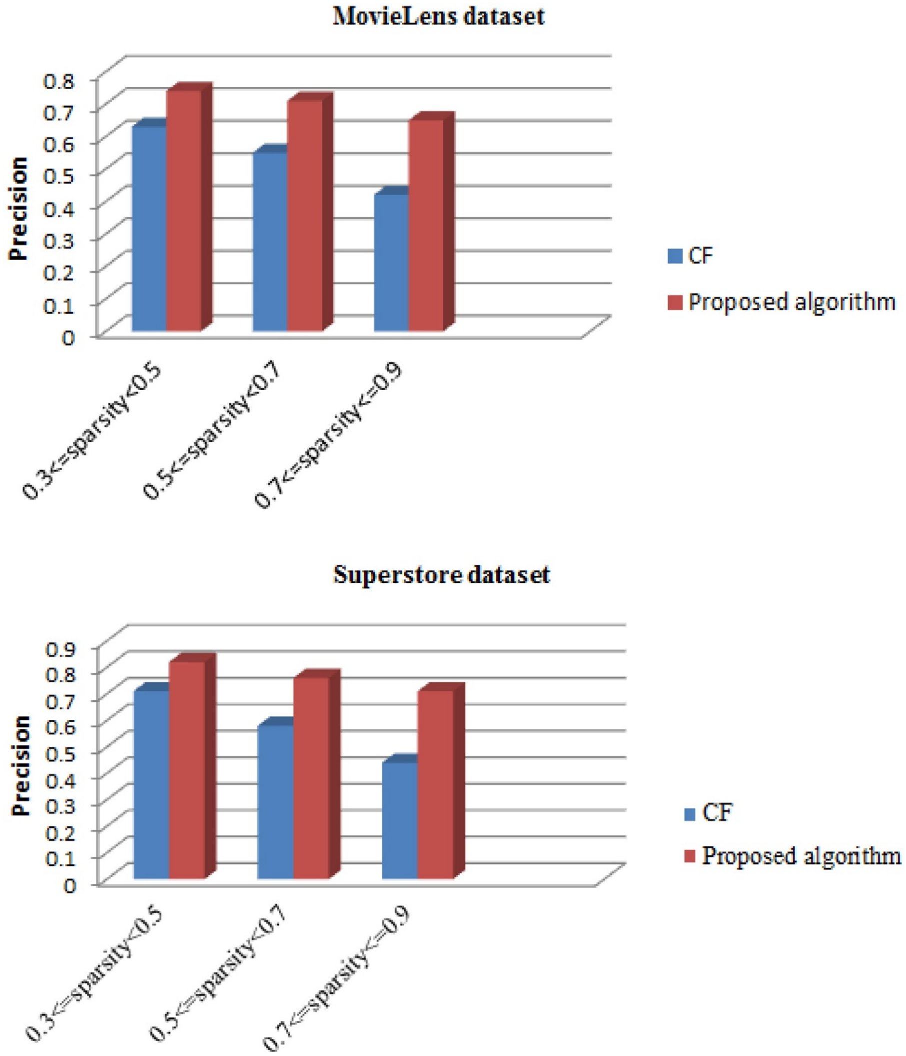


Fig. 5. Precision values for different sparsity measures using MovieLens and Superstore datasets.

