IEICE TRANS. INF. & SYST., VOL.E99–D, NO.6 JUNE 2016

PAPER

A Collaborative Filtering Recommendation Algorithm Based on Hierarchical Structure and Time Awareness

Tinghuai MA[†,††a)], **Limin GUO**[††], **Meili TANG**[†††], **Yuan TIAN**[††††], **Mznah AL-RODHAAN**[††††], ***Nonmembers***, ***and*** **Abdullah AL-DHELAAN**[††††], ***Member***

SUMMARY User-based and item-based collaborative filtering (CF) are two of the most important and popular techniques in recommender systems. Although they are widely used, there are still some limitations, such as not being well adapted to the sparsity of data sets, failure to consider the hierarchical structure of the items, and changes in users' interests when calculating the similarity of items. To overcome these shortcomings, we propose an evolutionary approach based on hierarchical structure for dynamic recommendation system named Hierarchical Temporal Collaborative Filtering (HTCF). The main contribution of the paper is displayed in the following two aspects. One is the exploration of hierarchical structure between items to improve similarity, and the other is the improvement of the prediction accuracy by utilizing a time weight function. A unique feature of our method is that it selects neighbors mainly based on hierarchical structure between items, which is more reliable than co-rated items utilized in traditional CF. To the best of our knowledge, there is little previous work on researching CF algorithm by combining object implicit or latent object-structure relations. The experimental results show that our method outperforms several current recommendation algorithms on recommendation accuracy (in terms of MAE).
key words: *recommender system, collaborative filtering, hierarchical structure, time weight*

1. Introduction

People quickly reach the point of information explosion with the rapid development of modern technology. However, when facing this vast explosive information resource, people cannot get really useful information for themselves, thereby reducing the utilization of information.

Under the above background, recommender system (RS) [1] came to be and has achieved great success, while gaining the attention of researchers who have developed classic collaborative filtering algorithms [2]–[5]. It helps users handle information explosion problem and obtain their own meaningful information, services, and recommendations more conveniently and quickly. Some examples would be Amazon and YouTube [4], [6]. For given users, items and ratings, recommender systems attempt to forecast ratings of the unseen items or generate a list of items (movies, music, news, others) that may interest users. The widely used recommendation method includes content-based (CB) [7], collaborative filtering (CF) [8] and hybrid approach [9]. Among these recommendation algorithms, collaborative filtering is commonly accepted as the most successful one. However, despite it is popular, the current collaborative filtering has its defects, for example, the sparseness, and the cold-start problem [5]. Additionally, traditional collaborative filtering does not consider item's hierarchical structure relation and user's interest shift, which cannot ensure the accuracy for complex items and multiple users. Some research use tag information for solving cold-start problem and achieve good results [10]. Recently, some leading researchers have paid attention to items' semantic similarities [11], [12], but the effectiveness is unremarkable.

In this paper, we explore the characteristics of the items themselves, to reduce the sparsity and enhance the precision of similarity measurement between items in collaborative filtering recommendation. We discovered that many items have distinctive features, with some hidden hierarchical structures that can tie items together. Our main method makes use of items' relations from two aspects. First, we use the obvious property feature to describe direct association relations. Second, we exploit the intrinsic hierarchical link to explore items' potential indirect connections. Furthermore, we incorporate the above connections into a comprehensive similarity function, which could take advantage of both methods. It is remarkable that our method can be efficiently utilized in any circumstances with hierarchical information.

Besides, in real recommendations, user's interests change naturally over a long period of time. It is common for their feedback data is ordered by time. Therefore, the basic recommendation technology should be able to dynamically track such changes accordingly and provide timely responses in order to make fast interactive experiences. However, most approaches don't consider the dimension of time and assume user's interests are static. They ignore the difference between current and historical data on the step of predicting current preferences. In order to make timely recommendations, we adopt a strategy which combine a time

Manuscript received September 26, 2015.
Manuscript revised January 26, 2016.
Manuscript publicized March 9, 2016.
[†]The author is with School of Computer, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210–044, China.
[††]The authors are with Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210–044, China.
[†††]The author is with School of Public Administration, Nanjing University of Information Science & Technology, Nanjing 210044, China.
[††††]The authors are with Computer Science Department, College of Computer and Information Science, King Saud University, Riyadh 11362, Saudi Arabia.
a) E-mail: thma@nuist.edu.cn
DOI: 10.1587/transinf.2015EDP7380

function on the first step of score predict. It can give more relevance to the current data, and less to the historical feedback information.

The remainder of this paper is arranged as follows. Initially, Sect. 2 demonstrates background information and related work. In Sect. 3, we address our own research method in details. Examples and related results analysis are described in Sect. 4. Finally, we conclude our paper and present future work in Sect. 5.

2. Review and Related Works

Recommended strategy is the most critical component in the recommender system, which determines the recommendation performance. Collaborative filtering is the most popular strategy and there has been a significant amount of research on it. Additionally, we also describe the technologies considering item's hierarchical structure and time-aware recommender systems, and emphasize the difference between our method and other work.

