

A MULTI-PERSPECTIVE EVALUATION OF MA AND GA FOR COLLABORATIVE FILTERING RECOMMENDER SYSTEM

Hema Banati¹ and Shikha Mehta²

¹Dyal Singh College, ^{1,2}Department of Computer Science,

^{1,2}University of Delhi, Delhi

¹banatihema@hotmail.com, ²mehtshikha@gmail.com

ABSTRACT

The rising popularity of evolutionary algorithms to solve complex problems has inspired researchers to explore their utility in recommender systems. Recommender systems are intelligent web applications which generate recommendations keeping in view the user's stated and unstated requirements. Evolutionary approaches like Genetic and memetic algorithms have been considered as one of the most successful approaches for combinatorial optimization. Memetic Algorithms (MAs) are enhanced genetic algorithms which incorporate local search in the evolutionary scheme. Local Search process on each solution after every generation helps in improving the convergence time of MA. This paper presents multi-perspective comparative evaluation of memetic and genetic evolutionary algorithms for model based collaborative filtering recommender system. Experimental study was conducted on MovieLens dataset to investigate the decision support and statistical efficiency of Memetic and genetic algorithms. Algorithms were analyzed from different perspectives like variation in number of clusters, effect of increasing the number of users, varying number of recommendations and using either one or more than one cluster for computing ratings of the unrated items. Results obtained demonstrated that from all perspectives memetic collaborative filtering algorithm has better predictive accuracy as compared genetic collaborative filtering algorithm.

KEYWORDS

Memetic collaborative filtering, Genetic collaborative filtering, Memetic recommender systems, Genetic recommender system, Evolutionary collaborative filtering.

1. INTRODUCTION

The explosion of information on World Wide Web has led to the development of intelligent web applications. These applications take into consideration heterogeneity of sources, user interests and mobility to intelligently utilize vast, rich and shared web resources and services available on the web. Examples of such web applications include Web Personalization, Web Mining, Semantic Web and Intelligent Agents etc. However in recent years, adaptive/personalized web applications popularly known as recommender systems have become more trendy. These systems try to identify the traits of the active user and then provide suggestions based on either the users personal past history or on the basis of opinions of friends, relatives and peers etc to facilitate their decisions. Among the various techniques used to develop the recommender systems, collaborative filtering, content based filtering and hybrid techniques are more common [1]. Other filtering techniques include Demographic filtering which uses demographic information like age, gender, qualification, etc. of the users to make recommendations. Utility based recommender systems suggest items based on their usefulness for the users. Knowledge based recommender systems take advantage of the knowledge about how a particular item meets a particular user needs, to make recommendations. Recently, several authors have proposed mixed approaches to

overcome the disadvantages of using any single method. These systems are called Hybrid recommender systems [17] [56]. All these approaches differ in the way how user profile and information products are described and compared to generate recommendations, like in content-based filtering; items are recommended based on the past preferences made by the user. On the other hand, collaborative filtering (CF) [47] technique primarily identifies the community of the active user by comparing the profile of active user with other users and then recommends items based on the community preferences. Generally profiles in collaborative filtering are represented by the explicit/implicit ratings made by the users on the various items of interest. Explicit rating/voting refers to a preference expressed by the user for the product, usually on a discrete numerical scale. For example the GroupLens system uses a scale of one to five for users to rate the Netnews articles. Users explicitly rate each article after reading it. Implicit rating refers to interpretation of user behavior or selections to input a vote or preference based on web browsing data or purchase history etc. Collaborative filtering does not exploit any information about the features of the products, thus it has been successfully applied to extensive range of applications [18]. Research in collaborative filtering turned vigorous with the development of Tapestry system [15] that was based on pull-based collaborative filtering approach. In such systems, users willing to have recommendations had to proactively pull out the recommendations from the database. With the immense success of this system, other researchers started contributing which led to the development of push based collaborative filtering system [33]. In contrast to pull based systems, users of these systems had the privilege to push the items of their interests to their friends and peers. With the great acceptance of semi-automatic systems, automatic collaborative filtering systems came into being like GroupLens system [30][45] developed in the domain of Usenet newsgroup articles, Bellcore's Video Recommender [20] for recommending movies and Ringo system [48] for music etc. Other examples include Amazon.com for books, Jester system [16] for recommending jokes and PHOAKS systems [49] to help users find pertinent information on the web. The anticipated statistics of varied recommender systems developed so far in both academia and industry across diverse application domains [43] are around 21 for recommending the Web, 13 for movies, 11 for news, 10 for Document and Information, 6 for music, 5 for information filtering and sharing etc. Other domains like travel, e-commerce, interesting changes on the web etc have 1-3 recommender systems. However these counts are too small as compared to the size of WWW. Several applications areas like Health, Education, and Geography etc. are still left to be explored.

In literature [7], collaborative filtering techniques are explored from two perspectives- Model-based CF and Memory-based CF. Memory-based CF systems use the whole user-item rating dataset to make predictions. These systems employ statistical techniques to identify a set of users known as neighbors whose past behavior was similar to the target user. Neighbors' preferences are then combined to compute recommendations for the target user. In contrast, Model-based CF systems split the complete dataset into source and target dataset. Source dataset is utilized to train the model using various machine learning algorithms like Bayesian network [24], clustering algorithms [29] etc. Thereafter the trained model is used to generate recommendations for the active user in the target dataset. Grouping of users into clusters helps in improving the accuracy of recommender systems, thus clustering algorithms have become more prevalent for creating the communities/classes of users who made similar purchases. The state of art clustering algorithms like K-means are though simple and take less time in clustering large datasets, they have good chance of getting trapped in the local optima. Moreover these techniques show inconsistent performance [51], either because data are often highly sparse or because people often have tastes which put them in multiple categories. Since these conventional clustering methods are ad hoc and the improvements are vague, one cannot simply take benefit of all the accessible knowledge.

