

Research Article

An Effective Recommender Algorithm for Cold-Start Problem in Academic Social Networks

**Vala Ali Rohani,¹ Zarinah Mohd Kasirun,¹
Sameer Kumar,² and Shahaboddin Shamshirband³**

¹ Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

² Asia-Europe Institute, University of Malaya, 50603 Kuala Lumpur, Malaysia

³ Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), Chalous 46615-397, Iran

Correspondence should be addressed to Vala Ali Rohani; vrohani@gmail.com

Received 22 January 2014; Accepted 30 January 2014; Published 18 March 2014

Academic Editor: Gerhard-Wilhelm Weber

Copyright © 2014 Vala Ali Rohani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abundance of information in recent years has become a serious challenge for web users. Recommender systems (RSs) have been often utilized to alleviate this issue. RSs prune large information spaces to recommend the most relevant items to users by considering their preferences. Nonetheless, in situations where users or items have few opinions, the recommendations cannot be made properly. This notable shortcoming in practical RSs is called cold-start problem. In the present study, we propose a novel approach to address this problem by incorporating social networking features. Coined as enhanced content-based algorithm using social networking (ECSN), the proposed algorithm considers the submitted ratings of faculty mates and friends besides user's own preferences. The effectiveness of ECSN algorithm was evaluated by implementing it in MyExpert, a newly designed academic social network (ASN) for academics in Malaysia. Real feedbacks from live interactions of MyExpert users with the recommended items are recorded for 12 consecutive weeks in which four different algorithms, namely, random, collaborative, content-based, and ECSN were applied every three weeks. The empirical results show significant performance of ECSN in mitigating the cold-start problem besides improving the prediction accuracy of recommendations when compared with other studied recommender algorithms.

1. Introduction

The prevalence of digital technology and rapid development of World Wide Web has revolutionized our society toward a culture based on the value of information [1]. The online environments, such as social networks and weblogs, have thus become an abundant information source evidencing a significant effect on users' lifestyle [2, 3]. Similar scenario has occurred in ASNs. A huge amount of e-content is produced on the web every single day, covering various types of academic items such as news, jobs, scholarships, and conferences. As recommending the most relevant information to users based on his/her needs is getting difficult [4], recommender systems (RSs) have emerged to alleviate above challenges by providing users with the most relevant items [5]. RSs utilize the users' past evaluations and interactions with system to predict potential further likes and interests of their users [6].

Two most popular approaches among RSs are collaborative and content-based recommender algorithms [5–8]. In collaborative algorithm, the recommendations are made based on the items that people with similar preferences and interests preferred previously [9], while, in content-based methods, recommended items are those with content similar to previously preferred items by a target user [6]. Since, in collaborative filtering techniques, the analysis of the actual content is not required, they are widely used in making predictions to filter any type of items such as text, photos, music, and videos [4]. Content-based RSs, on the other hand, analyze item descriptions to find items that are of particular interest to the user [10]. They adopt a well-structured framework for comparing user interests with the items' specifications to finally suggest the most suitable item to a target user [11]. Although content-based recommendation methods resolve the new items' issues, they still suffer from the cold-start problem in situations when new users are involved [5].

The cold-start problem is divided into two categories of cold-start items and cold-start users [9]. Cold-start items challenge is caused by new items that are supposed to be recommended to users while there are not enough previous submitted ratings about them [4, 12, 13]. Cold-start problem happens when a new user who has already joined an online environment has presented just few opinions. In such situations, there is no interaction between the new user and the other ones, and hence it is not possible to measure the similarity between them. As a result, the recommender systems are unable to make reliable recommendations [9]. Both collaborative and content-based recommender systems have a shortcoming related to cold-start problem [5]. For optimal recommendations, collaborative algorithms require strict records of previous item ratings. However, in some domains in which new items exist with no previous rating records, collaborative methods cannot function properly. Hence in collaborative filtering approaches, cold-start new items problem occurs in such conditions when new items are supposed to be recommended [9]. This issue has been mitigated to some extent by content-based recommender systems, which can predict item relevance even in the absence of prior ratings [10]. Nevertheless, even content-based recommender systems suffer from cold-start new users' challenge. They are unable to recommend items to new users in the absence of any history of previous interactions with the system [5].

To mitigate the above-mentioned problems, an enhanced version of the content-based algorithm is proposed in the current research whereby social networking techniques are utilized to not only solve the cold-start problem but also improve the prediction accuracy of the recommendation process. This model considers the interests and preferences of users' friends and faculty mates in addition to users' own preferences. In this novel approach, the interests and preferences of users are stored in a hierarchy tree structure. The subsequent sections are organized as follows. The next section presents some related research works in solving the cold-start problem. Section 3 introduces the ECSN algorithm which is proposed in the present research. The results of 12 weeks of experiments carried out in this research are presented and discussed in Section 4 and, finally, Section 5 concludes the paper by outlining our future research directions.

2. Related Works

The defining attribute of the Internet today is the abundance of information and choice. In such enormous online environment, recommendation systems were designed to alleviate this problem by providing personalized item recommendations to users [2]. Two widely used techniques in such systems are collaborative and content-based filtering but they both suffer from cold-start problem [5]. This issue has led to the emergence of some hybrid recommendation algorithms to sidestep this shortcoming and improve the recommendation quality.

By presenting a tagging method in collaborative filtering algorithms, Saini and Banda [14] were among the earliest researchers who suggested a solution for solving the cold-start problem. Park and Chu [15] also proposed predictive

feature-based regression models that leverage all available information of users and items. Some researchers have suggested the use of ratings agents, known as filterbots, for augmenting the ratings of items based on their content features [16, 17], while some others have integrated a user model with trust and distrust networks to identify trustworthy users [2]. In another research work, a recommendation algorithm is proposed that learns to conduct the interview process guided by a decision tree with multiple questions at each split [18]. Lam et al. [19] developed a hybrid model based on analysis of two probabilistic aspect models. Their study combined the pure collaborative filtering with users' information to solve the cold-start problem. Communities' information, extracted from different dimensions of social networks, was used in [20] to help recommendation systems in solving cold-start problem based on the found latent similarities. Proposing a new similarity measure is also another solution which was used in [21] to solve this problem. In their study, the authors used optimization based on neural learning to achieve better performance in recommending to new users. In [22], CCF-BURP was presented as a new method that constructs an algorithm with two steps, in the first of which they screen neighbors of the target user, using its personal attributes, while in the second of which they train the interview model on the dataset constituted of the neighbors and alternative projects. Then the recommender system forecasts the goal of optional project ratings of the target user.

