Research Article

International Journal of Advanced Computer Research, Vol 7(29)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2017.728008

An improvement on recommender systems by exploring more relationships

Hoang Lam Le*, Quoc Cuong Nguyen and Minh Tri Nguyen
Institute of Science and Technology of Industry 4.0, Nguyen Tat Thanh University, 300A Nguyen Tat Thanh, district 4, Hochiminh City, Vietnam

Received: 15-November-2016; Revised: 27-January-2017; Accepted: 03-February-2017
©2017 ACCENTS

Abstract
Recommender systems are systems that can filter a great number of pieces of data and suggest mostly similar interested items of the user's preference. A variety of approaches have been proposed to perform recommendation, including content-based, collaborative filtering and association-based, etc. A potential problem existing in a recommender system is cold start [1]; simply defined that a system cannot draw any inference for users. In this paper, we deal with one of cold start problems by proposing a hybrid approach which combines two distinct features to solve the problem. While a user is related to other users in product purchase behaviors or preference, an item is connected to different items by its inside information. Our recommender system utilizes both these relations instead of each individual one to ameliorate the quality of output suggestion. This procedure will be revealed and discussed through this paper.

Keywords
Cold start, Recommendation, Recommender, Collaborative filtering, Content-based, Hybrid approach.

1.Introduction
In recent years, the development of networking technology has not only accelerated up the distribution and access of online information, but also fostered conducting online marketing or business transactions on the internet. The huge data in this global information space are increasing so rapidly that a human cannot manually process or handle. This challenge has led to the establishment of recommender systems [2], which are typically personalized for each specific individual. Recommender systems represent user preferences by suggesting items closest matched to user's intention. They effectively prune huge information and direct users toward the items that best meet their needs and preferences. Because of this significant contribution, they have become the essential factor in electronic commerce and information access.

Several approaches have been proposed to implement these recommender systems, including content-based, collaborative filtering, association-based and other techniques [3]. All of the known recommendation approaches have strengths and weaknesses.

A common problem with these systems is cold start which refers to situations where the system cannot draw any recommendations because of the lack of sufficiently necessary information [4]. The cold start problem can occur in these following three scenarios:
New user: when a new user has just joined to the system and his preferences are not yet known.
New item: when a new item has just been added to the database and not yet received enough ratings.
New system: when a new recommender system has launched, the average number of ratings per user and item is low; it thus significantly decreases the performance of collaborative algorithms.

In this paper, we contribute a new approach to resolve the new item problem by using a combination of user and item features. This hybrid method's empirical result shows a better performance in comparison to single feature usage. The rest of this paper is organized as follows: firstly, we mention related works on this research field. Secondly, we introduce three possible approaches to solve cold start sub-problem new item. Next section present experiment outputs on Netflix dataset. Finally, the conclusion part summarizes this research, then nominates an improvement in the current approach as well as proposes a new solution for another Cold start sub-problem new user, based on this research's experiment procedure.

*Author for correspondence

2.Related works

2.1Collaborative filtering recommendation approach

2.1.1User-based collaborative filtering recommendation approach

The basic idea of these systems (as depicted in *Figure 1*) is that if some users shared the similar interests in the past-for instance: users checked, purchased or subscribed a channel-they will also incline to do the same things in the future [5]. This algorithm will firstly select a set of target user's acquaintances who ever rated several items with the target user. Additionally, it is compulsory that these acquaintances rated the target item before. Then, for the target product that the target user has not yet seen, a prediction is computed based on the ratings for this item made by the peer users. Several different similarity measurements of the users a and b have been proposed in [6, 8-11]

