Fuzzy-Bayesian Network Approach to Genre-based Recommender Systems

Soheila Ashkezari-T.[1] and Mohammad-R. Akbarzadeh-T.[2], *Senior Member*, *IEEE*

Abstract—**The World Wide Web has created a new media for mass marketing that can also be highly customized to online customers' needs and expectations. Recommender Systems (RS) play an important role in this area. Here, we aim to establish a genre-based collaborative RS to automatically suggest and rank a list of appropriate items (movies) to a user based on the user profile and the past voting patterns of other users with similar tastes. The contribution of this paper is using genre based information in a hybrid fuzzy-Bayesian network collaborative RS. The interest to the different genres is computed based on a hybrid user model. The similarity of like-minded users according to the fuzzy distance and also Pearson correlation coefficient is involved in a Bayesian network.**

I. INTRODUCTION

OVER the past few years, electronic commerce has grown quickly and an enormous amount of information is now available. Most of this information is, however, not accessible to the users, because either they are unaware that it exists or exploring the entire web is too time consuming for them. This explosive growth of data has generated an urgent need for powerful automated web personalization tools that can intelligently deliver the right information to the right people at the right time. In this framework, recommender systems (RS) have emerged to help people deal with this information overload. They are implemented to analyze users' data and extract useful information for prediction of the user preferences. Examples of such applications include recommending books, CDs and other products at Amazon.com [1], movies by MovieLens [2], books at LIBRA [3], electronic television program guides [4],[5].

"Recommender systems use four information filtering techniques that consist of demographic filtering (DMF) [6], content-based filtering (CBF) [7], collaborative filtering (CF) [8],[9], and hybrid filtering techniques [10]" [12]. DMF classifies the user based on the user personal attributes. In this way, the user is recommended items similar to the ones other people with same demographic features prefer. In CBF, items similar to the ones which user has preferred in the past are suggested. CBF makes recommendations just based on the user profile and items' content and does not rely on information provided by other customers. In CF, after finding similar users, the user is recommended items that other people with similar tastes liked in the past. The most common method is the nearest-neighbourhood [11]. Hybrid filtering techniques combine more than one filtering technique to enhance performance.

Basically, RS keeps a profile for each user which contains raw information about the user's preferences, background, personal details, and interactions with the system. Many of these features can be described as fuzzy sets. So it is reasonable to use fuzzy measures to find like-minded users. On the other hand, a Bayesian network is a graphical structure that allows us to represent and reason about an uncertain domain. Employing Bayesian network enables us to intuitively represent users' effect on each other and make a probabilistic model. In this paper, we develop a hybrid Fuzzy-Bayesian network RS to suggest appropriate movies to the users.

The user model implemented in our work is the one introduced in [12] which is a hybridization at four different levels, i.e. feature level, model-level, CF algorithm-level, and approach-level. The first level is composed of hybrid features that exploit both user ratings for highly rated items and content descriptions of the items. At the model-level, a user model is built from the set of hybrid features and DMF profile. At the CF algorithm-level, hybridization between model-based and memory-based algorithms of CF is done. The user model then finds users with similar patterns to perform a memory based search.

The objective of this paper is to predict and automatically rank movies to the users. For predicting a vote, in contrast to most of the recent works in this area, we did not use explicit ratings of like-minded users. Instead, we concentrate on determining a vote based on level of interestingness of similar users to the movie's genres. The rest of this paper is organized as follows: a review of related works in this area is discussed in Section 2. Some background on the RS is given in Section 3. In Section 4, the properties of user model at first introduced in [12] are discussed, while the fuzzy and hybrid fuzzy-Bayesian network approaches are given in Sections 5 and 6 respectively. The experimental results of

S. Ashkezari is with the Department of Computer Engineering, Cognitive Computing Lab, Center for Applied Research on Soft computing and Intelligent systems, Ferdowsi University of Mashhad, Mashhad, 91775-1111, Iran(e-mail: Soheila.Ashkezari@stu-mil.um.ac.ir).

