

A genre-based fuzzy inference approach for effective filtering of movies

I.-Chin Wu^{a,*} and Wei-Hao Hwang^b

^a*Department of Information Management, Fu Jen Catholic University, Taipei, Taiwan*

^b*Institute of Information Science, Academia Sinica, Taipei, Taiwan*

Abstract. Recommendation techniques are utilized in electronic commerce because of their potential commercial value. Many e-commerce sites employ collaborative filtering techniques to provide recommendations to customers based on the preferences of similar users. However, as the number of customers and the range of products increase, the prediction accuracy of memory-based collaborative filtering algorithms declines because of sparse ratings. In addition, the time complexity of such algorithms is quite high in the prediction phase. To resolve these issues, we propose a genre-based fuzzy inference filtering approach for predicting movie preferences. We use content-based and collaborative filtering algorithms as baseline methods to evaluate the performance of our approach. The results of experiments demonstrate that the hybrid approach exploits the strengths of the content-based and collaborative filtering algorithms to achieve effective filtering in terms of precision. Moreover, the computation time can be reduced by using the α -cut approach. The findings have implications for the design of an interactive movie recommendation system for the World Wide Web.

Keywords: Hybrid filtering, fuzzy inference, recommendation, sparse rating

1. Introduction

Recommender systems predict users' preferences based on the users' search behavior patterns on websites and recommend items that may be of interest to them. Since the mid-1990s, a great deal of research has focused on how to derive sufficient numbers of explicit ratings to construct users' profiles, which are then used to predict users' preferences [2,7,16,24]. Personalized recommendation systems, which provide users with one-to-one guidance, have become very important in electronic commerce. Various methods for learning users' interests or preferences from text documents or Web pages have been proposed for real-world applications based on recommender systems; for example, adaptive Web page recommendation services [2,5,32], personalized online newspapers [24], movie recommendations [7], joke recommendations [11] and review article recommendations [20]. These approaches are based on learning algorithms that modify users' profiles to reflect the users' interests or preferences.

Various information filtering approaches have been proposed to predict users' preferences. Generally, they can be classified as content-based filtering or collaborative filtering approaches. The latter, which are widely used in a large number of diverse applications [1,10], enable websites to recommend products based on the preferences of peer groups whose interests are similar to those of the target users. GroupLens [16] and MovieLens [7] are two well known collaborative recommender systems that recommend

*Corresponding author: I.-Chin Wu, 510 Chung Cheng Road, Xinzhuang Dist., Xinbei City 24205, Taiwan. Tel.: +886 2 2905 2739; Fax: +886 2 2905 2182; E-mail: icwu.fju@gmail.com.

news and movies respectively. The GroupLens project applies a personalized neighborhood-based collaborative filtering technique to Usenet news, a high-volume and high-noise set of discussion groups distributed across the Internet. In the MovieLens system, users provide ratings for movies they have seen. Then, the system applies a collaborative filtering algorithm to recommend movies that similar users liked. Although collaborative filtering is a promising way to expand a user's profile of interests, it cannot resolve or mitigate the sparse ratings problem, which occurs because an individual can only vote for a small fraction of all the items available on a website. The problem is usually resolved by employing hybrid methods, i.e., by combining content-based and collaborative filtering techniques [1,2,4]. Ma et al. [20] exploited users' social trust networks and social tags, i.e., social context information, to alleviate the sparsity problem and improve the quality of recommendations. They conducted a series of experiments on the Epinions and Movielens datasets to demonstrate the effectiveness of their approach.

The computational efficiency of collaborative filtering algorithms has also motivated a substantial amount of research in recent years. Memory-based collaborative filtering algorithms are simple and make reasonable recommendation results; however, they have limited scalability and they are not efficient in the training and prediction phases [9,12,27]. By contrast, model-based collaborative filtering algorithms, such as regression-based methods and Latent Semantic Indexing (LSI), are efficient in the prediction phase of recommendation, but constructing the model requires a substantial amount of time. Moreover, model-based algorithms are more suitable for static data [27,31]. To satisfy the increasing requirements of interactive recommender systems and information retrieval systems, an effective recommendation system must be able to 1) produce accurate predictions based on low density data; 2) make recommendations in a timely manner; and 3) fit the Web context to learn users' dynamic preferences.

To improve the performance of webpage searches and personalize the retrieved results, Kim and Cho [14] utilized a fuzzy inference method to construct a fuzzy concept network based on the user's profile. In this work, we propose a genre-based fuzzy (G-Fuzzy) inference and grouping approach that estimates the unknown values in a sparse matrix and identifies appropriate neighborhoods by applying an α -cut operation. The contribution of our research is threefold. 1) To resolve the cold start and sparse rating problems, we propose a novel hybrid movie preference filtering approach called G-Fuzzy. 2) We integrate the fuzzy inference method described in [15] with the proposed genre-based filtering method to locate users' with similar interests and identify the neighborhood of users. 3) We evaluate the content-based, collaborative and hybrid filtering methods under various levels of sparse ratings for movies. To evaluate the tradeoff between the methods, we conduct a series of experiments that verify the effectiveness of the proposed approach in terms of the precision rate and computation time.

The remainder of this paper is organized as follows. The next section contains a review of related work. In Section 3, we describe the framework of the proposed approach; and in Section 4, we introduce a genre-based fuzzy inference approach in detail. Sections 5 and 6, we consider the experimental methods, and results; and Section 7 contains some concluding remarks.

2. Related works

A recommendation system uses a customer's purchase history to determine his preferences and identify products that he may wish to purchase. They have been integrated with e-commerce applications to support product recommendations [28]. Most researchers have focused on recommendation technologies that rely explicitly on users' ratings. Content-based and collaborative filtering are usually used to make recommendations. Hybrid techniques that exploit the strengths of content-based and collaborative filtering methods have also been proposed.

2.1. Content-based filtering

In content-based systems, products are described by a common set of attributes extracted from the available product descriptions [6]. Customers' preferences are predicted by analyzing the relationship between the product ratings that an individual has given and the corresponding product attributes. The major limitation of content-based filtering methods is that they may not have a serendipitous finding to the user because they rely on the user's historical preferences to build the profile or prediction model. Another problem with such systems is that a large set of key attributes must be identified [6]. If the set is too small, there will be insufficient information to build the customer's profile. However, too many attributes may also be problematic because there will be a large number of trainable parameters in the model, which will result in a poor performance [23]. Blanco-Fernández et al. [3] improved the performance of content-based filtering by compiling user profiles with the aid of an ontology of items and investigated a TV program domain. Their approach integrates content-based filtering with a spreading activation process and semantic reasoning mechanisms in the ontology to improve the accuracy of recommendations. Although the approach alleviates the overspecialization problem in content-based filtering and resolves the sparsity problem in collaborative filtering, it requires additional information about the target domain and must maintain detailed user profiles.

2.2. Collaborative filtering

Usually, collaborative filtering (CF) methods exploit nearest-neighborhood algorithms to recommend products to target customers based on the preferences of neighbors; that is, customers who have similar preferences to the target customers [13,18]. The methods can be classified as either memory-based or model-based. Memory-based CF methods, such as neighborhood-based and item-based approaches, use the entire user-item database, or a sample of it, to generate predictions [30]. Users are classified into several groups based on similar interests. The neighborhood-based CF algorithm is a widely used memory-based CF filtering technique that (1) calculates the similarities between users; (2) determines the neighborhood of the target user; and (3) produces a prediction for the active user [12,25,30]. Normally, the cosine measure and the correlation-based similarity measure are used to calculate the similarities between users based on their previous ratings, as shown in Eq. (1). The equation formulates the similarity between the target user l and a neighbor k .

$$w_{l,k} = \frac{\sum_{i \in I} (r_{l,i} - \bar{r}_l) \times (r_{k,i} - \bar{r}_k)}{\sqrt{\sum_{i \in I} (r_{l,i} - \bar{r}_l)^2} \times \sqrt{\sum_{i \in I} (r_{k,i} - \bar{r}_k)^2}}, \quad (1)$$

where I is the item set co-rated by user l and k ; and \bar{r}_l and \bar{r}_k are the average item ratings given by l and k respectively. Equation (2) shows the formula used to predict an item for the target user.

$$P_{l,i} = \bar{r}_l + \frac{\sum_{k=1}^n [w_{l,k} \times (r_{k,i} - \bar{r}_k)]}{\sum_{k=1}^n w_{l,k}}, \quad (2)$$

where, $P_{l,i}$ represents the predicted item i for the target user l , and n is the number of neighbors that rated item i .

Model-based collaborative filtering methods, such as Bayesian models or clustering models, use classification algorithms as CF models if the user ratings are categorical. Regression models and SVD methods can be used for numerical ratings [30]. Both memory-based and model-based methods are affected by the cold-start and rating sparsity problems.