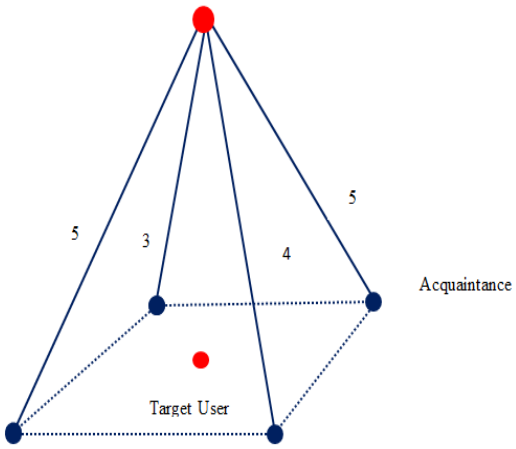

Figure 1 User-based collaborative filtering

2.1.2Item-based collaborative filtering recommendation approach

Item-based collaborative filtering algorithms capture the fundamental relationships among items, but not users as shown in *Figure 2*. Two items are related if the community agrees about their ratings by giving the same or almost similar score [7]. First, we look for items in the dataset that have ratings similar to the target product. The closer they are, the stickier these items belong to the same class or quality standard. After that, the algorithm will compute a weighted average that the target user has given to these similar items. This approach could be considered as the user-based method, but the roles of user and item are interchangeable. Hence, the equations used to calculate similarity and mean are applicable to not only user-based but also item-based collaborative filtering, too.

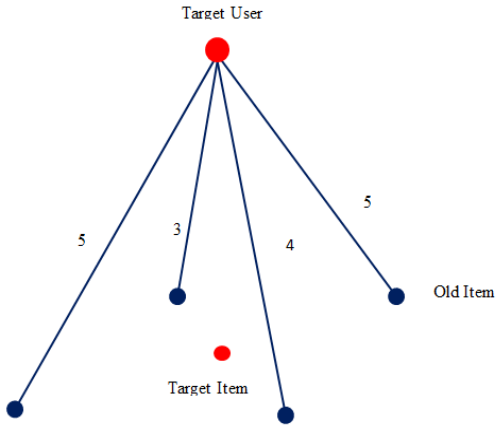

Figure 2 Item-based collaborative filtering

2.2Content-based recommendation approach

The previously presented user-based and item-based techniques do not exploit the existing information on the items themselves. Therefore, these methods could avoid either providing or updating item descriptions that are normally considered as a tough and costly task. However, with these simple and naïve collaborative filtering approaches, complex and detailed requirement tasks or user's practically specific preferences on a product's characteristics are hardly possibly completed. As a result, content-based approach, the method of selecting recommendable items based on their characteristics and the specific preferences of a user are employed to tackle this drawback. This recommender system needs, the availability of two pieces of information: item characteristic description and user's past interest profile [5]. The recommendation system will compare the two pieces of information and suggest some items which are closest matching to the user's preferences. This process is commonly called content-based recommendation, which basically recommend items similar to what the user used to make [9]. Thus, this approach automatically learns and adaptively updates the interest of the user's profile accordingly, which is also known as relevance feedback. Even though such way has to count on item's providing information and user's preference profile, it does not need the existence of a big user dataset or a rating history. Consequently, a recommendable item list could be generated even if there is only one individual. The content-based recommendation approach (*Figure 3*) contains four phases:

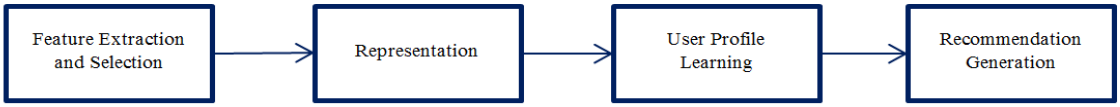

Figure 3 Content-based recommendation approach phases

Feature extraction and selection (first phase) – processes all items in collecting the dataset to take out representative vector.
Representation (second phase) –replaces each item by the feature vector in the dataset.
User profile learning (third phase) –upgrades the user profile model to suit different conditions based on the training examples directly related to the user.
Recommendation generation (fourth phase) –produce suggestions using the latest user profile model.

2.3Association-based recommendation approach

This process looks for and discovers the co-occurrence rules of two or more items in a transaction [6]. A typical example of this approach is the product pair disclosure in the supermarket, which means an item highly possibly appears together with another product. As a result, if a customer purchase product A, he is also concerned to buy B with high probability if A and B items are closely correlative.

2.4Hybrid approach

There are a few hybrid approaches [12] which combine two or more single methods to build a better model such as: weighted, switching, cascade, mixed, feature combination, feature augmentation and meta-level, etc.

- *Figure 4* shows the weighted hybrid recommender which aggregates weights of collaborative methods. It calculates the mean value of other recommenders' outcomes.