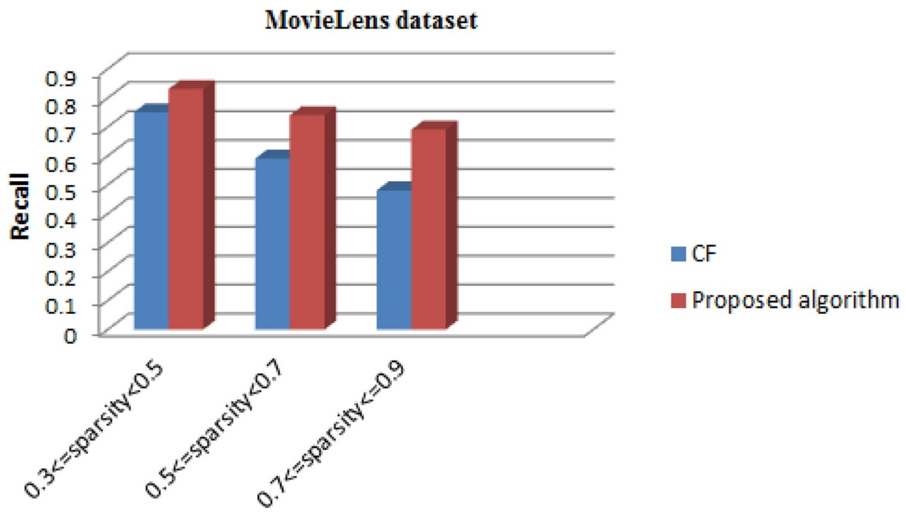
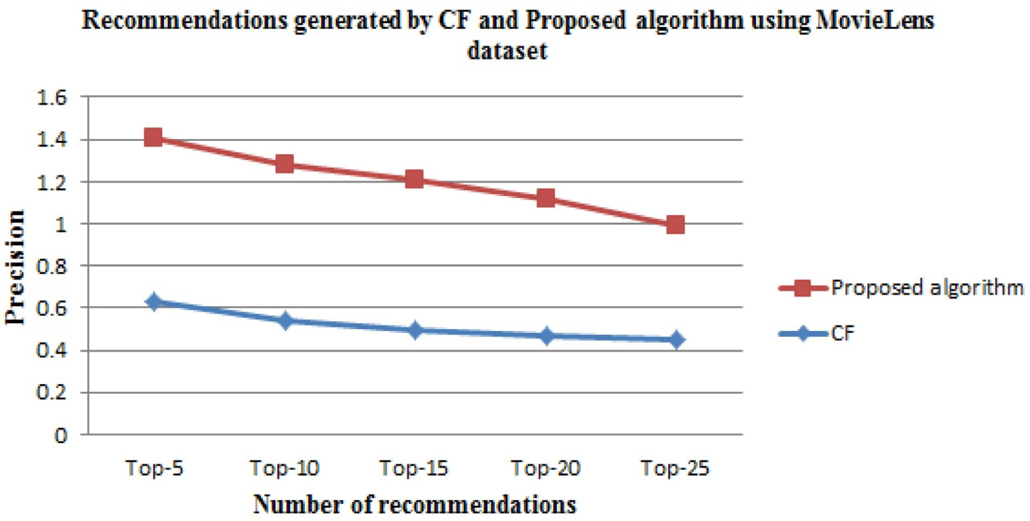


Fig. 6. Recall values for different sparsity measures using MovieLens and Superstore datasets.



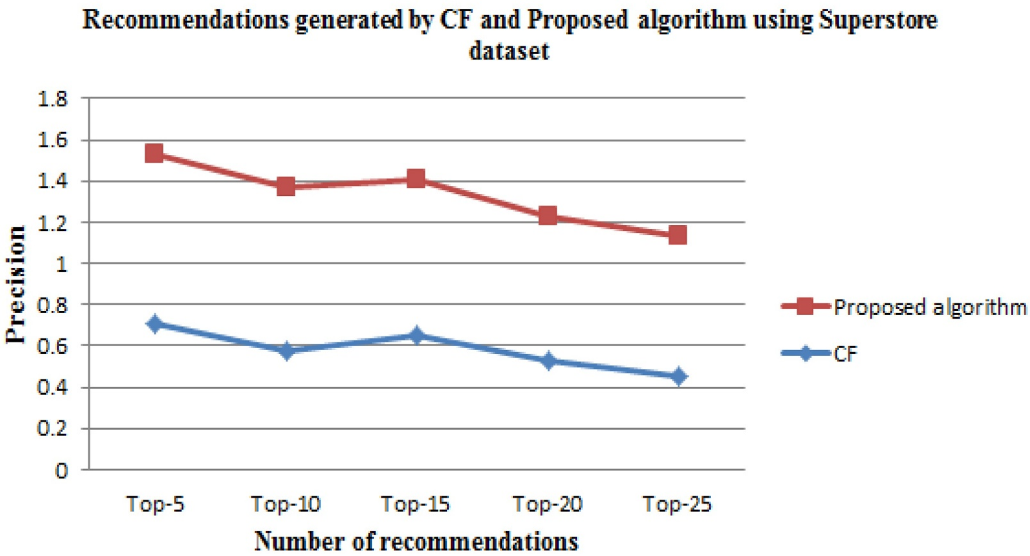


Fig. 7. Precision comparison of basic CF and proposed algorithm using MovieLens and Superstore datasets.

increase. It should be noted that precision and recall values for both algorithms cannot be simultaneously high.