M.-R. Akbarzadeh-T. is with the Departments of Electrical Engineering and Computer Engineering, Cognitive Computing Laboratory, Center for Applied Research on Soft computing and Intelligent systems, Ferdowsi University of Mashhad, 91775-1111, Iran (e-mail: akbarzadeh@ieee.org).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

the proposed approaches are discussed in Section 7. Finally in the last section, we conclude our work with a review of our contributions along with future research directions.

II. Related Works

In [12] a hybrid fuzzy-genetic RS is developed by employing GA to evolve appropriate weights for each feature of the user model. They built a hybrid user model based on CF, CBF and DMF filtering techniques. In our work we will use this compact user model. Also they proposed a novel fuzzy distance metric to match users that we will implement it. In [13] a soft computing-based collaborative recommender system is proposed which combined Bayesian networks and fuzzy set theory in order to model the uncertainties and the tolerance of imprecision related to the recommending process. They modelled the relationship between users and also the relationship between users and items by means of BNs. In our work instead of modelling the relationship between users and items, in a higher level of existing knowledge, we will model the relationship between users and movies related genres. The details of this modelling is discussed in Section 6 and by considering the results, its success is shown in Section 7. In [5] a personalized TV program recommendation system is introduced. They propose a hybrid approach, combining content-filtering techniques and collaborative filtering and used Singular Value Decomposition (SVD) to reduce the dimensionality and avoid sparsity problems of recommender system database. In [14] a new model-based CF approach, which is based on latent semantic indexing (LSI), is proposed to produce a condensed model for the user-item matrix that handles the scalability. In [15] a feature profile is constructed for the users to reveal their favourite features. Moreover, they grouped users into biclusters (i.e. groups of users which exhibit highly correlated ratings on groups of items) to exploit partial matching between the preferences of the target user and each group of users. In [16] a probabilistic generative model is used that unifies the collaborative and content-based data in a principled way. This model can explain the generative mechanism of the observed data in the probability theory. In [17] a hybrid K-means clustering and genetic algorithms is applied to carry out an exploratory segmentation of an online shopping market. To find the most effective clustering method, they adopted a number of clustering methods and compared the performance of each clustering method by using their suggested performance criteria.

III. Background

A. Recommender Systems

Formally, in CF recommenders, there is a set of users U = {u1,u2, . . . ,um} rating a set of items $S = \{s_1, s_2, \ldots, s_n\}$, such as books, movies, or CDs. The spaces *S* and *U* can be very large in some applications. Each user u_i,i=1,...,m has rated a subset of items S_j. The rating of user u_i for item S_j ,j=1,...,n is denoted by $r_{i,j}$. Explicit ratings from users follow a specified numerical scale indicating the degree of preference (e.g., 1–bad to 5–excellent. Four phases are required to perform the recommendation task in CF recommenders [12]:

(a) Data collection
(b) User model formation
(c) Neighbourhood set selection
(d) Making recommendations

1) Data Collection: There are three categories of data that can be collected about users. These categories consist of demographic data that is collected during registration, explicit rating about the items and implicit data about user behaviour.

In order to manage our experiments, we used the original *MovieLens*[3] dataset. The data was collected through the *MovieLens* web site during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up, i.e. users who had less than 20 ratings or did not have complete demographic information were removed from this data set. The dataset consists of 100,000 ratings, assigned by 943 users on 1682 movies. All ratings follow the 1–bad, 2–average, 3–good, 4–very good and 5–excellent numerical scales. Simple demographical data such as age, gender, occupation and zip code are included for all users, which are collected when a new user registers on the system. The movie title, release date, video release date, and genre data are given for each movie. The genre feature specifies if the movie is an action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, or western. A single movie can belong to more than one genre.

2) User Model Formation: A typical user model would properly reflect user's tastes, preferences, and needs. In real life, finding similar-minded people depends on some other factors in addition to their close opinions on a specific subject such as movie ratings. Features such as their background and personal details such as age, gender, and preferences of movie genres should be considered in addition to explicit ratings [12].