2.1 Collaborative Filtering

Different from content-based recommender systems, collaborative filtering recommendation methods attempt to predict the ratings for active users on target items based on the items formerly rated by other users with similar tastes [13], [14]. A large proportion of CF approaches only access the items' and users' identifiers. Thus, they are employed more than CB approaches because they don't require additional descriptions of items or users [5]. As we all know, various collaborative filtering methods have been developed in academic and commercial fields, where they are chiefly used as a tool to personally customize content for a certain customer's requirement in the online retail system. For the above reason that CF technologies afford to promote marketing and increase merchandise sales, of which the successful commercial website Amazon is a significant example [4], they have attracted great attention and achieved significant progress in the past decade. Collaborative recommendation is the most widely applied method according to Candillier [15]. Moreover, traditional collaborative filtering algorithms are still accessible [16] on account of the classic approach's simplification, reasonability, availability, and stability. In general, the two types of classic CF approaches are the user-based CF method and the item-based CF method.

User-based CF was first presented by group-lens research group [17]. The primary ideas of user-based CF are as follow: firstly, we seek others which have similar preferences previously for a given active user; secondly, for each item which the current user does not have seen, we will compute predicted scores exploiting the ratings of similar users (neighbors) for it. This method is based on two assumptions: (1) If a user has similar preferences in the past, they will continue to have similar preferences in the future. (2) Users' preferences will not change over time. We usually utilize the Person correlation coefficient as the similarity computation for the user-based CF algorithm:

$$sim(u,v) = \frac{\sum_{i\in ci}\left(r_{u,i}-\overline{r_u}\right)\left(r_{v,i}-\overline{r_v}\right)}{\sqrt{\sum_{i\in ci}\left(r_{u,i}-\overline{r_u}\right)^2}\sqrt{\sum_{i\in ci}\left(r_{v,i}-\overline{r_v}\right)^2}} \quad (1)$$

where $r_{u,i}$,$r_{v,i}$ denote respectively rating of user u, v on item, $\overline{r_u}$, $\overline{r_v}$ are treated separately as average score of user u, v on all items, ci shows items commonly rated by both user u and user v.

To predict rating of active user u for target item i, we use the aggregation function, defined as follows:

$$P_{u,i} = \overline{r_u} + \frac{\sum_{v\in N_u} sim\left(u,v\right)\left(r_{v,i}-\overline{r_v}\right)}{\sum_{v\in N_u}\left|sim\left(u,v\right)\right|} \quad (2)$$

where $P_{u,i}$ representatives the predicted preference of target user u on target item i, N_u represents neighbors of user u, $\overline{r_u}$, $\overline{r_v}$ indicate separately average scores of user u, v on all items, similarly, $r_{v,i}$ shows rating of every neighbor v on item i, $sim(i, j)$ means the adjusted cosine similarity between i and j, the $|sim(i, j)|$ is avoiding the offset each other because the $sim(i, j)$ range is from -1 to 1.

There are general optimization schemes proposed in order to improve user-based CF, for instance dividing users into different clusters and significance ranking. However, the progress of cluster users does not consider their evolving interests and intuitively users may have different characteristic degrees in different groups. Hence, in this paper we attempt to achieve efficiency of similarity computation according to essential hierarchical structure between items. Currently, many large-scale e-commerce sites possess millions of users and commodities, user-based CF confronts many serious challenges, such as its difficulty to accomplish the calculation of real-time prediction scores, especially when system scan a large number of potential neighbors. Hence, large-scale e-commerce sites adopt frequently another CF technology proposed by [2] known as item-based CF algorithm because that it is self-adaptability to conduct preprocess. The core idea of item-based CF is that it calculates ratings utilizing similarities between items rather than users. Additionally the algorithm consists of two phases: (1) Similarity Computation. Common Person correlation coefficient used in item-based CF is demonstrated as follow:

$$sim\left(i,j\right) = \frac{\sum_{u\in cu}\left(r_{u,i}-\overline{r_u}\right)\left(r_{u,j}-\overline{r_u}\right)}{\sqrt{\sum_{u\in cu}\left(r_{u,i}-\overline{r_u}\right)^2}\sqrt{\sum_{u\in cu}\left(r_{u,j}-\overline{r_u}\right)^2}} \quad (3)$$

where cu shows users who have rated on both item i and item j; $r_{u,i}, r_{u,j}$ are treated separately as rating of user u on items i, j; $\overline{r_u}$ displays average score of user u on all items.

(2) Preference prediction. The prediction rating for a given item can be defined including weighted sum and linear regression below:

$$P_{u,i} = \frac{\sum\limits_{j\in N_i} sim\left(i,j\right) r_{u,i}}{\sum\limits_{j\in N_i}\left|sim\left(i,j\right)\right|} \quad (4)$$

where N_i denotes the set of neighbors of target item i and $r_{u,i}$ indicates rating of user u on item j; $P_{u,i}$ shows the predicted score of target user u on target item i.

However, as far as we know, the pure collaborative filtering has its own fundamental barriers [18], especially reflecting on the data sparseness. In the actual websites, there are a large number of users and items, whereas users usually rate merely on a small part of items. In a consequence, available data for the calculation of similarity is very extremely limited, which makes it not accurate enough when systems select neighbors. And similarity is difficulty to be counted resulting from the absence of co-raters in traditional collaborative filtering. Even if similarity is calculated, reliability is also difficult to guarantee. Our approach is helpful for mitigating sparseness to a certain extent, since it considers adequately the hierarchical structure of items.