This motivated the research community to explore the utility of biologically inspired algorithms for global optimization of recommender systems performance. These algorithms have been successfully tested and applied in solving various combinatorial optimization problems. Biologically inspired computing techniques are stochastic search methods that mimic the metaphor of natural biological evolution to provide solutions to a variety of intricate problems. The most widespread computing techniques [12] [26] in this domain include the neural networks stirred from the functioning of the brain, the evolutionary algorithms (EAs) like genetic algorithms (GA) enthused by natural evolution, memetic algorithms (MA) integrated GA with local search, the ant colony optimization (ACO) inspired by the collective intelligence of social insects, particle swarm optimization (PSO) inspired by the social behavior of a flock of migrating birds and shuffled frog leaping algorithms (SFLA) that combine the benefits of both the genetic-based MAs and the social behavior-based PSO algorithms. Among these ACO, PSO and SFLA are swarm intelligence techniques whereas GA and MA are inspired by the notion of evolution. The present study is focused to investigate the relative contribution of evolutionary techniques like memetic and genetic algorithms towards recommender systems. Genetic algorithms (GAs) are a population-based Meta heuristics invented by John Holland in the 1960s [11]. They are search techniques used in computer science to find approximate solutions to optimization and search problems. These algorithms use operators such as inheritance, mutation, selection, and crossover etc inspired by evolutionary biology [37]. Genetic algorithms are normally implemented as a computer simulation in which evolution starts from a population of completely random individuals. In each generation, the fitness of the whole population is evaluated. Subsequently multiple individuals are stochastically selected from the current population based on their fitness, and modified (mutated or recombined) to form a new population. The new population is then used in the next iteration of the algorithm. This process of evolution generates successively better solutions from previous generations of solutions. The contribution of genetic algorithms towards information filtering applications has been studied in various scenarios like the application of GA K-means [27] to an online shopping market segmentation proved that GA K-means clustering performed better than K-means clustering and self-organizing maps (SOM). In another application Genetic algorithm [28] has been used to group users based on products categorized by Naive Bayes classifier. Consequently recommendations were made to the user on the basis of grouped user preferences and information of categorized items. The Genetic algorithm based approach [32] to determine the weight value of each feature of a customer has been found to be effective in recommendation systems. The efficiency of genetic clustering algorithm [57] has been discussed in tackling scalability problem of collaborative filtering algorithms. However, conventional genetic algorithms experience the limitation of early convergence [14]. In contrast memetic algorithms reduce the likelihood of the premature convergence occurring in genetic algorithms. MAs are population based evolutionary algorithms (EAs) inspired by Richard Dawkin's notion of a meme [9][38]. They are hybrid EAs that use local search techniques to accomplish exploitation and global search method to perform exploration. They are motivated by the models of adaptation in natural systems that unite evolutionary adaptation of population of individuals with individuals learning within a lifetime [35]. Memetic algorithms are an extension of the traditional genetic algorithms. They are similar to Genetic Algorithms (GA) but the unit of cultural evolution is meme in place of genes. A meme has been defined to be 'a noun that communicates the idea of a unit of cultural transmission, or a unit of imitation'. Memes are analogous to genes as they are self replicating, but they differ from genes as they are transmitted through imitation rather than being inherited. Examples of memes include stories, fashion and technology etc. MAs are different from GAs as they incorporate local search process to refine the solutions before they get involved in the evolutionary process. The worthiness of MA has already been tested in variety of applications like machine cell formation [39], portfolio optimization [3],

downlink spatial access division multiple access optimization [21], remote sensing image segmentation [23] and protein assignment problem [53] etc. This paper presents multi-perspective comparative evaluation of memetic algorithm with genetic algorithm for collaborative filtering recommender systems. Section 2 presents the methodology or algorithm followed by experiments and results in Section 3. Section 4 presents conclusion and future work.

2. METHODOLOGY

To compare the relative performance of Memetic and Genetic Algorithms, model based collaborative filtering recommender systems were developed respectively. User based clustering models were build using memetic and genetic clustering algorithms for respective collaborative filtering systems. Initially the data was maintained in the database in the form of user-item rating matrix as shown in Fig 1. Consequently, data was divided into Source/training ($s(i,j)$) dataset and test/target ($t(i,j)$) dataset. The training dataset was used offline to develop models using clustering algorithms whereas test dataset was utilized to measure the performance of the trained model.

uid	mid	rating
1	76	4
1	72	4
1	185	4
1	96	5
1	213	2
1	233	2
1	258	5
1	81	5
1	78	1
1	212	4
1	143	1
1	151	4
1	51	4
1	175	5
1	107	4
1	218	3
1	209	4
1	259	1
1	108	5
1	262	3

Fig 1: Snapshot of User-Item rating dataset

2.1 CLUSTERING ALGORITHM

Clustering is defined as a task which determines a finite set of categories (clusters) to describe a dataset according to similarities among its objects [13] [25]. The main objective of clustering is to maximize the homogeneity within each cluster and the heterogeneity among different clusters that is the objects that belong to the same cluster should be more analogous to each other than objects that belong to different clusters[2] [10]. The similarity among the clusters may be measured either directly using the similarity functions like cosine similarity, adjusted cosine similarity etc or indirectly using the distance measures like euclidean distance for quantifying the degree of dissimilarity among objects, in such a way that more similar objects have lower dissimilarity values [22]. The worthiness of Genetic Algorithm based clustering has been realized in various

application scenarios like production simulation.[41], microarray data analysis[42], clustering small regions in colors feature space[52], image compression problem[34], document clustering[8], text clustering.[55], mobile ad hoc networks[50] and gene ontology[44] etc. The efficacy of Memetic clustering algorithm has been proved in the variety of domains like clustering of gene expression profiles [36], traveling salesman problem [54] and for collaborative filtering recommender system [4]. In the present work memetic and genetic clustering algorithms were implemented for respective collaborative filtering as follows:

2.1.1 Initially K genes/memes were selected at random for every chromosome in the population P to create the clusters. In each chromosome, K memes or K genes represented the centroids of user based K clusters. Subsequently all other users were assigned to the meme/gene with maximum similarity to create the cluster. The similarity (Sim (i, Cent)) of the user i with the centroid of the cluster was computed using adjusted cosine similarity function [48] given in equation (1).