Figs. 7–8 are apparent whereby that they have shown that the proposed approach represents the improvements of over 20% in terms of Precision and Recall values under sparse data conditions compared to the basic CF. Combining of more than one method can avoid the drawbacks of a single recommendation method and provide more accurate recommendations. Moreover, since the proposed system find the correlation between items in graph structure without dependent on users' ratings, the Cold start problem of CF is removed.

To summarize, the advantages of the proposed algorithm are first; the enhancement of the profile of each user by having their choices of items based on the purchase time. The users' profiles are created based on their past and current interest in a way that interests of users can dynamically change with time and generate good quality recommendations. On the other hand, in real-world recommender systems, users are eager to receive the items which they are interested at the beginning of the recommendation list rather than to receive them at the very end of the recommendation list. Hence, our algorithm arranges the items in the recommendation list based on users' priorities. Second, in terms of the accuracy prediction, our proposed algorithm discovers the correlation between items in a graph-based model based on the interaction among all users on items instead of only focusing on interaction between two users. Third, the improvement that the proposed algorithm offers is capturing the correlation between items and applying users similarities based on availed data. Therefore, our algorithm achieves better performance when compared to the CF.

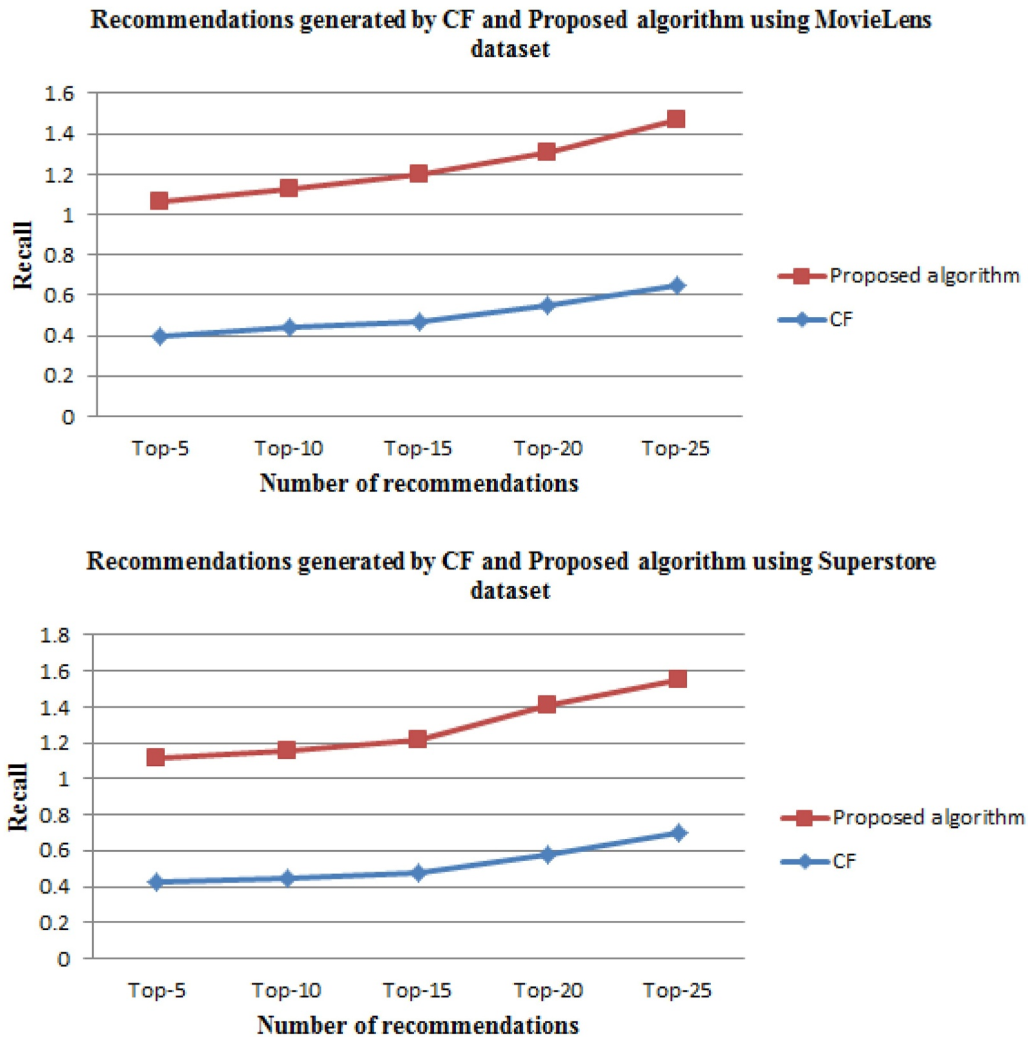


Fig. 8. Recall comparison of basic CF and proposed algorithm using MovieLens and Superstore datasets.

4.3.3. Comparison with state-of-the-art methods in terms of F-measure and MAP metrics