2.2 Hierarchical Structure Recommender System

When calculating the similarities of items, the traditional approaches do not take into account the influence of items own hierarchy and characteristics. However, in actual e-commerce recommendation system, hierarchical similarity plays an important role [28]. Hierarchical information is intended to reflect the similarity between items, to some extent. Some researchers think over the impact of taxonomy information on items [19], yet they ignore the internal relation of hierarchical structure. Additionally, many other researchers study hierarchy from the perspective of semantic similarity [20], but the effectiveness is not distinct. To the best of our knowledge, there is little work to integrate hierarchical structure similarity measure into recommender algorithms. Cognitive psychology enlightened us to the fact that items intuitively have hierarchical association. Therefore, in this paper, we will explore the innately hierarchical essence to characterize item-item association from two aspects of the direct (the comparison of common attributes) and indirect (construct hierarchical structure trees of items and their attributes). Then we measure the similarity between items by exploring a new calculative strategy that incorporates the above components and rating similarity in a comprehensive and absorbing way.

2.3 Time-Aware Collaborative Filtering

Recently, there is an emerging trend that users' behaviors with real recommender systems are becoming highly drifted. So it is rather unreasonable that most existing methods assume users' preferences and items' characters are static. Users' tastes, external conditions such as holidays, and recommender system self are all possible to change as time goes by [20], [21]. For these situations, it is crucial to dynamically track users' interests and response quickly for improving performances of recommender system. Traditional CF algorithms do not think of these and therefore require continuous training to trace the evolving data. Then exploiting time information has become an effective method to make timely recommendation. Hence, temporal recommendation methods are indeed receiving growing attention. The main feature of handing the time dimension in user profiles is the usage of time information at the rating prediction process, being able to make different recommendations at different recommendation time according to the users' preferences.

In this paper, we adopt a strategy to assign different weights to old and new data and transform the prediction rating formula in the traditional CF algorithm. The straight motivation of our approach is that users' interests naturally evolve over time.

3. Hierarchical Temporal Collaborative Filtering

In this section, we mainly display the following two aspects' content. One is the exploration of hierarchical structure between items to improve similarity, and the other is the improvement of the prediction accuracy by utilizing a time weight function.

3.1 Similarity Methods

This section introduce different methods for calculating the similarity between items. First method is direct hierarchical structure between attributes based on Jaccard similarity. Latent attributes relation between items namely indirect hierarchical structure is used as another method for measuring the similarity between items. Next, we make a linear combination of both aspects as hierarchical structure similarity. Finally, we combine hierarchical structure similarity and rating similarity used in traditional collaborative filtering.

3.1.1 Direct Hierarchical Structure Similarity

According to collaborative filtering in our previously mentioned methodology, we need to compute the similarity between items. In this paper, item hierarchical structure similarity derives from two different aspects mentioned above: item-item direct hierarchical similarity S_{DH} and item-item indirect hierarchical structure similarity S_{IH}.

Given two particular items, i and j, S_{DH} can be calculated by utilizing the Jaccard similarity between the attribute characteristics that the item belongs to. To this point, item-item direct association similarity S_{DH} can be represented as

$$S_{DH}(i,j) = \frac{A_i \bigcap A_j}{A_i \bigcup A_j} \tag{5}$$

where $A_i \bigcap A_j$ indicates the number of common attribute features between item i and item j, and $A_i \bigcup A_j$ denotes the sum of property features that these two items, i and j possess separately.

3.1.2 Indirect Hierarchical Structure Similarity Refinement

Generally speaking, a recommender system is associated

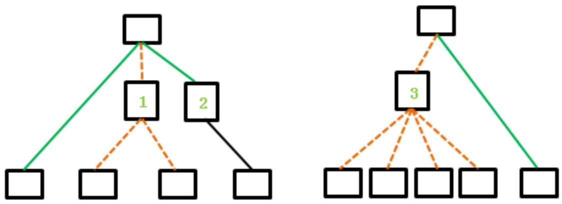


Fig. 1 Items' indirect hierarchical structure trees

with items that have indirect connections [22] generated from their natural hierarchical structure, by which users can be easily acquainted with different kinds of items. For a user, his/her rating not only expresses his/her direct preference about a certain item, but it also provides a hint about his/herinterest in related items. For example, if Alice rates a particular adventure/fantasy movie 4 stars, in addition to illustrating her opinion about that movie, her rating also reveals her taste. First, adventure/fantasy movies. Second, adventure and fantasy movies in a sense. Furthermore, the existence of these indirect relations could mitigate sparseness effectively.

Now, we will discuss the definition of indirect hierarchical structure. Through exploring the potential relationship between items' attributes, we can establish the hierarchical structure graph, shown in Fig. 1. The structure graph reveals the hierarchical relationships between items. For example, if two movies 1 and 2 have same attribute adventure, then the top layer rectangle indicates adventure. The bottom layer denotes the both items' other attributes apart from adventure. We view the straight line connecting both rectangles as edge, but different colors of edges having different weights. The weight mechanism of different edges is discussed in Formula 9.

Inspired by cosine formula, for any two given objects, we can now define their indirect hierarchical structure similarity formally in Eq. (6) below:

$$S_{IH}(i,j) = \frac{\sum_{a=1}^{m} w_a \left|e^i_{w_a}\right| + \sum_{b=1}^{n} w_b \left|e^j_{w_b}\right|}{\sqrt{\sum_{a=1}^{m} w_a^2 + \sum_{b=1}^{n} w_b^2}\sqrt{\sum_{a=1}^{m} \left|e^i_{w_a}\right|^2 + \sum_{b=1}^{n} \left|e^j_{w_b}\right|^2}} \tag{6}$$

where w_a, w_b denote the weight of different edges, and $\left|e^i_{w_a}\right|$ indicates the number of edges which have identical weight, w_a in item $i's$ hierarchical structure. In addition, $\left|e^j_{w_b}\right|$ represents number of edges which are equipped with weight w_b in item $j's$ hierarchy.