- *Cold-start*: This problem arises when a new product is introduced to the market because there are no previous ratings that can be used to compute the correlations between users or train the models [6, 16].
- *Rating sparsity*: Most commercial recommender systems have a large set of items and customers. Thus, the user-item rating matrix may be extremely sparse and will only contain a few items co-rated by users [1,27]. If a sparse matrix is used to calculate the similarity score and make predictions, the results will be inaccurate.

A number of hybrid CF recommendation algorithms have been proposed to address the shortcomings of content-based and collaborative filtering methods [12]. For example, Fab is a well known classic content-based CF recommender system that retrieves information available from the World Wide Web [2].

2.3. Hybrid filtering

Although CF algorithms are effective, they cannot resolve the cold-start and sparsity problems. However, a hybrid filtering approach, which exploits collaborative filtering as the primary technique and content-based filtering to make recommendations [19,22,34], is used to deal with the problems. In other words, the hybrid approach exploits the strengths of content-based and collaborative filtering. It has been found that hybrid CF algorithms, such as the content-boosted CF algorithm, can resolve the sparsity problem by using external content information to make predictions for new users or new items [30]. Hybrid filtering techniques may be linear or sequential combinations of collaborative and content-based filtering techniques [19]. Linear combinations use the prediction results of the CF and content-based filtering methods together. The relative weights of the two techniques depend on the number of users, as in the Tango system [1]. By contrast, sequential combinations apply a content-based filtering algorithm to find users with similar interests to the target user, and then use a CF algorithm to make predictions, as in the Fab system [2]. In this work, we adopt the concepts of the sequential combinations of hybrid filtering approach. Namely, we propose a novel hybrid movie preference filtering approach based on movie genres to take the strength of content-based and CF methods. A user's movie preferences can be expressed as a vector of genres with associated scores to mitigate the cold start problem. We also use a fuzzy inference method to resolve the sparse ratings problems and improve the computation time in the prediction phase.

3. The framework

Figure 1 shows the system framework of the proposed genre-based fuzzy inference approach.

Movie rating collection module: We use the MovieLens data sets collected by the GroupLens research group at the University of Minnesota. The data sets contain information about 943 users, 1682 movies and 100,000 ratings. We extract samples from the original data sets based on the number of ratings given by each user, which may be 150, 250, or 300 ratings. In addition, we capture users' on-line search behavior when they browse movies, and each user's explicit ratings for movies are recorded by the

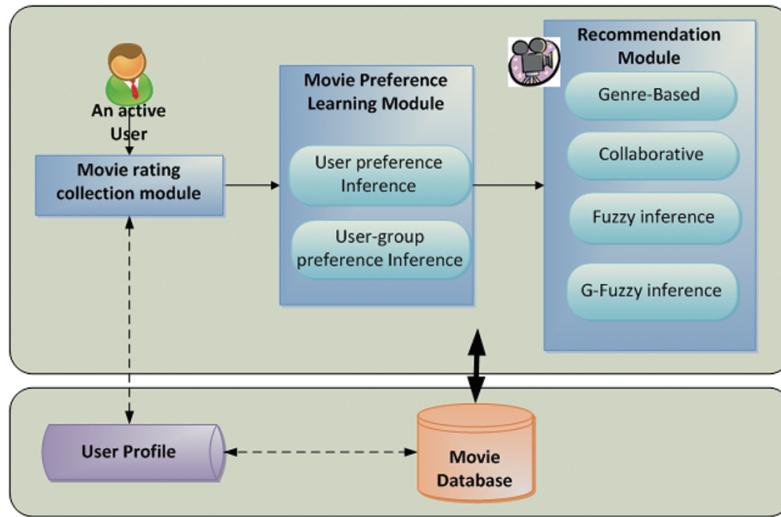


Fig. 1. The framework of the recommendation system. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

system. A log-parsing engine in the user behavior tracking module analyzes the log-files that track the user's interaction with the system. We discuss the sparseness rates of sample data in Section 4.

Movie preference modeling and inference modules: As mentioned earlier, the proposed content-based movie preference modeling approach is based on movie genres. Each user's movie preferences can be expressed as a vector of genres with associated scores, i.e., $UP_{userx} = (Genre_{x,1}, Genre_{x,2} \dots, Genre_{x,n})$. To resolve the sparse matrix problem, we utilize a fuzzy inference method to infer the implicit and transitive relationships among users. Moreover, to reduce the computation time required to identify neighborhoods of users, we use the α -cut method to group users that have similar movie preferences.

Recommendation module: This module matches a genre-based user's profile with the database to streamline the movie recommendation process. We use *Genre-based*, *Collaborative*, *Fuzzy Inference* and *Genre-based Fuzzy Inference* filtering methods to evaluate the recommendations in terms of the precision rate and computation time.

4. The genre-based fuzzy modeling and inference approach

4.1. Genre-based user movie preference modeling

User preference modeling: A movie may belong to more than one genre, for example, Toy Story III belongs to the animation, adventure, and comedy genres. Each user's movie preferences (UP) are expressed as a set of movie genres with associated scores, as shown in Eq. (3);

$$Genre_{userx,j} = \frac{\sum_{k=1}^{n_j} r_k}{n_j}, \quad (3)$$

where r_k denotes the ratings for movies belonging to genre j , and n_j is the number of movies that belong to the j th genre. Thus, $Genre_{j,useri}$ means the x th user's average movie preference for $genre_j$.

Table 1(a)
Users' ratings for different movie genres

User	Movie	Action	Art	Adventure	Crime	Romance	War	Comedy	Rating	Standardize
U1	M1	v		v				v	4	0.8
	M2	v		v	v		v		2	0.4
	M3		v			v		v	1	0.2
	M4			v	v		v		4	0.8
U2	M1	v		v				v	0	0.0
	M2	v		v	v		v		3	0.6
	M3		v			v		v	4	0.8
	M4			v	v		v		1	0.2
U3	M1	v		v				v	2	0.4
	M2	v		v	v		v		3	0.6
	M3		v			v		v	0	0.0
	M4			v	v		v		4	0.8
U4	M1	v		v				v	4	0.8
	M2	v		v	v		v		3	0.6
	M3		v		v		v		1	0.2
	M4			v	v		v		4	0.8

Accordingly, the UP of user_x can be expressed as follows:

$$UP_{userx} = (Genre_{x,1}, Genre_{x,2}, \dots, Genre_{x,n}), \quad (4)$$

Movie preference prediction: A user's preference for a target movie t can be derived from his/her historical UP , as shown in Eq. (5);

$$M_{t,usex} = \frac{\sum_{j \in SG_{x,t}} Genre_{x,j}}{|SG_{x,t}|}, \quad (5)$$

where M_t is the predicted preference score for a movie t , and SG denotes the total number of genres that the t belongs to. Note that SG is a subset of UP_x for movie t .

Example 1: Table 1(a) lists users' ratings for movies with the corresponding genres. We use the table to calculate a user's preferences for different movie genres. The user's preference for a target movie is calculated based on Eq. (3). The results are shown in Table 1(b). We predict user 4 s (U4) preference for movie 4 (M4) based on Eq. (5). Note that movie 4 belongs to the Adventure, Crime and War genres. Accordingly, the predicted preference rating is 0.633, which is calculated as follows:

$$M_{4,user4} = \frac{\sum_{t=1}^3 (0.7 + 0.6 + 0.6)}{3} = 0.633$$

4.2. Genre-based fuzzy inference and grouping method

4.2.1. Fuzzy inference method

We employ a fuzzy inference method that identifies users with similar movie interests by analyzing the results of genre-based preference modeling. The steps of the process are as follows.

Step one: A *user-movie* relation matrix is constructed according to *Definition 1*.

Table 1(b)
Users preferences for different genres

Genre \ user	Action	Art	Adventure	Crime	Romance	War	Comedy
UP _{user1}	0.600	0.200	0.667	0.600	0.200	0.600	0.500
UP _{user2}	0.600	0.800	0.400	0.400	0.800	0.400	0.800
UP _{user3}	0.500	0.000	0.600	0.700	0.000	0.700	0.400
UP _{user4}	0.700	0.200	0.700	0.600	0.200	0.600	0.500

$$S^2 = S \circ S = \begin{bmatrix} 1 & 0.7 & 0.4 & 0.2 & 0.7 \\ 0.7 & 1 & 0.4 & 0.5 & 0.8 \\ 0.4 & 0.4 & 1 & 0 & 0.4 \\ 0.2 & 0.5 & 0 & 1 & 0.5 \\ 0.7 & 0.8 & 0.4 & 0.5 & 1 \end{bmatrix} \quad S^3 = S \circ S \circ S = \begin{bmatrix} 1 & 0.7 & 0.4 & 0.5 & 0.7 \\ 0.7 & 1 & 0.4 & 0.5 & 0.8 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.7 & 0.8 & 0.4 & 0.5 & 1 \end{bmatrix}$$

Fig. 2. Inferring preferences based on users' similarity relations.