To evaluate the capability of the proposed algorithm on recommendations to users, we compared its performance with other the state-of-the-art methods. Therefore, a comparison of the experimental results obtained using state-of-the-art methods on MovieLens dataset in terms of the F-measure and MAP values are shown in Figs. 9–10. The comparison graphs of F-measure values are presented in Fig. 9. It shows that the proposed recommendation method achieved better performance compared to other recommendation methods, whereas user-rank (Gao et al., 2011) comes in second. It can be clearly seen that behavior of four methods in terms of F-measure has been relatively stable under different number of recommendations and their performance differ slightly for Top-10, Top-15 and Top-20 recommendations. This is due to the F-measure metric which is the combination of the Precision and Recall values, which has been used to find an optimal trade-off between Precision and Recall values. For example, when recommendation methods have achieved the highest precision, their recall would be at the lowest. Fig. 10 shows that the proposed recommendation method has outperformed the other two methods including Graph-based (Musto et al., 2017) and basic CF in terms of MAP, whereas Grank (Shams et al., 2017) comes in first. Getting back to the details, the proposed recommendation algorithm and Grank algorithm (Shams et al., 2017) proposed a ranking algorithm to model the users' priorities in a graph structure and suggest their priorities at the top of the recommendation list. This is reason why both algorithms have outperformed the other recommendation methods. The Fig. 10 reveals that the state-of-the-art methods control the MAP values' fluctuations for Top-10 and Top-20 recommendations and these MAP values have remained fairly static.

5. Conclusions and future work

Since CF recommendation methods compute the similarities between users or items based on the users' co-rated items, their performance in predicting the users' preferences is low when the data is sparse. In addition, most existing recommendation methods

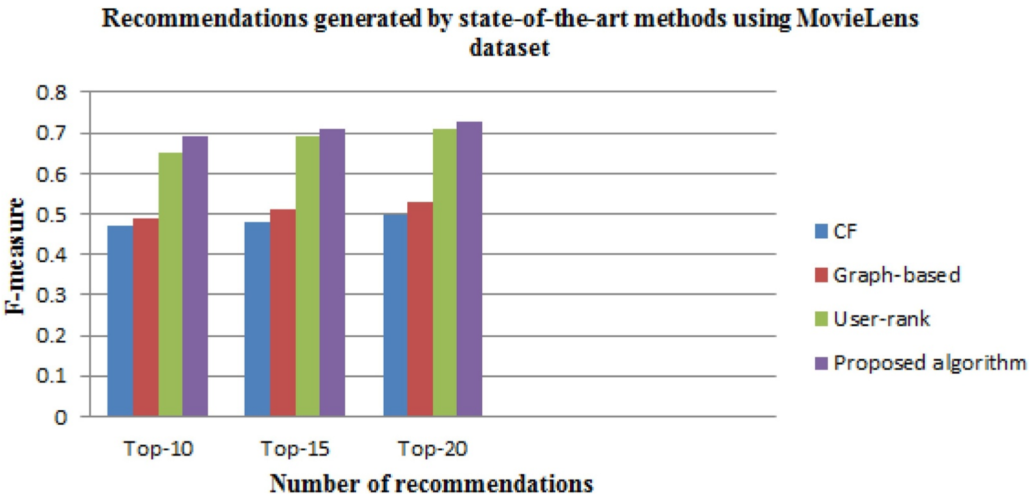


Fig. 9. F-measure metric comparisons of four methods using MovieLens dataset.