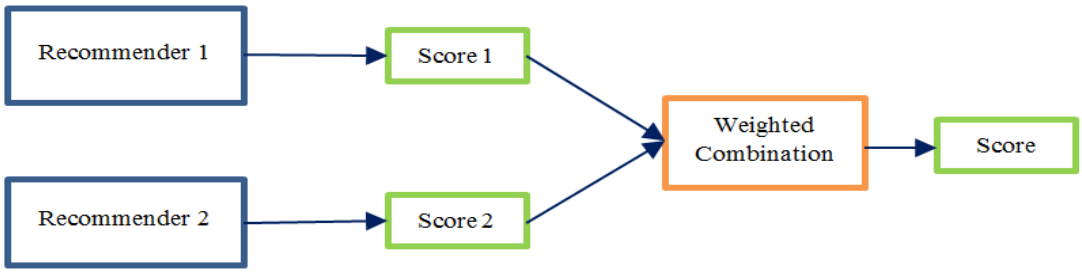

Figure 4 The weighted hybrid recommender system

- The switching system as shown in *Figure 5* switches distinct techniques relying on the specific circumstance. Depend on a certain condition, the system decides which recommender program is the most preferred choice among obtained candidates, and uses that recommender to perform the task.

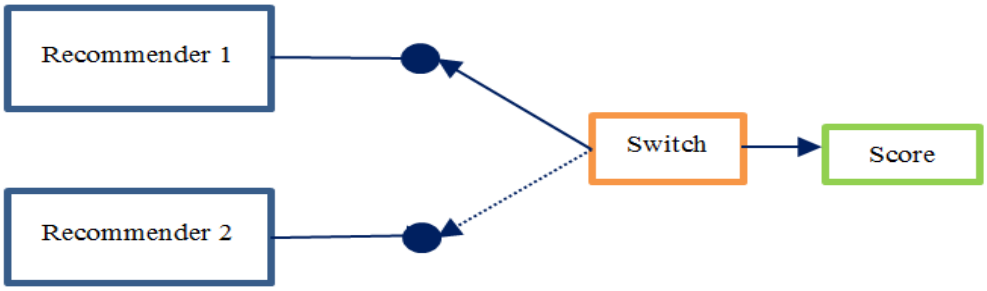

Figure 5 The switching system can switch and choose the most preferred recommender

- The mixed model provides suggestions from many models simultaneously. It just presents all sub-recommenders' outcomes as presented in the left part of *Figure 6*.
- The right part of *Figure 6* is the hybrid feature combination system. This hybrid approach concatenates multi single vectors to form a longer final representation and this vector is utilized in a single recommendation algorithm.

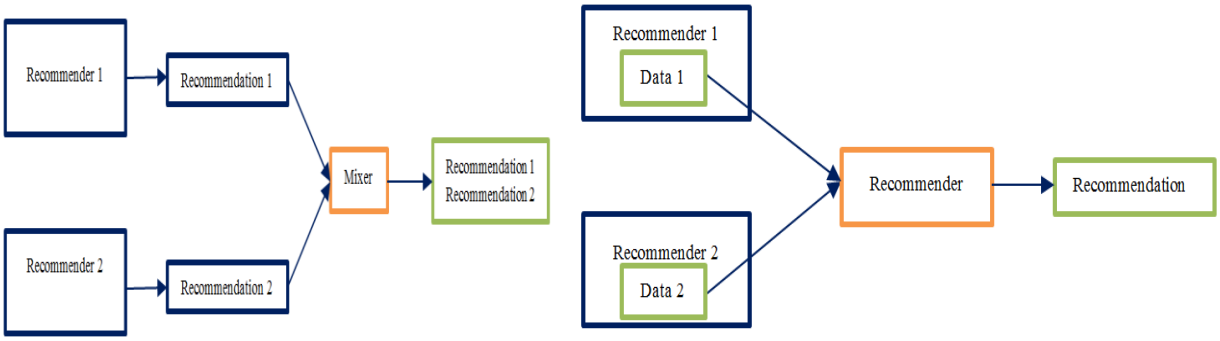

Figure 6 Mixed and feature combination system

- Cascade system model is depicted in *Figure 7*. This technique employs one method to yield recommendable candidates and the second method then refines to show the final list. Particularly, the first recommender tries to rank as many items as possible, then leaves the unranked items for its successors.

