ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

Modeling a Rule based Movie Recommender System for Group Preferences

Ashu A, Debasish Jena

Abstract*— When notifying information to a user not only his personalized user preferences but also preferences of the group in which he/she is part of should be taken into account. It is also the case that the recommendation which is made to him/her must be justified in the sense that an explanation is to be provided as to why that particular item/data is recommended to the user. In this paper we outline a content based group recommender system that can recommend movies to a group of users which can provide both accurate and justifiable recommendation. We propose a Decision List Rule Learner (D.L.R.L), a rule based approach which is based on the well known Ripper rule learner to learn the rule base by taking into consideration the users viewing history and a method called Repeat Combined Rule Strategy (R.C.R.S) which is based on social choice theory strategies to generate group ratings. We compare our learning algorithm with the existing c4.5 rule learner and the experimental results show that the performance of our rule learner is better in terms of literals learned (size of the rule set) and our rule learner takes time that is linear to the number of training examples. The proposed method has the ability of offering a range of justifications for the recommendations made for individual/group of users alike, based on the rule base generated.

***Index Terms*— Decision List Rule Learner, Repeat Combined Rule Strategy, Rule Base, Weightage of a group member**

I. INTRODUCTION

With the development of Web 2.0, the use of the web has become increasingly widespread and users have had the chance to express opinions about shared content updated daily. This generates an incredible amount of data that can't be handled directly by the users. So finding relevant information over the Internet nowadays is becoming more and more difficult. In a similar manner as online community activities have increased exponentially the need for a recommendation system has also become more and more imperative. So we need systems for recommending items (books, movies, websites etc.) that takes into consideration our own as well as a group's interests. Recommender systems aim to support users in their decision-making while interacting with large information spaces. These systems are generally divided into two categories namely personal recommender system and group recommender system wherein the former is effective in filtering useful information that fit each user's needs whereas the latter provides suggestions for group decisions and satisfy user's needs in group activities. They recommend items of interest to users based on preferences they have expressed, either explicitly or implicitly. Recommender systems help overcome the information overload problem by exposing users to the most interesting items in a brief and clear form, thereby offering novelty, surprise and relevance. Based on kind of recommendation techniques are used, personalized recommender systems are usually classified into three categories [5].

- Collaborative Filtering (C.F.): Recommends items to users that other user with similar tastes have liked in the past.
- Content based Recommending (C.B.): Provides recommendations by comparing representations of content describing an item to representations of content that interests a user.
- Hybrid Methods: Combines collaborative and content based methods.

In this paper we propose a Content based group recommendation system which uses Decision List Rule Learner (D.L.R.L) based on RIPPER to learn rule base and Repeat Combined Rule Strategy (R.C.R.S) strategy based on social choice theory strategies for group recommendation. Based on the rule base we provide justification for the recommended items to the user. The rest of the paper is organized as follows: Section II discusses previous work relevant to this paper and the problem description; In Section III, we describe the proposed model to group recommendation system; Section IV provides the performance analysis with the existing rule recommendation system; Section V contains the conclusion, future developments.

II. RELATED WORK

A (individual/personal) recommender system is a system which, through an information filtering technique, attempts to recommend information items - e.g.: music, movies, TV programs, videos on demand, books, news, images, web pages, research papers etc.), which are likely to be of interest to a single user. The way an individual recommender system works is that typically it compares a user profile to some reference characteristics, and tries to predict the 'rating' that a user would give to an item they had not yet considered based on these characteristics which may belong to the information item (the content-based approach) or the user's social environment (the collaborative filtering approach). A group recommender system is a recommender system aimed at generating a set of recommendations that will satisfy a group of users, with potentially competing interests. The challenges associated with this simple statement deal with, considering how to record and combine the preferences of many different users as they engage in simultaneous recommendation dialogs. The music recommender system, GroupFun, is designed to help a group of friends reach a common music play list for their distinct tastes. As far as personal recommendation is considered collaborative filtering (C.F.) techniques are widely used and have become the most

