A DISSERTATION

ON

OPTIMIZED COLLABORATIVE RECOMMENDATION ALGORITHM FOR AUTO DETECTED GROUPS

Submitted in Partial fulfilment of the Requirements

For the award of the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUBMITTED BY

NISHANT VERMA
ROLL NO: 2K13/SWT/09

UNDER THE ESTEEMED GUIDANCE OF

MR RAHUL KATARYA

ASSISTANT PROFESSOR, CSE DEPARTMENT, DTU

DEPARTEMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI-110042, INDIA

DELHI TECHNOLOGICAL UNIVERSITY, NEW DELHI

DECLARATION

I hereby undertake and declare that this submission is my original work and to the best of my knowledge and believe, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of any Institute or other University of higher learning, except where due acknowledgement has been made in the text. Project work associated to the chapters is well discussed and improved under the guide supervision.

DATE:

SIGNATURE:

NISHANT VERMA
ROLL NO: 2K13/SWT/09

DELHI TECHNOLOGICAL UNIVERSITY, NEW DELHI

CERTIFICATE

This is to certify that the thesis entitled, "**OPTIMIZED COLLABORATIVE RECOMMENDATION ALGORITHM FOR AUTO DETECTED GROUPS**", has been carried out by Nishant Verma (Roll no: 2K13/SWT/09) under my supervision and guidance in partial fulfilment of the degree of Master of technology in Computer Science & Engineering at Delhi Technological University, New Delhi. The matter embodied in the thesis has not been submitted to any other University/ Institute for the award of any degree or diploma to the best of my knowledge.

DATE:

SIGNATURE:

MR. RAHUL KATARYA (SUPERVISIOR)
ASSISTANT PROFESSOR, DEPARTMENT OF CSE,
DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGMENT

I am hereby presenting my work on “**Optimized Collaborative recommendation algorithm for auto detected groups**” with a lot of pleasure and satisfaction. I take this opportunity to thank my supervisor, Rahul Katarya, for guiding me and providing me with all the facilities, which paved way to the successful completion of this work. I would like to express my deep gratitude to my parents. Their continuous love and support gave me strength for pursuing my dream. I am thankful to my friends and colleagues who have been a source of encouragement and inspiration throughout the duration of this thesis. I am also thankful to the SAMSUNG who has provided me opportunity to enrol in the M.Tech. program and to gain knowledge through this program. This curriculum provided me knowledge and opportunity to grow in various domains of computer science. Last but not least, I am thankful to all the faculty members who visited the Samsung premises to guide and teach. Their knowledge and efforts helped me to grow and learn in the field of computer science. I feel proud that their contribution helped me to bring out new ideas in my professional life.

Abstract

Recommender Systems (RSs) are software tools and techniques used for providing suggestions to user based on either user's individual taste or are based on resemblance of a user to another user. Considering large number of items available to choose from it becomes difficult for user to make a choice, recommender system helps a user to make an informed decision by calculating the likelihood of user liking an item. Recommender systems are categorised into 6 different branches named: Content based, Collaborative filtering based recommender systems, Demographic, Knowledge-based, Community based recommender systems, Hybrid recommender systems. Collaborative filtering is the most successful and widely used technique in recommendation systems. This approach recommends to user the items that other similar users have liked in the past. Similarity is calculated between users based on similarity of rating history of users.

In this project the main focus has been on group recommendations based on collaborative filtering method incorporated with various group modelling strategies [1]. These group modelling strategies combines various user model into a single model and represent the available knowledge about user preference belonging to a group. The users are clustered into groups according to their ratings using a variant of k-means clustering algorithm EZ Hybrid [4]. After clustering the users and group formation the group ratings are predicted for all the items and finally the prediction accuracy is tested for the different strategies. Additive utilitarian, Approval voting with threshold 1 and 2, least misery and most pleasure strategy from existing literature were implemented. A new group modelling strategy "Median strategy" is proposed and its performance is compared with those present in the literature. 1M dataset from Movielens is used in the experiment. RMSE, MAE, Precision and Recall are the parameters used to measure the performance of prediction accuracy for auto detected groups. From results we come to a conclusion that the new proposed strategy gives better Precision, Recall, and MAE compared to already present in literature. MAE is improved by 6.67%, precision by 7.77% and recall by 9.69%. Also the RMSE values are better than all other strategies except additive utilitarian. Hence we conclude that using median strategy helps in making more accurate predictions.

Table of Contents

List of figures

List of tables

Chapter 1: Introduction

This chapter provides a general overview of the thesis and briefly explains the basic concepts of Recommender systems. It also talks about the motivation and various challenges and research issues that are currently faced in the field.

1. 1 Motivation

Due to the information overload and explosive growth and variety of information available making choices became difficult. RSs have in recent years become helpful in dealing with this information overload problem. Recommender Systems (RSs) are extensive class of web applications that are involved in prediction of user response to various options. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.

The main focus of recommender systems is in personalized recommendations, which are offered as ranked list of items. In order to make prediction for a user a recommender system collects from user their preferences which are either expressed explicitly, e.g. ratings given by a user to an item or implicitly by users browsing history or by cookies. Recommender systems rely on various types of input. Most convenient is high quality explicit feedback, where users directly report on their interest in products. The aim of service providers using RSs is to increase the number of items sold, sell more diverse items, increase user satisfaction and better understand what user wants.

When information systems are used to produce group recommendations and market segmentations, very large data of users and millions of items needs to be aggregated in order to produce suggestions for users. So it becomes very important for the system to analyse big data and able to produce inferences from that data which is useful to user.

In *Group recommendation* multiple users are involved in decision making process, where individual user preferences are combined to suggest items to a group. Various group modelling strategies that combine user models are used to develop a group recommender system. It was shown by Pizzutilo et al. that for every context there is no common strategy applicable and its choice must be made after careful analysis of context in which the groups are modelled [14].

Group recommendation is useful when it is not possible to produce a recommendation list for every user. So a possible solution is to group similar users together and then make common recommendation to them. However a common list may not be able to correctly map the user choice and user satisfaction can be low. So the users preferences are modelled by cluster them into a group.

1. 2 Research Issues

The various issues related to the field can be summarized broadly into below queries.

- Is it possible to make personalized preferences for each and every user, given that user count may be in millions?
- Can a group preference accurately map every user's personal choice?
- How to apply a Group modelling strategy to a particular context?
- Which Group modelling strategy gives best results?