Definition 1: Let $X = \{x_i\}_{i=1}^N$ be a set of users, and let $M = \{m_j\}_{j=1}^l$ be a set of movies. In addition, let $V = \{v_{ij}\}_{ij}$ be a user-movie matrix S in which entry v_{ij} denotes user x_i 's degree of interest in a movie.

Then, we calculate *Pearson's correlation coefficient* to measure the similarity of users' movie preferences and construct a *user-user* similarity matrix to represent their movie interests.

Step two: To resolve the sparse matrix problem, we use the fuzzy inference method to infer the implicit and transitive relationships among users' preferences. The method identifies users with similar interests based on their inherent transitive relationships. As a result, the system can identify similar users, even if there are very few explicit similarity scores in the similarity relationship matrix S . We employ Klir and Yuan's [15] transitive max-min closure method to derive a reflective, symmetric, and transitive matrix, which is a fuzzy equivalence matrix. The fuzzy equivalence relation can be used to classify the matrix's outputs. Note that a fuzzy relation \tilde{R} is a fuzzy equivalence relation if it satisfies the properties of reflexivity, symmetry, and transitivity. Klir and Yuan's definition of a transitive max-min closure S_T in a similarity matrix S is given in *Definition 2*.

Definition 2: Given an n -by- n fuzzy similarity relationship matrix S representing the fuzzy relations among a set of users X , where $|X| = n$, a transitive max-min closure S_T of S is derived as $S_T = S^y$ by applying a sequence of max-min operations on the relation matrix until $S^y = S^{y+1} = \dots = S^\infty$. Note that $S^y = S^{y-1} \circ S^{y-1}$, where y is an integer, $1 \leq y \leq n - 1$ and \circ denotes a fuzzy max-min operation. The max-min composition and max operator for set unions are used to derive the transitive max-min closure S_T . The fuzzy max-min operation is defined as follows:

$$\tilde{\zeta}^y(x_i, x_j) = \max_{E_x \in X} \min(\tilde{\zeta}^{y-1}(x_i, x_u), \tilde{\zeta}^{y-1}(x_u, x_j)), \quad (6)$$

where $\tilde{\zeta}^y(x_i, x_j)$ represents an element in S^y , and $\tilde{\zeta}^{y-1}(x_i, x_u)/\tilde{\zeta}^{y-1}(x_u, x_j)$ represents an element in S^{y-1} .

Example 2: If $\tilde{\zeta}^y(x_1, x_2) = 0.7$ and $\tilde{\zeta}^y(x_2, x_3) = 0.4$, but $\tilde{\zeta}^y(x_1, x_3) = 0.0 \leq \min(0.7, 0.4)$, the fuzzy transitive property will not be satisfied. Thus, the above similarity relation S does not satisfy the fuzzy transitivity property because s is not a fuzzy equivalence relation. After two more compositions, the fuzzy transitive relation $S^3 = S \circ S \circ S$ is obtained and all three conditions of the fuzzy equivalence relation are satisfied, as shown in Fig. 2.

Table 2
The MovieLens dataset and the sampled data

Number of ratings*	Users	Movies	Expected ratings	Actual number of ratings	Sparseness rate
The MovieLens dataset					
Unlimited	943	1682	1586126	100000	93.70%
Data samples taken from the MovieLens dataset					
RNs \geq 150	230	1653	380190	58071	84.76%
RNs \geq 250	91	1609	146419	31205	78.69%
RNs \geq 350	33	1576	52008	14363	72.38%

$$R_{0.7} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad R_{0.7} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

(a) (b)

Fig. 3. (a) Selection of groups with similar interests based on 0.7-cut values. (b) Selection of groups with similar interests based on 0.5-cut values.

4.2.2. Fuzzy clustering: Selection of groups with similar interests

Next, we apply the α -cut operation to generate a proper set of neighborhoods. The operation can be applied to the equivalence matrix S_T to cluster any degree of interest to group users in X , where $\alpha \in (0, 1)$. Different subsets of equivalence relations can be derived by setting different α degrees in the matrix S_T to partition set X . One way to improve the quality of recommendations is to provide collaborative recommendations based on a group of users that have similar movie preferences. A standard (non-fuzzy or crisp) equivalence relation S_α can be obtained from a fuzzy equivalence relation, say S_e , by applying an α -cut operation to the fuzzy relation S_e . The crisp relation S_α is defined as $S_\alpha = \{(i, j) | x_s(i, j) \geq \alpha\}$.

Example 3: If the strength of the relation between i and j is greater than or equal to α , it is set at 1; otherwise, it is set at 0. The outputs are classified based on the values of α . For example, if we set $\alpha = 0.7$, we obtain the crisp equivalence relation matrix shown in Fig. 3(a). We can identify three groups in the matrix, i.e., $\{\{x_1, x_2, x_5\}, \{x_3\}, \text{ and } \{x_4\}\}$. However, if we set $\alpha = 0.5$, we can only identify two groups and the crisp equivalence relation becomes $\{\{x_1, x_2, x_4, x_5\}, \{x_3\}\}$, as shown in Fig. 3(b).

Finally, the predicted value of a movie recommended to the target user is computed as the weighted sum of the neighbors' ratings.

5. Experiment setup and procedure

5.1. Dataset analysis

We used one of the MovieLens data sets, collected by the GroupLens research group at the University of Minnesota. The data sets have proven useful in research on algorithms for recommender systems, as well as in numerous collaborative filtering publications (<http://www.grouplens.org/node/12>) [7,22]. They contain information about 943 users and 1682 movies with 100,000 ratings, as shown in Table 2. In the MovieLens system, users rate movies they have seen on a scale of 1 to 5. Then, the system makes recommendations via a collaborative filtering algorithm that identifies movies that similar users liked. We sampled data from the MovieLens dataset based on the number of ratings (RNs) given by

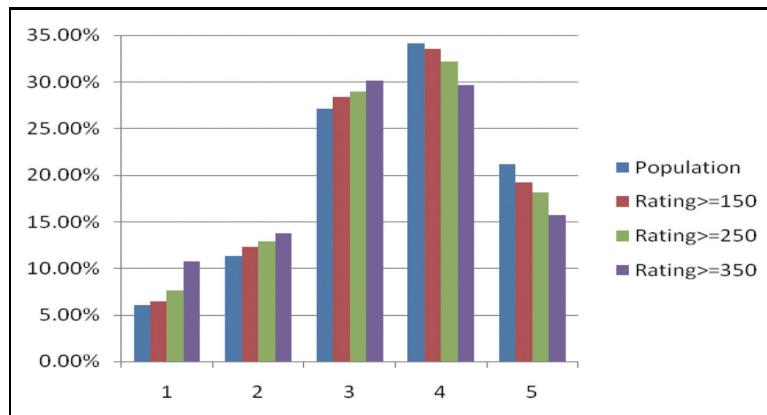


Fig. 4. The proportion data and sample data for different rating scales. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

each user, which were at least 150, 250, or 300 ratings. Accordingly, the total number of ratings for the three sample data sets were 58,071, 31,205, and 14,363, and the sparse rates were 84.76%, 78.69%, and 72.38%, respectively. Table 2 provides an analysis of the sparse rates of each sample data sets from the MovieLens dataset. We also calculated the distribution of the different rating scales with respect to the population and different data samples, as shown in Fig. 4. Clearly, the distribution of the sampled data was similar to that of all the data.

5.2. Experimental methods and procedures

5.2.1. Experimental methods and parameter settings

We conducted two experiments to compare the effectiveness of the proposed *Genre-based Filtering*, *Fuzzy Inference Filtering* and *Genre-based Fuzzy (G-Fuzzy) Inference Filtering* methods with that of the promising *Collaborative-based Filtering (CF)* method. The methods are described in detail in Table 3. In Experiment 1, we evaluated the effectiveness of the proposed *Genre-based Filtering* method and the traditional automatic *Collaborative Filtering (CF)* method. The latter is a statistical approach that automates all the recommendation procedures used to predict user preferences based on the preferences of similar users [10]. Recall that an individual may only vote for a small number of items in a database, so ratings tend to be sparse. To resolve the problem, we propose two hybrid methods, namely, the *Fuzzy Inference* method and the *G-Fuzzy Inference* method. In Experiment 2, we evaluated the effectiveness of the two hybrid methods and compared the precision and time complexity of all the methods. We also compared the performance of the correlation-based and cosine similarity measures applied by the two hybrid methods.

5.2.2. Experimental procedures

The experimental procedures for the *Fuzzy Inference* and *G-Fuzzy Inference* methods are shown in Fig. 5. The steps are as follows. (1) A genre-based profile is generated for each user, as mentioned in Section 4.1. (2) The *Fuzzy Inference* method infers the user's preferences, as explained in Section 4.2. The *G-Fuzzy Inference* method inputs the null values of the user's movie matrix, S , based on the genre preference values in advance. It then uses the *Fuzzy Inference* method to infer the user's preferences. (3) The α -cut operation is applied to generate a proper set of neighborhoods. (4) We sample the training set and testing set to predict the user's movie preferences.