As stated in [23, 24], the majority of previous recommender algorithms have focused on enhancing the performance of the recommendation process and mitigating the cold-start problem without considering the social elements of decision making and advice seeking. More specifically, traditional recommender systems ignore social relationships among users. It has been affirmed that recommender algorithms could be significantly improved by drawing on features from social systems [25]. Filling this gap, some researchers have conducted related studies by utilizing the social networking features. For example, Said et al. [26] presented a probabilistic approach to item recommendation in order to enhance the recommendations during the cold-start period in CiteULike community. The authors extended the previous models such as probabilistic latent semantic analysis (PLSA) by merging both user-item and item-tag observations into a unified representation. Also in this context, an improved method was proposed by introducing an item-oriented function, focusing on solving the dilemma of the recommendation accuracy between the cold and popular items [27]. This method was based on a hybrid algorithm incorporating the heat conduction and probability spreading processes. The experimental results indicate that their proposed algorithm significantly improves the recommendation accuracy of the cold items, while it keeps the recommendation accuracy of the overall and the popular items. Furthermore, Zhang et al. proposed a recommendation algorithm based on the user-tag-object tripartite which makes use of social tags. Besides enhancing the algorithmic accuracy and diversity, this method significantly resolves the cold-start problem in social tagging systems with heterogeneous object degree distributions [28]. Another algorithm [29] was

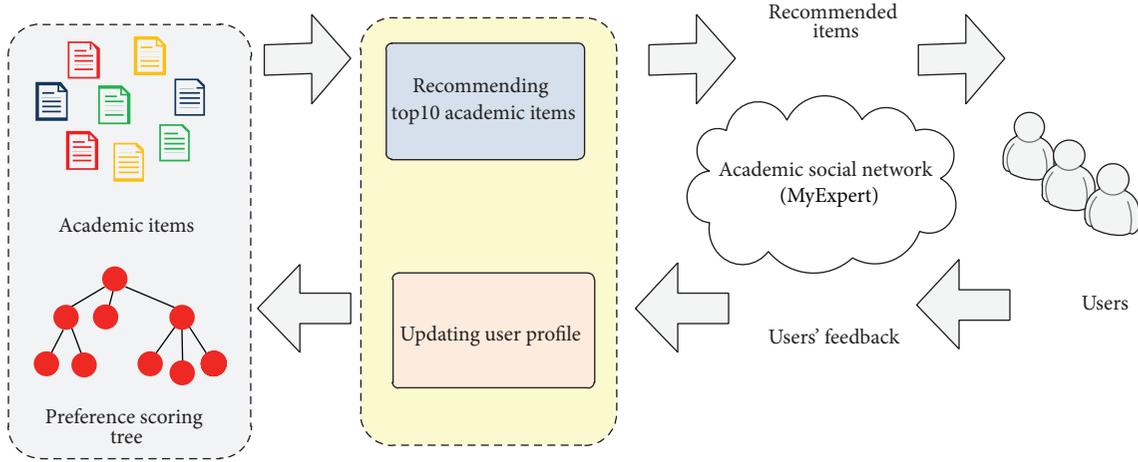


FIGURE 1: Overview of ECSN recommendation process.

introduced to leverage the rich social information in one platform to help the construction of a new user network in another platform on user level. Based on conducted data analysis, they showed that friend relations and common contact behavior can be better transferred to another social platform.

3. The Proposed ECSN Algorithm

In the present research, an enhanced version of content-based recommender systems is generated which incorporates social network-based factors to improve the performance of the recommendation process. Figure 1 depicts the overview of recommending top 10 items to members of MyExpert (<http://www.malaysianexperts.com.my/>). This process includes collecting the relevant feedback from MyExpert users, generating and keeping updated user profiles based on their elicited preferences during the interactions, and, finally, applying ECSN recommender algorithm to find the top 10 academic items among 100 submitted ones in each week of experiment and sending the weekly e-newsletter to each member of MyExpert.

More specifically, ECSN algorithm utilizes the “friends profile” and “faculty mates profile” besides the users’ own preferences. In doing so, all transactions records of a given user’s friends are analyzed by ECSN recommender engine and the most relevant nodes in preference tree structure of academic items would be elicited. Then, the rating value of elicited nodes will be updated in preference tree of the given user. The same process would be done regarding the faculty mates of the target user.

ECSN recommender algorithm manages the preference scores of users for academic items categories in a tree data structure which is designed in hierarchical form. Hence, in ECSN algorithm the users’ preference scores for each academic item category are stored based on following definition.

Definition 1. The preference tree $PT(i)$ of user i is isomorphic to the hierarchy tree of item categories, and the set of nodes

of the preference tree $PT(i)$ is as follows:

$$PT(i) = \{(UID, ICID, PS)\}, \quad (1)$$

where UID, ICID, and PS are the user identifier, the item category identifier of the hierarchy tree, and the preference score, respectively.

Definition 2. The preference scores (PS) are defined as follows:

$$\begin{aligned} PS(i, j) = & \alpha_1 \times SelfClickScore(i, j) \\ & + \alpha_2 \times SelfRankScore(i, j) \\ & + \alpha_3 \times FacultyMatesScore(i, j) \\ & + \alpha_4 \times FriendsScore(i, j), \end{aligned} \quad (2)$$

where $PS(i, j)$ is the total preference score of user i for the academic item category node j . Each element of this definition is described in following.