3) Neighbourhood Set Selection: After establishing the user model, a set of neighbours based on a suitable distance function will be formed for him. There are two methods for setting the neighbourhood set size. It could be considered fixed by selecting the top K users or variable by selecting the users with a higher similarity value of a certain threshold. There are different functions to compute the distance, d (x, y), between users u_x and u_y in CF recommenders. Pearson correlation coefficient is the most popular function for memory-based CF [8], where the

[3] http://www.movielens.umn.edu

distance of two users is computed based on the ratings of the items which both of them have seen. The Pearson correlation coefficient is given by

$$\mathrm{corr}(x,y)=\frac{\sum_{s\in S_{xy}}(r_{x,y}-m_x)(r_{y,s}-m_y)}{\sqrt{\sum_{s\in S_{xy}}(r_{x,y}-m_x)^2(r_{y,s}-m_y)^2}} \quad (1)$$

Where S_{xy} is the set of items rated by both users u_x and u_y , and m_x, m_y are the mean vote of users x and y. We name the RS, which uses (1) for similarity computations as Pearson RS (PRS).

Since (1), just consist of common items for both users, it is not appropriate if other mentioned features are also comprised in the model. So, another way to compute similarity is the modified Euclidean distance function [12], which takes into account multiple features:

$$d(\mathbf{x},\mathbf{y})=\frac{1}{z}\sum_{i=1}^{z}\sqrt{\sum_{j=1}^{N}(x_{i,j}-y_{i,j})^2} \quad (2)$$

Here $x_{i,j}$ is the jth feature for the common item S_i, N is the number of features, and $Z=|S_{xy}|$ is the cardinality of S_{xy} . Note that a vector of features represents each user and it is written bold in (2).

4) Making Recommendations: In this phase, RS assigns a predicted rating to all of the new items for u_x seen by their neighbourhood set. The predicted rating, $pr_{i,j}$, indicates the expected utility of the item S_j for the user u_x, and is usually computed as an aggregate of the ratings of user's (u_x) neighborhood set for the same item S_j [12].

$$pr_{x,j}=m_x+k\sum_{u_c\in C}d(x,c)\times(r_{c,j}-m_c) \quad (3)$$

where C denotes the set of neighbors who have rated item S_j . The multiplier k is a normalizing factor and is usually selected as $k=1\Big/\sum_{u_c\in C}|d(a,c)|$, m_c is the average rating of user u_c.

$$m_c=\frac{1}{|S_c|}\sum_{s\in S_c}r_{c,S} \quad (4)$$

In (3), a vote is produced based on the set of user u_x 's neighbours vote for the item S_j . In Section, 6 we will propose our method to make a prediction base on the knowledge we obtain from the users interested genres in Bayesian network, i.e. for making a recommendation there is no need to compute the difference between each neighbour's vote on the item j and the average of his vote (Equation 3), instead we just concentrate on the interest of the neighbours of u_x on the genres which the item (movie) j is belongs them and we will use BN to build a probabilistic model according to this idea. We will compare performance of these methods in Section 7.

IV. A Hybrid User Model[12]

In the user model introduced in [12] only one online filtering process (CF) is needed. The other offline filtering techniques (CBF and DMF) are used to combine the data by building a compact user model. Further, to remove the scalability problem caused by a sparse user-item matrix, many information sources are integrated. They build a set of hybrid features that combine some of the users' and items' properties based on the user's ratings for a set of high rated movies and the content descriptions of the genres corresponding to this set of movies. The hybrid features are utilized as the basis for formulating a genre interestingness measure (GIM). After making an appropriate formula for GIM, a user model will be constructed from DMF user profile and GIMs. Block diagram of the proposed work in [12] is given in Fig. 1. After building a hybrid user model by the hybrid features and DMF user profile, CF recommender generates a neighbourhood set according to model-based CF. Finally the entire database of this set is retrieved to recommend items according to memory-based CF.