$$Sim(i,j) = \frac{\sum_{n=1}^{nomovies} (U_{i,n} - \overline{o(n)})(U_{j,n} - \overline{o(n)})}{\sqrt{\sum_{n=1}^{nomovies} (U_{i,n} - \overline{o(n)})^2} \sqrt{\sum_{n=1}^{nomovies} (U_{j,n} - \overline{o(n)})^2}} \quad (1)$$

where U_i and U_j are the vectors containing ratings given by user i and j for movie n. $\overline{o(n)}$ refers to the average of the ratings given by all users on the nth movie. The centroids and the users belonging to the respective clusters for a chromosome of size $K=20$ are shown in Fig 2.

id	Centroid	users
1	472	54,58,69,71,122,126,127,133,136,137,204,247,314,333,359,378,381,383,397,408,425,446,472,498,531,
2	302	24,52,55,107,128,230,231,302,304,313,331,344,350,353,362,371,373,376,430,467,480,481,486,487,494,538
3	177	3,32,42,50,64,75,77,93,95,98,102,109,171,177,199,217,220,246,265,268,306,316,321,324,325,334,340,351,
4	295	295,236,363,339,285,189,308,305,141,530,73,556,5,252,185,219,208,192,515,240,343,539,108,194,193,85,
5	493	2,65,80,116,125,153,197,234,262,272,287,364,365,457,492,493,552,522,
6	36	357,22,36,40,59,76,88,92,138,156,159,187,225,228,237,238,253,261,266,271,327,337,423,444,452,482,483
7	448	260,4,7,17,20,150,165,175,180,200,223,244,254,273,282,291,345,384,387,407,415,424,437,448,451,453,45
8	181	256,111,25,29,63,89,97,135,139,149,155,162,169,181,190,196,227,248,280,317,375,379,391,400,434,474,4
9	465	31,51,56,113,114,163,218,224,286,315,346,348,394,411,417,432,445,461,465,510,519,555,560,
10	21	13,21,66,68,72,74,106,121,134,151,242,274,276,288,303,307,329,426,442,475,478,548,
11	389	79,463,298,140,330,441,37,377,184,110,232,489,33,124,57,178,146,310,87,221,438,213,112,389,
12	541	41,44,62,105,154,164,264,267,336,338,361,392,410,429,464,476,509,527,541,
13	245	28,48,245,290,318,319,322,326,413,440,554,
14	294	11,23,67,83,117,119,144,152,167,195,207,216,294,311,312,385,395,398,402,409,416,460,469,490,491,511,
15	123	123,186,198,201,226,279,520,536,
16	258	16,30,38,91,101,160,168,170,176,182,191,209,243,258,263,269,270,296,323,332,347,368,382,403,404,406,
17	148	35,39,43,46,49,78,86,100,132,147,148,166,222,239,289,293,328,354,439,449,466,495,500,517,518,542,559
18	14	10,14,19,26,84,103,120,129,188,206,211,233,241,299,342,388,412,427,454,496,
19	513	6,9,18,27,34,53,60,61,70,81,82,104,115,131,157,158,173,174,183,202,205,249,251,275,278,283,297,300,3
20	292	1,8,45,96,130,142,143,179,212,214,215,235,255,257,292,349,358,399,477,485,523,546,

Fig 2: Snapshot of Chromosome structure for K=20

2.1.2: Both the memetic and the genetic algorithms are population based algorithms. In these algorithms, only the fitter solutions are taken to the next generations. To evolve population for the next generation, the fitness of the whole population is computed. Subsequently multiple solutions are randomly selected from the current population based on their fitness, and mutated or recombined to form a new population. The new population is then used in

the next iteration of the algorithm. The fitness function used for evaluating the fitness of a particular chromosome (Chr) is shown in equation (2):

$$\text{Fitness (Chr)} = \frac{\sum_{i=1}^k C_i}{K} \tag{2}$$

where C is the fitness of the cluster i and K is the number of clusters in the solution. The function used to evaluate the fitness C of a cluster is given in equation (3)

$$\text{Fitness(C)} = \frac{\sum_{\substack{i=1 \\ i \neq j}}^m \text{Sim}(i, \text{Cent})}{m} \tag{3}$$

where Cent represents the centroid of the cluster and m refers to the number of users in the cluster. Fig 3 depicts the fitness of each of the clusters in a chromosome of size K=20.

id	users	fitness	Centroid
0	115,433,173,34,81,309,82,528,18,418,202,297,157,323,60,26,61,9,401,174,320,405,131,6,27,183,70,258,20...	0.2792821495992238	513
1	43,58,69,71,122,126,127,133,136,137,204,247,314,333,359,378,381,383,397,408,425,446,472,498,531,	0.41558484435081483	472
2	25,29,63,89,97,135,139,149,155,162,169,181,190,196,241,248,255,280,317,342,375,379,391,399,400,474,4...	0.33274038664756284	181
3	24,52,55,91,107,128,168,230,231,302,304,313,331,332,344,350,353,362,371,373,376,430,449,467,480,481,...	0.33204396019379295	302
4	12,14,15,35,73,84,85,90,108,129,141,161,185,188,192,193,194,203,208,211,219,240,252,259,277,281,295,...	0.29804389781895135	295
5	2,65,80,116,153,197,234,262,272,287,299,364,365,457,493,552,	0.3739187438040972	493
6	22,30,36,40,59,76,88,92,120,138,156,159,182,187,225,228,237,238,253,261,266,271,327,337,406,423,436,...	0.2695596605539322	36
7	4,7,17,20,150,165,175,180,223,244,254,270,273,282,291,345,384,387,404,407,415,424,437,448,451,453,45...	0.3117729091161006	448
8	56,86,113,114,163,286,315,328,346,348,411,417,445,461,465,510,519,540,542,549,555,560,	0.32478807731108233	465
9	13,21,66,68,72,78,121,130,134,151,329,426,439,442,475,478,495,	0.33597292268977447	21
10	32,96,103,109,224,246,257,276,293,316,324,351,497,508,551,	0.3562026302019755	551
11	10,41,44,105,154,164,264,267,336,349,361,392,410,429,464,476,509,527,541,	0.33366309106349945	541
12	11,23,67,83,117,119,144,152,160,167,176,195,207,243,294,311,312,347,368,385,395,398,402,403,409,416,...	0.2997673526406288	294
13	3,45,49,54,74,125,142,147,189,212,222,236,242,274,288,289,303,352,354,358,370,420,456,466,473,492,51...	0.2837722922364871	242
14	37,62,200,216,227,235,256,260,308,338,422,434,485,	0.3202834507593742	62
15	42,46,50,64,75,77,93,95,98,102,132,148,171,177,179,199,209,217,220,265,268,306,321,325,386,390,512,533,	0.33063471264072825	177
16	28,48,245,290,318,319,322,326,413,435,440,454,554,	0.3441038681910588	245
17	123,186,198,201,206,226,233,279,520,536,	0.35158259719610213	123
18	31,39,51,100,106,166,218,307,334,340,360,367,372,393,394,428,432,447,500,514,534,	0.31054322563466574	393
19	1,5,8,19,33,38,47,57,79,87,94,99,101,110,111,112,118,124,140,143,145,146,172,178,184,210,213,214,215,...	0.28837917535974267	284