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012
preferred method. Memory-based and Model-based algorithms which are subdivisions of C.F. methods have been extensively studied in this regard. In memory-based technique a subset of users are chosen based on their similarity to active user and a weighted combination of their ratings is used to produce predictions for this user. Similarity measures like Pearson correlation, cosine similarity, Spearman rank correlation, Kendall's t correlation, mean squared differences, entropy and adjusted cosine similarity have been used [7,4]. It was shown that the conventional memory-based C.F. algorithms do not scale well when applied to millions of users and items due to the computational complexity of the search for similar users. To overcome this drawback item-to-item Collaborative Filtering [8] was proposed; where rather than matching similar users, they match a user's rated items to similar items. It was shown that this approach leads to faster online systems, and often results in improved recommendations [9]. The similarity fusion between the user-based and item-based methods [10] was proposed which also uses data from a third source (ratings of other similar users on other similar items). Other notable extensions to similarity-based Collaborative Filtering include weighted majority prediction [11] and imputation-boosted C.F. [6].

When model-based C.F. algorithms are concerned latent factor and matrix factorization models [13] have emerged as the state of the art methodology in this class of techniques. Cluster-based smoothing method [12] was proposed wherein clusters are created and predictions for a target user are made by averaging the opinions of the other users in the cluster he/she participates and is weighted by the degree of participation. Another clustering algorithm put forward in [14] uses the decoupled model wherein user preferences from its rating is decoupled but allows user/item to be in multiple clusters and performs separate clustering of users and items. All the methods discussed above fall into the category of C.F. algorithms and the problem with pure collaborative filtering recommenders is that it treats all users and items as atomic units, where predictions are made without regard to the specific of individual users or items. Whereas many pure C.B. systems have tried to provide explanations by knowing more about a user, such as demographic information or about an item, such as the director or genre of a movie [15]. Pure C.B. to recommend news articles to users [16] provided explanations for the reasoning behind their recommendations. This method exploits user's feedback to improve the recommendation process. A method for recommending books based on pure C.B. was proposed in [17] where a machine learning algorithm was used for text categorization and explanations were provided for the recommendations made. In order to leverage the strengths of content-based and collaborative recommenders, there have been several hybrid approaches proposed that combine the two. A general framework for content-boosted collaborative filtering is proposed in [15] where content-based predictions are applied to convert a sparse user ratings matrix into a full ratings matrix, and then a C.F. method is used to provide recommendations. A web recommender system is proposed in [18] in which collaborative and content features are integrated under the maximum entropy principle. In fact, hybrid recommendations were extended to contain knowledge-based techniques for the purposes of improving the quality of recommendations and reducing the effect of the traditional C.F. cold-start problem. More recent approaches [19] allow users to create their own profile by crafting a list of their own questions/topics. Such system differs from the traditional recommender system ones, since it recommends content for users to create, rather than consume. They deploy two different algorithms (Network-based and Content-based), with the aim of recommending a set of meaningful questions to the user by looking at the behavior of the users which are similar to the target one.

In the case of group recommender systems only a few have been designed as of now. All these systems assume that the input of the system is comprised of item's ratings given by individuals and group recommendation is obtained by aggregating or combining the individual recommendations of the members in the group.GRS with Consideration of Interaction among Group Members [22] is a novel group recommendation approach based on collaborative filtering and genetic algorithm to predict the possible interactions among group members so that it can correctly estimate the rating that a group of members might give to an item. TV program Recommendation for Multiple Viewers based on User Profile Merging [23] is a group recommendation system which is build on merging the user profiles. Merging of user profile is done based on total distance minimization technique. A TV Anytime Metadata Approach to GRS [24], is a method to elaborate personalized recommendations for a group of users watching TV together, they start from an OWL ontology that (i) describes TV programs, (ii) classifies them in a content hierarchy, and (iii) relates them to other programs through their semantic characteristics (attributes such as cast members, location, dates....). Social choice Theory Strategies [5] (also called as group decision making)-deciding what is best for a group given the opinions of individuals. The strategies are Plurality Voting, Utilitarian Strategy, Least Misery Strategy, Most Pleasure Strategy and Average without Misery Strategy.