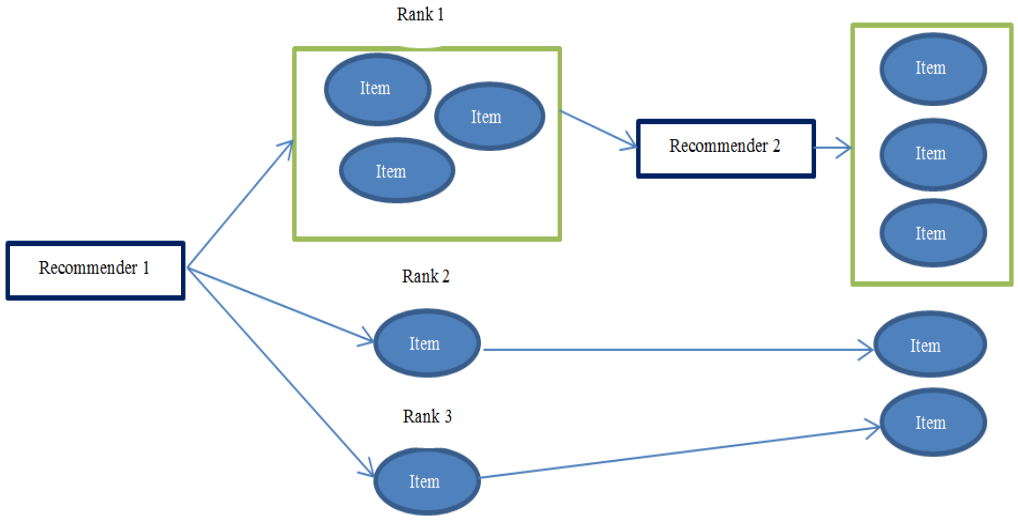

Figure 7 Cascade method

- *Figure 8* shows the method of feature augmentation. This approach takes advantage of a technique to supply a rating score and that information is merged into the processing procedure of the other technique.

Figure 8 Feature augmentation method

- Meta-level hybrid approach: one entire model becomes the input for another model, so that two recommendation techniques are jointed into a hybrid model as illustrated in *Figure 9*.

Figure 9 Meta-level method

After consideration, we conclude that it is impossible to arrive at the best prediction in the cold start situation. Because of the lack of information about a user's preference over new item, achieving a perfect prediction at first try is really a big challenging task to complete. That's why we decide to start with some not-so-wrong predictions at first, then utilize them as a foundation to arrive at a better result. In other words, our experimental navigation is that a better prediction will be conducted on the previously simple predictions. In later discussion, we will show you how we used item-based collaborative filtering combined with items' descriptions to find an initial prediction. The next step is to use user-based collaborative filtering to connect those predictions which form a solid foundation to achieve a better subsequent prediction.

3.Three experimental approaches

3.1First approach

The first experiment is executed on item-based collaborative filtering. The algorithm (*Figure 10*) looks into the set of items the target user has rated and calculates the similarity between new item and old items. In order to determine the similarity value between item and item, a list of users who have rated both items are selected and a similarity method is then applied to determine the similarity measure between two items.

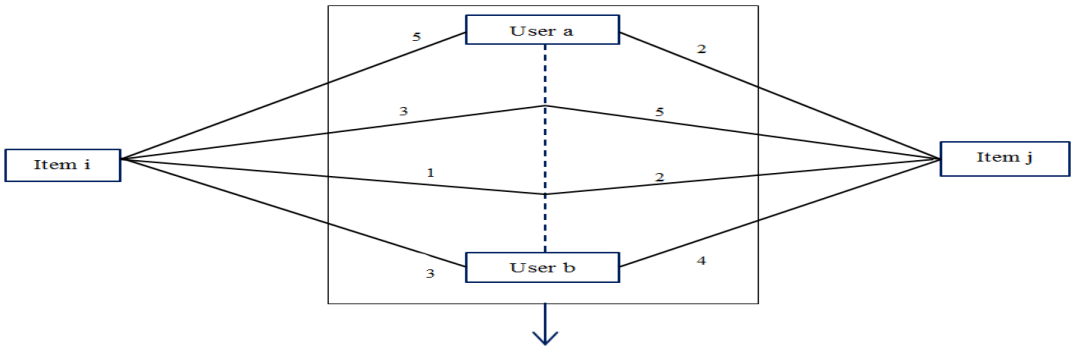

Figure 10 Items similarity computation

In case of the new item has not been rated by any user, we propose a solution as drawn in *Figure 11* that the new item's features are taken into consideration instead of using rating scores.