Fig 3: Snapshot of Fitness of the cluster and the chromosome

2.1.3: System adapted the crossover operator [36] for creating the population for next generation.

The steps followed to generate an offspring of the two chromosomes A and B are:

2.1.3.1: Primarily for all memes/genes b_i in parent B, determine the nearest meme a_i in parent A, where $1 \leq i \leq K$, K is number of memes/genes in each solution.

2.1.3.2: Assign b_i to the memes of a_i with maximum similarity.

2.1.3.3: Finally the offspring chromosome contains memes/genes of A which are closer to more than n memes/genes of B where n lies between 0 to K. To retain size of

chromosome, rest of the memes were selected randomly from the list of memes in parent B which were closer to A.

To maintain the size of initial population, second offspring was produced by applying same recombination strategy in the reverse order.

2.1.4: Local search optimization is the key quality of Memetic algorithms. The most widely used algorithms for performing local search in Memetic algorithms are K-means and simulated annealing [46]. The strategy used to perform the local search in our system is as follows:

2.1.4.1: Primarily from an individual chromosome with K memes, the meme cluster with maximum fitness is selected.

2.1.4.2: Then the n worst users from the chosen cluster are extracted and redistributed in the other K-1 clusters.

2.1.4.3: Thereafter from the left out K-1 clusters, t worst clusters are identified.

2.1.4.4: All the users from these t worst clusters were reassigned among themselves based on their similarity with the centroids of the t clusters.

This process of local search optimization was followed for defined number of iterations for every individual in the population.

2.1.5: In this step an additional K-means operator was used to update the centroids of whole population after every generation.

For creating the clusters using Memetic algorithm, Step1 to Step 5 were followed for defined number of iterations. However for Genetic algorithm based clustering, Step 1-Step 5 except step 4 were used to create the model.

2.2 PREDICTIONS

The main task of collaborative filtering is to predict the utility of items to an active(online) user, based on a dataset of user ratings(votes) provided by the other users in the database. The prediction task in collaborative filtering has been broadly categorized into two types- rating prediction and choice prediction [40]. Rating prediction endeavors to estimate the rating score that a user may assign to an item. This is generally evaluated by holding out certain amount of user ratings in the form of test dataset and then comparing the predicted ratings with the true ratings. Rating scores measure the extent to which a user likes an item, thus the performance of rating prediction reflects an algorithm's ability at capturing user's preference over items. In contrast choice prediction task involves predicting which users would rate what items, and is often based on implicit user feedbacks for example The who rated what type of task in KDDCUP 2007 [6] is an example of choice prediction. The present study is performed on rating prediction. Thus after creating the models using Memetic and Genetic clustering algorithms, test dataset was used to evaluate the predictive accuracy of the model for collaborative filtering. Ratings for the unrated items were predicted using two methods.

2.2.1: In this method, the cluster having highest similarity with the active user was used to predict the ratings. The algorithm [48] modified to predict how user i in the test dataset t(i,j) would

rate item j is :

- Primarily the similarity (Sim (i, u)) of active user i with the centroids of all clusters was computed using the similarity function given in equation (3).
- Then the cluster whose centroid had maximum similarity with the active user was selected for prediction.
- The users of the selected cluster comprised the neighborhood (Ni) people similar to the active user i.
- Thereafter the weight w(i,k) of each user k in Ni was computed using equation(4)

$$w(i,k) = \frac{(constant - Sim(i,k))}{constant} \quad (4)$$

where constant \leq Number of users in the chosen cluster.

- Finally equation (5) was used to compute the predicted value p(i,j) for t(i,j) .

$$p(i,j) = \frac{\sum_{k=1}^{Ni} (w(i,k) * s(k, j))}{\sum_{k=1}^{Ni} w(i,k)} \quad (5)$$

2.2.2: This methodology [5] utilizes more than one cluster to predict the ratings of the unrated items. It primarily involves selection of top 3 clusters having maximum similarity with the active user. Afterwards the rating quality of each item in the selected cluster was computed using function given in equation (6).

$$Q = \frac{(UB + avgrating)}{2 \times UB \times \sqrt{var}} \quad (6)$$

where UB is upper bound of the ratings, avgrating represents average rating of the item in the chosen cluster and var is the variance of the ratings given by individual users for the item in the chosen cluster. After computing the quality (Q) of each unrated item in the chosen cluster, the clusters in which Q for each item lies in the interval $((\max imum - .005) \leq Q \leq \max imum Q)$ are selected for predicting the ratings using equation (7) as follows:

$$Rating = \frac{\sum_{c=1}^{nc} Q_c * avgrating)}{\sum_{c=1}^{nc} Q_c} \quad (7)$$

where Qc is the quality of item in selected cluster and nc is the number of clusters selected.