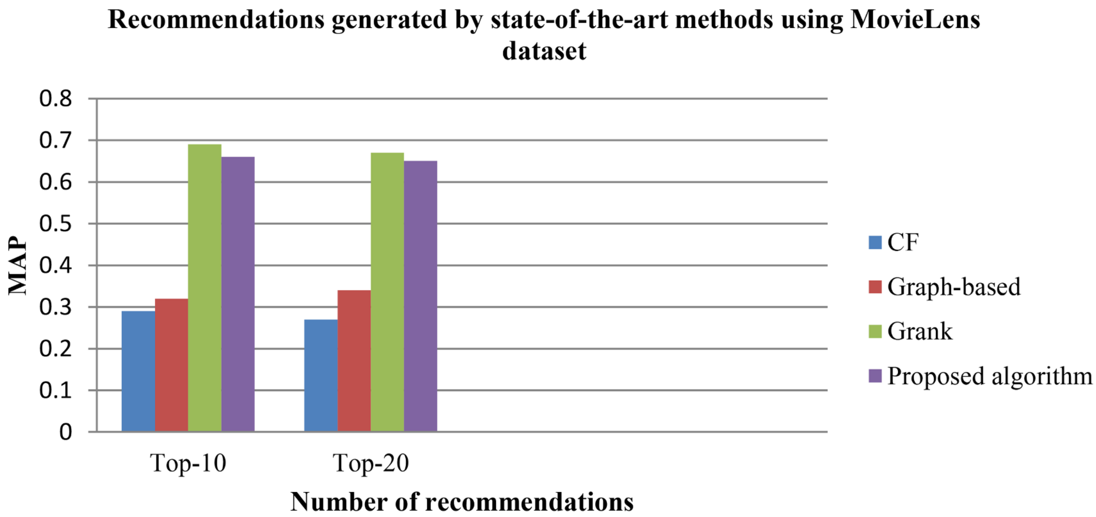


Fig. 10. MAP values of four methods using MovieLens dataset.

have made the efforts to improve traditional CF methods with augmenting users' interesting items to their recommendation list. However, they are vulnerable in capturing users' priorities to the sequence of items in the recommendation list. It means they fail to analyze the users' priorities in inferring a recommendation list. In this research, to overcome the shortcomings of CF and current recommendation methods in ranking preference dataset, we have used a graph-based structure to model the users' priorities and capture relationships between users and items. In real-world recommender systems, users are eager to receive the items which they are interested in at the beginning of the recommendation list rather than to receive them at the very end of the list. Hence, the goal of this research was to arrange the order of preferences of users in a recommendation list in such a way that users' preferred items come at the beginning of the list.

We proposed a hybrid recommendation model that is using a new graph-based approach in incorporating timestamps of selected items by users to improve recommendation performance. Our proposed approach has improved the effectiveness of recommendations by employing an improved PageRank algorithm to determine the possible connections among the items and compute the influencer score of each item in next purchase of users. This was done using the Cosine similarity measurement which had been used in CF to define the degree of similarities between users. Then, we conducted a series of experiments on movie and e-commerce dataset. As a result, the proposed algorithm improved Recall to 25% when the data was sparse and improved Precision to 18% on higher sparsity level in comparison with basic CF. The proposed system also demonstrated around 20% improvement on average of Precision and Recall values for top- n recommendations. The uniqueness of proposed approach is the enhancement of the profile of each user by having choices of items based on purchase time. The users' profiles were created based on their past and current interest as their interests can dynamically change with time and thus, this generated good quality recommendations. The proposed algorithm has its own unique feature that discovered the association between items based on the interaction among all users on items instead of only focusing on interaction between two users. Moreover, it generated a list of ranked items for the active users in collaboration with degree of similarities between the users and influencers scores of items on other users' choices. Consequently, all these unique features with evidence gathered from the experiments conducted helped our algorithm to gain significant improvement compared to other recommendation methods in terms of F-measure and MAP metrics. In future, we intent to involve a big data framework in order to distribute and process a huge number of data while generating recommendations.

Acknowledgments

The authors would like to thank Institute of Research Management & Innovation (IRMI) Universiti Teknologi MARA and the Malaysian Ministry of Education for their support and cooperation including researches and other individuals who are either directly or indirectly involved in this study.

References

Alp, Z. Z., & Öğüdücü, Ş. G. (2018). Identifying topical influencers on twitter based on user behavior and network topology. *Knowledge-Based Systems, 141*, 211–221.