Table 3
The methods evaluated in each experiment

Method	Description	Equations and parameter settings
Experiment 1: Traditional movie recommendation methods		
Collaborative filtering (CF)	<ul style="list-style-type: none"> The CF method is based on the traditional automatic CF method. We set different user preference similarity (UPS) thresholds to select neighborhoods. 	<i>Collaborative filtering method with A similarity threshold of 0.2, 0.3, or 0.4 is set for the user preference similarity (UPS) when selecting the proper neighborhoods in Eq. (2).</i>
Genre-based filtering	<ul style="list-style-type: none"> The Genre-based filtering method, which is based on the user's preference for a specific genre of movies constructs a profile and then makes recommendations. It is similar to the traditional content-based filtering method. 	
Experiment 2: Comparison of Hybrid methods		
Fuzzy inference filtering (with correlation-based and cosine similarity)	<ul style="list-style-type: none"> The fuzzy inference filtering method utilizes the transitive closure to infer the null value of the similarity matrix, and the α cut operation to group users. The recommendations are made within the groups. 	<i>Fuzzy inference filtering method with α set at 0.4, 0.5, or 0.6 to group users.</i>
Genre-based fuzzy inference (G-fuzzy inference) (with correlation-based and cosine similarity)	<ul style="list-style-type: none"> The <i>G</i>-fuzzy method is similar to the <i>Fuzzy inference</i> method. It utilizes the results of the <i>Genre-based</i> method to rate movies without ratings based on the genre values, and then applies the Fuzzy-based inference method, as shown in Fig. 6. 	<i>G-Fuzzy inference filtering method with α set at 0.4, 0.5, or 0.6 to group users.</i>

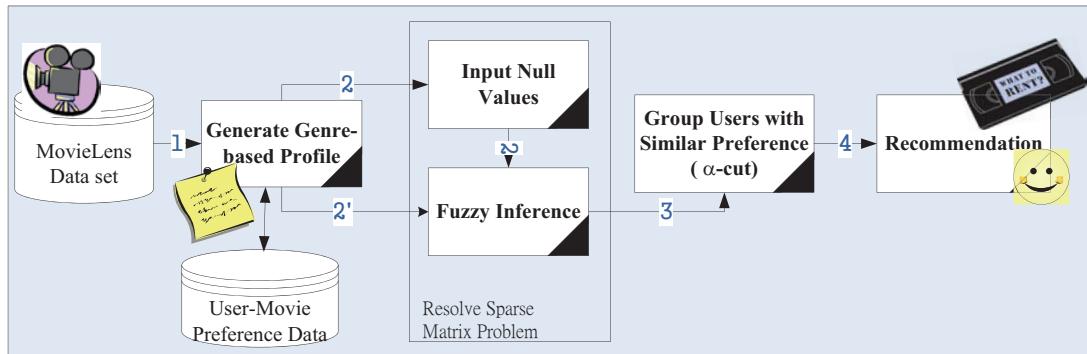


Fig. 5. Experimental procedure for the *genre-based fuzzy inference* method. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

6. Experiment results and discussion

6.1. Experiment 1: Comparison of genre-based and CF methods

In this experiment, we compared the performance of the *Genre-based Filtering* method and *CF* method for different numbers of $p@n$ recommended movies, i.e., the precision of the first n movies recommended. For the *Genre-based Filtering* method, we aggregated a user's ratings for movies in terms of movie genres (e.g., action, comedy, drama, etc.) in order to predict the user's ratings for unrated movies

Table 4
Precision of *genre-based filtering* under different training/testing proportions

	Average			The best case			The worst case		
	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150
60/40	0.5589	0.6621	0.6048	0.5789	0.6956	0.6231	0.5276	0.6264	0.5955
70/30	0.5627	0.6972	0.6500	0.5917	0.7179	0.6699	0.5302	0.6723	0.6363
80/20	0.6082	0.6082	0.6646	0.6266	0.7328	0.6740	0.5872	0.6656	0.6580
90/10	0.6436	0.7264	0.6906	0.6694	0.7727	0.7013	0.6266	0.7090	0.6740

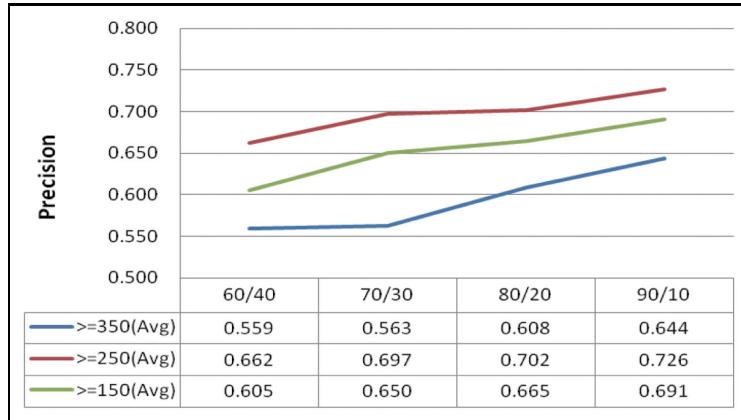


Fig. 6. Comparison of different RNs under the *genre-based filtering* method. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

based on Eq. (4) and make recommendations. The CF method, which is based on the traditional automatic collaborative filtering (CF) technique, makes a recommendation by computing the similarity between the target user's preference and the preferences of other users. The rationale behind the method is that recommendations about products or services are often passed on by word of mouth [25,29]. CF algorithms must be able to (1) deal with very sparse data, (2) scale up with increasing numbers of users and items, and (3) make satisfactory recommendations in a timely manner [30]. We use Pearson correlation coefficient to measure the similarity of two users, as shown in Eq. (2). In the equation, the similarity threshold is set at 0.2, 0.3, or 0.4 for the user preference similarity (*UPS*). As shown in Tables 5 and 6, we varied the proportions of the training and testing sets using 60/40, 70/30, 80/20, and 90/10 splits for 10-fold cross validation (CV) of the three RN settings, i.e., 150, 250, and 350 ratings.

6.1.1. *Genre-based filtering results*

Observation 1: Table 4 shows that the 90/10 split achieves the best performance in terms precision in all cases. It yields the best results under various numbers of ratings. In addition, the precision of the best cases and the average cases is similar under the 90/10 split of the sampling data set.

Observation 2: Figure 6 shows that the best average performance is achieved when the number of ratings (RNs) is greater than 250, irrespective of the proportions of the training and testing sets.

6.1.2. *The CF method*

In the CF method, we set the threshold to select the proper number of neighborhoods. In this experiment, we chose users whose correlation values were in the top 20%, 30% and 40% of the targeted users as our thresholds. The groups of targeted users are called *user preference similarity groups (UPS)*, i.e., *UPS(0.2)*, *UPS(0.3)*, and *UPS(0.4)* respectively. The experiment results are shown in Table 5 and Fig. 7.

Table 5
Precision of *CF* under different training/testing proportions for various UPS

	UPS(0.2)			UPS(0.3)			UPS(0.4)		
	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150
The average (10-runs)									
60/40	0.9351	0.9593	0.9635	0.9239	0.9615	0.9651	0.9028	0.9634	0.9653
70/30	0.9358	0.9580	0.9680	0.9345	0.9627	0.9666	0.9165	0.9597	0.9656
80/20	0.9405	0.9631	0.9654	0.9429	0.9639	0.9670	0.9341	0.9605	0.9672
90/10	0.9436	0.9686	0.9651	0.9451	0.9641	0.9655	0.9403	0.9642	0.9668
The best case									
60/40	0.9469	0.9645	0.9667	0.9313	0.9653	0.9683	0.9137	0.9632	0.9664
70/30	0.9458	0.9664	0.9667	0.9460	0.9654	0.9676	0.9246	0.9632	0.9667
80/20	0.9518	0.9659	0.9665	0.9481	0.9664	0.9685	0.9379	0.9637	0.9687
90/10	0.9489	0.9717	0.9664	0.9508	0.9674	0.9663	0.9536	0.9653	0.9677
The worst case									
60/40	0.9235	0.9541	0.9602	0.9173	0.9531	0.9611	0.8843	0.9578	0.9643
70/30	0.9275	0.9510	0.9638	0.9145	0.9591	0.9658	0.9080	0.9527	0.9632
80/20	0.9354	0.9588	0.9645	0.9397	0.9621	0.9654	0.9318	0.9551	0.9662
90/10	0.9361	0.9659	0.9637	0.9347	0.9629	0.9645	0.9310	0.9629	0.9662