A. Problem Description

In the collaborative filtering method items are recommend to an active user(The user who wants recommendations is called an active user) as follows, it simply searches the database to find other similar users for the active user (similar users are found based on their user profiles). Based on those similar users, it will recommend the items. But there are cold start, first rater and popularity bias problems with collaborative recommender systems. Cold start occurs when there are no enough similar users in the database. First rater problem occurs when the item is new or it hasn't been rated earlier by any user. Popularity bias occurs when an item is recommended based on the opinion of other similar users.

In recent years, a few group recommender systems have been designed. These systems have been developed based on collaborative filtering technique [22] [24] [25]. Hence, they suffer with the problem of collaborative filtering as stated

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

above. Our proposed approach uses Content based recommendation.

In Content based approach, recommendations are based on information on the content of items rather than on other users opinions. It uses a machine learning algorithm to induce the profile of the user preferences from training examples based on a feature description of content. To learn user profile, many machine learning algorithms have already been used. Among those, popular algorithms are (i) decision tree rule learners and (ii) Naive Bayes classifier. But Movie content is a kind of structured data, so we tried to use rule learners. Decision tree rule learners have already been used to develop recommender systems. But there are some problems with decision trees:

- Decision trees are often quite complex and hard to understand. Even a pruned decision trees may be too cumbersome, complex, and inscrutable to provide insight into the domain at hand [26].
- There is a replicated sub tree problem. It often happen that identical sub trees have to be learned at various places in a decision tree [26].
- Sequential Covering algorithms (Decision List Learners) are better than Simultaneous covering algorithms (Decision Tree rule learners) when plenty of data available [27].
- Decision list (Ordered set of rules) with at most k-conditions per rule strictly more expressive than decision tree of depth k [28].

When the training data is structured, rule based recommendation is better [29].Here the movie data is structured because the information about the movie like director, hero etc are related. Most recommendation techniques as cited above have mainly aimed to enhance the accuracy of recommendations by fine-tuning the respective algorithms. At the same time it is also important that the acceptance of a recommender system is increased when users receive along with a recommendation the reasoning behind it. Such recommendations are called justified recommendations and in an e-commerce like scenario they help to improve customer attraction/retention and sales boosting because customers can evaluate the provided recommendations more easily and accept them if satisfactory. In this paper we propose a novel approach for group recommendation system which uses Decision List to generate the rule base and combine the social choice theory strategies for group recommendation. This is an advantage when compared to the existing group recommender systems wherein recommendations are made by aggregating or combining the individual recommendations of the members in the group or by aggregating the group's rating of similar items through the item-based collaborative filtering algorithm.

III. PROPOSED MODEL

The aim of our work is to develop a group recommender system that is able to recommend a Movie to an individual as well as group of users. In our proposed model we propose Decision List Rule Learner (D.L.R.L) based on RIPPER [30] and FOIL rule learning techniques for generating the rules. We propose Repeat Combined Rule Strategy (R.C.R.S) which is based on social choice theory strategies [5] to generate group recommendations. R.C.R.S. takes the rule set generated by the D.L.R.L as input for recommending new movies to the group of users. It also takes the weightage into account the weightage of the each user in the group before recommending movies to the group. Initially, the system has no idea to recommend any MOVIE except if we add some external rules. For few weeks, It collects ratings for each and every the user has watched. These are the training examples to the learning algorithm. From these training examples the learning algorithm learns the set of rules that cover all training examples. This process will be done for every user i.e., for each user, we get a separate rule base (User profile).

Movies are classified according to their 19 genres and these genres are the features of the movies in our data set. The 19 genres are: Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western. Here, the rating is 5-scaled one 0, 1,2,3,4. 0 indicates bad , 1 indicates average , 2 indicates above average , 3 indicates a good , 4 indicates an excellent .

There are N-users; we will get the rating for the movies they have watched. These collected movies with corresponding rating are shown as an N-user viewing histories. D.L.R.L uses the collected user viewing history and learns the ordered rules called as RuleBase, for each user. Whenever a new movie releases it consists of information like action, actor etc. This new MOVIE will be classified by the each user's Rule base; hence we get predicted ratings for each MOVIE in the MOVIE set. These movies with predicted ratings are nothing but recommendation list for each individual user. Finally, R.C.R.S uses the rule set generated by the D.L.R.L and provides a combined (group) recommendation list to a group of users. The complete process of our method is shown in Fig.1.