Colace, F., De Santo, M., Greco, L., Moscato, V., & Picariello, A. (2015). A collaborative user-centered framework for recommending items in Online Social Networks. *Computers in Human Behavior, 51*, 694–704.

Da Costa, A. F., & Manzato, M. G. (2016). Exploiting multimodal interactions in recommender systems with ensemble algorithms. *Information Systems, 56*, 120–132.

Gao, M., Wu, Z., & Jiang, F. (2011). Userrank for item-based collaborative filtering recommendation. *Information Processing Letters, 111*(9), 440–446.

Koohi, H., & Kiani, K. (2016). User based Collaborative Filtering using fuzzy C-means. *Measurement, 91*, 134–139.

Li, W., Jiang, S., & Jin, Q. (2018). Overlap community detection using spectral algorithm based on node convergence degree. *Future Generation Computer Systems, 79*, 408–416.

Li, X., & Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems, 54*(2), 880–890. https://doi.org/10.1016/j.dss.2012.09.019.

Liu, J., Sui, C., Deng, D., Wang, J., Feng, B., Liu, W., et al. (2016). Representing conditional preference by boosted regression trees for recommendation. *Information Sciences, 327*, 1–20.

Lee, S., Park, S., Kahng, M., & Lee, S. G. (2013). Pathrank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Systems with Applications, 40*(2), 684–697.

Moradi, P., Ahmadian, S., & Akhlaghian, F. (2015). An effective trust-based recommendation method using a novel graph clustering algorithm. *Physica A: Statistical Mechanics and its Applications, 436*, 462–481.

Musto, C., Basile, P., Lops, P., de Gemmis, M., & Semeraro, G. (2017). Introducing linked open data in graph-based recommender systems. *Information Processing & Management, 53*(2), 405–435.

Najafabadi, M. K., Mahrin, M. N. R., Chuprat, S., & Sarkan, H. M. (2017). Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior, 67*, 113–128.

Najafabadi, M. K., Mohamed, A. H., & Mahrin, M. N. R. (2017). A survey on data mining techniques in recommender systems. *Soft Computing,* 1–28.

Najafabadi, M. K., & Mahrin, M. N. R. (2016). A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback. *Artificial Intelligence Review, 45*(2), 167–201.

Polatidis, N., & Georgiadis, C. K. (2016). A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications, 48*, 100–110.

Rana, C., & Jain, S. K. (2014). An evolutionary clustering algorithm based on temporal features for dynamic recommender systems. *Swarm and Evolutionary Computation, 14*, 21–30.

Schall, D. (2014). Who to follow recommendation in large-scale online development communities. *Information and Software Technology, 56*(12), 1543–1555.

Shams, B., & Haratizadeh, S. (2017). Graph-based collaborative ranking. *Expert Systems with Applications, 67*, 59–70.

Salah, A., Rogovschi, N., & Nadif, M. (2016). A dynamic collaborative filtering system via a weighted clustering approach. *Neurocomputing, 175*, 206–215.

Tewari, A. S., & Barman, A. G. (2018). Sequencing of items in personalized recommendations using multiple recommendation techniques. *Expert Systems with Applications, 97*, 70–82.

Tong, C., Qi, J., Lian, Y., Niu, J., & Rodrigues, J. J. (2017). TimeTrustSVD: A collaborative filtering model integrating time, trust and rating information. *Future Generation Computer Systems*.

Tyagi, S., & Bharadwaj, K. K. (2013). Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining. *Swarm and Evolutionary Computation, 13*, 1–12.

Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications, 69*, 29–39.

Xu, Y., & Yin, J. (2015). Collaborative recommendation with user generated content. *Engineering Applications of Artificial Intelligence, 45*, 281–294.

Xie, F., Chen, Z., Shang, J., & Fox, G. C. (2014). Grey forecast model for accurate recommendation in presence of data sparsity and correlation. *Knowledge-Based Systems, 69*, 179–190.

Zhou, X., He, J., Huang, G., & Zhang, Y. (2015). SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences, 81*(4), 717–733.

Zheng, N., & Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications, 38*(4), 4575–4587.

Zhang, Z. K., Zhou, T., & Zhang, Y. C. (2010). Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications, 389*(1), 179–186.