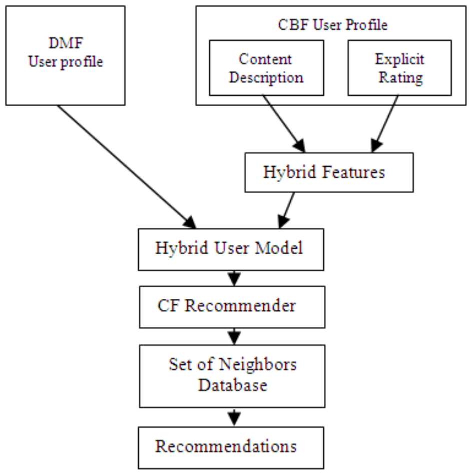

Fig. 1. Block diagram of the work proposed in[12]

A. Necessary Equations

The total ratings (TR) of user u_i is

$$TR(i)=\sum_{s\in S_c}r_{i,S} \quad (5)$$

Here s_i is the set of movies rated by user u_i. The genre rating (GR) (resp. genre frequency (GF)) for high rated items of genre G_j corresponding to user u_i is computed using (6) (resp. (7))

$$GR(i,j)=\sum_{s\in G_j\subset S_i,r\geq 3}r_{c,S} \quad (6)$$

$$GF(i,j)=\sum_{s\in G_j\subset S_i,r}\delta_k(r_{i,s}) \quad k\in\{3,4,5\} \quad (7)$$

$$\delta_k(r_{i,s})=\begin{cases}1 & \mathrm{k}=\mathrm{r}_{\mathrm{i,s}}\\ 0 & \mathrm{k}\neq\mathrm{r}_{\mathrm{i,s}}\end{cases} \quad (8)$$

It is to be noted that only the items rated as good (3), very good (4), or excellent (5) are considered. Such items called high rated. Finally, relative genre rating (RGR) (resp.

relative genre frequency (RGF)), the ratio of u_i's ratings (resp. frequency) for high rated items of G_j to his total ratings (resp. frequency), is computed as

$$RGR(i,j) = \frac{GR(i,j)}{TR(i)} \tag{9}$$

$$RGF(i,j) = \frac{GF(i,j)}{TF(i)} \tag{10}$$

Where $TF(i) = |S_i|$, the cardinality of s_i [12].

B. *The Hybrid User Model*

In RGF Equation all high ratings have the same weight, so exact degree of priority is not captured. The following definition captured from [12] introduces a modified version of RGF Equation, which tries to reflect the exact preference for items with high ratings.

Definition 1. For a rating-based movie recommender system, the modified relative genre frequency (MRGF) of genre G_j for user u_i is defined as

$$MRGF(i,j) = \frac{\sum_{s \in G_j \subset S_{i,r}} \delta_3(r_{i,s}) + 2 \times \delta_4(r_{i,s}) + \delta_5(r_{i,s})}{3 \times TF(i)} \tag{11}$$

To develop more accurate GIM, the relative genre rating needs to be considered also. Indeed, RGR identifies the preferences for genres with some drawbacks. However, a combined Equation of RGR and MRGF will bring out the best in both Equations. The following definition gives one possible form of such a Equation [12].

Definition 2. For a rating-based movie recommender system, the genre interestingness measure (GIM) of genre G_j for user u_i is defined as

$$GIM(i,j) = \frac{2 \times nf \times RGR(i,j) \times MRGF(i,j)}{RGR(i,j) + MRGF(i,j)} \tag{12}$$

Here *nf* is the normalization factor for a given system. Equation (12) gives the harmonic mean of *RGR(i,j)*, and *MRGF(i,j)* multiplied by *nf* . The range of *GIM(i,j)* is [0, MAX] that agrees with the system's rating structure, i.e. 1 – bad . . . MAX – excellent. The normalization factor *nf* could take the value of MAX. The advantages of GIM Equation are discussed in [12]. Actually GIM represents user i interestingness for genre *j*. However, as noted earlier that two people are said to be similar is not based solely on whether they have close opinions on a specific subject, but also on other factors, such as their background and personal details. Therefore, we construct the user model from DMF user profile and GIMs. The proposed user [12] model consists of age and gender as demographical data and GIMs for 18 genres.

V. Fuzzy Approach to Recommender System [12]

A. *User Model Fuzzification*

The crisp description of the age and genre interestingness measure does not reflect the actual case for human decisions. So, first of all age is fuzzified into three fuzzy sets, young, middle-aged and old as shown in Fig. 2.