In this method, we are collecting the ratings for each and every movie the user has watched. But some members are unlike to give ratings; therefore we need a method to generate the ratings for the movies in viewing history. We suggest a method that is used for each and every movie to calculate probability of the movie in viewing history as the occurrence of the similar movies in the viewing history and based on that we can give ratings. If a similar kind of movie occurs many times than we can give high rating for those movies and so on

A. *Decision List Rule Leaning Algorithm*

Learning algorithm plays major role in content based recommendation approach. It is used to learn user profiles. Our learning algorithm, D.L.R.L is based on RIPPER and FOIL rule learners. It is a multi-class rule learner. In our case, there are five classes: bad, average, above average, good, excellent.

Initially, all training examples are divided into two sets: train data and prune data. Train data is used to learn the set of rules. Prune data is used to prune the rules to avoid over fitting. FOIL Information gain is as follows:

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

$$FoilGain(L,R) = t\left(\log_2\left(\frac{p1}{p1+n1}\right) - \log_2\left(\frac{p0}{p0+n0}\right)\right) \qquad (1)$$

Where **L** is the candidate literal to add to rule **R**, **p0** is the number of positive bindings of **R**, **n0** is the number of negative bindings of **R**, **p1** is the number of positive bindings of **R + L**, n1 is the number of negative bindings of **R + L**, **t** is the number of positive bindings of **R** also covered by **R + L**. The pruning criteria (v) used here can be defined as

$$v = \frac{(p-n)}{(p+n)} \qquad (2)$$

Where **p** is the number of positive examples covered by the rule in prune data set and **n** is the number of negative examples covered by the rule in the prune data set.