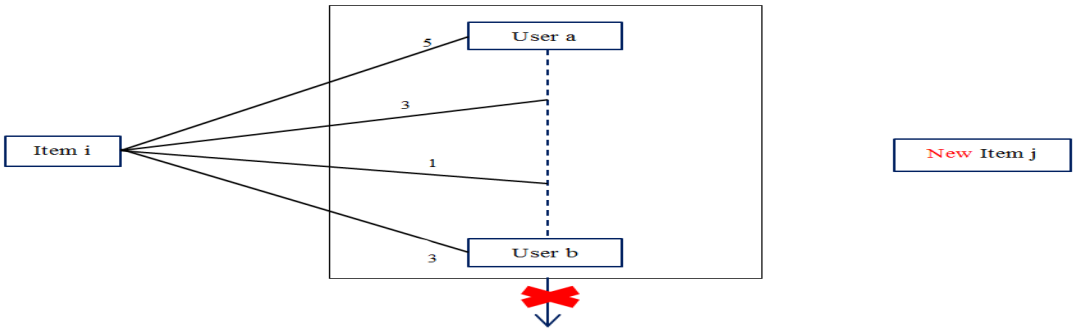

Figure 11 Items similarity computation with new item

The formula of computing similarity is written as:

$$sim(i,j) = \vec{f_i} \cdot \vec{f_j}$$

Where:

$sim(i,j)$: The similarity between item i and item j.

$\vec{f_i}$: The feature vector of item i.

$\vec{f_j}$: The feature vector of item j.

The value of this formula is the count of shared features between item i and item j. The below *Figure 12* explains in details this formula.

	$\vec{f_1}$	$\vec{f_2}$
Action	0	1
Sci-fi	1	1
Adventure	1	0
Drama	1	1
Horror	0	1

Figure 12 Example of calculation item-item similarity: sim(i,j)=2

After computing similarity scores between new item and old items, the predicted rating is computed by using the below weighted average expression:

$$p_{ai} = \frac{\sum_{j=1}^{k} sim(i_i, i_j) p_{aj}}{\sum_{j=1}^{k} sim(i_i, i_j)}$$

3.2Second approach

While the first approach only takes the item-item relationship into consideration, the second approach is going to consider the user-item correlation by using user profile.

$$sim(i,j) = \vec{f_i} \cdot \vec{f_j} \cdot \vec{p_u}$$

Where:

$sim(i,j)$: The similarity between item I and item j.

$\vec{f_i}$: the feature vector of item i.

$\vec{f_j}$: the feature vector of item j.

$\vec{p_u}$: the profile vector of target user.

The value of this formula is the sum of the profile features of the shared features between items i and item j. This makes the similarity score between two items become more subjective and personalized since it depends much on the target user's opinion. *Figure 13* is an example of calculating item-item similarity by using user profile.

	$\vec{f_1}$	$\vec{f_2}$	$\vec{p_u}$
Action	0	1	15
Sci-fi	1	1	32
Adventure	1	0	14
Drama	1	1	0
Horror	0	1	22

Figure 13 Example of calculating item-item similarity by using user profile: $sim(i,j) = 32$

3.3Third approach

As the method user-based collaborative filtering [7] presented, a set of acquaintances of target user are selected. These acquaintances and target user used to co-rate the same items in the past. However, the target item might be brand new and has not received any rating score yet. We propose the predicted rating score, which means that a friend "may give this score to new item". This approach is executed by following procedure.

- Firstly, select a set of old items which the target user has already rated, as depicted in *Figure 14*.

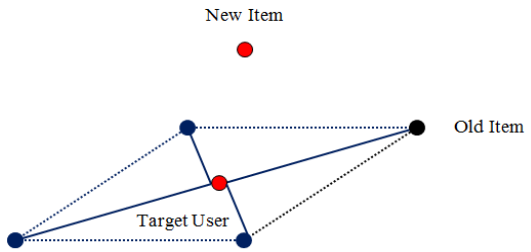

Figure 14 Select a set of old items the target user has already rated

- Secondly, find the users who also rated those items as shown in *Figure 15*.

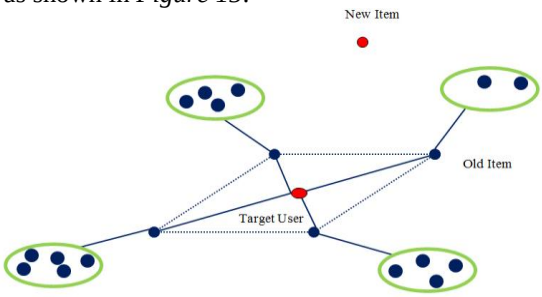

Figure 15 The users who rated the same items as the target user did

- Thirdly, join all users at each item to obtain a set of acquaintances. Then, we calculate the similarity between a target user and each set acquaintances using Pearson's correlation coefficient.