From Eq. (6), we can observe that the more similar the hierarchical structure, the more edge weights, numerators get bigger and denominator is relatively smaller according to set theory, the higher value is. Thus, it satisfies the calculation principle of hierarchical structure similarity.

Next, we will discuss the additional core problem of how to calculate the weights of different edges. Apparently, the weight is connected to the level of the edge, so we take the function of each edge's level into consideration. In this work, we adopt Eq. (7) to count weight as follows:

$$w_a = f(x) = \varepsilon^x = \varepsilon^{lev(e_a)} \tag{7}$$

Here, $f(x) = \varepsilon^x$ shows a certain computation rule, ε is the result related to the corresponding structure, and the computational formula is demonstrated in Eq. (8) as follows:

$$\varepsilon = \frac{N_{DC}}{N_{DC} + N_V} = \frac{N_{DC}}{N_{DC} + 1} \tag{8}$$

where N_{DC} represents the directly connected nodes, that is to say, the sum of a node's parent's number and its children's number is the current node and is assigned the value 1. N_V is determined by the item's structure and remains unchanged in its respective similarity matching calculation.

$x = lev(e_a)$ denotes calculation theorem associated with edge's level, as shown below in Eq. (9):

$$x = lev(e_a) = \begin{cases} 0 \text{ , directly connected node edge} \\ \frac{1}{2} \text{ , brother node edge} \\ 1 \text{ , others} \end{cases} \tag{9}$$

To clarify our idea about the indirect hierarchical structure of the items, we illustrate it with the following digestible and accessible example: We assume a simple movie recommender system containing several movies whose corresponding inherent hierarchy is shown above. In Fig. 1, the bottom layer denotes the items other attributes apart from the common character, the middle layer represents items and siblings (other items covered in the item category), the top layer indicates the items' common category. The top category of the left tree and the right tree are same. Different colors of edge correspond to different weights.

Then, the calculation process of the items' indirect hierarchical structure similarity is as follows:

Firstly, according to the Eq. (8), we can obtain $\varepsilon_1 = \frac{3}{4}$, $\varepsilon_3 = \frac{5}{6}$.

Secondly, the different edges' weights of items 1 and 3 are calculated according to Eq. (7) and (9), respectively.

For item 1, the green edge's weight is $(3/4)^{1/2}$, the brown edge's weight is $(3/4)^0$ and the black edge's weight is $(3/4)^1$. For item 3, the green edge's weight is $(5/6)^{1/2}$, the brown edge's weight is $(5/6)^0$.

Finally, acording to Eq. (6), we can calculate:

$$S_{IH}(1,3) = \frac{(3/4)^0*3+(3/4)^{1/2}*2+(3/4)^1*1+(5/6)^0*5+(5/6)^{1/2}*1}{\sqrt{((3/4)^0)^2+((3/4)^{1/2})^2+((3/4)^1)^2+((5/6)^0)^2+((5/6)^{1/2})^2}+\sqrt{3^2+2^2+1^2+5^2+1^2}} = 0.88$$

For the sake of comparing the calculated results, we also compute that the value of indirect hierarchical structure similarity between items 1 and 2 is 0.96, which is more than 0.88. Apparently, in the items' hierarchical structure trees, items 1 and 2 have more similar structures on account of sharing a common category.

3.1.3 Item-Item Comprehensive Similarity

In this section, we enlighten the item-item comprehensive similarity through combining hierarchical structure similarity and rating similarity. Next, we introduced the general formulations of hierarchical structure similarity and comprehensive similarity respectively.

Firstly, the hierarchical structure similarity $S_H(i, j)$ between two items, i and j, can be denoted formally as the linear combination of $S_{DH}(i, j)$ and $S_{IH}(i, j)$ by parameters α, β, such that $\alpha + \beta = 1$:

$$S_H(i, j) = \alpha S_{DH}(i, j) + \beta S_{IH}(i, j) \tag{10}$$

Ultimately, deriving from the above components, we eventually express the modified similarity between items based on Eq. (3) and (10) in Eq. (11) as follows:

$$S(i, j) = (1 - \lambda)\, sim(i, j) + \lambda S_H(i, j) \tag{11}$$

In this improvement, item-item's similarity is enriched, improving accuracy because it integrates not only rating similarity, but also items' potential hierarchical structure similarity. While there is no hierarchical relationship ($\lambda = 0$), the traditional similarity is adopt.

3.2 Score Prediction

As we describe previously, the user's preferences are sensitive to time. Therefore, changes of user interest over time should be taken into consideration in final rating prediction. Following this idea, we designed a function $f_{u,i}^{\omega}(t)$ to the time t for gradually reducing the influence of prior data and assign greater significance to recent data.

The core of our time-aware collaborative filtering strategy is the concept of temporal relevance function $f_{u,i}^{\omega}(t)$. It measures the relation of each rating for recommending at time t, and based on the hypothesis that a user's current taste correlates less with the older rating, should decrease from the perspective of intuition from the time the rating was input $t - t_{u,i}$). Our temporal decay function is defined as follows:

$$f_{u,i}^{\omega}(t) = \frac{1}{1 + e^{\omega(t - t_{u,i})}} \tag{12}$$

where t represents the right now time-stamp, $t_{u,i}$ is the moment when user u rates item i. Actually, we need the t-$t_{u,i}$, which denotes the time from now (t) to the moment when user u rates item ($t_{u,i}$). The t-$t_{u,i}$ is labeled as from 1 to 6 as referenced from [23]. ω is the delay rate.