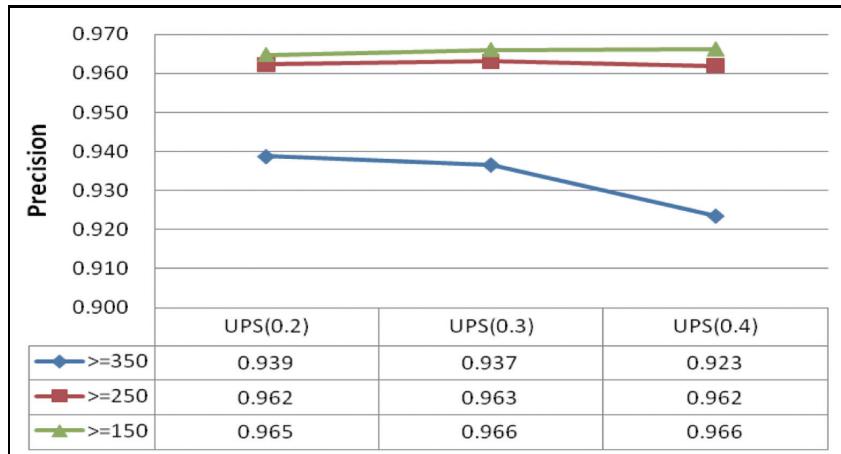


Fig. 7. Comparison of different RNs under the *CF* method (average case). (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

Observation 1: Table 5 shows that the 90/10 split achieves the best performance in terms of precision when the number of ratings (RNs) is greater than 250 or 350 in terms of precision in nearly all cases.

Observation 2: Figure 7 shows that the best average performance is achieved when the number of ratings (RNs) is greater than 150, irrespective of the proportions of the training and testing sets. When the number of RNs is greater than 250, the performance is slightly worse than that of 150. Interestingly, more ratings, i.e., $RNs \geq 350$, do not improve the performance of collaborative filtering. Thus, the results are influenced by the number of users used to recommend items.

Observation 3: Table 5 and Fig. 8 show that, on average, *UPS(0.2)* and *UPS(0.3)* perform slightly better than *UPS(0.4)*. Varying the number of neighborhoods seems to influence the result of a dataset with higher density and a smaller number of neighborhoods, i.e., when the number of *RNs* is greater than 350.

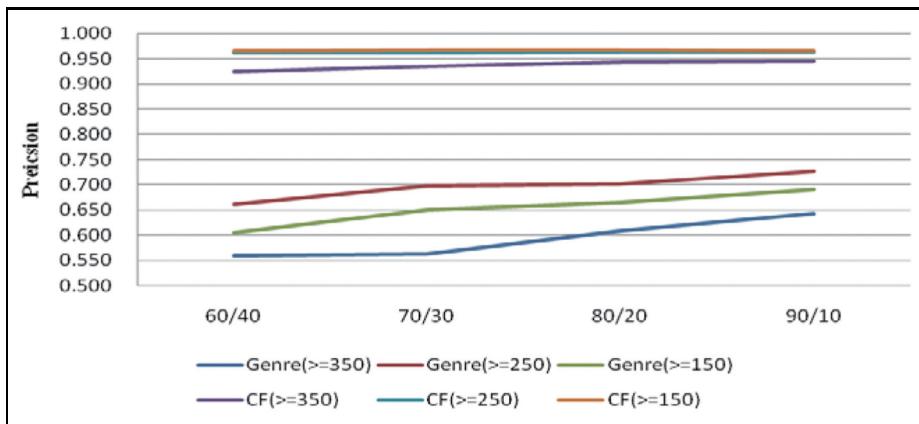


Fig. 8. Comparsion of the genre-based and CF methods in terms of precision. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

6.1.3. Discussion

Figure 8 shows that the *CF* method outperforms the *Genre-based Filtering* method in terms of precision, regardless of the sparseness rates. However, it requires much more computation time, especially for a dataset with a higher sparseness rate and more users, as shown in Fig. 9. Note that although a sample in which the set number of ratings (*RNs*) is greater than 150 yields the best performance, it requires much more computation time than the other sample sets. On the other hand, the computation time of the *Genre-based Filtering* method is not sensitive to the sparseness rates, as shown in Fig. 9. Accordingly, there can be a trade-off between the precision rate and computational cost when selecting a method to build the recommendation model. In the next section, we introduce and compare two hybrid methods that incorporate the strengths of the proposed genre-based and *CF* methods.

6.2. Experiment 2: Comparison of hybrid methods

6.2.1. Experiment 2-1: The pearson correlation-based hybrid methods

Based on the results of Experiment 1, we try to minimize the sparsity problem and the computation time. Specifically, we propose two hybrid methods: the *Fuzzy Inference Filtering* method, and the *Genre-based Fuzzy Inference Filtering* method (*G-Fuzzy Inference*). The methods combine the features of the *CF* and the *Genre-based Filtering* methods, which were introduced in Sections 2.2 and 4.2. We used Pearson's correlation coefficient to predict users' movie preferences. For the two hybrid filtering methods, we set different α degrees to cluster users into groups. In this experiment, we set α at 0.4, 0.5, and 0.6 respectively to group users. Table 6, Figs 10 and 11 show the experiment results.

Observation 1: Table 6 shows the results of varying the proportions of the training and testing sets as 60/40, 70/30, 80/20, and 90/10 for 10-fold cross validation (CV) for the three *RNs* settings. Interestingly, in contrast to the results of the *Genre-based Filtering* and *CF* methods, these results show that the 90/10 split yielded the worst average precision performance. The 60/40 and 70/30 splits achieved better average precision performances. The finding suggests that the proposed hybrid methods can achieve a better performance without significant amounts of training data.

Observation 2: Figure 10 shows that the average precision performance is the worst when the number of ratings (*RNs*) is greater than 150, regardless of the α -cut level. In addition, $RNs \geq 250$ and $RNs \geq 350$ returned similar results under different levels of α -cut settings.

Table 6
Precision of the two hybrid filtering methods under different training/testing splits with various α settings

	$\alpha \geq 0.4$			$\alpha \geq 0.5$			$\alpha \geq 0.6$		
	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150
Fuzzy inference (Hybrid 1): The average (10-runs)									
60/40	0.9424	0.9721	0.4603	0.9284	0.9767	0.4555	0.9283	0.9579	0.4715
70/30	0.9459	0.9742	0.4952	0.9545	0.9767	0.4940	0.9303	0.9598	0.4995
80/20	0.9444	0.9697	0.5038	0.9623	0.968	0.5047	0.9280	0.9590	0.5079
90/10	0.9317	0.9569	0.4917	0.9383	0.9553	0.4952	0.9143	0.9510	0.4934
Genre-based fuzzy inference (Hybrid 2): The average (10-runs)									
60/40	0.9943	0.9901	0.5332	0.9921	0.9917	0.5350	0.9919	0.9900	0.5317
70/30	0.9914	0.9914	0.5375	0.9947	0.9930	0.5368	0.9941	0.9912	0.5404
80/20	0.9888	0.9883	0.5392	0.9920	0.9901	0.5364	0.9921	0.9886	0.5379
90/10	0.9884	0.9807	0.5287	0.9859	0.9828	0.5302	0.9882	0.9791	0.5262

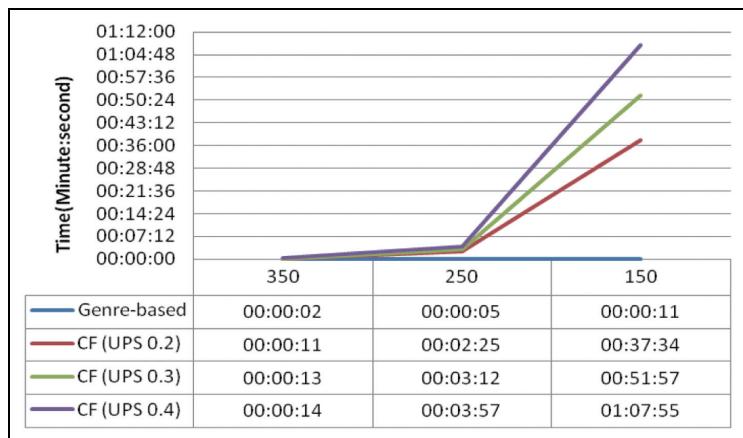


Fig. 9. Cmparsion of the *Genre-based Filtering* and *CF* methods in terms of computational efficiency. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

Observation 3: Finally, we compare the *Fuzzy Inference* and *G-Fuzzy Inference Filtering* methods. Figure 11 shows that, in terms of precision, the *G-Fuzzy* method outperformed the *Fuzzy Inference Filtering* method under various α -cut settings. The *Fuzzy Inference* method achieved the best performance under various training/test dataset splits when α was set at 0.4. For the *G-Fuzzy Inference Filtering* method, the results are similar regardless of the α -cut levels. The results show that the Pearson correlation-based *G-Fuzzy Inference Filtering* method yields better precision rates and more stable results than the *Fuzzy Inference* method, regardless of the parameter settings.