- *Figure 16* shows the similarity of a target user and his acquaintances as well as a formula to compute it in this step.

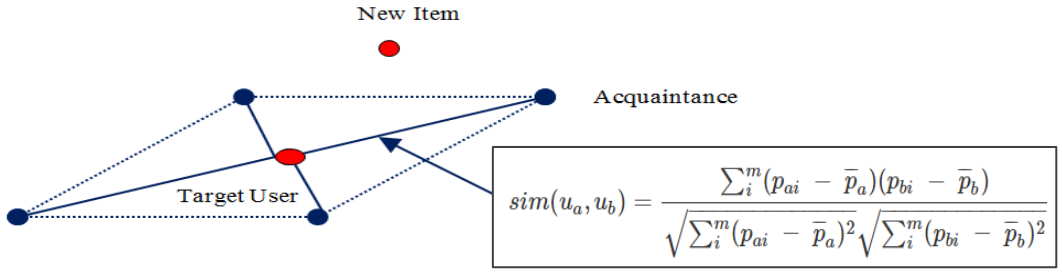

Figure 16 Compute the similarity between target user and a set of acquaintances

- Fourthly, k (in our experiment k = 100) closed sets of acquaintances – called friends - will be selected based on the similarity score. Each friend will predict a rating value for the new item as the explanation in the second approach (see *Figure 17*).

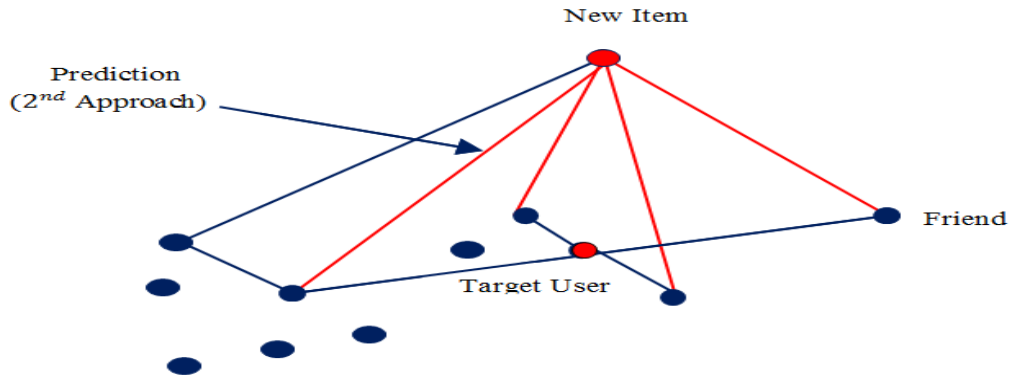

Figure 17 Each friend predicts a rating value for the new item

- Finally, we calculate the rating value that the target user may give to new item using z-score average. This step is summarized in *Figure 18*

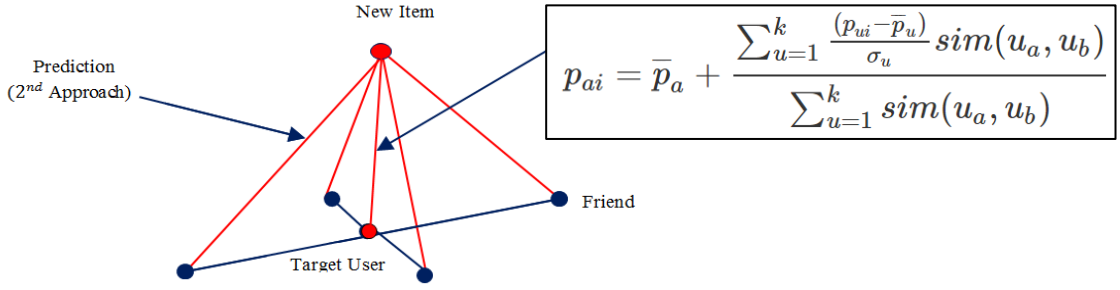

Figure 18 The target user rates the target item

This approach (*Figure 19*) takes not only user-item, item-item but also user-user relationship into consideration and applies the weighted method in feature augmentation hybrid method.