$f_{u,i}^{\omega}(t)$ is a mathematical dlcay function(a monotonic decreasing function). The longer ago the user rates an item, the bigger t-$t_{u,i}$ will be, which makes $f_{u,i}^{\omega}(t)$ smaller. We emphasize the user's latest purchase interest and focus on the most recent data. Inspired by the reference [23], $f_{u,i}^{\omega}(t)$ is an exponential time function, which is very suitable in this case. Following the research of Paula [23], we reference its method for the calculation of time weights values.

Table 1 Time weights

t	1	2	3	4	5	6
f(t)	0.18	0.23	0.29	0.35	0.43	0.50

In the concrete implementation process, the whole data set is divided into six periods according to timestamp. Thus, the possible time function values are shown in Table 1, if $\omega = 0.3$.

Now, we will present how the score prediction of the target user on every target item is counted. Firstly, the whole similarities between a target item and each other item must be calculated. Secondly, top K items with the highest similarities are chosen as neighbors of the target item. Next, we predict the target user's rating for each target item. In our study, the Eq. (4) is revised in Eq. (12) as follows:

$$P_{u,i} = \frac{\sum_{j \in N_i} sim(i, j)\, r_{u,j} f_{u,i}^{\omega}(t)}{\sum_{j \in N_i} |sim(i, j)| f_{u,i}^{\omega}(t)} \tag{13}$$

4. Experimental Results and Evaluation

To examine the effectiveness of our proposed HTCF algorithm, we have done several experiments. We compare simultaneously the algorithm's performance with several collaborative filtering recommendation means. In this section, we begin with a description of the experimental datasets and the metric. Then, we proceed to the design and interpretation of experiments. In these tests, we adopt a 5-fold cross verification method. Experimental results are finalized via values averaged five times.

4.1 Dataset

We make use of two different datasets. One is the MovieLens dataset, which is the most standard test data of the research subjects concerning collaborative filtering technologies. It consisted of 1,000,209 ratings, where more than 900 users rated about 1,680 movies. In addition, every user averages ratings of no fewer than 20 movies. Scores are integers ranging from 1 to 5. The higher score indicates the user's greater like of the movie. In particular, every score is accompanied by an obvious timestamp, which allows us to address the issue of changes in user interest over time. Additionally, we also incorporated extremely valuable characters regarding categories and attributes, such as romantic or comedy and actors, to investigate hierarchical structure hidden within the items.

The other is Yahoo!Music dataset, which used in KDD CUP 2011. It is a relatively new dataset, which has 262,810,175 ratings of 624,961 music items by 1,000,990 users and each user and item have at least 20 ratings and genres. A distinctive nature of it is that there are four kinds of musical items: tracks, albums, artists, genres, tied together within a known taxonomy, and forming a four level hierarchy.

All the experiments were conducted with datasets divided as an 80%-20% train-test ratio.

Table 2 The optimal *MAE* with α, λ, ω changed as *step*=0.1 for MovieLens dataset

	C1	C2	C3	C4	C5	C6	C7	C8	C9
step	α	λ	ω	change of α	change of λ	change of ω	ABS(C4)	ABS(C5)	ABS(C6)
	0.76289	0.748153	0.741314	-0.11354	0.023351	-0.02458	0.113545	0.023351	0.024583
0.1	0.751536	0.750488	0.738856	-0.08222	-0.14388	-0.01701	0.082225	0.14388	0.017009
0.2	0.743313	***0.7361***	0.737155	-0.07213	0.000351	-8.8E-05	0.072132	0.000351	8.84E-05
0.3	***0.7361***	0.736135	0.737146	0.011521	0.003729	-0.01046	0.011521	0.003729	0.010462
0.4	0.737252	0.736508	***0.7361***	0.045001	0.007306	0.007579	0.045001	0.007306	0.007579
0.5	0.741752	0.737238	0.736858	0.080362	0.014642	0.415731	0.080362	0.014642	0.415731
0.6	0.749788	0.738703	0.778431	0.128628	0.013413	-0.19982	0.128628	0.013413	0.199818
0.7	0.762651	0.740044	0.758449	0.193729	0.016259	-0.09667	0.193729	0.016259	0.096671
0.8	0.782024	0.74167	0.748782	-0.01652	0.017804	-0.04538	0.016523	0.017804	0.045381
0.9	0.780372	0.74345	0.744244	-0.08705	0.021053	-0.01693	0.087054	0.021053	0.016933
1	0.771666	0.745556	0.742551	———-	———-	———-	**0.083072**	**0.026179**	**0.083425**

Table 3 Liner regression results

Model		Unstandardised Coefficients		Standardised Coefficients		
		B	Std. Error	Beta	t	Sig.
1	Constant	.747	.001		521.419	.000
2	α	.029	.002	.442	18.424	.000
3	λ	.000	.002	-.005	-.189	.850
4	ω	.013	.002	.201	8.394	.000

4.2 Evaluation Criteria

To estimate the accuracy of the prediction scores, we employ the Mean Absolute Error (MAE), which is the most classical and commonly practical measurement benchmark in the territory of recommender system [24]–[27] as its simplification and perceptual comprehension in calculations. The MAE is a metric of deviation between predicted ratings and user's actual values. For a given dataset, we will practice model in the training set and predicate in the test set, and then compute the MAE, so it is determined by calculating average value of all the unseen item's predicted grades compared to real values. It can be given formally (14) by:

$$MAE = \frac{\sum_{i=1}^{n} \left|p_{u,i} - r_{u,i}\right|}{n} \tag{14}$$

where $p_{u,i}$ identifies the predicted rating the user generated to the item i and $r_{u,i}$ is the real rating of a hidden item i the user existed in the test dataset and n stand for the number of tested ratings. As we known, the lower the value of MAE is the higher prediction accuracy the recommendation algorithm achieves.