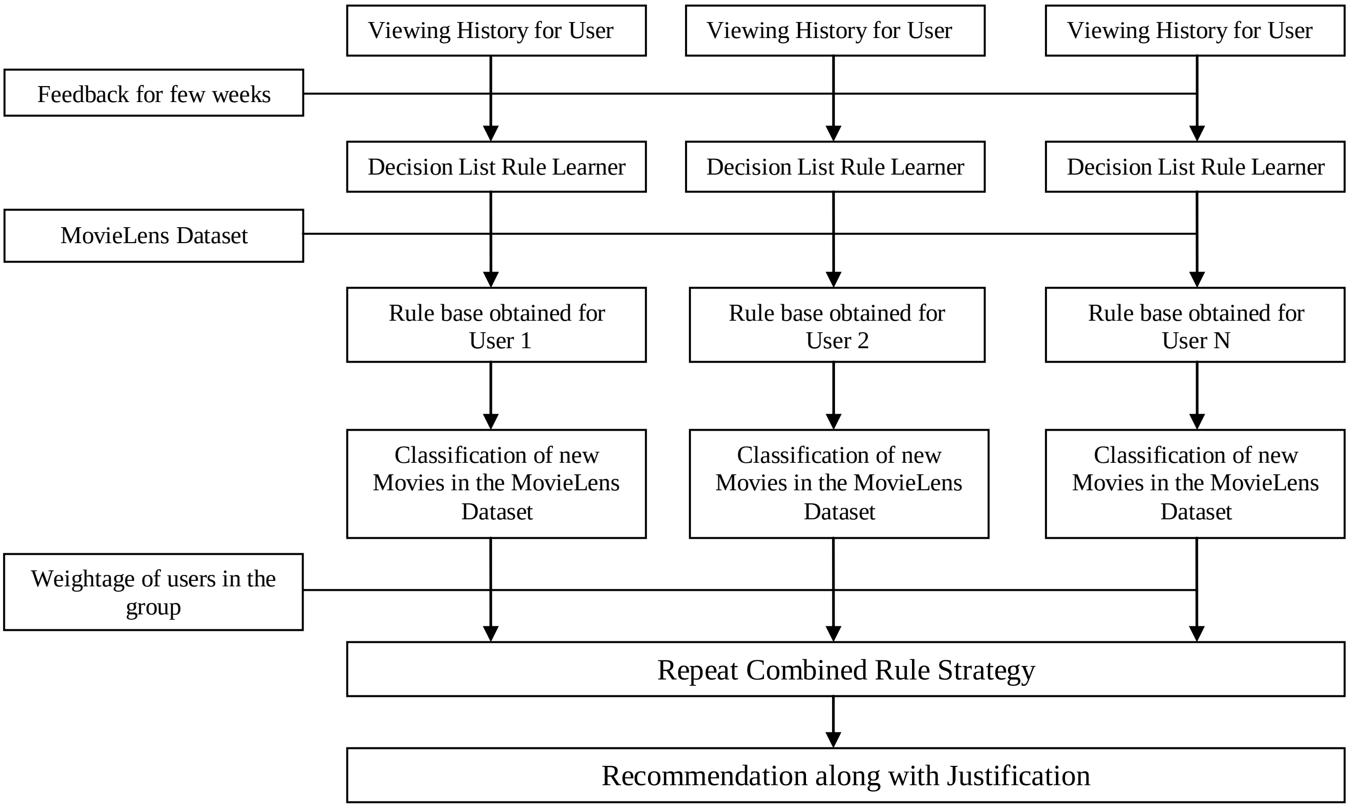

Fig.1. Working of the Proposed Model

Pruning criteria is deleting the final sequence of conditions that maximizes v. The steps in our algorithm are shown in Algorithm-1.

Algorithm 1: Decision List Rule Learning Algorithm

Input: Train Data
Prune Data

Output: Set of rules

START:

1. Find the number of training examples for each individual class

2. Consider an empty RuleSet

3. For no class has left

//Consider the next smallest class;

//Take class (training examples) → positive;

// Others → negative;

4. For all positive examples

i. Take empty Rule;

ii. Add conjuncts to rule when it improves FOIL Information gain

Do rule pruning by deleting any final condition sequence

iii. Mark covered positive examples by this rule

iv. Then add this rule to the Rule Set;

5. Add Default rule to Rule set and Return Rule Set

END

B. R.C.R.S Strategy to Generate Group Ratings

A single strategy alone wouldn't be sufficient to get the most accurate result as far as ruled based group recommenders are concerned. Therefore, we need a combined strategy that considers three factors:

i. Least group member happy (like least misery strategy)
ii. Most group member happy (like most pleasure strategy)
iii. Total group happy (like Utilitarian strategy).

We propose a method by combining these three factors. The three factors which we consider are taken on Social Choice Theory Strategies which decides what is best for a group given the opinions of individuals. We name the proposed strategy as Repeat Combined Rule Strategy

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

(R.C.R.S.). The working process of R.C.R.S is shown in Algorithm 2.

Let G be a group consists of N users. Here the method is to calculate the sum of least happiness and total happiness for each instance and calculate the maximum of these summed values and then recommend that instance. If we have maximum value for multiple instances than, take only that instances, remove the minimum values (least happiness) from these instances and apply the same process for the new set of instances repeatedly. So we name it as Repeat Combined Rule Strategy.

Algorithm 2: Repeat Combined Rule Strategy Algorithm

Input: InstanceRatingVector (I)
    Number of users (N)
Output: Recommended Instances
START:
1. if all Instances in I has " 0" user rating then goto step 2;
  else
    //remove the instances in I which has " 0 " user rating
    //and goto step 2;
2. CountNoOfRepeat=0;
  for each instance i in I
      // calculate Ci and max.
      // Where Ci= sum of all user rating + minimum user
          // rating for instance "i";
        // max = maximum value in Ci;
  if countNoOfRepeat = N then
      for each Instance i in I
          if max=Ci then
          // return i;
  else
  if max appears many times in Ci and
    countNoOfRepeat < N then
      for each instance i in I
      // Remove the least rating entries for the instance "i"
      //countNoOfRepeat++;
      //repeat step2;
  else
      //return i
END

C. Weight age of a Group Member

Another intuition we had is to assign weights to group members. We thought that in real life, weight age of group members needs to be considered in certain situations. For instance

- In a family, channels are chosen based on the decision of the father or grandfather or some other senior member of that family
- There might be an occasion like Birthday, Wedding day of members in a group. These members should be having more weight age on that day.

To apply weightage to R.C.R.S, we just modified the Instance Rating Vector by two things:

- For user who has the minimum rating for that instance divide his rating by his weight age.
- For other users, multiply user ratings with their weight age and apply the same Algorithm 2.