1.3 Challenges in Recommender Systems

The main challenges faced in development of RSs have been pointed below:

- Scalability of algorithms to meet requirements of real world dataset.
- Provide accurate predictions even in case of shilling attacks.
- Ability to proactively make recommendations even if not asked explicitly.
- Preserving privacy of users.
- Emphasize on diversity of recommendations.
- Enhancing cross domain recommendations.
- Implementing scalable and flexible distributed systems.
- Designing of recommender system to work efficiently in case of mobile devices.

1.4 Thesis outline

This thesis is organized into 6 chapters. In this section a brief description of each chapter is given.

- **Chapter 2 – Recommender systems.** In this chapter brief introduction of recommender system is given explaining each type to systems present. Also the background and related work is presented.

- **Chapter 3 – Group Modelling Strategies.** In this chapter the various group modelling strategies present in literature are explained briefly.
- **Chapter 4 – Implementation.** This chapter presents the implementation steps and the new strategy proposed to predict group rating.
- **Chapter 5 – Results and Analysis.** In this chapter the results of the experiment are presented and compared with existing work.
- **Chapter 6 – Conclusion and future work.** This chapter briefly presents the scope of future work and summarize the work done in this thesis.

Chapter 2: Recommender Systems

Recommender systems (RSs) are computer-based techniques that attempt to present information about products that might be of interest to a user. These techniques are mainly used in *e-Commerce* in order to provide suggestions of items that a customer is, might like. RS have also found use in other applications, such as social networks and community-building processes.

2.1 Introduction to Recommender Systems

Item is the general term used to denote what the system recommends to users. A RS focuses normally on a specific type of item (e.g., movies, or news) and accordingly its design, its graphical user interface, and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item. RSs are directed primarily towards individuals who lack sufficient personal experience or competence to evaluate the overwhelming large number of alternatives that are being offered and thus are unable to make an informed decision.

For example is a case where a movie recommender system helps a user to select which movie can be of interest to him based on his previous history or based on the movies liked by his peers. Also the recommendations can be non-personalized which are easy to generate and normally features top rated or bestselling items.

Users of a RS are one for whom recommendations are made. Users have very diverse goals and preferences. So in order to personalize the recommendations the RSs exploit a range of information about the users mostly based on individual taste or past history. This information can be structured in various ways and again the selection of what information to model depends on the recommendation technique used.

Transactions are the recorded interaction between users and RS, most popular form of transaction is ratings given by user to an item. Rating can take various forms like:

- Numerical ratings like 1-5 stars
- Ordinal ratings like “ agree, neutral, disagree”
- Binary ratings i.e. good or bad

- Unary ratings indicates user have rated an item, absence of rating indicate no information for that user is available.

The function of RSs is to predict that a particular item is worth recommending for a user. The system must be able to compare the utility of item with each other and be able to decide which item is to be recommended. The degree of utility of user *u* for an item *i* is modelled as R(u, i) i.e. rating of user u for item i. In collaborative filtering RS the main task is to predict value of R over pairs of users and items i.e. compute R'(u, i), where R' is the predicted rating computed by an RS.

Recommendation systems use a number of different technologies. We can classify these systems into following broad groups.

- Collaborative filtering
- Content based
- Demographic
- Knowledge-based
- Community based
- Hybrid recommender systems

All these techniques are briefly explained below.

2. 2 Collaborative Filtering

Collaborative filtering is the most popular and widely implemented technique in RS. It recommends to a user the items that other users with similar taste have liked in the past. The similarity between users is calculated on the basis of the similarity in the rating history of a user. This approach is sometimes referred as people to people correlation. In order to make recommendations, CF systems need to relate two fundamentally different entities: items and users. There are two primary approaches to facilitate such comparison, which constitute the two main techniques of CF: *the neighbourhood approach* and *latent factor models*.

- Neighbourhood methods focus on relationships between items or, between users. An item-item approach models the preference of a user to an item based on ratings of similar items by the same user.
- Latent factor models, such as matrix factorization (aka, SVD), comprise an alternative approach by transforming both items and users to the same latent factor space. The

latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback.

Robustness in recommender systems refers to the ability of a system to provide stable recommendations when its rating database is contaminated with some portion of noisy or attack profiles [4]. Although some robust collaborative recommendation algorithms have been proposed, they still have the following limitations:

- The robustness of the recommendation algorithms is relatively poor against shilling attacks, thus leading to great prediction shift.
- The robustness of the recommendation algorithms acquires at a cost of recommendation accuracy, which causes the poor quality of recommendation

Nearest neighbour based recommendation can be classified into 2 broad categories explained as below:

2.2.1 User-based Recommendation

User-based neighbourhood recommendation methods predict the rating r_0 of a user u for a new item I using the ratings given to i by users most similar to u, called nearest-neighbours. Suppose we have for each user v=u a value w_{uv} representing the preference similarity between u and v. The k-nearest-neighbours (k-NN) of u, denoted by N(u), are the k users v with the highest similarity w_{uv} to u. However, only the users who have rated item I can be used in the prediction of r_{ui}, and we instead consider the k users most similar to u that have rated i.

In Eq. 1 $N_i(u)$ is the set of neighbours. The rating r' can be estimated as the average rating given to i by these neighbours

$$r' = \frac{1}{|N_i(u)|} \sum_{v \varepsilon N_i} r_{vi} \tag{1}$$

2.2.2 Item-based Recommendation

While user-based methods rely on the opinion of like-minded users to predict a rating, item-based approaches look at ratings given to similar items. So instead of consulting with peer user determines whether an item is right for him by considering the items he has already seen and finding items similar to it.

2. 3 Content based systems

This method takes a personalized approach in recommending items and the items similar to those rated in the past are recommended to the user. Content based systems analyse the user history and builds a profile or model of user interest based on the features of the items rated/liked by the user. The recommendation process consists of matching the attribute of items with that of user's interest. If the profile accurately expresses user preferences then the prediction accuracy is high.

2. 4 Demographic based systems

In these types of system user's demographic attributes like age, gender, religion, education, language etc. are used. The recommendations can vary depending on the demographic attribute. For example the recommendation list can vary as per different age groups.

2. 5 Knowledge based systems

They recommend items based on the specific domain knowledge about how accurately the items features meets user's preferences and needs. They basically are concerned with the usefulness of the item to the user. Most of knowledge based systems are case based where a similarity function estimates how recommendation meets the user needs. Constraint based systems are also knowledge based systems. Major difference between case and constraint based system is that case based system determine recommendation on basis of similarity metrics whereas constraint based systems used predefined explicit rules.

2. 6 Community based systems

These types of systems recommends based on preferences of user friends. This type of system acquires user's social relations information and friend's preferences and then makes recommendations on basis of this information.