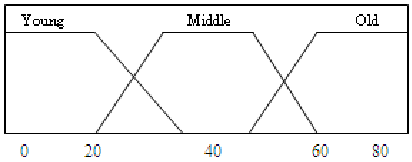

Fig. 2. Membership functions for age [12]

The gender value is considered as fuzzy points with membership value of one. Finally, the genre interestingness measure is fuzzified into six fuzzy sets, very bad (VB), bad (B), average (AV), good (G), very good (VG), and excellent (E) as shown in Fig. 3.

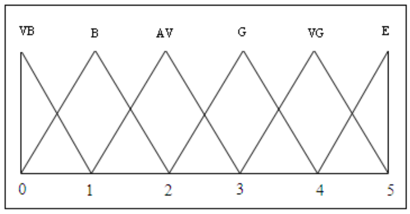

Fig. 3. Membership functions for Genre Interesting Measure [12]

B. *Fuzzy Distance Function*

Definition 3. Let x_i and y_i be the membership vectors correspond to two crisp values, x_i and y_i for a given feature with l fuzzy sets. The fuzzy distance [12] between x and y is defined as

$$fd(x_i, y_i) = d(\mathbf{x}_i, \mathbf{y}_i) \times d(x_i, y_i) \tag{15}$$

Where *d(x_i, y_i)* is simply the difference operator, *x* and *y* are vectors of size l, and $d(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is any vector distance metric. In our experiments, the Euclidean distance function is used for $d(\boldsymbol{x}_i, \boldsymbol{y}_i)$:

$$d(\mathbf{x}_i, \mathbf{y}_i) = \sqrt{\sum_{j=1}^{l} (x_{i,j} - y_{i,j})^2} \tag{16}$$

Where l is the total number of fuzzy sets for the ith feature, and $x_{i,j}$ is the membership value of the ith feature in the jth fuzzy set. The global fuzzy distance is computed by the aggregation operator. The aggregation operator may be the average of the 20 local fuzzy distances [12]:

$$fd(\mathbf{x}, \mathbf{y}) = \frac{1}{20} \sum_{i=1}^{20} fd(x_i, y_i) \tag{17}$$

This included 18 different movie genres, age and gender. We call the RS uses (17) for similarity computations as Fuzzy RS (FRS).

VI. BN-based Collaborative RS

When we are interested in representing our knowledge by means of BNs, the first task is to select those variables which are relevant to the problem we are tackling. In our case, we want to model the relation between users' vote (***U*** = *{u_i, i=1,...,943}*) and their preferred genres (***G****={G_j, j=1,...,18}*). So, we must model both the relation $\mathbf{G} \to \mathbf{U}$ by modelling the database of users' favourite genres for the set of observed items and also the relation $\mathbf{U} \to \mathbf{U}$ by modelling the relationships between users. Therefore, at the preliminary stage, we consider the set of genres ***G*** and the set of users ***U*** as variables in the BN (nodes in the graph). Our first objective is to model the user's preferred genres pattern which represents the dependence relationships between genres, ***G***, and user votes, ***U***. At the beginning we include an arc from each genre G_j, to each user u_i. The strength of each connection in this level (genres layer) depends on the quality of genre's conformation by the user. To understand this quality we will use GIM concept introduced in Section 4.2 and also fuzzy distance between users according to Section 5.2.