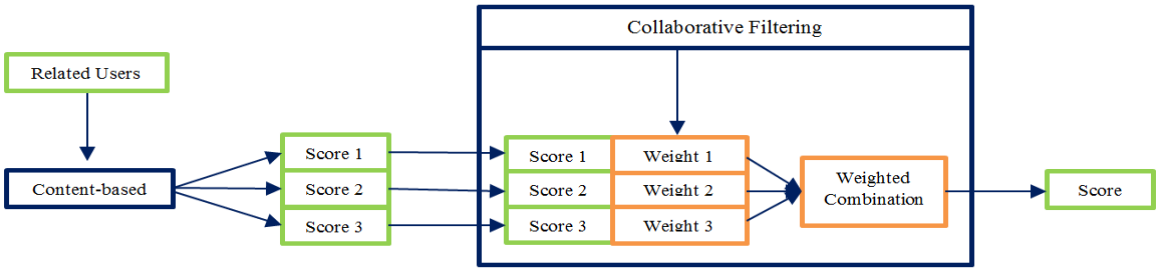

Figure 19 The third approach model

4.Experiment

4.1Dataset description

We do experiments on the movie rating dataset Netflix, containing over 100 million comments from 480 thousand randomly-chosen, anonymous customers over 17 thousand movie titles. The data were collected from October, 1998 to December, 2005 so that it sufficiently reflects the distribution of all ratings on a scale from 1 to 5 (integral) stars. To avoid customer privacy violation, each user id has been replaced by a randomly assigned id but we still keep data rating date, title and year of release. Three above explained approaches are taken into consideration for comparison and utilize max error and root mean squared error (RMSE) as the evaluation metric, defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - r_i)^2}{n}}$$

Where:

p_i the i-th predicted value.

r_i the i-th real value.

n the number of pairs of predicted and real values.

4.2Evaluation

The result of our experiments in *Figure 20* is shown through the scattered error data and error comparison table. The first experiment explored the item-item relationship based on item similarity; the second approach took advantage of the item-item and user-item relationships, while the third method utilized multi-correlations of the item-item, user-item and user-user. *Figure 19* depicts blue triangles gathered under y=2 while black rectangle and the red circle are also spread from y=2 to y=4. This diagram expresses that the third approach which explored more correlative relations among items can reduce the errors.

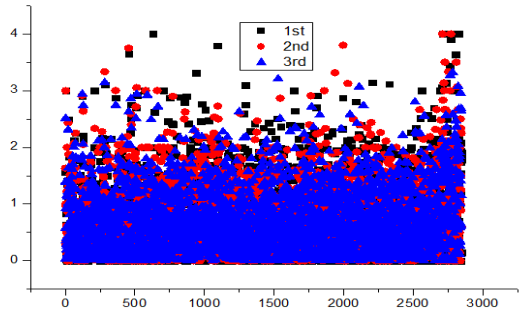

Figure 20 Scattered error of three approaches

Table 1 Error comparison

Approach	RMSE	Max error
1st approach	1.01	4.0
2nd approach	0.97	4.0
3rd approach	0.92	3.31

Observing the values of max error and RMSE as in *Table 1*, there is a relative improvement in accuracy from the first experiment to the second one, but the max error is still unchanged; however, it decreases significantly in the third experiment and the obtains a small max error less than 3.4. These results indicate a point that the more relationships we explore between entities in dataset, the better performance a program can execute. Our new approach has shown the much enhancement in our recommender system.

5.Conclusion and future work

Cold start problem occurs when: (1) a new user has joined the system, but their preferences are not yet known, (2) a new item has been added to the database but has not yet received sufficient ratings to be recommendable, (3) a new system has been built recently, the average number of ratings per user and item is low. This research focuses on solving the sub-problem (2): new item by comparing three approaches' performance. The result showed a better output from the first to the third approach since more relationships are employed. Our future work in this

field of studying is to refine the third approach because the predictions from acquaintances have not yet been explored. The fourth approach keeps the same scope, but further analysis as the repetition of the third approach. Each prediction value will be refined by repeating multiple times of user-user relationship. This procedure is described as the below *Figure 21*.