6.2.2. Experiment 2–2: The cosine similarity-based hybrid methods

In this experiment, we compared the performance of the cosine similarity measure with that of the Pearson correlation-based method. We evaluated two cosine similarity-based hybrid methods, namely the *Fuzzy Inference Filtering* method and the *Genre-based Fuzzy Inference Filtering* method (*G-Fuzzy Inference*). In both methods, we set different α degrees to cluster users into groups. Specifically, we set α at 0.4, 0.5, and 0.6 respectively to group users. Table 7, Figs 12 and 13 show the experiment results.

Observation 1: Table 7 shows the results of varying the proportions of the training and testing sets as 60/40, 70/30, 80/20, and 90/10 for 10-fold cross validation (CV) under three *RN* settings for the two hybrid methods. In contrast to the results of the *Genre-based Filtering* and *CF* methods, the 90/10 split

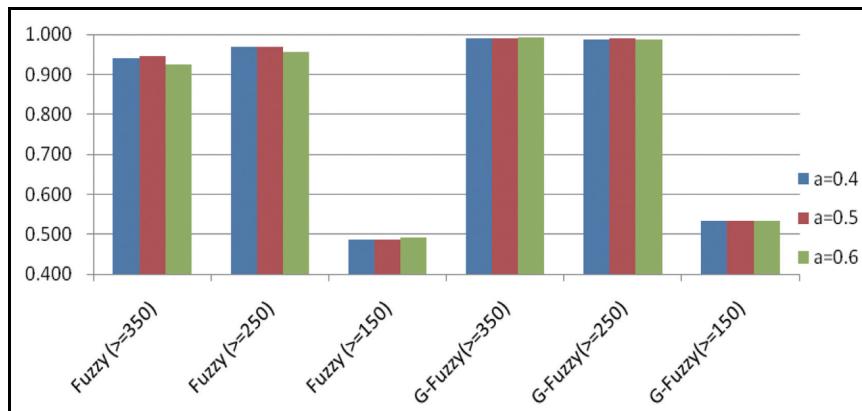


Fig. 10. Comparison of the average cases under the two hybrid methods using different α degree settings (Pearson correlation-based similarity). (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

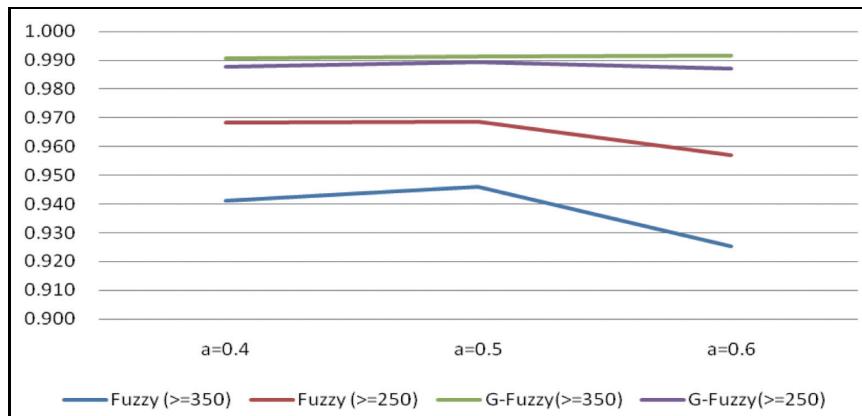


Fig. 11. Comparsion of the precision rates of the two hybrid filtering methods (Pearson correlation-based similarity). (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

yielded the worst performance in terms of the average precision. The 60/40 and 70/30 splits achieved the best average precision performance. The results are similar to those of the correlation-based hybrid methods.

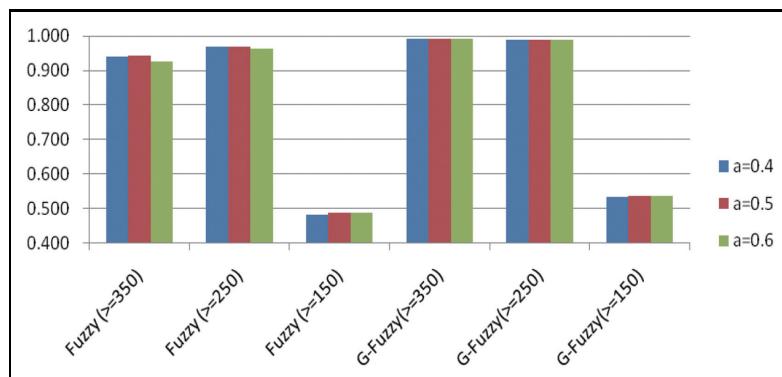
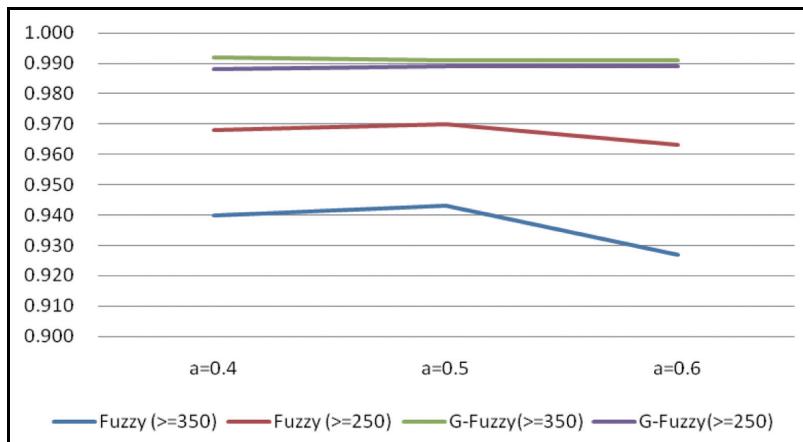
Observation 2: Figure 12 shows that the average precision performance is the worst when the number of ratings (RNs) is greater than 150, regardless of the α -cut level. In addition, the $RNs \geq 250$ and $RNs \geq 350$ results were similar under different α -cut levels, especially for the *Genre-based Fuzzy Inference Filtering method* with the cosine measure. The *Genre-based Fuzzy Inference Filtering method* performed slightly better than the *Fuzzy Inference Filtering method*.

Observation 3: Finally, we compare the *Fuzzy Inference* and *G-Fuzzy Inference Filtering* methods. Figure 13 shows that the *G-Fuzzy Inference Filtering* method achieved a better precision performance than the *Fuzzy Inference* method under various α -cut levels. The *Fuzzy Inference* method achieved the best performance under various training and testing splits when we set α at 0.4. The results of the *G-Fuzzy Inference Filtering* method are similar, regardless of the α -cut level. We conclude that the cosine similarity-based *G-Fuzzy Inference Filtering* method yields a better recommendation performance and more stable results than the *Fuzzy Inference* method under all parameter settings.

Table 7

Precision rates of the two hybrid filtering methods under different training/test dataset splits and different α degrees settings

	$\alpha \geq 0.4$			$\alpha \geq 0.5$			$\alpha \geq 0.6$		
	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150	≥ 350	≥ 250	≥ 150
Genre-based fuzzy inference (Hybrid 1): The average (10-runs)									
60/40	0.9367	0.9736	0.4529	0.9133	0.9717	0.4548	0.9278	0.9675	0.4518
70/30	0.9685	0.9704	0.4858	0.9625	0.9754	0.4985	0.9288	0.9650	0.5011
80/20	0.9309	0.9722	0.4998	0.9567	0.9732	0.5032	0.9263	0.9646	0.5024
90/10	0.9236	0.9553	0.4932	0.9401	0.9595	0.4923	0.9233	0.9545	0.4978
Genre-based fuzzy inference (Hybrid 2): The average (10-runs)									
60/40	0.9931	0.9890	0.5321	0.9913	0.9918	0.5346	0.9913	0.9918	0.5320
70/30	0.9938	0.9914	0.5374	0.9932	0.9905	0.5372	0.9913	0.9911	0.5386
80/20	0.9915	0.9900	0.5363	0.9916	0.9900	0.5378	0.9910	0.9891	0.5387
90/10	0.9882	0.9797	0.5273	0.9887	0.9833	0.5282	0.9911	0.9830	0.5297

Fig. 12. Comparison of the average performance of the two hybrid methods (cosine similarity) under different α degrees settings. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)Fig. 13. Comparison of the two hybrid methods in terms of precision (cosine similarity). (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

6.2.3. Discussion

In Fig. 14, we compare the precision of the correlation-based and cosine similarity measures for the two hybrid methods. Clearly, the *Genre-based Fuzzy Inference* method outperforms the