2. 7 Hybrid systems

These types of systems combine more than one type of above mentioned techniques. It uses the advantage of one technique to fix the disadvantage of other technique. For example collaborative filtering suffers from new item problem they cannot recommend item that do not have ratings , however content based systems do suffer from this problem as they can match the items attribute with user profile to recommend it or not. So normally these two are combined to produce a new hybrid system.

2. 8 Background and related work

Collaborative filtering (CF) methods produce user specific recommendations of items based on patterns of ratings or usage without any need of external information about items or users. Recommender systems rely on various types of input. Most convenient is high quality explicit feedback, where users directly report on their interest in products. A variety of methods have been proposed for developing robust recommendation algorithms. O'Mahony et al proposed an intelligent neighbourhood formation scheme by modeling the usefulness of user profiles using profile utility [6]. This method can improve the quality of selected neighbours, but its robustness is limited. Sandvig et al. [7] proposed a recommendation algorithm based on association rule mining. This algorithm can get better robustness, but the robustness is acquired at the cost of coverage.

In collaborative filtering recommender systems, let R be the user-item rating matrix, which includes m users and n items. Let U be the set of all users and I be the set of all items. The rating of user u on item i is r_{ui}. The predicted rating is represented by r'_{ui}. If user U has not rated item i, we represent that as r_{ui} =¢

Baseline predictors: CF models try to capture the interactions between users and items that produce the different rating values. However, much of the observed rating values are due to effects associated with either users or items, independently of their interaction. A principal example is that typical CF data exhibit large user and item biases – i.e., systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others.

These effects are encapsulated, which do not involve user-item interaction, within the baseline predictors (also known as biases).Because these predictors tend to capture much of the observed signal, it is vital to model them accurately. Such modelling enables isolating the part of the signal that truly represents user-item interaction, and subjecting it to more appropriate user preference models.

In Eq. 2, 3 and 4 we denote the overall average rating by μ. A baseline prediction for an unknown rating r_{ui} is denoted by b_{ui} and accounts for the user and item effects. The parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average.

$$b_{ui} = \mu + b_u + b_i \quad (2)$$

$$b_u = \frac{1}{|I|}\sum_{i=0}^{i=I}(r_{ui} - \mu) \quad (3)$$

$$b_i = \frac{1}{|U|}\sum_{u=0}^{u=U}(r_{ui} - b_u - \mu) \quad (4)$$

For example, suppose that we want a baseline predictor for the rating of the movie Titanic by user A. Now, say that the average rating over all movies, μ, is 3.7 stars. Furthermore, Titanic is better than an average movie, so it tends to be rated 0.5 stars above the average. On the other hand, A is a critical user, who tends to rate 0.3 stars lower than the average. Thus, the baseline predictor for Titanic's rating by A would be 3.9 stars by calculating 3.7−0.3+0.5.

Similarity measure: One of the preferred approaches to collaborative filtering (CF) recommenders is to use the kNN. This classification method–as most classifiers and clustering techniques–is highly dependent on defining an appropriate similarity or distance measure.

The simplest and most common example of a distance measure is the Euclidean distance

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (5)$$

In Eq. 5 n is the number of dimensions (attributes) and x_k and y_k are the k^{th} attributes (components) of data objects x and y, respectively.

Sampling: Sampling is the main technique used for selecting a subset of relevant data from a large data set. It is used both in the pre-processing and final data interpretation steps. Sampling may be used because processing the entire dataset is computationally too expensive. It can also be used to create training and testing datasets. In this case, the training data set is used to learn the parameters or configure the algorithms used in the analysis step, while the testing data set is used to evaluate the model or configuration obtained in the training phase, making sure that it performs well(i.e. generalizes)with previously unseen data.

It is common practice to use standard random sampling without replacement with an 80/20 proportion when separating the training and testing datasets. This means that we use random sampling without replacement to select 20% of the instances for the testing set and leave the remaining 80% for training. The 80/20 proportion should be taken as a rule of thumb; in general, any value over 2/3 for the training set is appropriate.

Nearest Neighbours: Instance based classifiers work by storing training records and using them to predict the class label of unseen cases. A trivial example is the so-called rote-learner. This classifier memorizes the entire training set and classifies only if the attributes of the new record match one of the training examples exactly. A more elaborate, and far more popular, instance-based classifier is the nearest neighbour classifier (kNN).

Given a point to be classified, the kNN classifier finds the k closest points (nearest neighbours) from the training records. It then assigns the class label according to the class labels of its nearest-neighbours. The underlying idea is that if a record falls in a particular neighbourhood where a class label is predominant it is because the record is likely to belong to that very same class.

Perhaps the most challenging issue in kNN is how to choose the value of k. If k is too small, the classifier will be sensitive to noise points. But if k is too large, the neighbourhood might include too many points from other classes.

Euclidean distance: For any user's $u_a \mathcal{E} U$ and $u_b \mathcal{E} U$ Euclidean distance between user u_a and u_b is given by Eq. 6. Whwre $I(u_a, u_b)$ be the co-rated item set of users u_a and u_b. $R(u_a)$ and $R(u_b)$ be the rating item set of users u_a and u_b, respectively.

$$\mathrm{dist}(u_a,u_b) = (|R(u_a)| + |R(u_b)|) \sqrt{\sum_{i=1}^{n}(r_{ai} - r_{bi})^2} \quad , R(u_a) \cap R(u_b) = 0 \qquad (6)$$

$$= \frac{(|R(u_a)| + |R(u_b)|)}{|I(u_a,u_b)|} \sqrt{\sum_{i=1}^{n}(r_{ai} - r_{bi})^2} \quad , R(u_a) \cap R(u_b) \neq 0$$

Nearest neighbour is one of the most common approaches to CF – and therefore to designing a RS. One of the advantages of this classifier is that it is conceptually very much related to the idea of CF: Finding like-minded users (or similar items) is essentially equivalent to finding neighbours for a given user or an item. The other advantage is that, being the kNN classifier a lazy learner, it does not require learning and maintaining a given model.

Therefore, in principle, the system can adapt to rapid changes in the user ratings matrix. Unfortunately, this comes at the cost of re computing the neighbourhoods and therefore the similarity matrix. This is why we proposed a neighbourhood model that uses a reduced set of experts as the source for selecting neighbours.

The main advantages of neighbourhood – based methods are

- Simplicity
- Efficiency
- Stability
- Justifiability

Table 1 sample rating matrix shows a typical user item rating matrix. It shows the ratings given by users to different items (movies in this case).