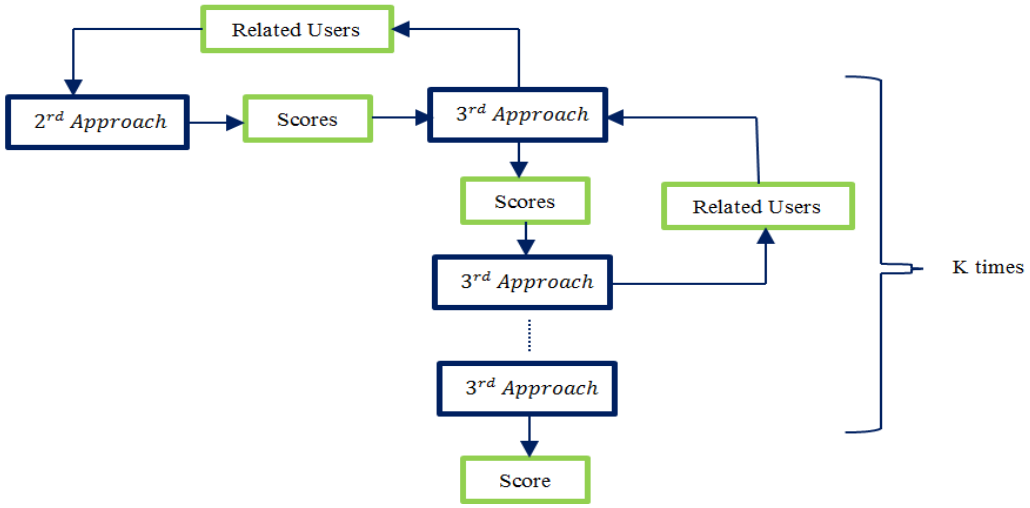

Figure 21 The fourth approach model

Acknowledgment
This research is funded by NTTU Foundation for Science and Technology Development under grant number 2016.02.04

Conflicts of interest
The authors have no conflicts of interest to declare.

References
[1] Cremonesi P, Turrin R. Analysis of cold-start recommendations in IPTV systems. In proceedings of the third ACM conference on recommender systems 2009 (pp. 233-6). ACM.
[2] Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. Springer US; 2011.
[3] Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation. Communications of the ACM. 1997; 40(3):66-72.
[4] Ekstrand MD, Riedl JT, Konstan JA. Collaborative filtering recommender systems. Foundations and Trends in Human–Computer Interaction. 2011; 4(2):81-173.
[5] Jannach D, Zanker M, Felfernig A, Friedrich G. Recommender systems: an introduction. Cambridge University Press; 2010.
[6] Wei CP, Shaw MJ, Easley RF. Recommendation systems in electronic commerce. E-Service: new directions in theory and practice. 2002.
[7] Andale. Pearson correlation: definition and easy steps for use. http://www.statisticshowto.com/what-is-the-pearson-correlation-coefficient/. Accessed 10 May 2016.
[8] Spearman's rank-order correlation. https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php. Accessed 16 May 2016.
[9] Perone CS. Machine learning: cosine similarity for vector space models (part iii). http://pyevolve.sourceforge.net/wordpress/?p=2497. Accessed 16 May 2016.
[10] Herlocker J, Konstan JA, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval. 2002; 5(4):287-310.
[11] Weighted average. http://www.financeformulas.net/Weighted_Average.html. Accessed 10 June 2016.
[12] Moreno A, Ariza-Porras C, Lago P, Jiménez-Guarín CL, Castro H, Riveill M. Hybrid model rating prediction with linked open data for recommender systems. In semantic web evaluation challenge 2014 (pp. 193-8). Springer International Publishing.

Hoang Lam Le graduated from the University of Science University, HoChiMinh national university with a degree in computer networking in 2011. He received a Master of Computer Science in the electronics department, Myongji University, Korea in 2015. He is currently a researcher at the institute of science of technology industry 4.0. His research interests include data mining, text mining, ontology and machine learning.
Email: lehoanglam20000@gmail.com

Quoc Cuong Nguyen received the Bachelor of Science degree from Dong A University, Da Nang city, Vietnam in 2016. He is currently a researcher in the institute of science and technology Industry 4.0. His research interests include data mining and machine learning.

Minh Tri Nguyen received the Bachelor of Science degree in Computer Sciences from Ho Chi Minh City International University, Vietnam in 2010 and a Master of Engineering degree in Information Technology Management from Ho Chi Minh City International University, Vietnam in 2013. He is currently a software engineer at the KMS Technology company. His research interests include data warehouse, data mining and machine learning.