3. EXPERIMENTS AND RESULTS

Evaluation of recommender systems depends upon the task for which they are used. The two main end user tasks identified in collaborative filtering literature are Annotation in context and Find Good Items [19]. First task refers to the annotations of recommendations with prediction information. Users use this information to decide which information to read or follow foremost whereas the second task provides users with ranked list of the recommended items along with predictions for how much users would like them. Extensive experiments were carried out to compare the performance of memetic algorithm approach with genetic algorithm approach for collaborative filtering recommender system. Performance of the both algorithms was measured for both the end user tasks using the two most popular evaluation metrics:

3.1: Decision support metrics- These metrics measure the rate with which a recommender system makes correct or incorrect decisions about whether an item is good. These metrics are suitable for the task of finding good items when users have true binary preferences. These parameters are used to determine the capabilities of the systems in predicting high quality items that is the items that would be rated highly by the active user for example precision, recall, f1 measure as given in equation (8), (9) and equation (10) respectively.

$$\text{Precision} = \frac{N}{P} \times 100 \quad (8)$$

$$\text{Recall} = \frac{N}{R} \times 100 \quad (9)$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (10)$$

where P denotes the number of recommendations or predicted ratings, N be the subset of P with number of predicted ratings greater than the average ratings and R be the number of ratings given by active user which are greater than the average ratings. The F1-measure combines the precision and recall ideas from information retrieval for cluster evaluation [31]. F1-measure values lie in the interval [0, 100], and larger values of f1 measure indicate the better clustering quality. However these metrics do not endeavor to directly measure the capability of an algorithm to accurately predict ratings.

3.2: Statistical Accuracy metrics: This metrics is particularly important for evaluating the task in which predicted ratings are displayed to the user like Annotation in Context. For example MovieLens recommender predicts the number of stars that a user will give to each movie and displays that prediction to the user. This metric establishes the statistical strength of the recommender algorithm by comparing the estimated (predicted) stars against the actual stars given by the user to each movie. Thus a recommender system may fail if the predicted ratings it displays to the user are incorrect even if a recommender system was able to correctly rank users movie recommendations. Mean Absolute Error (MAE) metrics was used to compute the predictive accuracy of both the recommender algorithms. Let $\{a_1, a_2, \dots, a_n\}$ be the actual user ratings, $\{p_1, p_2, \dots, p_n\}$ be the predicted values of same ratings then the mean absolute error (MAE) is computed as given in equation (11):

$$|E| = \sum_{i=1}^N \frac{|pi - ai|}{N} \tag{11}$$

Experiments were conducted on a Core2 Duo 1.67 GHz processor, 2-GB RAM computer using Java1.6 and MYSQL5.1. All experimental were performed on widely used MovieLens [58] recommender dataset, with population of 100 chromosomes evolved over 100 generations. MovieLens collaborative filtering dataset was collected by the GroupLens Research Project at the University of Minnesota. Dataset contained 100,000 ratings ranging from 1-5 for 1682 movies rated by 943 users. Rigorous experiments were performed from different perspectives to compare the efficacy of collaborative filtering recommender systems developed using memetic and genetic algorithms. In all the experimental evaluations M and G are used as prefix for memetic and genetic algorithm respectively. The following perspectives were identified to evaluate the recommendation and predictive accuracy of both the genetic and the memetic algorithm:-

- 3.2.1: Variation in the number of clusters
- 3.2.2: Effect of varying the number of users by varying the number of movies rated by the user.
- 3.2.3: Impact of using more than one cluster for making predictions.

Experiment 1: This experiment was conducted to evaluate the impact of varying number of clusters K=10, 20 and 30 on recommendation accuracy. It was performed by training the clustering model for users whose rated movies were greater than or equal to 30 in the MovieLens dataset. The results as shown in Fig 4, 5 and Fig 6 depict the efficiency of precision, recall and f1 parameters over top N number of predictions.