4.3 Parameter Adjustment

Recall that the parameter α and β indicate the weight of direct and indirect hierarchical structure similarity respectively. And the sum of both is equal to 1. λ controls the ratio of traditional rating similarity and hierarchical structure similarity in our proposed comprehensive similarity calculation. When λ is equal to 0, we could obtain the standard item-based collaborative filtering. Otherwise, we could adjust the parameter values of λ and α to take into account hierarchical structure similarity. Apart from the above parameters, ω also plays an important role. Time function give more weight to recent rated items, less relevance to the past, and ω controls the decline rate related to time.

Table 4 Methods for comparison

Method	Description
UBCF	User-based CF (Sect. 2.1)
IBCF	Item-based CF (Sect. 2.1)
HCF	CF with hierarchical structure (Sect. 3.3)
TCF	CF with temporal relevance (Sect. 4.1)
GIS-GD	Combination of GIS,GD and proposed by Parivash [27]
HTCF	Hierarchical structure and temporal relevance (Sect. 3)

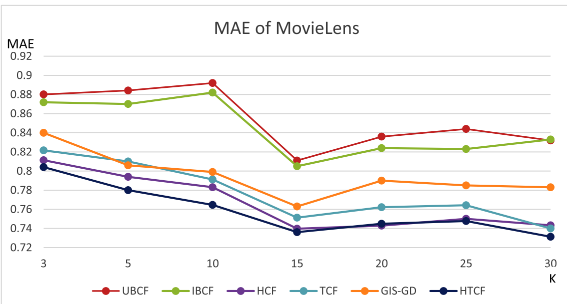
In order to evaluate the effectiveness of proposed algorithm HTCF, we test all the value of α, β, λ and ω from 0 to 1 with step is 0.1. Trough 11*11*11 times experiments, we find the best optimal MAE is got while $\alpha = 0.3, \beta = 0.7$, $\lambda = 0.2$, and $\omega = 0.4$, using MovieLens dataset. Similarly, the best results were obtained when $\alpha = 0.3, \beta = 0.7$, $\lambda = 0.3$, and $\omega = 0.4$ with music dataset.

To understand how these parameters work and which parameters are important to accomplish the better performance than the traditional methods, we measured the performances of HTCF as we increased the value of α from 0 to 1 by 0.1, while holding the other parameters at their optimal values which are obtained by the experiments. The MAE with the step of 0.1 of α, λ, ω as shown in table 2. For example, on the evaluation of α in MovieLens dataset, we keep $\lambda = 0.2$ and $\omega = 0.4$ while changing α from 0 to 1 by 0.1. Similarly, we keep $\alpha = 0.3$ and $\omega = 0.4$ while changing λ from 0 to 1 and we keep $\alpha = 0.3$ and $\lambda = 0.2$ while changing ω from 0 to 1.

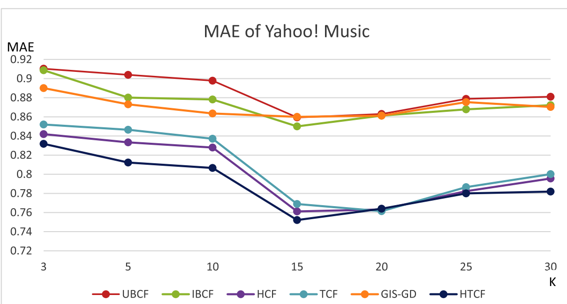
In Table 2, we also calculate the change rate of MAE according to α, λ, ω separately. The change rate is calculated to verify the effect to MAE while the parameters changed. We also use the absolute change rate to represent the effect. In the bottom of table, the bold numeric is the average rate with parameters from 0 to 1. The average change

Table 5 *MAE* comparison of proposed methods with the others

	MovieLens						Yahoo!Music					
	UBCF	IBCF	HCF	TCF	GIS-GD	HTCF	UBCF	IBCF	HCF	TCF	GIS-GD	HTCF
K=3	0.8801	0.8719	0.8113	0.8216	0.8400	0.8041	0.9103	0.9087	0.8419	0.8520	0.8900	0.8318
K=5	0.8842	0.8701	0.7939	0.8100	0.8060	0.7800	0.9039	0.8801	0.8333	0.8465	0.8731	0.8123
K=10	0.8920	0.8820	0.7831	0.7912	0.7990	0.7646	0.8978	0.8782	0.8279	0.8371	0.8635	0.8066
K=15	0.8110	0.8050	0.7397	0.7513	0.7630	0.7361	0.8594	0.8500	0.7612	0.7689	0.8600	0.7521
K=20	0.8360	0.8240	0.7429	0.7622	0.7900	0.7449	0.8630	0.8613	0.7632	0.7614	0.8612	0.7640
K=25	0.8440	0.8230	0.7501	0.7643	0.7850	0.7477	0.8788	0.8678	0.7823	0.7865	0.8754	0.7800
K=30	0.8320	0.8330	0.7433	0.7400	0.7830	0.7313	0.8810	0.8722	0.7956	0.8001	0.8703	0.7819



(a) Movielens(MAE)



(b) Yahoo!Music(MAE)

Fig. 2 *MAE* comparison of proposed HTCF with the UBCF, IBCF, HCF, TCF, GIS-GD

rate shows that the α, ω are important to MAE decreasing compared to λ.