Now if user C has to decide whether or not to rent the movie "Titanic" that he has not yet seen. He knows that B has very similar tastes when it comes to movies, as both of them hated "The Matrix" and loved "Forrest Gump", so he asks her opinion on this movie. On the other hand, C finds out him and D have different tastes, D like's action movies while he does not, and he discards her opinion or considers the opposite in his decision.

	Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
A	5	1		2	2
B	1	5	2	5	5
C	2	?	3	5	4
D	4	3	5	3	

Table 1: Sample rating matrix

3. Group Modelling Strategies

It is necessary to first model a group in order to provide and manage information related to a group. A group is a composition of like-minded users that come together for a decision making. The individual user model is the first aspect to be considered before mapping with group model and after that a group model is made by combining preferences of individual users in a group.

By taking into account the opinion of likeminded users a collective choice is made. The individuals are aggregated using various strategies and after that the usefulness of the strategy is evaluated. The existing group modelling strategies are described below in brief along with reference to group recommender system using them. In example for these strategies 3 users (u1, u2, u3) are considered that rate 5 items (i1, i2, i3, i4, i5) on a ratings of scale from 1 to 5.

3.1 Additive utilitarian strategy

In this group modelling strategy the rating of an item for a particular group is predicted by taking the average of all the ratings given by all users in that group for a particular item. As per [15] taking average of ratings is the best way to aggregate the individual preferences and have been used in many recommender systems.

As can be seen in Table 2 for additive utilitarian matrix the user preferences are being given for 5 items. The group rating for a particular item is predicted by taking the sum of all predictions and dividing it by number of users. Also a ranked list of items can be prepared by arranging the group ratings in decreasing order which will same as would be in case average rating is considered. Top N items can also be recommended from the ranked list.

	I1	I2	I3	I4	I5
U1	5	1	3	2	5
U2	1	4	4	5	5
U3	3	1	5	5	5
Group	9	6	12	12	15

Table 2: Additive utilitarian matrix

3.2 Multiplicative utilitarian strategy

In this strategy for every item the ratings given by the user are multiplied and the group rating value is used to make an ordered list i.e. higher the product is earlier the item will appear in the list. This strategy has been used in various music recommender systems.

Table 3 shows the group rating assigned using this strategy. The given ratings of users are multiplied and assigned to that particular group.

	I1	I2	I3	I4	I5
U1	5	1	3	2	5
U2	1	4	4	5	5
U3	3	1	5	5	5
Group	15	4	60	50	125

Table 3: Multiplicative utilitarian matrix

3.3 Borda count

In this strategy each item gets point according to its position in the list of each user. The least favourite get 0 and a point is added each time next item is added into the list. If two items have same ratings then the points are distributed equally among them. After this the group preference is generated by adding up all the points for a particular item. Then a ranked list is prepared in descending order of rankings.

Table 4 Borda count matrix shows the group rating assigned using this strategy. For user u1 items I2 gets 0 as it is the least favourite item, next item is I4 and I2 so it gets 1 and 2 respectively. Now item I1 and I5 have same ratings so they are assigned (3+4)/2 = 3.5 i.e. the mean of next possible values.

	I1	I2	I3	I4	I5
U1	3.5	0	2	1	3.5
U2	0	1.5	1.5	3.5	3.5
U3	1	0	3	3	3
Group	4.5	1.5	6.5	7.5	9.5

Table 4: Borda count matrix

3.4 Copeland rule

In this strategy the items are sorted according to their Copeland index that is calculated as number of times one alternative beats other minus number of times it loses against other alternatives.

3.5 Plurality voting strategy

In this strategy all the users votes for their favourite options. The item which receives highest number of votes wins. The item which gets selected maximum number of times is recommended to that particular group.

Table 5: Plurality rating matrix shows that item I5 is the most selected item so this item is recommended to the group i.e. item 5 as it is most favourite of all.

User	Item
U1	i1,i5
U2	i4,i5
U3	i3,i4,i5
Group	i5

Table 5: Plurality voting strategy matrix

3.6 Approval Voting

A user votes for all the items that are above certain threshold i.e. 1 or 2 and all other items with ratings below threshold are ignored. After that the group preference is obtained by adding the individual points given to an item.

Table 6 Approval voting matrix shows the group rating predicted using this strategy. Threshold 2 is considered so for all items rated less than 2 are given 0 and others are given 1.

	I1	I2	I3	I4	I5
U1	1	0	1	0	1
U2	0	1	1	1	1
U3	1	0	1	1	1
Group	2	1	3	2	3

Table 6: Approval voting matrix

3.7 Least Misery strategy

The rating assigned to an item for a group is the lowest rating given to that item by the members of that group. If the minimum rating is 1 then 1 is assigned to that group for that particular item.

Table 7 least misery matrix shows the ratings assigned to the items using this strategy. For every item the least given rating is given to the group.

	I1	I2	I3	I4	I5
U1	5	1	3	2	5
U2	1	4	4	5	5
U3	3	1	5	5	5
Group	1	1	3	2	5

Table 7: Least misery matrix

3.8 Most Pleasure strategy

The rating assigned to an item for a group is the maximum rating given to that item by the members of that group. If the maximum rating is 4 then 4 is assigned to that group for that particular item.

Table 8 most pleasure matrix shows the ratings assigned to the items using this strategy. For every item the maximum given rating is assigned to the group.

	I1	I2	I3	I4	I5
U1	3	1	3	2	5
U2	1	4	4	5	5
U3	3	1	5	5	5
Group	3	4	5	5	5

Table 8: Most Pleasure strategy

3.9 Average without misery

In this strategy the rating assigned to an item in a group is average of all the ratings given by the users in that group, but all the ratings below a particular threshold are not considered.

Table 9 Average without misery matrix shows the ratings assigned with threshold 2. For a group the items having rating below this threshold are not considered.

	I1	I2	I3	I4	I5
U1	5	1	3	2	5
U2	1	4	4	5	5
U3	3	1	5	5	5
Group	-	-	12	12	15

Table 9: Average without misery matrix

3.10 Fairness strategy

The idea in this strategy is that users can be recommended items they do not like as long they are being recommended items that they like. The highest rated item for each user is selected if two items have same ratings then it is decided by other users preference. This process is continued until everyone has made their choice. In next round the user who selected last gets the first turn to pick his/her favourite. Continuing in this way the rating for an item in a group is calculated.

3.11 Most respected person strategy

This strategy takes into consideration the influence on users from other users in the group. More often it happens user choice is influenced by others. In this strategy from a group a most respected user is taken and the ratings assigned to the items in that group are same as that of this user. This is a form of dictatorship in which only one person decides the group behaviour.

4. Implementation

In this section the details of implementation and the tasks performed by the system are discussed. C++ language was used to implement this group recommender system. The tasks performed by the system are explained in details below.