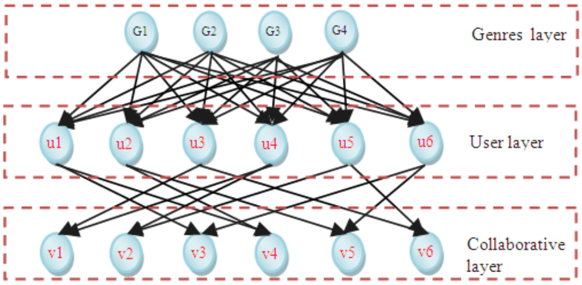

Fig. 4. **BN**-Based Collaborative Recommender System Topology

In a collaborative RS, the prediction of the vote for a user also depends on the votes of people with similar tastes or preferences. Our model therefore might be able to represent relations between users, $\mathbf{U} \to \mathbf{U}$. Regardless of the mechanism used to find these relationships; they should be modelled in the BN by the inclusion of arcs between any two related users. Thus, whenever a dependence (similarity) between the preferences of user u_a and user u_b has been found, an arc connecting both nodes should be included in the BN. However, taking into account that similarities between users' tastes tend to be symmetric (when u_a is highly related with u_b, it is also common for u_b to be related with u_a), a cycle could be included in the BN, which is forbidden in a BN topology. In order to facilitate the presence of these relationships in the model, we therefore use a new set of nodes ***V*** be included to denote collaborative votes [13]. There is one collaborative node for each user in the system, i.e. ***V****={v_i,i=1,..,943}*. Following the performance of a collaborative RS, these nodes are also used to predict the vote that the user could give to an unseen item. The topology of this network is shown in Fig.4 that is assumed there are 4 genres and 6 users in our database and a constant neighbour size 2 is considered to select users who are more like-minded.

In Section 3.1.4 we mentioned about how should predict a vote for an unseen item and we said we also will try another Equation. Here, because we focus on genres, for predicting a vote to recommend to user v_i we should consider that the desired item (unseen movie) is related to which genres. Then we compute the probability of giving a high rate vote to these genres by the user v_i neighbours using (18). Then we make a recommendation for this unseen item according to its genres according to (22):

$$P(v_i \geq 3 \mid G_j) = \frac{\sum_{k=1}^{NB} (u_k \geq 3 \mid G_j) \times fd(v_i, u_k)}{\sum_{k=1}^{NB} fd(v_i, u_k)} \quad (18)$$

$$P(u_k \geq 3 \mid G_j) = \frac{PRG(u_k \geq 3, G_j)}{PR3(u_k \geq 3)} \quad (19)$$

$$PRG(u_k \geq 3, G_j) = \frac{GF(k,j)}{\sum_{j=1}^{GC} GF(k,j)} \quad (20)$$

$$PR3(u_k \geq 3) = \frac{N(u_k \geq 3)}{TF(k)} \quad (21)$$

$$\Pr_{i,j} = \frac{\sum_{j=1}^{GC} P(v_i \geq 3 \mid G_j) \times MV(v_{i,j})}{\sum_{j=1}^{G} P(v_i \geq 3 \mid G_j)} \quad (22)$$

Where *NB* is the cardinality of neighbours set of v_i, *GC* is the number of related genres to the unseen item and *MV* is the mean of user v_i high rated votes ($v_i \geq 3$)for genre G_j.

After computing the value of *pr* by (22), we should identify the proper label for it (1-bad, …, 5-Excelent). First we fuzzify *pr* to understand its dependency to the fuzzy labels and then select the label corresponding to the largest membership value of the fuzzy sets (Fig. 3).

$$Finalvote = \arg\max_{l} \{\mu_l(pr_{i,j})\} \quad (23)$$

We call the RS uses (23) as Bayesian network RS (BNRS). If we use Pearson for similarity computations named it PBNRS and using fuzzy distance for similarity computation called it FBNRS.

The difference between this method (22) and the Equation in Section 3.1.4 (3) is that in the proposed BN-based RS, we just concentrate on genres without pay attention to the vote of the user's neighbours for the new item.

VII. Experiments

The dataset is divided into 80% training and 20% testing subsets. The process is repeated 5 times for a 5 fold cross validation. For each user a neighbour of size 30 is considered who are the most similar ones.

To evaluate the effectiveness of different RS, two type of evaluation metrics are used, the mean absolute error (MAE) as predictive accuracy metric[18], and precision, recall and relevant measure as classification accuracy metrics[18]. Also we compute the total coverage of the system and the percent of correct predictions in each spilt.

A. Predictive Accuracy Metrics

The MAE measures the divergence of predictions generated by the RS from the true ratings specified by the user. The MAE is given by the following Equation:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | pr_i - r_i | \tag{24}$$

MAE is able to describe the accuracy of predictions, the lower MAE, the more accurate RS predictions is. But it is not demonstrative for the accuracy of the recommendations i.e. since our goal is to find good items this metric may be less appropriate because in this kind of RS users may only care about errors in items that are ranked high [18].