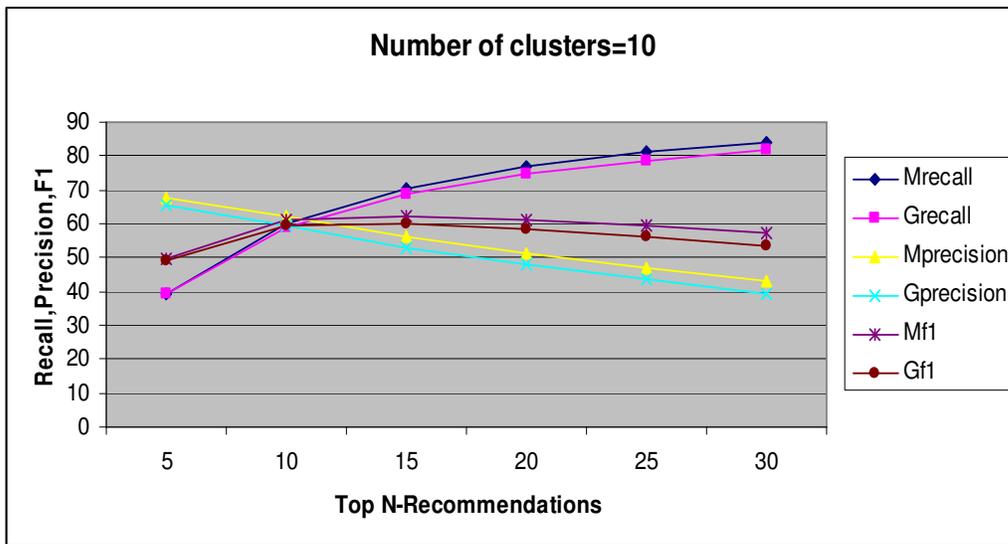


Fig: 4 Number of clusters=10

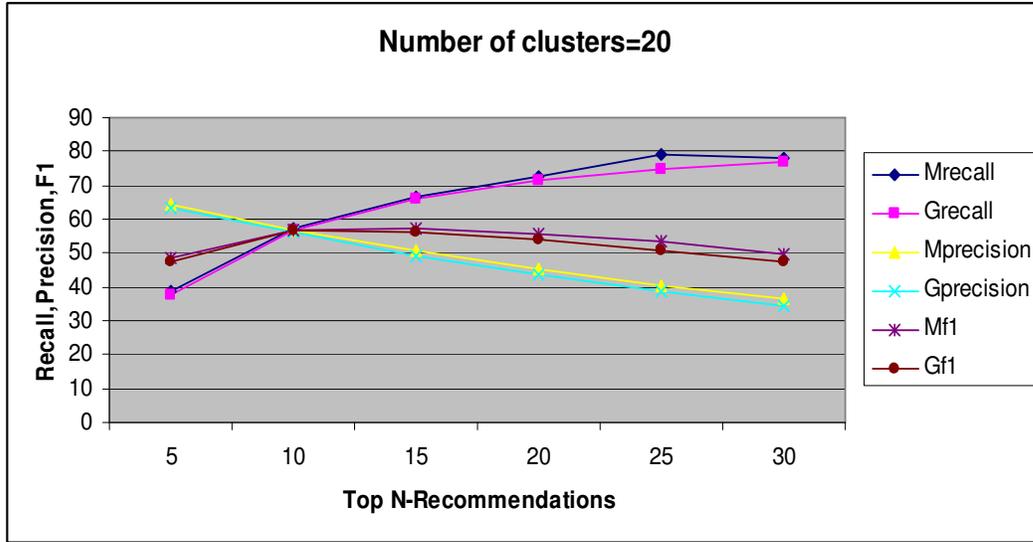


Fig: 5 Number of clusters=20

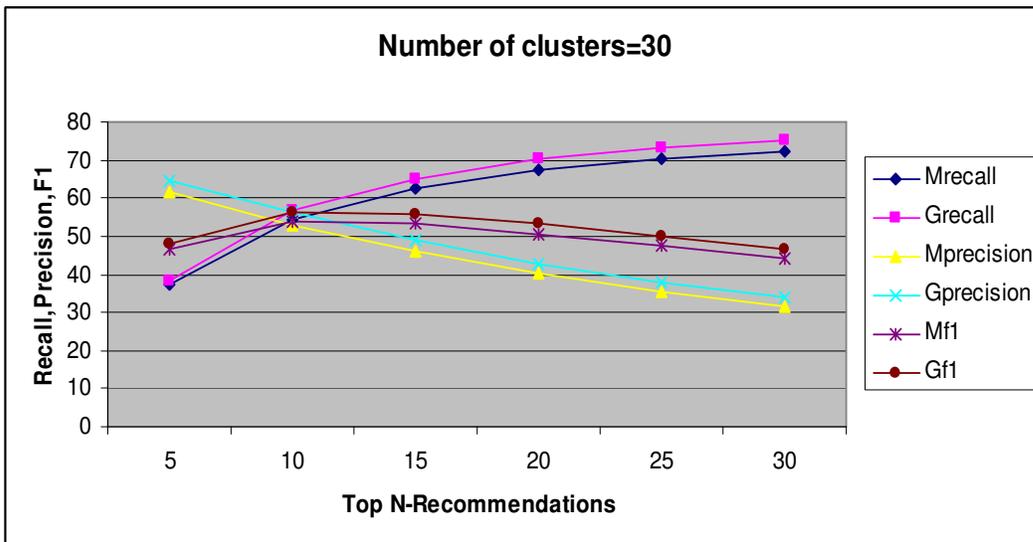


Fig: 6 Number of clusters=30

Fig 4 and fig5 illustrate that memetic recommender system has slightly better accuracy as compared to genetic recommender system. However for N = 30 as shown in Fig 6, genetic recommender system performed better. To validate these results, mean absolute error was computed as shown in Fig 7 which precisely establishes that for all values of N, memetic collaborative filtering algorithm significantly outperforms genetic collaborative filtering algorithm.

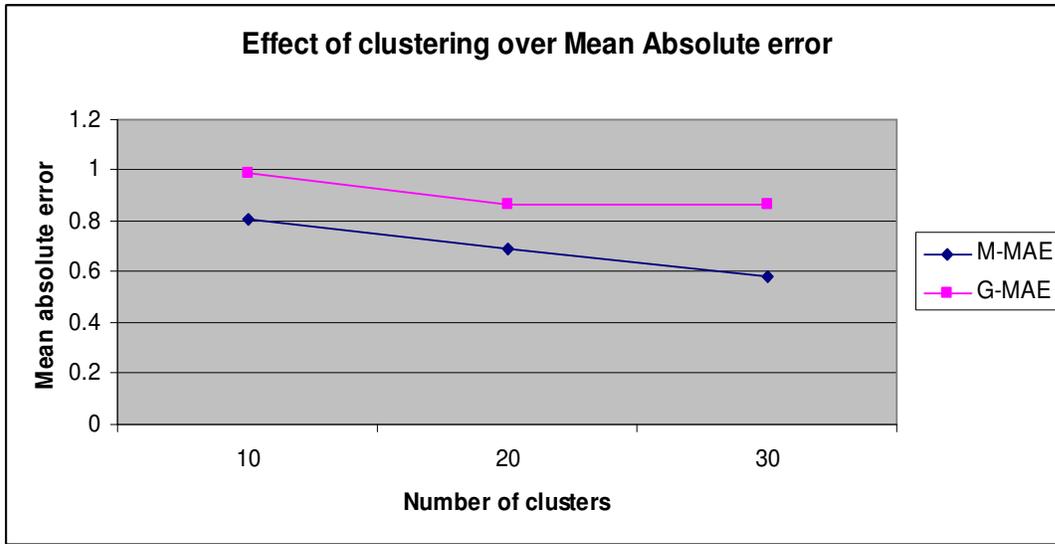


Fig 7: Mean Absolute error for N=10, 20, 30

Experiment 2: This experiment was performed to measure the effect of varying the number of users and predicted ratings, on precision recall and f1 metrics. The variation in numbers of users was made by varying the threshold value that is the number of movies rated by the user. Consequently only the users, whose rated movie count was above threshold, were considered for training. The threshold values of 200,100, 50 and 30 movies reduced the number of users (U) 148, 361,563 and 730 respectively. The results obtained for U= 148, 361, 563, 730 are shown in Fig8, Fig9 Fig10 and Fig11 respectively.