4.1 Predicting missing ratings

As there are many users and the items the initial rating matrix is very sparse as users have rated only very few items initially. The missing ratings are predicted with a user based nearest neighbour collaborative filtering algorithm for all users. For all the items that were not rated by the user u a rating p_{ui} is predicted by considering n nearest neighbours who are similar to user u.

Eq. 7 gives the formula used to predict the ratings for user u.

$$p_{ui} = \overline{r_u} + \frac{\sum_{neighbours(u)} userSim(u,n) * (r_{ni} - r_n)}{\sum_{neighbours(u)} userSim(u,n)} \tag{7}$$

r_u and r_n are the average rating of user u and n. userSim(u, n) is calculated using pearson correlation is used to calculate the user similarity between two user.

Pearson Correlation: A popular measure that compares ratings where the effects of mean and variance have been removed is the Pearson Correlation (PC) similarity. The sign of a similarity weight indicates whether the correlation is direct or inverse, its magnitude (ranging from -1 to 1) represents the strength of the correlation. -1 indicates that similarity is minimum, 1 indicates that similarity is maximum, 0 indicates that two users have no common interest.

$$\text{PC}(u, v) = \frac{\sum_{i=0}^{I}(r_{ui} - r'_u)(r_{vi} - r'_v)}{\sqrt{\sum_{i=0}^{I}(r_{ui} - r'_u)^2}\sqrt{\sum_{i=0}^{I}(r_{vi} - r'_v)^2}} \tag{8}$$

4.2 Group detection

The set for all the users' needs to be partitioned into groups but as groups do not exist by default unsupervised clustering is done to detect the groups. As initially the rating matrix is very sparse it affects the performance of group recommender system more specifically of clustering. So in order to improve the system accuracy the individual predictions made are also included into the rating matrix for clustering, thereby negating the effect of data sparseness.

A variant of k-means clustering algorithm called EZ Hybrid was used to cluster the users based on the user ratings [3]. In k-means clustering we are given n data points in d dimensions and an integer k i.e. number of data points. A k-means algorithm can get stuck in locally minimal solution thus not providing optimal results. So to improve on this a heuristic based local search is considered in which the data points are swapped in and out of existing solutions. If the swap decreases the average distortion then it is considered else ignored. The EZ hybrid algorithm does one swap followed by some iteration of k-means algorithm.

4.3 Group modelling

The existing strategies additive utilitarian, approval voting, least misery and most pleasure present in literature were considered and implemented [16]. There are certain strategies that do produce explicit ratings and only produce a ranked list of items for a group e.g. plurality voting, Copeland rule, fairness strategy and in most respected person strategy only one user derive group behaviour so it is not applicable for most scenarios. In multiplicative utilitarian ratings are produced by multiplying all ratings of item the product cannot be stored as it becomes very large given that there are 3952 items.

So for this project Additive Utilitarian, Approval voting, least misery and most pleasure strategy were implemented. Also as the user ratings are given on scale of 1 to 5 the ratings produced by the modelling strategies are on completely different scale of representation. So in order to measure the prediction accuracy we need to bring them down to same scale. Formula mentioned in equation (9) does this job.

$$\text{new\_rating} = \frac{group\ rating * \max\_user\_rating}{\max\_group\_rating} \tag{9}$$

- *group_rating* is the rating produced by modelling strategy
- *max_group_rating* is maximum value of group_rating that can be obtained for an item
- *max_user_rating* is the maximum rating that can be given by an user to an item

4.4 Proposed changes

A new group modelling strategy using median is proposed for predicting the ratings for an item for a particular group. The strategy is described briefly below with the flowchart on next page.

Median strategy: After the group formation is completed using k-means clustering algorithm the ratings of users belonging to a group are arranged in increasing order and the median of the group is calculated. If there are n items in a group then if n is even then the predicted rating is $(r_{n/2} + r_{n/2+1})$ / 2 and if n is odd then the predicted rating is $r_{n/2+1}$. The median value is thus assigned as the group rating to that item.

The stepwise implementation steps of the group recommendation process are given in form of flowchart on next page in Fig 1 and explained in brief below.

Step 1: The rating matrix is initialized from the 1M Movielens dataset consisting of 6040 users and 3952 items.

Step 2: The similarity between all users is calculated using Pearson correlation. Nearest neighbour value k = 100

Step 3: The missing ratings for the items not rated by the users are predicted using the formula mentioned in Eq. 7

Step 4: Using EZ Hybrid algorithm the users are clustered into 1, 20, 50 and 200 groups.

Step 5: Using the modelling strategies mentioned in chapter 3 the group ratings are predicted for every item.

Step 6: Root mean square error, Mean absolute error, Precision and Recall are calculated for the predicted group ratings and the performance of the proposed strategy is compared with that of others present in the literature.