$SelfClickScore(i, j)$ is the score related to clicks of given user i for the item category node j , which is specified by counting the number of clicks for given customer during the research experiments.

$SelfRankScore(i, j)$ is calculated by adopting the submitted rates of given user i to academic items which classified in category node j .

The value of $FacultyMatesScore(i, j)$ is calculated by considering the top 3 interesting item nodes of preference tree among the members who have registered in the same faculty that the given user i belongs to.

Last element, $FriendsScore(i, j)$, is dedicated to preferences of friends for given member i . The strategy that is described above for calculating the $FacultyMatesScore(i, j)$ is adopted here for $FriendsScore(i, j)$ with this difference that it takes into account the top 3 item nodes which are mostly interesting among friends of the given member.

As mentioned in previous researches [30], some weights might be assigned to each parameter of the formula for computing the preference scores. Accordingly, in Definition 2, the (α_k) represent relative weights for each element. As SelfClickScore and SelfRankScore are the most important personal elements that should be counted for given users i , the value of 5 is considered for α_1 and α_2 . Relatively, the weight of 3 has been considered for α_3 since FacultyMatesScore is less significant than user's own preferences. And, finally, α_4 is set by 1 as FriendsScore has the lowest influence in Definition 2. The assigned weights are subjective values for considering the levels of importance among users' own preferences, their faculty mates, and friends. Although the experimental results of this study indicate that these settings work well in improving the prediction accuracy of recommendations, but, as the future works, even these weights might be optimized by applying some other techniques such as genetic algorithms.

After calculating the preferences scores PS for each user i , Definition 3 is applied for some nonleaf nodes of preference tree whose values are still 0.

Definition 3. The preference scores (PS) a nonleaf-level product category j are defined as follows:

$$\begin{aligned} \text{PS}(i, j) = & \text{Average}_{k \in \{k | k \text{ is a child node of product category } j\}} \\ & \times \text{PS}(i, k). \end{aligned} \quad (3)$$

The preference tree of a certain user i is initialized to zero when a user creates a profile in academic social network (MyExpert):

$$\text{PS}(i, j) = 0 \quad j \in \{j | j \text{ is a node in preferences scores tree structure for academic item categories}\}. \quad (4)$$

The values that are illustrated in Table 1 are used for updating the preference scores.

- (1) When the given user i rates the academic items related to category node j ,

$$\text{PS}(i, j) \cdot \text{SelfRankScore} = \text{PS}(i, j) \cdot \text{SelfRankScore} + \text{RV}. \quad (5)$$

- (2) When the given user i clicks the academic items related to category node j ,

$$\text{PS}(i, j) \cdot \text{SelfClickScore} = \text{PS}(i, j) \cdot \text{SelfClickScore} + 1. \quad (6)$$

The above update procedure does not require the update of preference scores for all nodes of the tree; rather it requires only the update of the preference scores of nodes related to visited and rated items.

After updating SelfRankScore, the preference scores (PS) need to be updated by considering the faculty mates and friends of given user i :

$$\begin{aligned} & \text{PS}(i, j) \cdot \text{FacultyMatesScore} \\ & = \text{PS}(i, j) \cdot \text{FacultyMatesScore} \\ & \quad + \text{Average}(\text{FS}), \end{aligned} \quad (7)$$

TABLE 1: Illustration of assigned points to different rates.

| Given rate | Description | The rate value (RV) |
|------------|-------------|---------------------|
| ★★★★★ | Excellent | +3 |
| ★★★★☆ | Good | +2 |
| ★★★☆☆ | Fair | +1 |
| ★★☆☆☆ | Not bad | -1 |
| ★☆☆☆☆ | Weak | -2 |

where

$$\text{FS} \in \left\{ \begin{array}{l} \text{Top 3 Preference scores of user} \\ i' \text{ s friends to product category } j \end{array} \right\}. \quad (8)$$

Similarly, the FriendsScore value is updated as

$$\begin{aligned} \text{PS}(i, j) \cdot \text{FriendsScore} = & \text{PS}(i, j) \cdot \text{FriendsScore} \\ & + \text{Average}(\text{FS}), \end{aligned} \quad (9)$$

where

$$\text{FS} \in \left\{ \begin{array}{l} \text{Top 3 preference scores of user} \\ i' \text{ s friends to product category } j \end{array} \right\}. \quad (10)$$

As mentioned in the above formulas, Average(FS) is the average value of top 3 preference scores which were assigned to product category j by friends or faculty mates of target user (i). In this study, top 3 scores were considered instead of all recorded scores to make the proposed algorithm more applicable in real situations facing millions of items and users. As another reason, considering assigned weights in (2), the preferences of faculty mates and friends are mostly effective in cold-start situations when there are not enough recorded preferences for target user. In such conditions, for making recommendations, it is preferred to find the top items which are the most interesting for friends and classmates.

In each week of experiments, 100 academic items were submitted at MyExpert academic social network. Each studied recommender algorithm aimed to select the top 10 items for each user and recommend them through an e-newsletter. For first three algorithms, the selection process was implemented based on recommender algorithms that were studied through the review of literature. In random recommender algorithm, the random 10 items were selected. Collaborative algorithm made predictions based on the items that people with similar preferences and interests preferred previously [31, 32]. For implementing the pure content-based recommendation, the preference tree approach [30] was followed. Finally, an enhanced selection process was used in ECSN recommender algorithm which is illustrated in Algorithm 1.