B. Classification Accuracy Metrics

Precision and recall are the most popular metrics for evaluating information retrieval systems [18]. Precision is defined as the ration of relevant items selected to the number of items selected, shown in (2).

$$P = \frac{N_{rs}}{N_s} \tag{25}$$

Where N_{rs} is the number of selected relevant items by the recommender system, and N_s is the total number of selected items. It shows the probability that a selected item is relevant. The criteria of be a relevant items is including a rating equal or greater than 3 (i.e. 'good') in *MovieLens*.

Recall is given in (26), is defined as the ration of relevant items selected to total number of relevant items available. It shows the probability that a relevant item will be selected.

$$R = \frac{N_{rs}}{N_r} \tag{26}$$

Precision and recall must be considered together [18]. One approach to combine these metrics is the F1 metric according to the (27).

$$F = \frac{2PR}{P+R} \tag{27}$$

C. Coverage and Correct Percent

The percentage of items for which a RS can provide predictions is determined by Coverage. Low coverage value indicates that the RS will not be able to aid the user with many of the new items. It will be computed by the bellow Equation:

$$Coverage = \frac{\sum_{i=1}^{N} p_i}{\sum_{i=1}^{N} n_i} \tag{28}$$

Where, p_i is the total number of predicted items for user u_i, and n_i is the cardinality of the test rating set of user u_i.

The percent of correct prediction via all predicted voting (high and low rated) is also computed for each split.

In the following tables, we first summarize the properties of presented methods in Table 1 and then show results of the mentioned metrics for average of the five splits of the *MovieLense* dataset in Tables 2-7.

TABLE I
SUMMARY OF METHODS PROPERTIES

	PRS	FRS	PBNRS	FBNRS
Related Equations	1,3	17,3	1,18,22,23	17,18,22,23

TABLE II
MAE FOR PRS,FRS,PBNRS , FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	0.9138	0.8897	0.8639	0.8552

TABLE III
COVERAGE FOR PRS,FRS,PBNRS , FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	0.3658	0.6794	0.9914	0.9932

TABLE IV
CORRECT PREDICTION FOR PRS,FRS,PBNRS AND FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	34.1203	39.972	48.4998	47.0376

TABLE V
PRECISION OF RECOMMENDATION FOR PRS,FRS,PBNRS AND FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	0.8328	0.8300	0.8264	0.8266

TABLE VI
RECALL OF RECOMMENDATION FOR PRS,FRS,PBNRS AND FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	0.3257	0.6662	0.9837	0.9836

TABLE VII
F1 OF RECOMMENDATION FOR PRS,FRS,PBNRS AND FBNRS

Spilt	PRS	FRS	PBNRS	FBNRS
Avg of 5 splits	0.4682	0.7392	0.8983	0.8983

In the following table we compare the ratio of the average of MAE and F1 metrics to show the usefulness of our method. Each column's numbers show the improvement of the related method comparing with others. For MAE, we should consider the coverage percent. For example, averagely PRS is able to cover 36.58% of users' ratings and MAE is 0.9138. The ability of FBNRS to cover ratings is 99.32% and MAE is 0.8552, so it is deduced that FBNRS is 2.9012 times better than PRS.

TABLE VIII
TOTAL COMPARISION OF PRS,FRS,PBNRS AND FBNRS

PRS		FRS		PBNRS		FBNRS		
MAE	F1	MAE	F1	MAE	F1	MAE	F1	
0	0	1.91	0.28	2.87	0.43	2.90	0.43	PRS
		0	0	1.53	0.16	1.52	0.16	FRS
				0	0	1.01	0	PBNRS

VIII. CONCLUSION

This paper proposes a hybrid fuzzy BN approach to genre-based recommender systems based on the *Movielense* dataset. Its contribution is involving movies genre's interestingness into a three layered BN to predict user preferences. Forming the neighbourhood set is based on the Pearson correlation coefficient and also a sort of novel fuzzy distance introduced in [12]. The usefulness of this method in comparison with pure Pearson and pure fuzzy RS is confirmed in the experiments by analyzing the predictive and classification accuracy metrics. As part of our future works, we aim to consider the added uncertainty that rises when handling of sparse data set i.e. where there is little information about users.