Further more, we uss liner regression to analysis the 11*11*11 *MAE* data, and we get the result as shown in Table 3. From Table 3, the Sigs of α and ω are close to 0. So, the parameters of α and ω more important to to accomplish the better performance.

We use Table 2 and Table 3 from two view to proof that the α and ω are the main improvement to increase the recommendation accurate.

4.4 Performance Comparison

In this section, we estimate several recommender approaches listed in Table 4. We make experiments for various neighbors (K) from 0 to 30 by 5, and the results on MAE are shown in Table 5 and Fig. 2.

According to Table 3, we proposed the combination of hierarchical structure and time information based collaborative filtering (**HTCF**) outperforms other approaches. And $K = 15$ is the optimal value for two datasets. Additionally, we can see from Fig. 2 that HCF (which only considers hierarchical structure) outperforms TCF (which only considers temporal relevance based on the traditional collaborative filtering).

5. Conclusions and Future Work

In traditional CF recommendation methods, item's hierarchical structure and user's preference drifts are not considered. However, they shouldn't be ignored in real recommender system. Therefore, we explore a novel similarity measure method based on item's hierarchical structure and exploit a time decay function for weighting rating. A distinct feature of HTCF is that we propose a new structure similarity measure for item hierarchical structure. Such evolution can overcome above barriers of traditional collaborative filtering technology. We perform a series of experiments to evaluate HTCF and compare our method with several state-of-art CF recommender algorithms. According to the results of experiments, we demonstrate the advantages of HTCF. In particular, our system results in the improvement of recommendation quality of CF-based technology.

However, our approach also has a limitation. We need to select the most optimal values of given parameters by a series of experiments. However, they are different for different datasets. Therefore, the performance of our proposed algorithm will decrease when selecting most proper parameters' values.

In future work, we will enhance our algorithm by considering parallel computing method for improving the efficiency of our algorithm on larger scale datasets. That is to say, how to combine HTCF algorithm and hadoop so as to enhance the speed of similarity calculation.

Acknowledgments

This work was supported in part by National Science Foundation of China (No. 61572259), Special Public Sector Research Program of China (No. GYHY201506080) and was also supported by PAPD.

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-264.

References

[1] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending

and evaluating choices in a virtual community of use," Proceeding CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp.194–201, 1995.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web, ACM New York, NY, USA, pp.285–295, 2001.

[3] B. Marlin, Collaborative filtering: A machine learning perspective, Master's thesis, University of Toronto, 2004.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," Internet Computing, IEEE, vol.7, no.1, pp.76–80, 2003.

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

[6] J. Davidson, B. Livingston, D. Sampath, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, and M. Lambert, "The YouTube video recommendation system," Proceeding RecSys '10 Proceedings of the fourth ACM conference on Recommender systems, ACM New York, NY, USA, pp.293–296, 2010.

[7] M.J. Pazzani and D. Billsus, Content-based recommendation systems, The Adaptive Web Volume 4321 of the series Lecture Notes in Computer Science, Springer, pp.325–341, 2007.

[8] J.B. Schafer, D. Frankowski, J. Herlocker et al., Collaborative filtering recommender systems, The Adaptive Web Volume 4321 of the series Lecture Notes in Computer Science, Springer, pp.291–324, 2007.

[9] M.A. Ghazanfar, A. Prugel-Bennett, "A scalable, accurate hybrid recommender system," Knowledge Discovery and Data Mining, pp.94–98, 2010.

[10] T. Ma, J. Zhou, M. Tang, Y. Tian, A. Al-dhelaan, M. Al-rodhaan, and S. Lee, "Social network and tag sources based augmenting collaborative recommender system," IEICE Trans. Inf. & Syst., vol.E98.D, no.4, pp.902–910, 2015.

[11] J.-H. Li, X.-S. Li, H.-L. Liu, X.-J. Han, and J. Zhang, "Fuzzy collaborative filtering approach based on semantic distance," Fuzzy Information and Engineering Volume 2, Springer, vol.62, pp.187–195, 2009.

[12] M.I. Martln-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J.J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores, "A semantic approach to improve neighborhood formation in collaborative recommender systems," Expert Systems with Applications, vol.41, no.17, pp.7776–7788, 2014.

[13] J. Herlocker, J.A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," Information retrieval, vol.5, no.4, pp.287–310, 2002.

[14] M.R. McLaughlin and J.L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," Proceeding SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM New York, NY, USA, pp.329–336, 2004.

[15] L. Candillier, F. Meyer, and M. Boullé, Comparing state-of-the-art collaborative filtering systems, Machine Learning and Data Mining in Pattern Recognition, Springer, pp.548–562, 2007.

[16] C. Desrosiers and G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, Recommender systems handbook, Springer, pp.107–144, 2011.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, Proceeding CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work, ACM New York, NY, USA, pp.175–186, 1994.