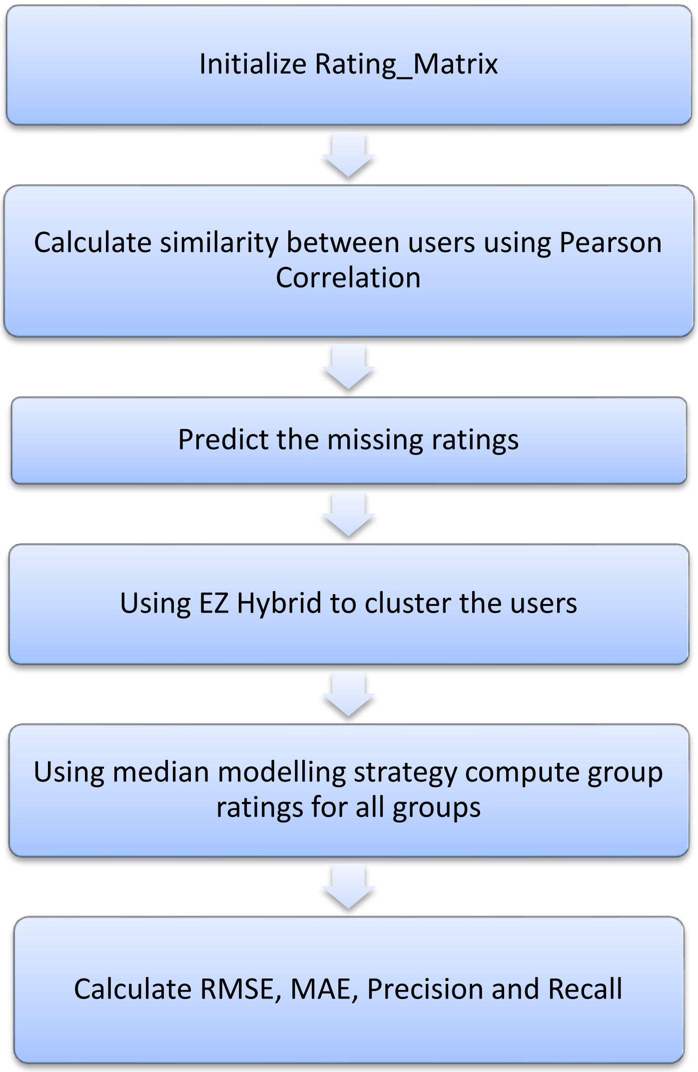

Fig 1: Median strategy implementation steps

5. Experiment and *Results*

Datasets: In order to evaluate the performance of the algorithm, we select the 1M dataset provided by MovieLens site (http://movielens.umn.edu/) as the experimental data, which contains 1000209 ratings on 3952 movies by 6040 users. In the experiment ratings.dat was used which contains user IDs in range 1 to 6040, Movie ID in range of 0 to 3952. All ratings are integer values between 1 and 5, where 1 is the lowest (disliked) and 5 is the highest (most liked). For the number of items which one user has rated in the dataset, the minimum is 20. K i.e. nearest neighbour is taken as 100.

5.1 Evaluation metrics

5.1.1 Root Mean Squared Error (RMSE)

The Root mean square error (RMSE) was used to measure the performance of algorithm. RMSE represents the standard sample deviation of difference of predicted and observed values. It compares the rating r_{ui} given by the user with the predicted rating of the item for group to which the user belongs.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui}-p_{gi})^2}{n}} \qquad \text{Eq. (10)}$$

Eq. 10 shows the formula being used, where r_{ui} is the actual rating given by user u to item i, and p_{gi} is the rating predicted for item I for group g to which the user belongs and n is the total number of the ratings available. A low value of RMSE signifies higher accuracy.

5.1.2 Mean Absolute Error (MAE)

The mean absolute error (MAE) is used to measure the performance of algorithm. MAE is a metric used in evaluating the accuracy of algorithms and measures the average of absolute errors.

$$\text{MAE} = \frac{\sum |r_{ui} - r'_{ui}|}{n} \qquad \text{Eq. (11)}$$

Eq. 11 shows the formula used for MAE where r_{ui} is the real rating of user U on item i, r'_{ui} is the predicted rating of user U on item i, n is the total number of predictions. Lower the MAE higher is prediction accuracy

5.1.3 Precision

Precision also called positive predicted value measures the fraction of retrieved instances that are relevant. Precision is generally represented in %age and higher value signifies more accuracy.

$$\text{Precision} = \frac{number\ of\ correct\ predictions}{total\ predictions + number\ of\ wrong\ predictions} \qquad \text{Eq. (12)}$$

5.1.4 Recall

Recall measures how accurately we are able to predict ratings. Recall is generally represented in %age and higher value signifies more accuracy.

$$\text{Recall} = \frac{number\ of\ correct\ predictions}{total\ predictions} \qquad \text{Eq. (13)}$$

5.2 Experimental Results

Below given figures contains the RMSE, MAE, Precision and Recall values obtained in the experiment for different modelling strategies, for different group number.

Results show that with the proposed Median strategy we get the best MAE, Precision and Recall and RMSE is also very close to that of additive utilitarian strategy and outperform other strategy.

5.2.1 RMSE values

Table 10 shows the RMSE values obtained during for 1, 20, 50, 200 and 500 groups. The results show that additive utilitarian performs the best in this case. However proposed Median strategy performs better than other strategies. On an average RMSE for additive utilitarian is 0.87590, for median is 0.93541, for approval voting (1) is 1.33112, for approval voting (2) is 1.13363, for least misery is 2.15606 and for most pleasure 1.52262

EZ-Hybrid	***1 group***	***20 groups***	***50 groups***	***200 groups***	***500 groups***
Additive Utilitarian	0.97452	0.94509	0.91558	0.8312	0.7131
Median	1.02495	1.0021	0.97662	0.8903	0.7831
Approval Voting[1]	1.13564	1.24273	1.31833	1.4497	1.5092
Approval Voting[2]	1.04647	1.07286	1.10632	1.1871	1.2554
Least Misery	2.80512	2.53702	2.31094	1.7784	1.3488