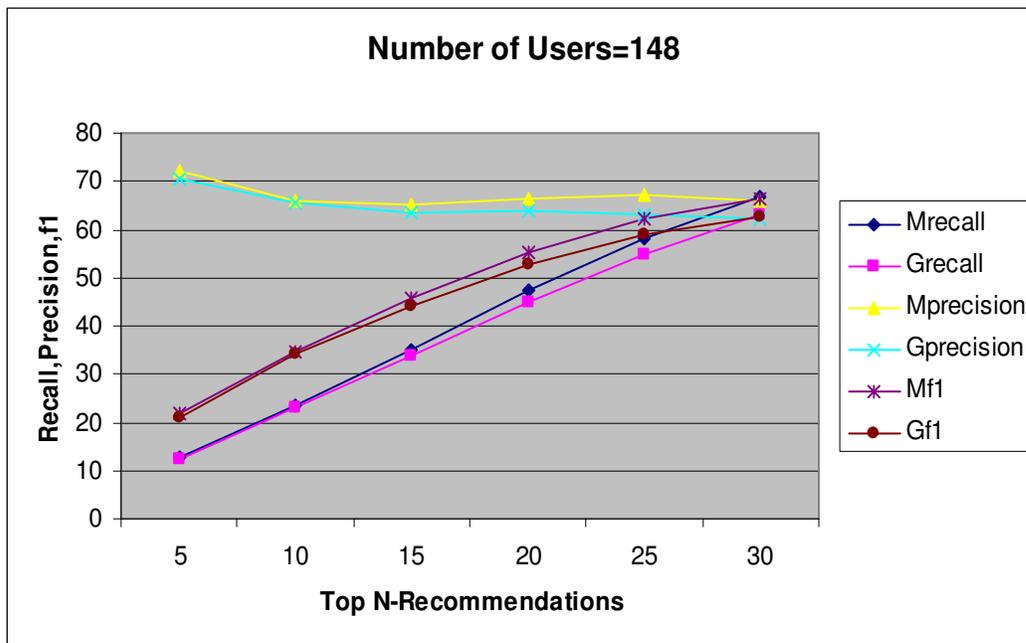


Fig 8: Precision, Recall and F1 for U=148

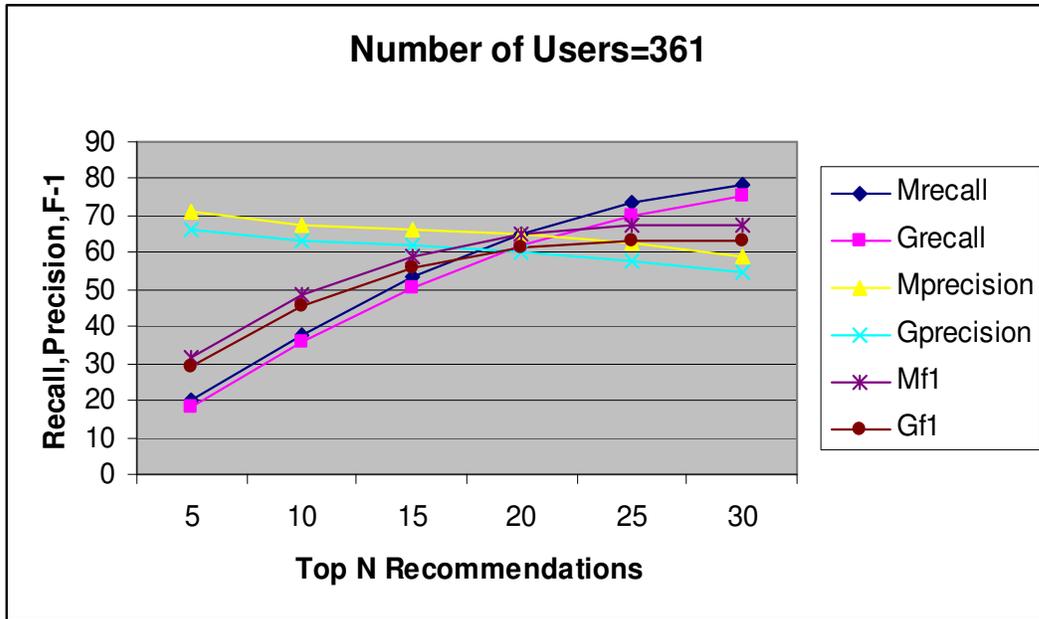


Fig 9: Precision, Recall and F1 for U=361

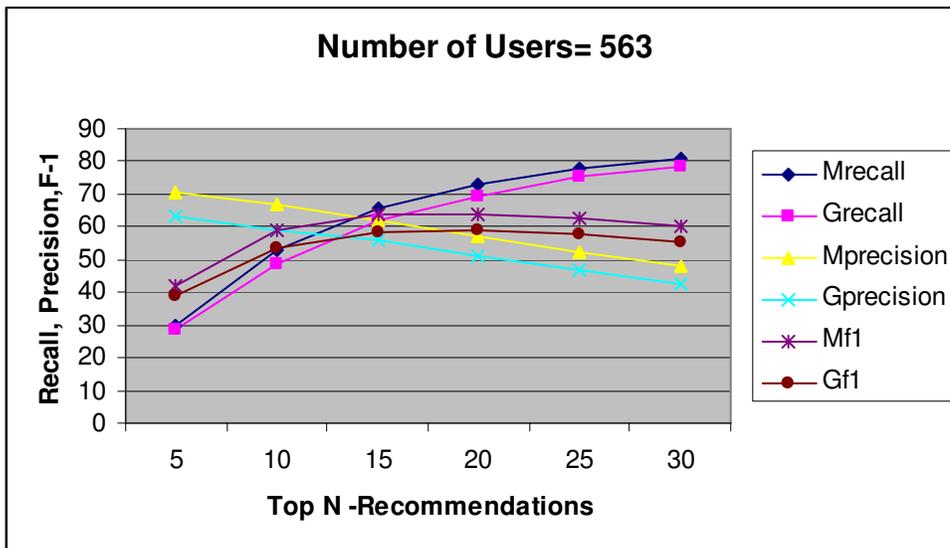


Fig 10: Precision, Recall and F1 for U = 563

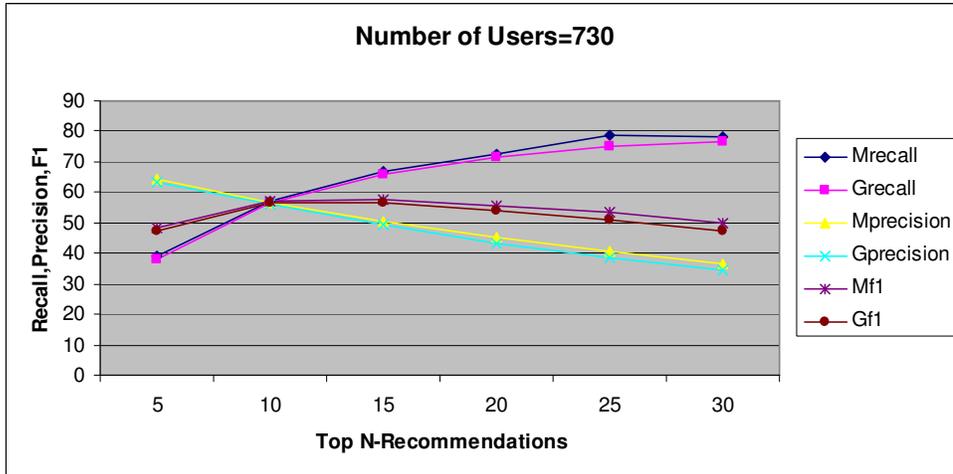


Fig 11: Precision, Recall and F1 for U = 730

Fig8-Fig11 illustrate that there is only trivial difference in the decision support accuracy of memetic algorithm and genetic algorithm for top N recommendations. To substantiate these results statistical accuracy of the algorithms was evaluated using Mean absolute error (MAE). The MAE results obtained for U=148, 361, 563 and 730 as shown in Fig12, precisely demonstrate that for all values of N, MAE is less for memetic algorithm. This confirms that memetic collaborative filtering algorithm has more likelihood of providing good quality items to the user as compared to genetic collaborative filtering algorithm.