Table 8
The differences between the compared methods

Information filtering categories	Main advantages	Main shortcomings	Usage timing
Genre-based	<ul style="list-style-type: none"> - Recommendation performance is not sensitive to the sparseness rates - Shortest computation time 	<ul style="list-style-type: none"> - Yields the worst recommendation performance in terms of precision 	Interactive recommendation
CF	<ul style="list-style-type: none"> - Reasonable recommendation results 	<ul style="list-style-type: none"> - Requires the longest computation time in the prediction phase - Recommendation performance is sensitive to the number of users 	Batch processing
Fuzzy Inference (correlation-based or cosine similarity measure)	<ul style="list-style-type: none"> - Reasonable recommendation results - α-cut can reduce the computation time required to identify similar users 	<ul style="list-style-type: none"> - Recommendation performance is sensitive to the sparseness rates 	Interactive recommendation
G-Fuzzy Inference (correlation-based or cosine similarity measure)	<ul style="list-style-type: none"> - Achieves the best performance in terms of precision - α-cut can reduce the computation time required to identify similar users 		

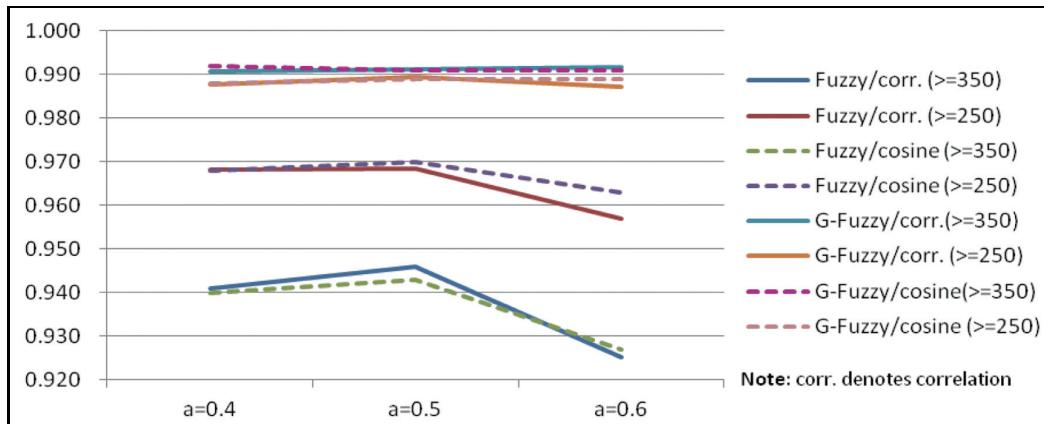


Fig. 14. Comparsion of the precision of the correlation-based and cosine similarity measures in the two hybrid methods. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

Fuzzy Inference Filtering method on the two sampled data sets whose sparseness rates are 78.69%, and 72.38% respectively. The sparseness rate affects the performance of the *Fuzzy Inference Filtering* method; however, the *Genre-based Fuzzy Inference Filtering* method's performance is not influenced by the sparseness rate. Moreover, the results of the *Fuzzy Inference Filtering* method vary under different α -cut levels, but the *Genre-based Fuzzy Inference Filtering* method's results are similar under different α -cut levels. Overall, the *Genre-based Fuzzy Inference Filtering* method yields a better recommendation performance and more stable results under both similarity measures.

6.3. Overall discussions among methods

In Fig. 15, we compare the performance of the recommendation methods under two sparseness rates: 78.69%, and 72.38%, i.e., $RNs \geq 250$ or $RNs \geq 350$ respectively. The proposed *G-Fuzzy Inference*

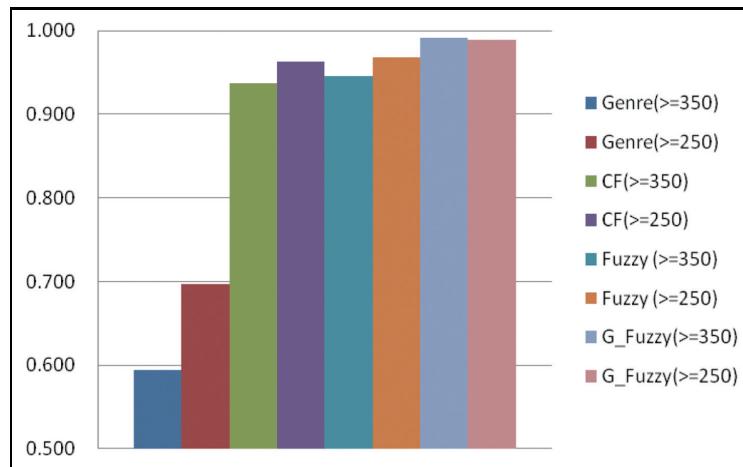


Fig. 15. The precision rates of the compared methods. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

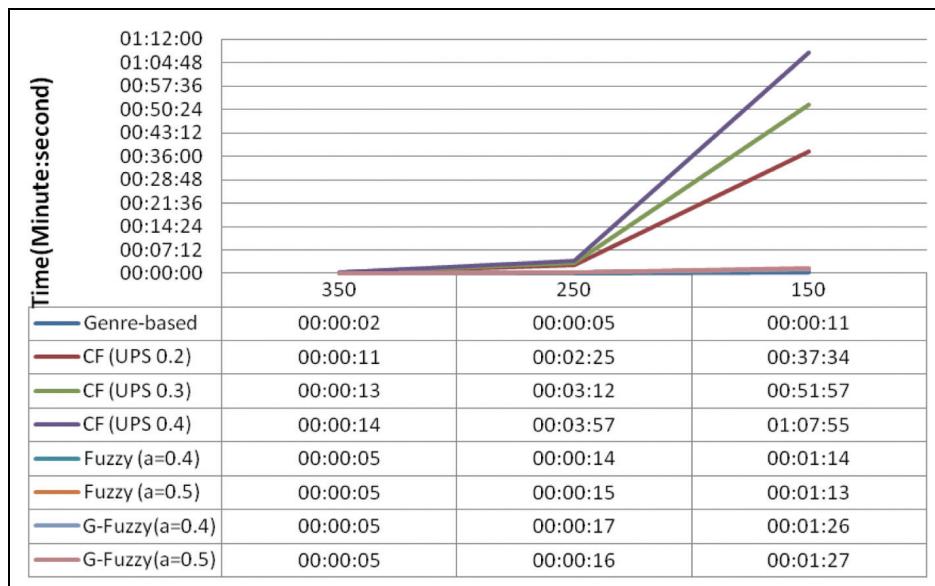


Fig. 16. The time complexity of the compared methods. (Colours are visible in the online version of the article; <http://dx.doi.org/10.3233/IDA-130622>)

Filtering method outperforms the other methods in terms of precision. Moreover, the *Fuzzy Inference Filtering* and *CF* method achieve similar results. In contrast, the *Genre-based Filtering* method yields the worst performance. Basically, the genre-based fuzzy inference approach is a model-based framework that including training and testing phases to build and validate the model. For the training phase, the method infers the implicit and transitive relationships among users' preferences by the transitive max-min closure method [15]. It then identifies appropriate neighborhoods, i.e., group users, by applying an α -cut operation. For the prediction phase, the method only needs to compute the most relevant cluster and then select the scores of the required genres from the target cluster of the training sets to obtain the prediction ratings. The computation complexity of the training phase and prediction phase is $O(n^2)$ and

$O(c)$ respectively, where n denotes the number of users and c represents the number of cluster. However, for the prediction phase in the collaborative filtering (CF) method, the time complexity is approximately $O(mn)$, where n denotes the number of users and m is the number of items. Thus, compared to the other approaches, the CF method requires a significant amount of time to make predictions, as shown in Fig. 16. The presented hybrid approach, i.e., *G-Fuzzy Inference*, outperformed the other methods in terms of precision and computational efficiency. With regard to the density of the data, the hybrid methods achieve the best performance on datasets with 78.69% and 72.38% sparseness rates, i.e., the RNs are higher than 250 and 350 respectively. The experiment results confirm that it is practical to integrate users' preferences for different genres of movies into the collaborative filtering process. In addition, the simple α -cut grouping method reduces the computation time required to select neighborhoods without reducing the precision of the recommendations. Table 8 summarizes the advantages, disadvantages, and usage timing (context) for make effective recommendation.

7. Conclusion

Recommendation techniques are used in electronic commerce to predict users' preferences based on the users' ratings for items on websites. Then, items that may be of interest are recommended to the target user. In this work, we propose a genre-based fuzzy inference filtering approach that exploits the strengths of content-based and collaborative filtering techniques to predict users' movie preferences. The approach classifies users' ratings for movies into genres to mitigate the sparse ratings problem and make accurate predictions. It also uses (1) transitive closure to infer the null value of the similarity matrix to further mitigate the sparse matrix problem; and (2) the α -cut approach to group users in order to reduce the computation time required to select neighborhoods. The results of experiments on a dataset with different sparseness rates show that the proposed approach exploits the strengths of content-based and collaborative filtering to achieve a better performance than the CF and content-based filtering methods in terms of precision. In addition, the α -cut approach reduces the computation time. Note that an effective recommendation system must be able to (1) collect users' preferences for items explicitly, and (2) determine users' search behavior patterns implicitly. Thus, the higher the number of ratings collected from users, the better will be the performance of the system. The proposed method can be applied to the on-line product-based recommendation, e.g., movies and books, as well as text-based recommendation, e.g., review articles, and webpages. We will also seek data sets for use in extensive experiments to investigate the effectiveness of the proposed method.