For each user of MyExpert, the item categories get ordered based on PS value and stored in a stack data structure (ItemStack) such that the category with the biggest PS is accessible at the top of the stack. To produce the recommendation list for user u , the highest ranked category at the top of stack is moved to TopItemCat using POP

```

for each  $u \in U$ 
{
  (1) Generating the ordered stack of item categories (ItemStack) based on PS value computed by Definitions 2 and 3.
  (2)  $SIC \leftarrow 0$  // Initializing the Selected Items Count (SIC) by 0
  (3)  $SC \leftarrow 0$  // Initializing the selected category (SC) by 0
  (4) While (SelectedItems < 10)
  (5) {
  (6)  $TopItemCat \leftarrow POP(\mathbf{ItemStack})$ ,
  (7)  $TopItemsList \leftarrow FindNewItems(TopItemCat, PrioritizedCount(SC))$ 
  (8)  $SIC -= count(TopItemsList)$ 
  (9) Adding  $TopItemsList$  to RecommendationList
  (10) }
}

```

ALGORITHM 1: The ECSN selection process.

(ItemStack). Then, the new submitted items in MyExpert (100 items per week) are searched to find academic items using category ID of TopItemCat. The recommendation list that is supposed to be suggested to each user u includes the 10 most relevant items. To have more items with highest PS value in this list, the PrioritizedCount array has been considered to identify the number of items for each top scored item category:

$$\text{int PrioritizedCount} = \{3, 2, 2, 1, 1, 1, 1, 1, 1, 1\}. \quad (11)$$

Based on this identified priority, the highest scored category can have up to 3 items while the two next highest ones come with at most 2 items in recommendation list. The others have the same value of one item. In the body of while loop, the ordered recommendation list (TopItemsList) is generated for each user $u \in U$.

To compare the efficiency of ECSN algorithm with previous approaches, collaborative and content-based recommendation algorithms were implemented and applied in this study.

In collaborative filtering approach, prediction is done based on the items previously preferred by people with similar preferences and interests. There are two main classifications for collaborative filtering recommender systems, that is, memory-based (user-based) and model-based (item-based). To mitigate some shortcomings of former approach, the model-based method was developed that looks for similar items instead of making groups of similar users [6, 32]. For this reason, the model-based approach was implemented in this research to utilize its advantages. In this method, the similarity of items is calculated as

$$\text{sim}(\vec{a}, \vec{b}) = \frac{\sum_{u \in U} (r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,a} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,b} - \bar{r}_u)^2}}, \quad (12)$$

where U indicates the set of all users who rated both items a and b . Accordingly, $r_{u,a}$ and $r_{u,b}$ are the ratings assigned by user u to items a and b , respectively [33]. Also, to predict which items are the best candidates to be recommended to

each given member of MyExpert, the following predictor function was used at this stage:

$$\text{pred}(u, p) = \frac{\sum_{i \in \text{ratedItem}(u)} \text{sim}(i, p) * r_{u,i}}{\sum_{i \in \text{ratedItem}(u)} \text{sim}(i, p)}. \quad (13)$$

To implement the content-based recommendation algorithm in this study, the preference scoring structure was used to model the user profiles [30]. Similar to ECSN algorithm, this approach also manages the preference scores of users for academic items categories in a tree data structure which is designed in hierarchical form. Hence, the data structure which was identified in Definition 1 is used for this method:

$$PT(i) = \{(UID, ICID, PS)\}, \quad (14)$$

where UID, ICID, and PS are the user identifier, the item category identifier of the hierarchy tree, and the preference score, respectively.

But in this recommendation algorithm, only the user's own preferences are used for calculating the preference score:

$$PS(i, j) = \text{SelfClickScore}(i, j) + \text{SelfRankScore}(i, j). \quad (15)$$

Comparison of the above formula with its enhanced version in Definition 2 illustrates the fact that in ECSN algorithm the preferences of friends and faculty mates are considered in addition to the target user's own preferences. The experimental results in the next section show the positive influence of this enhancement in improving the prediction accuracy of recommendations and solving the cold-start problem.

4. Experiments and Results

The experiments in this research ran for 14 weeks from the 7th of September until the 26th of December, 2012. In this duration, four recommender algorithms (random, collaborative, content-based, and ECSN) were applied and their performance in making the most effective recommendations was compared among them. In doing so, after running the first two weeks of experiments as pretest, the

TABLE 2: The experimental statistics in the context of cold start.

| Algorithms | Data gathering series number | Cold-start parameters | | | | | | |
|---------------|------------------------------|-----------------------|-----------|----------------|-----------|-----------|--------------|--------------|
| | | New items | New users | Existing users | totNRI_EU | totNRI_NU | avgNRI_EU | avgNRI_NU |
| Collaborative | CF_Series1 | 100 | 8 | 864 | 0 | 0 | 0 | 0 |
| | CF_Series2 | 99 | 5 | 869 | 0 | 0 | 0 | 0 |
| | CF_Series3 | 98 | 6 | 875 | 0 | 0 | 0 | 0 |
| Content-based | CB_Series1 | 99 | 7 | 882 | 3112 | 0 | 3.528 | 0 |
| | CB_Series2 | 101 | 9 | 891 | 3156 | 0 | 3.542 | 0 |
| | CB_Series3 | 100 | 8 | 899 | 3190 | 0 | 3.548 | 0 |
| ECSN | ECSN_Series1 | 99 | 7 | 906 | 3720 | 27 | 4.106 | 3.857 |
| | ECSN_Series2 | 100 | 5 | 911 | 3769 | 18 | 4.137 | 3.6 |
| | ECSN_Series3 | 98 | 9 | 920 | 3792 | 34 | 4.122 | 3.778 |

random recommender algorithm was used for three weeks to recommend academic items to MyExpert users. Next, the collaborative and content-based recommender systems were adopted over the following 6 weeks of data gathering. Finally, during the last 3 weeks, MyExpert members received recommendations by the ECSN algorithm to conclude the experiments for this study. In these 14 weeks of feedback collection, 1390 records of academic items were submitted to MyExpert including 346 academic jobs, 339 conferences, 355 scholarships, and 350 academic news. These items were sent to 920 MyExpert registered members from 10 universities in Malaysia. To assess the prediction accuracy of studied recommender algorithms, precision, recall, fallout, and $F1$ were used as well-known measurements in this context [6, 33]. The details of experimental results are presented in following in terms of solving cold-start problem and making the effective recommendations.