Table 10: RMSE values

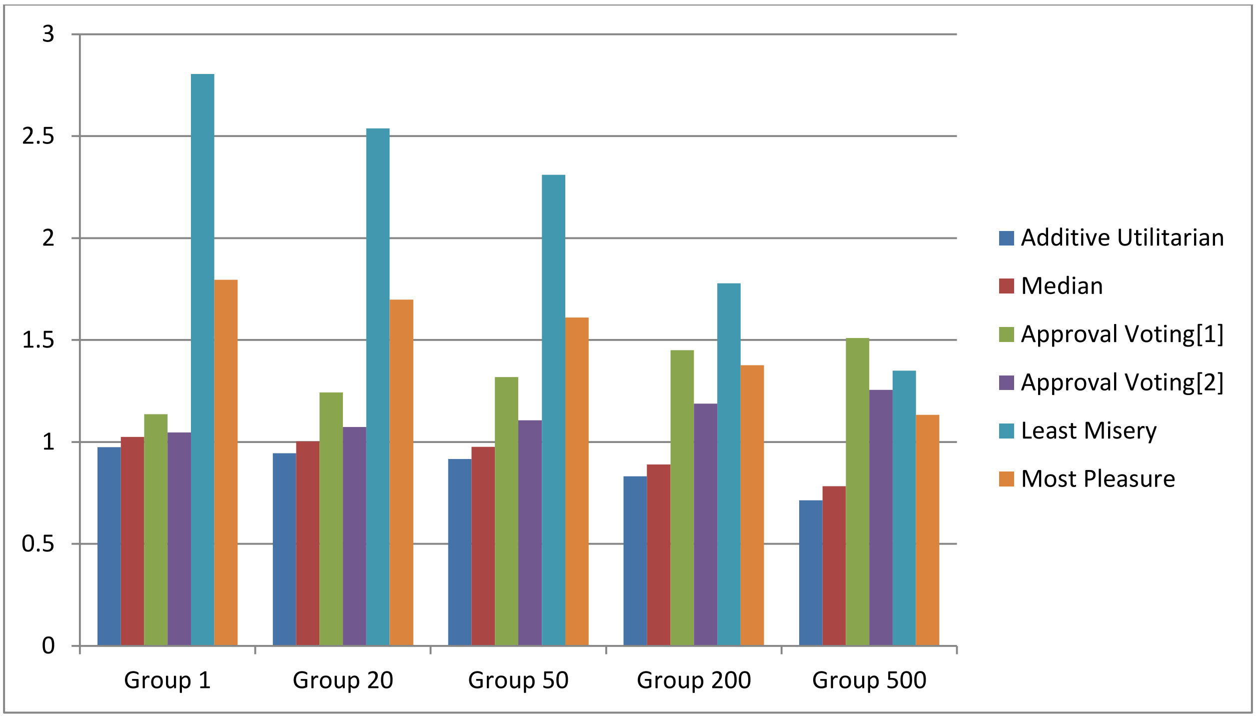

Fig 2: RMSE chart

5.2.2 MAE values

Table 11 shows the MAE values obtained during for 1, 20, 50, 200 and 500 groups. The results show that median strategy out performs all others. On an average MAE for median shows improvement of 6.67% over additive utilitarian which was the best performing among existing group modelling strategies. On an average MAE for additive utilitarian is 0.6691, for median is 0.6245, for approval voting (1) is 1.1628, for approval voting (2) is 0.8324, for least misery is 1.8822 and for most pleasure 1.1511.

EZ-Hybrid	***1 group***	***20 groups***	***50 groups***	***200 groups***	***500 groups***
Additive Utilitarian	0.778773	0.74993	0.71716	0.61132	0.4888
Median	*0.738301*	*0.70486*	*0.66982*	*0.56356*	*0.4464*
Approval Voting[1]	0.832517	0.92208	0.99226	1.12708	1.9401
Approval Voting[2]	0.766154	0.79167	0.82447	0.79171	0.9882
Least Misery	2.571811	2.2318	1.94401	1.81721	0.8465

Table 11: MAE values

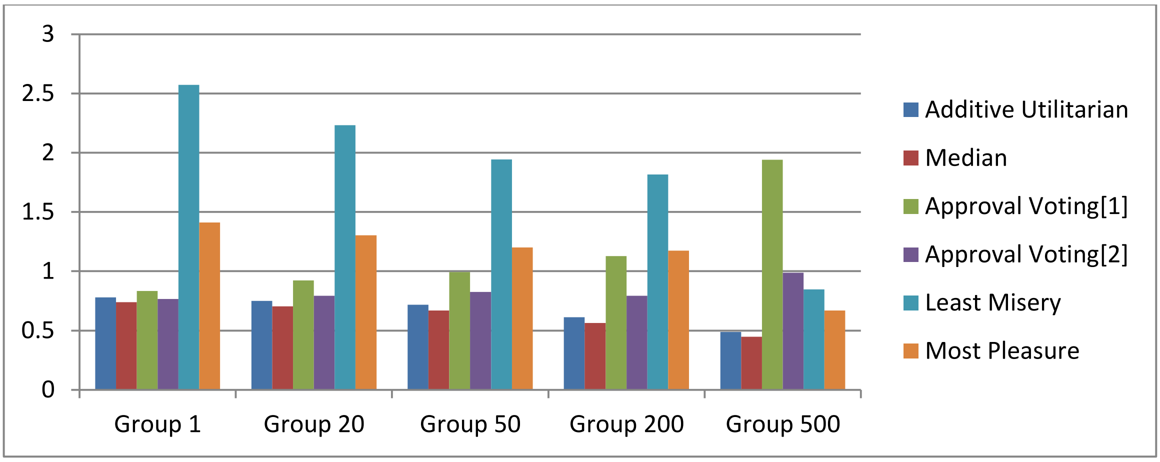

Fig 3: MAE chart

5.2.3 Precision values

Table 12 shows the precision values obtained during for 1, 20, 50, 200 and 500 groups. The results show that median strategy out performs all others. On an average Precision for median shows improvement of 7.77% over additive utilitarian strategy which was the best performing among existing group modelling strategies.

EZ-Hybrid	*1 group*	*20 groups*	*50 groups*	*200 groups*	*500 groups*
Additive Utilitarian	18%	19.41%	21.24%	27.94%	37.04%
Median	24.90%	27.08%	29.41%	36.74%	45.72%
Approval Voting[1]	21.67%	20.11%	18.71%	15.86%	14.35%
Approval Voting[2]	23.12%	22.44%	21.57%	19.29%	17.10%
Least Misery	2.94%	5.81%	9.36%	20.84%	34.90%

Table 12: Precision values

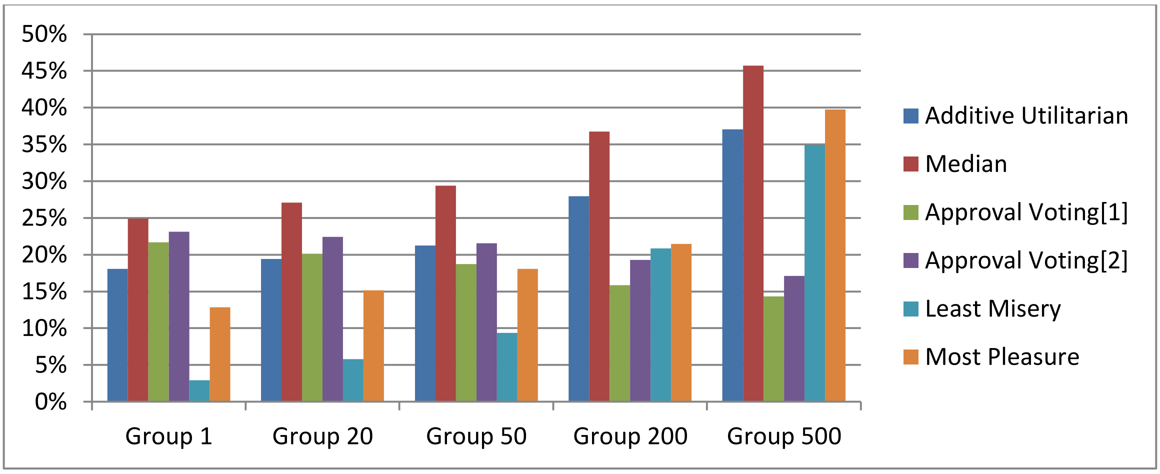

Fig 4: Precision chart

5.2.4 Recall values

Table 13 shows the recall values obtained during for 1, 20, 50, 200 and 500 groups. The results show that median strategy out performs all others. On an average recall for median strategy shows improvement of 9.69% over additive utilitarian strategy which was the best performing among existing group modelling strategies.