IV. PERFORMANCE ANALYSIS

To demonstrate that our recommendation method works, we have performed experiments with a real data set called Movie Lens which is commonly used as the bench mark data set for movie recommender systems. These data sets were collected by the GroupLensResearch Project at the University of Minnesota. This data set consists of: 100,000 ratings from 943 users on 1682 movies. The range of rating is between 1 (bad) and 5 (excellent). Each user has rated at least 20 movies. Movies are classified according to their 19 genres and these genres are the features of the movies in our data set. '1' under a particular genre indicates that movie is of that genre and '0' indicates it is not; movies can be in several genres at once.

For instance, we selected a group of random users among the 943users of the data-set and our method has recommended two movies to this group as shown in Table I. The first row of the figure can be interpreted as saying that item TrueLies has been recommended because it contains features Action, Comedy, Adventure, and Romance which are included in items X, Y the group members has already rated separately. Our method achieved good results by recommending user with items that match his interest based on the evaluation metrics of precision and recall which is explained in the next section *A*.

Table I. Movies Recommended to a group of random users.

RECOMMENDED ITEMS	THE JUSTIFICATION OF RECOMMENDED ITEMS
TRUE LIES (1994)	ACTION, ADVENTURE, COMEDY, ROMANCE
STAND BY ME (1996)	ADVENTURE, COMEDY, DRAMA

A. Evaluation Metrics

First we have to divide our data into two disjoint sets, the training set and the test set. The algorithm employed by the system works only on the training set and generates rules for all the ratings provided by the users in the training set, which we will refer to as Rule set. The main goal is to scan through the instance set, representing the portion of the initial data set which was not used by the Recommender System, and match items in the test set with items included in the generated Rule set. Items that appear in both sets will become members of a special set, called the hit set.

The existing studies of recommendation systems use several different measures to evaluate the quality of recommendation produced [33]. The metrics we adopted and used to calculate the performance of our recommender systems are recall and precision measure which is one of popular evaluation metrics in information retrieval systems. We can now define recall and precision for recommendation systems in the following way: Recall describes the idea of all items which are relevant to the query that are successfully retrieved. Here, it can be defined as the ratio of relevant retrieved items over the relevant items.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

$$\mathrm{Recall} = \frac{\mathrm{RelevantItemsRetrieved}}{\mathrm{ItemsRetrieved}} \quad (3)$$

Precision describes the idea of only those items which are relevant to the user's information need. Therefore, we have used the precision for measuring the correctness of recommendation as the ratio of the relevant retrieved items to the number of retrieved recommended items (Rule set).

$$\mathrm{Precision} = \frac{\mathrm{RelevantItemsRetrieved}}{\mathrm{RetrievedItems(RuleSet)}} \quad (4)$$

That is, precision represents the probability that recommended items are chosen for each user. The difference among the different groups was rather small, and our system consistently resulted in better recommendations for all group sizes that have been taken in our examples. The result for the MovieLens data set for different values of 'v' which was used in Equation (2) is shown in Fig. 2. As expected, precision decrease when 'v' increases. This decrease is less steep because in this data set, there is no deviation between the users viewing history and their rating behavior. However, our method attains good precision because the similarity measure is based on Rule set and thus, being able to detect partial matching of users rating and can provide accurate recommendations

B. Comparison with C4.5 and Cluster based approach

The number of literals learned for these datasets for Decision tree rule learner (C4.5) [30], Cluster based approach [12], Decision list rule learner (D.L.R.L: Our method) and clustered based approach are shown in Fig.3. From this graph we can say that D.L.R.L learns less number of literals (conditions) than the decision tree rule learner (C4.5) and cluster based approach that are sufficient to classify the training data. Anyway, we know that cluster based approach requires to store all the examples in order to recommend the items. Hence the graph of this method is having high values. We noted the time required in learning the different datasets is shown in Fig.3. (No. of training examples and Time). From these experimental results, we can say that the performance of D.L.R.L is almost linear to the training examples.