4.1. Solving the Cold-Start Problem. In situations where collaborative and content-based methods suffer from the cold-start problem when new items or new users are involved [33], the ECSN recommender algorithm utilized social networking features to solve this issue. Collected feedback from 14 weeks of experiments was analyzed to show how the proposed recommender algorithm in present research mitigated this issue.

Table 2 demonstrates the detailed statistics in this context. The second column of this table, data gathering series number, lists 9 series of the data collection phase where three main recommender algorithms (collaborative, content-based, and ECSN) were applied. The next three columns (new items, new users, and existing users) present the updated situation of MyExpert social network for each week of experiments based on the number of users and items. Finally, the last four columns represent the values of some parameters used to measure the cold-start problem. The four used parameters for evaluating the performance of recommender algorithms in solving cold-start problem are defined as follows:

totNRI_EU: total number of recommended new items to existing users with prediction value > 1 ;

totNRI_NU: total number of recommended new items to new users with prediction value > 1 ;

avgNRI_EU: average number of recommended new items to existing users with prediction value > 1 ;

avgNRI_NU: average number of recommended new items to new users with prediction value > 1 .

To clarify the status of the cold start, this research focused on the number of recommended new items to existing and new users. In this context, totNRI_EU assisted in investigating the new items problem while totNRI_NU concentrated on the new user perspective of the cold-start problem. The average value of these measurements for each user shows to what extent the adopted recommender algorithm succeeded in solving the cold-start issue.

As illustrated in Table 2, the collaborative recommender algorithm was not able to contribute at all regarding new users and new items in the cold-start problem. Any new item was suggested even to existing MyExpert users. Thus, this method only applies when previously rated items are supposed to be recommended to existing users. The above-mentioned statistics therefore prove that the collaborative algorithm poses the cold-start problem in both cases of new items and new users.

The content-based technique found an average of 3 new items for existing users. Nevertheless, it still encountered a problem with new users. As shown in Table 2, 24 new users were added to MyExpert academic social network during the experimental series CB_Series1 to CB_Series3. According to column totNRI_NU (total number of recommended items with prediction value > 1 to new users), the content-based approach was unable to recommend the new items to new users. So it is concluded that the new items part of the cold-start problem was solved with an average of 3 recommended new items to existing users. However, the shortcoming related to new users remains in this approach.

The ECSN recommender algorithm could solve both the new users and new items issues with regard to the cold-start problem. The statistics of experimental series ECSN_Series1 to ECSN_Series3 clearly show that roughly 4 new items were recommended to existing users. In the case of number of recommended items with prediction values higher than 1 to new users, the average value was around 3.7 based on column avgNRI_NU. This means that not only has the cold-start problem of previous recommender systems been solved

TABLE 3: Cold-start problem situations in different recommender algorithms.

| Recommender algorithms | Cold-start problem | |
|------------------------|--------------------|-----------|
| | New items | New users |
| Random | NA | NA |
| Collaborative | ✗ | ✗ |
| Content-based | ✓ | ✗ |
| ECSN | ✓ | ✓ |

NA: not applicable.

✗: problem exists.

✓: problem is solved.

by the ECSN algorithm but it is also clear that the average number of recommended items to existing users with values above 1 improved by 15%.

In conclusion, the cold-start status in the four examined recommender algorithms (random, collaborative, content-based, and ECSN) is illustrated in Table 3. This problem is not applicable to the random recommender algorithm, as it only selected 10 random items and sent them to users. The collaborative algorithm suffers from the new item and new user problem, while the content-based approach solved the new item shortcoming but not the new user issue. As depicted in the last rows of Table 2, both the new user and new item issues pertaining to the cold-start problem were resolved by the ECSN recommender algorithm.

4.2. Improving the Prediction Accuracy. We now present the experimental results and also the evaluation of the performance of the proposed algorithm (ECSN) based on four previously mentioned measurements. During 14 weeks of online experiments carried out among 920 members of MyExpert academic social network, precision, recall, fallout, and $F1$ were used for evaluating the performance of each studied recommender algorithm.

Precision is one of most prevalent metrics for assessing usage prediction in recommender systems and information retrieving studies. It plays a great role in instances where some sets of the best results are required out of several possible alternatives [33].

Precision considers the number of a user's relevant objects ranked in the top- L places [6]. More specifically, it measures the share of top results that are relevant. In this study, the relevant items defined include academic items visited by users and rated with more than 2 stars.

Table 4 illustrates four possible conditions based on the selection and usage situations.

According to the notations in Table 4, precision is defined as

$$\text{Precision} = \frac{N_{rs}}{N_{rs} + N_{is}}. \quad (16)$$

It can also be stated that precision is the probability that a recommended item corresponds to a user's interests and preferences.

TABLE 4: The possible conditions of item recommendation to a user.

| | Selected | Not selected | Total |
|------------|----------|--------------|-------|
| Relevant | N_{rs} | N_{rn} | N_r |
| Irrelevant | N_{is} | N_{in} | N_i |
| Total | N_s | N_n | N |

Recall is recognized as another metric for measuring usage prediction in recommender systems and other information retrieval domains. It determines the proportion of all relevant results included in the top results. In studies where a fixed number of recommendations are suggested to each user (such as the current study in which the top 10 items are recommended to MyExpert users in every week of experiments) precision and recall can be computed at each recommendation list length N for each user. Then the average value of precision and recall can be computed for all users involved in the experiment [6, 33].

In the recommendations domain, a perfect recall score of 1.0 indicates that all excellent items were recommended in the list. Consequently, a higher precision value is better. Recall, or the true positive rate, is calculated as the ratio of selected (recommended) items used (relevant) to the total number of items used [6, 33]:

$$\text{Recall} = \frac{N_{rs}}{N_{rs} + N_{rn}}. \quad (17)$$

Precision and recall are inversely related. In most cases, increasing the size of the recommendation set will increase recall but decrease precision [33].