REFERENCES

[1] G. Linden, B. Smith, J. York, "Amazon.com recommendations: item-to-item collaborative filtering, " in *proc. 2003 IEEE Internet Computing conf.*, vol. 7, no. 1, pp. 76-80.

[2] B. Miller, I. Albert, S. Lam, J. Konstan, J. Rield, "Movielens unplugged: experiences with an occasionally connected recommender systems," in *2002 Proc. Internat. Conf. Intelligent User Interfaces*, pp. 263–266.

[3] R.J. Mooney, L. Roy, "Content-based book recommending using learning for text categorization," in *DL '00: Proc. 2000 Fifth ACM Conf. on Digital Libraries*, ACM Press, New York, NY, USA, pp. 195–204.

[4] D. O'Sullivan, B. Smyth, D.Wilson, K. McDonald, A. Smeaton, "Improving the quality of personalized electronic program guide, User modeling User-Adapted Interact ," *Kluwer Academic Publishers Hingham, MA, USA* ,v.14 n.1, pp. 5–35, 2004.

[5] A. B. B. Martínez, J. J. P. Arias, A. F. Vilas, J. G. Duque, M. L. Nores, "What's on TV Tonight? An Efficient and Effective Personalized Recommender System of TV Programs," *IEEE Trans. on Consumer Electronics*, vol. 55, no. 1, Feb. 2009.

[6] B. Krulwich, "Lifestyle Finder: Intelligent user profiling using large-scale demographic data," *Artificial Intelligence Magazine*, vol. 18(2), pp.37–45, 1997.

[7] Lang, K. "NewsWeeder: Learning to filter netnews," in *Proc. 1995 the 12th international conference on machine learning, Tahoe City,CA.*

[8] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, & J. Riedl, "GroupLens: An open architecture for collaborative filtering of Netnew,." in Proc. *1994 of ACM conference on computer supported cooperative work* ,pp. 175–186.

[9] U. Shardanand,P. Maes, "Social information filtering: Algorithms for automating 'Word of Mouth'. " in Proc. *1995 the conference on human factors in computing systems (CHI'95), Denver* , pp. 210–217.

[10] Pazzani, M. "A framework for collaborative, content-based, and demographic filtering." *Artificial Intelligence Review*, 393–408, 1999.

[11] Thomas M. Cover and Peter E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, (1967) Vol. 13 (1) pp. 21-27.

[12] M. Yahya, H. Al-Shamri , K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model, " *Expert Systems with Applications* vol. 35, pp.1386–1399, 2008.

[13] L. M. de Campos, J. M. Fernandez-Luna, J. F. Huete, "A collaborative recommender system based on probabilistic inference from fuzzy observations," *Fuzzy Sets and Systems*, vol. 159, pp.1554 – 1576, 2008.

[14] P. Symeonidis, A. Nanopoulos, A. N. Papadopoulos,Y. Manolopoulos, "Collaborative recommender systems: Combining effectiveness and efficiency, " *Expert Systems with Applications* vol. 34, 2008.

[15] P. Symeonidis, A. Nanopoulos, , Y. Manolopoulos , "Providing Justifications in Recommender Systems," *IEEE Trans. on Systems, Man, And Cybernetics*, vol. 38, no. 6, 2008.

[16] K. Yoshii, M. Goto, K. Komatani, T. Ogata , H. G. Okuno, "An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model", *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, vo. 2, Feb. 2008.

[17] K. Kyoung-jae, A. Hyunchul , "A recommender system using GA K-means clustering in an online shopping market ", *Expert Systems with Applications*, vol.34 pp.1200–1209, 2008.

[18] J. L. Herlocker, J. A. Konstant, L. G. Terveen, J. T. Riedl, "*Evaluating Collaborative Filtering Recommender Systems*" , *ACM*, 2003.