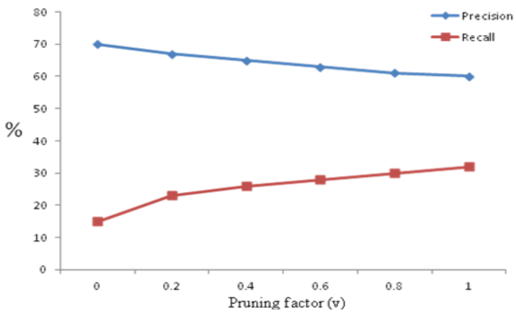

Fig.2. Precision and recall versus 'v' for MovieLens

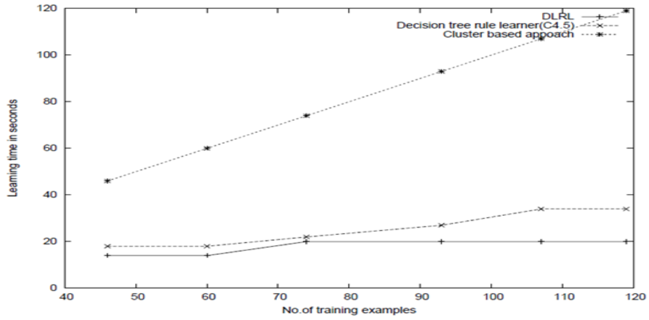

Fig.3. Comparison between cluster based, C4.5 and D.L.R.L

V. CONCLUSION

In this work, we developed a content based group recommender system for movies. Our proposed method D.L.R.L generates rule base for the users and these rules are used by another method called R.C.R.S to generate group ratings, which is based on social choice theory strategies. We compare our learning algorithm with the existing c4.5 rule learner and cluster based approach. Results shows that performance of proposed rule learner is better in terms of literals learned (size of the rule set) and our rule learner takes time that is linear to the number of training examples. In R.C.R.S, we consider the weightage of the group members which is used in recommending movies to the group. Our proposed D.L.R.L approach is evaluated by applying precision and recall evaluation metrics on MovieLens dataset.

REFERENCES

[1] Shunichi Seko, Takashi Yagi , Manabu Motegi, Shinyo Muto. "Group recommendation using feature space representing behavioral tendency and power balance among members", ACM Conference on Recommender Systems; 2011, p.101-108.

[2] J. Wang, A. Vries, and M. Reinders. "Unifying user-based and item-based collaborative filtering approaches by similarity fusion", in Proc. SIGIR Conf.; 2006,p. 501–508

[3] Gediminas Adomavicius and Alexander 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; 2005,p.734–749

[4] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl ."An algorithmic frame-work for performing collaborative filtering", In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999, p. 230–237

[5] Judith Masthoff. "Group modeling: Selecting a sequence of television items to suit a group of viewers. User Model". User-Adapt. Interact.; 2004 ,p.14(1):37–85

[6] Xiaoyuan Su, Taghi M. Khoshgoftaar, Xingquan Zhu, and Russell Greiner. "Imputation-boosted collaborative filtering using machine learning classifiers", In SAC '08: Proceedings of the ACM symposium on Applied computing; 2008,p. 949–950.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

[7] Xiaoyuan Su and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques". Advances in Artificial Intelligence; 2009,p.1–20

[8] Robert Bell Yehuda Koren and Chris Volinsky. "Matrix factorization techniques for recommender systems." In IEEE Computer,; 2009 ,volume 42 (8), pages 30–37.

[9] Greg Linden, Brent Smith, and Jeremy York. "Amazon.com recommendations: Item-to-item collaborative filtering", IEEE Internet Computing; 2003,7(1):p.76–80.

[10] J. Wang, A. Vries, and M. Reinders ".Unifying user-based and item-based collaborative filtering approaches by similarity fusion", in Proc. SIGIR Conf.; 2006,p. 501–508.

[11] Atsuyoshi Nakamura and Naoki Abe. "Collaborative filtering using weighted majority prediction algorithms". In ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, p. 395–403

[12] Gui R.Xue,Chenxi Lin, Qiang Yang, WenSi Xi, Hua. J., Zeng, Yong Yu, Zheng Chen. "Scalable Collaborative filtering using cluster based smoothing", ACM SIGIR Conference on Research and development in Information Retrieval; 2005, p.114-121

[13] Robert Bell Yehuda Koren and Chris Volinsky. "Matrix factorization techniques for recommender systems." In IEEE Computer; 2009, volume 42 (8), pages 30–37.