To further assure the performance of the ECSN algorithm, two other metrics were applied to evaluate the accuracy prediction of all studied recommender systems in this research, namely, fallout and $F1$. The upcoming sections focus on analyzing the relevance feedback based on these metrics.

Fallout, or the false positive rate [33], is measured as the ratio of selected (recommended) items that are not used (irrelevant) to the total number of unutilized items:

$$\text{Fallout} = \frac{N_{is}}{N_{is} + N_{in}}. \quad (18)$$

It is the probability that an irrelevant (not used) item will be recommended to a user. According to this definition, a lower fallout rate indicates better recommender algorithm performance.

To evaluate the overall performance of a recommender algorithm it makes sense to consider precision and recall together [6]. Various researches have pointed out that precision and recall are inversely related and dependent on the length of the result list returned to the user [6, 33]. So under these circumstances, a vector of precision/recall pairs may describe recommender system performance. Several methods have been assessed to combine precision and recall into a single metric. One approach is the $F1$ metric which amalgamates precision and recall into a single value.

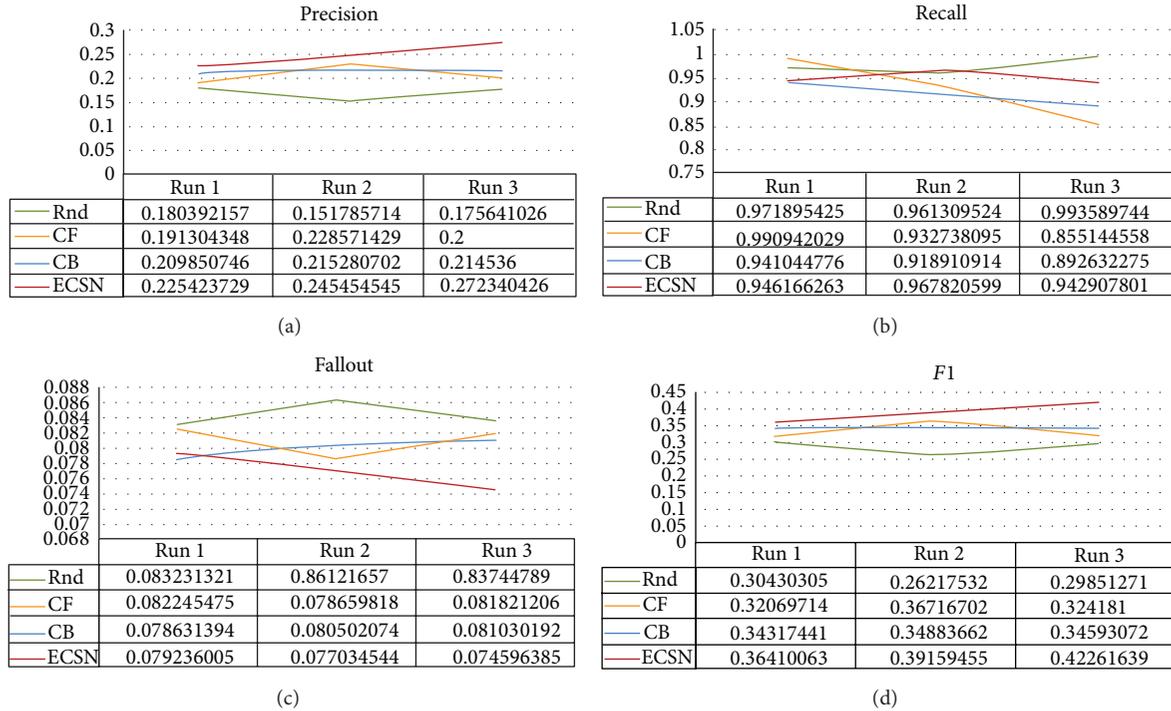


FIGURE 2: Comparison of studied recommender algorithms in terms of prediction accuracy.

The $F1$ score, or $F1$ measure, is defined as the standard harmonic mean of precision and recall:

$$F1 = \frac{2}{(1/Precision) + (1/Recall)} = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (19)$$

As shown in (19), the values of both precision and recall are considered when calculating the $F1$ score for measuring the accuracy prediction of a given recommender algorithm.

A complete view of the results based on prediction accuracy measurements is presented in Figure 2. The first two runs of 14 weeks experiments were considered pretest stages that MyExpert members received the recommended items through weekly e-newsletters and accordingly were not included in measurements. During these two weeks, they tried rating the academic items. After this pilot test, each recommender algorithm of random, collaborative, content-based, and ECSN was applied in three subsequent weeks. For each series of experiments, the average value of four different measurements (precision, recall, fallout, and $F1$) was calculated.

Figure 2 illustrates the mean value of prediction accuracy based on four different measurements. The precision value had an upward trend in the 12 weeks of experiments. It started at an average of 0.169 with the random algorithm and steadily rose to 0.207 for the collaborative and 0.213 for the content-based approach. In the last stage of the experiments, the ECSN algorithm reached a peak of 0.248. The ECSN algorithm enhanced the precision value of other recommender

algorithms by 32% (random, MD = 0.00741), 20% (collaborative, MD = 0.00395), and 21% (content-based, MD = 0.00310).

The recall value comparison for all four recommender algorithms can be seen in Figure 2. The highest rate was attained by the random (0.976) and the ECSN algorithm (0.952), while the collaborative and content-based approaches had lower recall values, at 0.926 and 0.918, respectively. Although the ANOVA test results do not show any significant differences in case of recall metrics, even based on this measurement, the contribution of the ECSN method is clear with improvement over the collaborative by 3% and content-based method by 4%.

The fallout values of the studied recommender algorithms are compared in Figure 2, where it is obvious that the fallout rate had a decreasing trend in the 12 weeks of experiments. Referring to the definition of fallout, a lower fallout rate indicates better recommender algorithm performance. Thus, this diagram shows that the prediction accuracy of recommender algorithms improved from random (0.084) to collaborative (0.081), content-based (0.080), and finally ECSN (0.077) algorithm while the values of the fallout metric declined steadily.