[18] W.S. Lee, "Collaborative learning for recommender systems," Proc. 18th International Conference on Machine Learning, pp.314–321, 2001.

[19] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy," Proceeding RecSys '11 Proceedings of the fifth ACM conference on Recommender systems, ACM New York, NY, USA, pp.165–172, 2011.

[20] Y. Blanco-Fernández, J.J. Pazos-Arias, M. López-Nores, A. Gil-Solla, M. Ramos-Cabrer, J. García-Duque, A. Fernández-Vilas, and R.P. Díaz-Redondo, "Incentivized provision of metadata, semantic reasoning and time-driven filtering: Making a puzzle of personalized e-commerce," Expert Systems with Applications, vol.37, no.1, pp.61–69, 2010.

[21] Y. Blanco-Fernández, M. López-Nores, J.J. Pazos-Arias, and J. García-Duque, "An improvement for semantics-based recommender systems grounded on attaching temporal information to ontologies and user profiles," Engineering Applications of Artificial Intelligence, vol.24, no.8, pp.1385–1397, 2011.

[22] A.N. Nikolakopoulos, M. Kouneli, J. Garofalakis, A novel hierarchical approach to ranking-based collaborative filtering, Engineering Applications of Neural Networks, Springer, pp.50–59, 2013.

[23] P.C. Vaz, R. Ribeiro, D.M. de Matos, "Understanding temporal dynamics of ratings in the book recommendation scenario," Proceeding ISDOC '13 Proceedings of the 2013 International Conference on Information Systems and Design of Communication, ACM New York, NY, USA, pp.11–15, 2013.

[24] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems (TOIS), vol.22, no.1, pp.5–53, 2004.

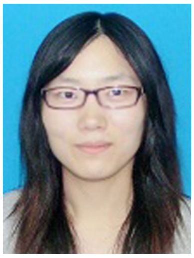
[25] B. Xie, P. Han, F. Yang R.-M. Shen, H.-J. Zeng, and Z. Chen, "DCFLA: A distributed collaborative-filtering neighbor-locating algorithm," Information Sciences, vol.177, no.6, pp.1349–1363, 2007.

[26] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, and S. Li, "Incremental Support Vector Learning for Ordinal Regression," IEEE Trans. Neural Netw. Learning Syst., vol.26, no.7, pp.1403–1416, 2015.

[27] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding," IIEEE Trans. on Broadcast., vol.61, no.2, pp.166–176, 2015.

[28] T. Ma, H. Rong, C. Ying, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Detect structural-connected communities based on BSCHEF in C-DBLP, Concurrency and Computation: Practice and Experience, 2015.

Tinghuai Ma is a professor in Computer Sciences at Nanjing University of Information Science & Technology, China. He received his Bachelor in 1997 (HUST, China), Master in 2000 (HUST, China), Ph.D. in 2003 (CAS, China) and was Post-doctoral in 2004 (AJOU University, Korea). From Feb. 2009 to Aug. 2009, he was a visiting professor in Ubiquitous computing Lab, Kyung Hee University. His research interests are data mining, grid computing, ubiquitous computing, privacy preserving etc.



Limin Guo received her Bachelor degree in Software Engineering from Nanjing University of Information Science & Technology, China in 2013. Currently, she is a candidate for the degree of Master of Computer Science in Nanjing University of Information Science & Technology. Her research interest is recommender system.

Meili Tang is an associate professor in School of Public Administration at Nanjing University of Information Science & Technology, China. She received her master degree from Huazhong University of Science & Technology, China, 2000. Her main research interests are in the areas of e-government and data publishing.

Yuan Tian has received her master and Ph.D. degree from KyungHee University and she is currently working as Assistant Professor at College of Computer and Information Sciences, King Saud University, Kingdom of Saudi Arabia. She is member of technical committees of several international conferences. In addition, she is an active reviewer of many international journals. Her research interests are broadly divided into privacy and security, which are related to cloud computing, bioinformatics, multimedia, cryptograph, smart environment, and big data.

Mznah Al-Rodhaan has received her B.S. in Computer Applications (Hon) and M.S. in Computer Science both from King Saud University on 1999 and 2003 respectively. In 2009, she received her Ph.D. in Computer Science from University of Glasgow in Scotland, UK. She is currently working as the Vice Chair of the Computer Science Department in College of Computer & Information Sciences, King Saud University, Riyadh, Saudi Arabia. Moreover, she has served in the editorial boards for some journals such as the Ad Hoc journal (Elsevier) and has participated in several international conferences. Her current research interest includes Mobile Ad Hoc Networks, Wireless Sensor Networks, Multimedia Sensor networks, Cognitive Networks, and Network Security.

Abdullah Al-Dhelaan has received B.S. in Statistics (Hon) from King Saud University, on 1982, and the M.S. and Ph.D. in Computer Science from Oregon State University on 1986 and 1989 respectively. He is currently the Vice Dean for Academic Affairs, Deanship of Graduate Studies and a Professor of Computer Science, King Saud University, Riyadh, Saudi Arabia. He has guest edited several special issues for the Telecommunication Journal (Springer), and the International Journal for Computers and their applications (ISCA). Moreover, he is currently on the editorial boards of several journals and the organizing committees for several reputable international conferences. His current research interest includes: Mobile Ad Hoc Networks, Sensor Networks, Cognitive Networks, Network Security, Image Processing, and High Performance Computing.