[14] R. Jin, L. Si, and C. Zhai." A study of mixture models for collaborative filtering," Inf. Retr.;2006, vol. 9, no. 3, pp. 357–382.

[15] Prem Melville, Raymond J. Mooney, and RamadassNagarajan. "Content-boosted collaborative filtering for improved recommendations," In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02); 2002, p.187–192

[16] D. Billsus and M. Pazzani. "A personal news agent that talks, learns and explains" in Proc. Auton. Agents Conf.;1999,p. 268–275.

[17] R. Mooney and L. Roy. "Content-based book recommending using learning for text categorization" in Proc. ACM DL Conf.; 2000, p. 195–204.

[18] X. Jin, Y. Zhou, and B. Mobasher. "A maximum entropy web recommendation system: Combining collaborative and content features" in Proc. ACM SIGKDD Conf.;2005 ,p. 612–617.

[19] Werner Geyer, Casey Dugan, David R Millen, Michael Muller and Jill Freyne." Recommending topics for self-descriptions in online user profiles", In RecSys, 2009, p. 59-66.

[20] Mastho, J. " Modeling a Group of Television Viewers" In Proceedings of the workshop future TV in intelligent tutoring systems conference, 2002 ,p. 34–42.

[21] McCarthy, J. F., &Anagnost, T. D. "MusiC.F.X: an arbiter of group preferences for computer supported collaborative workouts",in Proceedings of the ACM conference on computer supported cooperative work;1998 ,p. 363–372

[22] Yen-Liang Chen,Li-Chen Cheng,Ching Nan Chuang," A group Recommender System with consideration of interaction among group members ,Expert Systems with applications ;2008, p. 2082-2090

[23] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. Tv program recommendation for multiple viewers based on user profile merging." User Model. User-Adapt. Interact; 2006, p. 16(1):63–82

[24] R. Tubio, R. Sotelo, Y. Blanco, M. Lopez, A. Gil, J. Pazos, and M. Ramos. A tv-anytime metadata approach to tv program recommendation for groups. In Consumer Electronics. ISCE , IEEE International Symposium ;2008,p. 1–3,

[25] Luis M. de Campos, Juan M. Fernandez-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Group recommending: " A methodological approach based on Bayesian networks". In ICDE Workshops;2007, p. 835–844

[26] Johannes Furnkranz. "Separate-and-conquer rule learning". Artif. Intell. Rev.;1999, p. 13(1):3–54

[27] Tom M. Mitchell. Machine Learning. McGraw Hill, USA, 1997.

[28] Ronald L. Rivest. "Learning decision lists. Machine Learning"; 1987,p. 2(3):229–246.

[29] Asela Gunawardana and Guy Shani. "A survey of accuracy evaluation metrics of recommendation tasks". The Journal of Machine Learning Research, 2009,p. 10:2935–2962

[30] William W. Cohen. "Fast effective rule induction". In ICML; 1995,p 115–123

[31] J. R. Quinlan. "Improved use of continuous attributes in C4.5". Journal of Artificial Intelligence Research, 4:77–90, 1996.

[32] http://movielens.umn.edu/login

[33] Jonathan L. Herlocker **,** Josheph A. Konstan, Loren G. Terveen, and John T Riedl." Evaluating collaborative filtering recommender systems**",** ACM Transactions on Information Systems; 2004, Vol. 22, No. 1, p. 5–53

AUTHOR BIOGRAPHY

Ashu A received the B.Tech degree in 2009 from Jawaharlal Nehru Technological University, Hyderabad, India. Currently he is pursuing his M.Tech degree from Department of Computer Science and Engineering, International Institute of Informational Technology, Bhubaneswar, Orissa, India. His current research interests include Machine Learning and Data-Mining.
E-mail: a110003@iiit-bh.ac.in

Debasish Jena received the B.E degree in 1990 from Gulbarga University, the M.Tech degree in 2002 from Osmania University at Hyderabad and the Ph.D degree in 2010 from National Institute of Technology, Rourkela, India. His current research interest includes Information Security.
E-mail: dr.djena@gmail.com