The final section of the diagram corresponds to the $F1$ score. As mentioned previously, the values of both precision and recall are combined to calculate the $F1$ score for measuring the accuracy prediction of a given recommender algorithm. It is thus considered an overall metric that includes both recall and precision. Figure 2 shows the steady rise of $F1$ values during 12 weeks of experiments. The lowest value belongs to the random algorithm (0.288) while the peak of 0.393 corresponds to when the ECSN algorithm was

TABLE 5: One-way ANOVA tests for examining the difference of means between recommender algorithms.

| (a) ANOVA test on precision value. | | | | |
|------------------------------------|----------|----------|----------|------|
| | Rnd | CL | CB | ECSN |
| Rnd | 1 | | | |
| CL | 0.03735 | 1 | | |
| CB | 0.04395* | 0.00660 | 1 | |
| ECSN | 0.07847* | 0.04111* | 0.03452* | 1 |
| (b) ANOVA test on recall value. | | | | |
| | Rnd | CL | CB | ECSN |
| Rnd | 1 | | | |
| CL | 0.4932 | 1 | | |
| CB | 0.05807 | 0.00875 | 1 | |
| ECSN | 0.02330 | 0.02602 | 0.03477 | 1 |
| (c) ANOVA test on fallout value. | | | | |
| | Rnd | CL | CB | ECSN |
| Rnd | 1 | | | |
| CL | 0.00346* | 1 | | |
| CB | 0.00431* | 0.0085 | 1 | |
| ECSN | 0.00741* | 0.00395* | 0.00310 | 1 |
| (d) ANOVA test on F1 value. | | | | |
| | Rnd | CL | CB | ECSN |
| Rnd | 1 | | | |
| CL | 0.04902* | 1 | | |
| CB | 0.05765* | 0.00863 | 1 | |
| ECSN | 0.10444* | 0.05542* | 0.04679* | 1 |

*The mean difference is significant at the 0.05 level.

applied. In other words, the ECSN recommender algorithm significantly contributes to the $F1$ values of the random ($MD = 0.10444$), collaborative ($MD = 0.05542$), and content-based ($MD = 0.04679$) algorithms, by 26%, 14%, and 12%, respectively.

The one-way analysis of variance (ANOVA) test can be used for the case of a quantitative outcome with a categorical explanatory variable that has two or more levels of treatment [34]. As four different measurements (precision, recall, fallout, and $F1$) were used in this study for computing the prediction accuracy of recommender algorithms, the four ANOVA tests were run to examine if there were any between-group differences of means between studied recommender algorithms (Table 5). According to the results of LSD post hoc tests, the mean value of precision is significantly different between ECSN and three other algorithms (Table 5(a)) while this variation is not clear in terms of the recall measurement (Table 5(b)). The differences are obvious also based on fallout and $F1$ as shown in Table 5(c) and Table 5(d). Consequently, in exception of recall, the other measurements (precision, fallout, and $F1$) show the significant difference of prediction accuracy between four studied recommender algorithms.

5. Conclusion and Future Directions

Although recommender systems have been studied in the past decade, the study of social-based recommender systems is a recent phenomenon. The purpose of this research was to determine how the cold-start problem of recommender systems could be solved in academic social networks by applying an enhanced content-based algorithm utilized by social networking features (ECSN). To test the effectiveness of the proposed recommender algorithm, a 14-week experiment was performed to compare the ECSN algorithm, content-based approach, collaborative filtering, and random method. The investigation results for the collected relevance feedback from MyExpert users show that the ECSN algorithm may significantly contribute to solving cold-start problem. In addition to addressing both new user and new item issues in this context, the proposed recommender algorithm in the present research seems to improve the average value of recommended items to new users by 15% compared to the pure content-based algorithm. Besides, the results of the experiment based on four different measurements indicate that ECSN recommender algorithm provides better overall prediction accuracy than other studied methods. In relation to precision, the ECSN algorithm had a significant contribution of 32% (random), 17% (collaborative), and 14% (content-based). In terms of the recall measurement, the contribution of the ECSN method is evident with improvement over the collaborative by 3% and content-based method by 4%. The fallout rate had a decreasing trend in the 14 weeks of experiments. Referring to the definition of fallout, a lower fallout rate indicates better recommender algorithm performance. Thus, based on this measurement, the prediction accuracy of recommender algorithms improved from random (0.084) to collaborative (0.081), content-based (0.080), and finally ECSN (0.077) algorithm while the values of the fallout metric declined steadily. $F1$, as an overall metric that includes both recall and precision, is the last measurement that was used in this research. The experimental results show the steady rise of $F1$ values during 14 weeks of experiments. The lowest value belongs to the random algorithm (0.288) while the peak of 0.346 corresponds to the case when the ECSN algorithm was applied. In other words, it is clear that the ECSN recommender algorithm significantly contributes to the $F1$ values of the random, collaborative, and content-based algorithms by 27%, 14%, and 11%, respectively. To conclude, 14 weeks of evaluations based on the four most familiar metrics, namely, precision, recall, fallout, and $F1$, demonstrate that the proposed recommender algorithm in this research (ECSN) successfully enhanced the prediction accuracy compared to the other studied and implemented recommender approaches. The application of ECSN recommender algorithm facilitated MyExpert existing users' receiving of more relevant academic items. In addition, the new users of academic social networks have this chance to be recommended by more related items and, also, the problem of making recommendations for new items without any previous ratings is solved by applying ECSN algorithm.