EZ-Hybrid	*1 group*	*20 groups*	*50 groups*	*200 groups*	*500 groups*
Additive Utilitarian	30.65%	32.52%	35.04%	43.68%	54.06%
Median	39.86%	42.62%	45.45%	53.73%	62.75%
Approval Voting[1]	35.62%	33.49%	31.52%	27.39%	25.10%
Approval Voting[2]	37.56%	36.66%	35.49%	32.34%	29.20%
Least Misery	5.71%	10.98%	17.12%	34.49%	51.76%

Table 13: Recall values

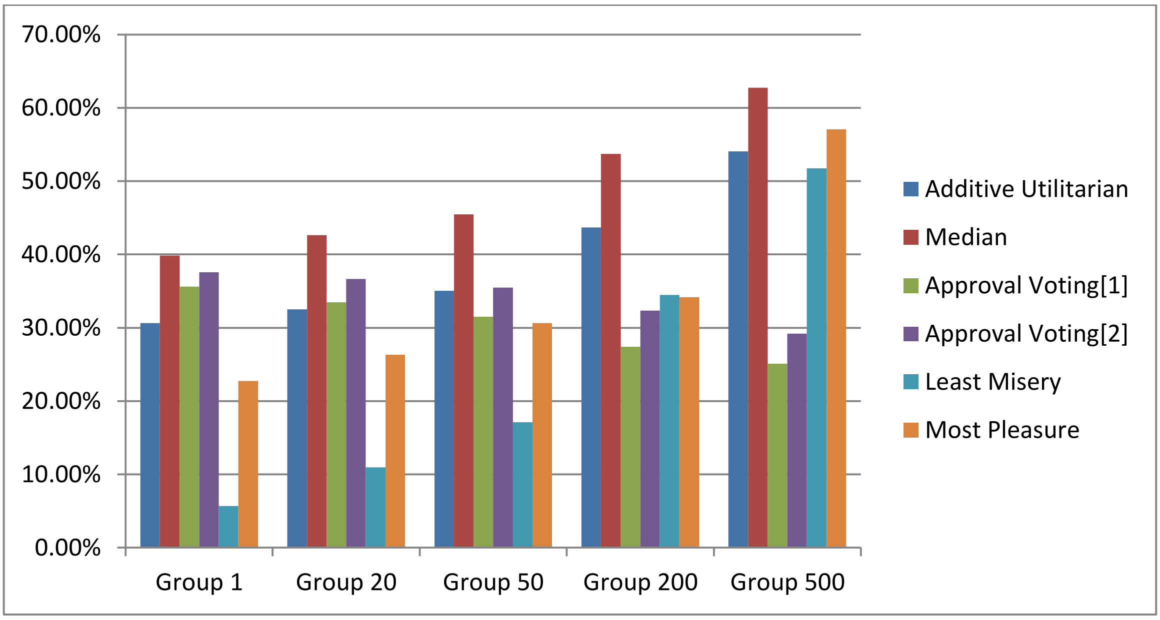

Fig 5: Recall chart

6. Conclusion and Future work

In this project the performance of group recommender system was measured using various modelling strategies. Depending on the modelling strategy the performance of the system varies. Also a new modelling strategy was proposed and its performance was measured and found better than other strategies. MAE is improved by 6.67%, Precision by 7.77% and Recall by 9.69%. Also the RMSE values are better than all other strategies except additive utilitarian. Hence we conclude that using median strategy helps in making more accurate predictions.

In future work more modelling strategies could be worked out and existing ones could be further improved by considering the link with user demography, item category etc. Also the accuracy of the system can be improved by using various other clustering algorithms. Furthermore the clustering accuracy can be enhanced by more accurate predictions of user ratings.

6. References

[1] L.Boratto, et al., Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modelling and clustering. Future Generations Computer Systems (2015)

[2] L Boratto, S. Carta, Using collaborative filtering to overcome the curse of dimensionality when clustering users in a group recommender system. Proceedings of 16^{th} International Conference on Enterprise Information Systems

[3] T.Kanungo, D.M, Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Anal. Mach, Intell.

[4] A Hybrid Trust Based Recommender system for online communities of Practise. Xiao-Lin Zheng, Chao Chao Chen IEEE transactions on learning technologies [2015]

[5] Burke, R.: Hybrid web recommender systems. In: The Adaptive Web, pp. 377–408.Springer Berlin/ Heidelberg (2007)

[6] YI Huawei, Zhang Fuzhi, Lan Jei: Robust collaborative recommendation algorithm based on k-distance and Tukey M-estimator [2014]

[7] GUNES I, KALELI C, BILGE A, et al. Shilling Attacks against Recommender Systems: A Comprehensive Survey [J]. Artificial Intelligence Re view, 2012: 1-33.

[8] O'MAHONY M, HURLEY N, KUSHMERICK N, et al. Collaborative Recommendation: A Robustness Analysis [J]. ACM Transactions on Internet Technology (TOIT), 2004, 4(4): 344-377.

[9] BURKE R, O'MAHONY M P, HURLEY N J. Robust Collaborative Recommendation [M]. Recommender Systems Handbook, Springer US, 2011: 805-835

[10] O'MAHONY M P, HURLEY N J, SILVESTRE G C M. Efficient and Secure Collaborative Filtering Through Intelligent Neighbor Selection[C] 11 Proceedings of the 16th European Conference on Artificial Intelligence. Valencia, Spain: lOS Press, 2004: 383-387

[11] SANDVIG J J, MOBASHER B, Burke R. Robust ness of Collaborative Recommendation Based On Association Rule Mining[C]11 Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07). Minneapolis, Minnesota, USA: ACM Press, 2007: 105-111

[12] Meng jeng et al., Social Recommendation with cross-domain transferrable knowledge, IEEE transactions on knowledge and data engineering 2015.

[13] Ilham Esslimani, Armelle Brun, Anne Boye, A collaborative filtering approach combining clustering and navigational based correlations

[14] S.Pizzutilo et.al, Group modelling in a public space: Methods, techniques and experiences, in: Proceedings of WSEAS AIC 05, ACM, Malta 2005.

[15] T. Pessemier, et al. Comparison of group recommendation algorithms, Multimedia Tool Appl (2013)

[16] J.Mastoff, Group recommender systems: Combining individual models, Recommender systems handbook, Springer, 2011

[17] Francesco Ricci, Lior Rokach and Bracha Shapira "Introduction to recommender systems handbook"

[18] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro "Content based Recommender Systems: State of the Art and Trends"

[19] Badrul M. Sarwar et al "Recommender Systems for Large-scale E-Commerce: Scalable Neighbourhood Formation Using Clustering"

[20] Lyle H. Ungar et al "Clustering Methods for Collaborative Filtering"