The experimental results confirm that the proposed *G-based Fuzzy Inference Filtering* method can be integrated with an interactive movie recommendation system for use in a dynamic Web context. Some recent studies have shown that an individual's preferences are correlated with his/her personality and prior knowledge [13,26]. Hu and Pu [13] proposed a personality-based system and demonstrated that users prefer to use a quiz-based recommendation system rather than a traditional ratings-based recommendation system. For example, Whattorent.com is a service that provides recommendations based on users' personality traits and moods. The user's input can also be used to fine tune the recommendation results. For example, MovieGEN [8] an expert system for making movie recommendations, utilizes a well-trained support vector machine (SVM) to predict the genres and the period of the movies that the user prefers based on his/her personal information. The system can ask the user questions to further refine the recommendation results. In a recent work, we proposed an anchor-based hybrid filtering approach that makes interactive movie recommendations. We utilized the proposed G-fuzzy inference filtering method in the back-end of the system. For the front-end, we introduced an interactive anchor-based user

preference elicitation method that determines a user's preference for a specific genre of movies via a series of quizzes [33]. In addition, Ma et al. [20] showed that integrating social context information into the filtering and recommendation process is effective. In a future work, we will also investigate whether additional social context information can yield better fuzzy inference results and facilitate more effective recommendations.

Acknowledgments

This research was supported by the National Science Council of Taiwan Government and Fu-Jen Catholic University under the grant NSC99-2815-C-030-013-H and No. 409931074078 respectively.

References

- [1] 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* **17**(6) (2005), 734–749. doi: 10.1109/TKDE.2005.99.
- [2] M. Balabanović and Y. Shoham, Fab: content-based, collaborative recommendation, *Communications of the ACM* **40**(3) (1997), 66–72. doi: 10.1145/245108.245124.
- [3] Y. Blanco-Fernández, M. López-Nores, A. Gil-Solla, M. Ramos-Cabrera and J.J. Pazos-Arias, Exploring synergies between content-based filtering and spreading activation techniques in knowledge-based recommender systems, *Information Sciences* **181**(21) (2011), 4823–4846. doi: 10.1016/j.ins.2011.06.016.
- [4] R. Burke, Hybrid recommender systems: Survey and experiments, *User Modeling and User-Adapted Interaction* **12**(4) (2002), 331–370. doi: 10.1023/A:1021240730564.
- [5] G. Castellano, A.M. Fanelli and M.A. Torsello, NEWER: A system for Neuro-fuzzy web recommendation, *Applied Soft Computing* **11**(1) (2011), 793–806. doi: 10.1016/j.asoc.2009.12.040.
- [6] K.W. Cheung, J.T. Kwok, M.H. Law and K.C. Tsui, Mining customer product ratings for personalized marketing, *Decision Support Systems* **35**(2) (2003), 231–243. doi: 10.1016/j.asoc.2009.12.040.
- [7] B.J. Dahlen, J.A. Konstan, J.L. Herlocker, N. Good, A. Borchers and J. Riedl, Jump-starting MovieLens: User benefits of starting a collaborative filtering system with “dead data”, University of Minnesota, Technical Report, 1998, pp. 98–017.
- [8] A.E. Eyrun, T. Gaurangi and L. Nan, Moviegen: A movie recommendation system, Hewlett-Packard, August 2008.
- [9] V. Formoso, F. Cacheda and V. Carneiro, Algorithms for efficient collaborative filtering, in: *Proceedings of the Efficiency Issues in Information Retrieval Workshop in European Conference of Information Retrieval*, Glasgow, UK, (2008), 17–28.
- [10] D. Goldberg, D. Nichols, B.M. Oki and D. Terry, Using collaborative filtering to weave an information tapestry, *Communications of the ACM* **35**(12) (1992), 61–70. doi: 10.1145/138859.138867.
- [11] D. Gupta, M. Digiovanni, H. Narita and K. Goldberg, Jester 2.0: A new linear-time collaborative filtering algorithm applied to jokes, in: *Proc ACMSIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA, (1999).
- [12] J.L. Herlocker, J.A. Konstan and J. Riedl, An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, *Information Retrieval* **5**(4) (2002), 287–310. doi: 10.1023/A:1020443909834.
- [13] R. Hu and P. Pu, A comparative user study on rating vs. personality quiz based preference elicitation methods, in: *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*, Sanibel Island, FL, USA, (2009), 367–372. doi: 10.1145/1502650.1502702.
- [14] K.J. Kim and S.B. Cho, Personalized mining of web documents using link structures and fuzzy concept networks, source, *Applied Soft Computing* **7**(1) (2005), 398–410. doi: 10.1016/j.asoc.2005.09.002.
- [15] G.J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: Theory and applications*, U.S. River ed., NJ: Prentice Hall, 1995.
- [16] J. Konstan, B. Miller, D. Maltz, J.L. Herlocker, L. Gordon and J. Riedl, GroupLens: Applying collaborative filtering to usenet news, *Communications of the ACM* **40**(3) (1997), 77–87. doi: 10.1145/245108.245126.
- [17] Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, *ACM Transactions on Knowledge Discovery from Data* **4**(1) (2010), 1–24. doi: 10.1145/1644873.1644874.
- [18] G. Lekakos and P. Caravelas, A hybrid approach for movie recommendation, *Multimedia Tools and Applications* **36**(1–2) (2008), 55–70. doi: 10.1007/s11042-006-0082-7.
- [19] Q. Li and B.M. Kim, Clustering approach for hybrid recommender system, in: *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, Beijing, China, (2003), 33–38.

- [20] H. Ma, T.C. Zhou, M.R. Lyu and I. King, Improving recommender systems by incorporating social contextual information, *ACM Transaction on Information System* **29**(2) (2011), 9–31. doi: 10.1145/1961209.1961212.
- [21] P. Melville, R.J. Mooney and R. Nagarajan, Content-boosted collaborative filtering, in: *ACM SIGIR Workshop on Recommender Systems*, 2001.
- [22] M.P. O’Mahony, N.J. Hurley and G.C.M. Silvestre, Efficient and secure collaborative filtering through intelligent neighbor selection, in: *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI’04)*, Valencia, Spain, (2004), 383–387. doi: 10.1023/B:AIRE.0000036256.39422.25.
- [23] T. Mitchell, *Machine learning*, New York: MacGraw-Hill, 1997.
- [24] M. Pazzani and D. Billsus, Learning and revising user profiles: The identification of interesting web sites, *Machine Learning* **27** (1997), 313–331. doi: 10.1023/A:1007369909943.
- [25] P. Renick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, in: *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, 1994, 175–186.
- [26] P.J. Rentfrow and S.D. Gosling, The do re mi’s of everyday life: The structure and personality correlates of music preferences, *Journal of Personality and Social Psychology* **84** (2003), 1236–1256.
- [27] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, in: *Proceedings of 10th International Conference on World Wide Web (WWW’01)*, Hong Kong, (2001), 285–295. doi: 10.1145/371920.372071.
- [28] J.B. Schafer, J.A. Konstan and J. Riedl, E-Commerce recommendation applications, *Journal of Data Mining and Knowledge Discovery* **5**(1–2) (2001), 115–153. doi: 10.1023/A:1009804230409.
- [29] U. Shardanand and P. Maes, Social information filtering: Algorithms for automating “Word of Mouth”, in: *Proc of ACM Conf on Human Factors in Computing Systems (CHI’95)*, (1995), 210–217. doi: 10.1145/223904.223931.
- [30] X. Su and T.M. Khoshgoftaar, A survey of collaborative filtering technique, *Advances in Artificial Intelligence* (2009), 1–20. doi: 10.1155/2009/421425.
- [31] S. Vucetic and Z. Obradovic, A regression-based approach for scaling-up personalized recommender systems in e-commerce, in: *ACM WebKDD Workshop on Web Mining for Ecommerce Workshop*, 2000.
- [32] R.W. White, P. Bailey and L. Chen, Predicting user interests from contextual information, in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM SIGIR 09’)*, Boston, Massachusetts, USA, 2009, 363–370.
- [33] I.C. Wu, W.H. Hwang, K.Y. King and M.Y. Lin, An interactive quiz-based approach for inferring users’ movie preference, *International Symposium on the Transformation and Innovation of Library and Information Science*, 2010.
- [34] L. Yu, L. Liu and X. Li, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce, *Expert Systems with Applications* **28**(1) (2005), 67–77. doi: 10.1016/j.eswa.2004.08.013.

Copyright of Intelligent Data Analysis is the property of IOS Press and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.