As a novel topic for future researches, the social network based features utilized in this study could be combined

with social network analysis (SNA) concepts to propose a new model for enhancing the organizational behaviors and recommendations. The results of such researches can be used, for example, in decision support systems (DSS). Another question that needs more extensive research is whether the weights that were considered for calculating the node scores in Definition 2 could be optimized. In this research, based on the degree of importance, weight of 5 was considered for user's own preferences, weight 3 is for faculty mates, and weight 1 is for applying the preferences of friends. Although applying these weights could make a significant contribution in solving the cold-start problem and also in improving the prediction accuracy of recommendations, it may be better to apply fuzzy logic or neural networks techniques to achieve even better optimum weights [35, 36]. Furthermore, MyExpert academic social network, which was developed for this study as the runtime environment for establishing the online experiments, has this potential to be used in future researches. Anomaly detection and community studies are two fields of related researches that can use this runtime environment for establishing their experiments and archiving online results [37].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the reviewers for their comments. This work was supported under Grants PS063-2010A and UM.C/625/1/HIR/MOHE/SC/13.

References

- [1] Y. Xiao, B. Wang, Y. Liu et al., "Analyzing, modeling, and simulation for human dynamics in social network," *Abstract and Applied Analysis*, vol. 2012, Article ID 208791, 16 pages, 2012.
- [2] C. C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2013.
- [3] V. A. Rohani and O. S. Hock, "On social network web sites: definition, features, architectures and analysis tools," *Journal of Advances in Computer Research*, vol. 1, no. 2, pp. 41–53, 2010.
- [4] H.-N. Kim, A. El-Saddik, and G.-S. Jo, "Collaborative error-reflected models for cold-start recommender systems," *Decision Support Systems*, vol. 51, no. 3, pp. 519–531, 2011.
- [5] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, pp. 1–35, Springer, 2011.
- [6] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [7] G. Yuan, S. Xia, and Y. Zhang, "Interesting activities discovery for moving objects based on collaborative filtering," *Mathematical Problems in Engineering*, vol. 2013, Article ID 380871, 9 pages, 2013.
- [8] H. A. Jalab and N. A. Abdullah, "Content-based image retrieval based on electromagnetism-like mechanism," *Mathematical Problems in Engineering*, vol. 2013, Article ID 782519, 10 pages, 2013.
- [9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [10] M. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, vol. 4321 of *Lecture Notes in Computer Science Web*, pp. 325–341, Springer, Berlin, Germany, 2007.
- [11] P. Lops, M. Gemmis, and G. Semeraro, "Content-based recommender systems: state of the art and trends," in *Recommender Systems Handbook*, pp. 73–105, Springer, New York, NY, USA, 2011.
- [12] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR '02)*, pp. 253–260, 2002.
- [13] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.
- [14] S. Saini and L. Banda, "Improving scalability issues using gim in collaborative filtering based on tagging," *International Journal of Advances in Engineering & Technology*, vol. 4, no. 1, pp. 600–610, 2012.
- [15] S.-T. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proceedings of the 3rd ACM Conference on Recommender Systems*, pp. 21–28, Association for Computing Machinery, Silicon Valley, Calif, USA, 2009.
- [16] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD '06)*, pp. 699–705, Association for Computing Machinery, Philadelphia, Pa, USA, 2006.
- [17] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press, MIT Press, London, UK, 1999.
- [18] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, "Learning multiple-question decision trees for cold-start recommendation," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 445–454, Association for Computing Machinery, Rome, Italy.
- [19] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC '08)*, pp. 208–211, Siem Reap, Cambodia, 2008.
- [20] S. Sahebi and W. W. Cohen, "Community-based recommendations: a solution to the cold start problem," in *Proceedings of the Workshop on Recommender Systems and the Social Web (RSWEB '11)*, 2011.
- [21] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012.

- [22] P.-Y. Zhu and Z. Yao, "Cold-start collaborative filtering based on user registration process," in *Proceedings of the 19th International Conference on Industrial Engineering and Engineering Management*, 2013.
- [23] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11)*, pp. 287–296, 2011.
- [24] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.
- [25] P. Bonhard, C. Harries, J. McCarthy, and M. A. Sasse, "Accounting for taste: using profile similarity to improve recommender systems," in *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '06)*, pp. 1057–1066, Association for Computing Machinery, Québec, Canada, 2006.
- [26] A. Said, R. Wetzker, W. Umbrath, and L. Hennig, "A hybrid PLSA approach for warmer cold start in folksonomy recommendation," in *Proceedings of the Workshop on Recommender Systems & the Social Web (RecSys'09)*, pp. 87–90, New York, NY, USA, 2009.
- [27] T. Qiu, G. Chen, Z.-K. Zhang, and T. Zhou, "An item-oriented recommendation algorithm on cold-start problem," *Europhysics Letters*, vol. 95, no. 5, Article ID 58003, 2011.
- [28] Z.-K. Zhang, C. Liu, Y.-C. Zhang, and T. Zhou, "Solving the cold-start problem in recommender systems with social tags," *Europhysics Letters*, vol. 92, no. 2, Article ID 280002, 2010.
- [29] M. Yan, J. Sang, T. Mei, and C. Xu, "Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '13)*, pp. 1–6, San Jose, Calif, USA, 2013.
- [30] J. W. Kim, K. M. Lee, M. J. Shaw, H.-L. Chang, M. Nelson, and R. F. Easley, "A preference scoring technique for personalized advertisements on Internet storefronts," *Mathematical and Computer Modelling*, vol. 44, no. 1-2, pp. 3–15, 2006.
- [31] P.-Y. Wang and H.-C. Yang, "Using collaborative filtering to support college students' use of online forum for English learning," *Computers & Education*, vol. 59, no. 2, pp. 628–637, 2012.
- [32] F. CACHEDA, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web*, vol. 5, no. 1, article 2, 2011.
- [33] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, pp. 257–297, Springer, New York, NY, USA, 2011.
- [34] R. C. Littell, SAS for mixed models2006: SAS institute.
- [35] M. Ji, F. Xie, and Y. Ping, "A dynamic fuzzy cluster algorithm for time series," *Abstract and Applied Analysis*, vol. 2013, Article ID 183410, 7 pages, 2013.
- [36] R. R. Yager, "Fuzzy logic methods in recommender systems," *Fuzzy Sets and Systems*, vol. 136, no. 2, pp. 133–149, 2003.
- [37] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, article 15, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