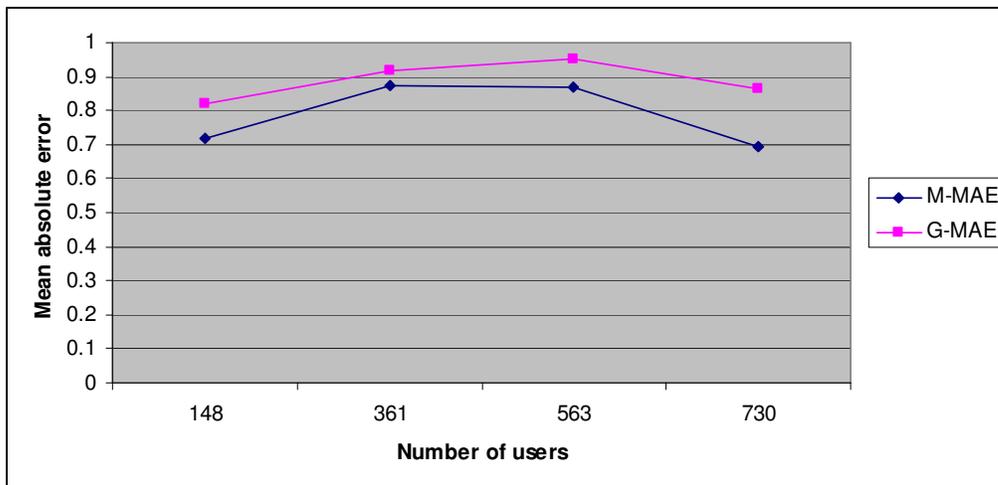


Fig12: Mean Absolute Error over number of users

Experiment 2: This experiment was conducted to study the impact of using more than one cluster for making predictions. Both the memetic and genetic algorithms were evaluated based on decision support metrics for small and large number of recommendations. Following parameters were set for the experiments:

Number of rated movies: ≥ 30

Number of Users: 730

Number of clusters(C) = 20

Number of Recommendations(R) = 15, 30.

As shown in Fig13 and Fig14 respectively for smaller as well as larger number of recommendations, values of recall, precision and f1 metrics improved considerably for predictions made using more than one cluster.

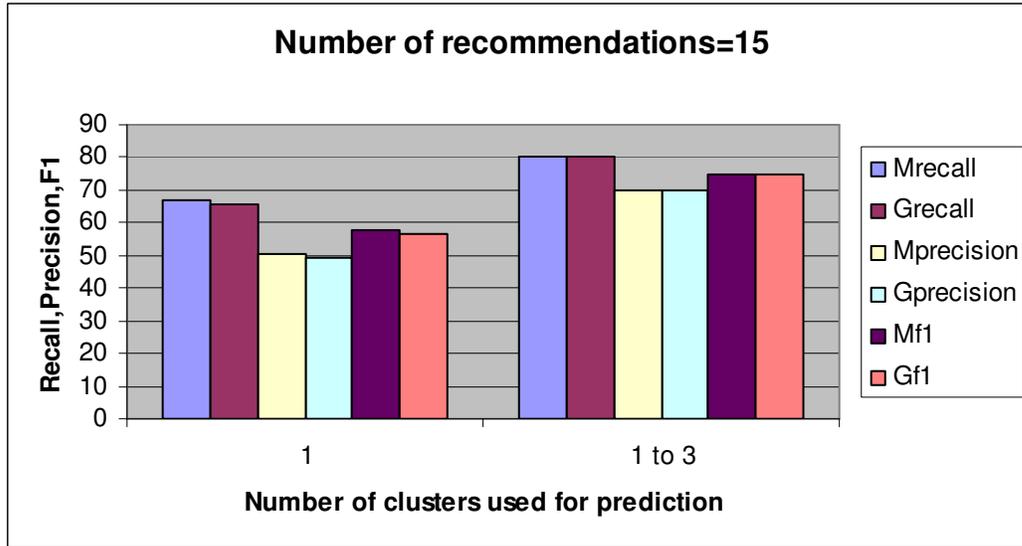


Fig13: Precision, Recall and F1 for C=20, R=15

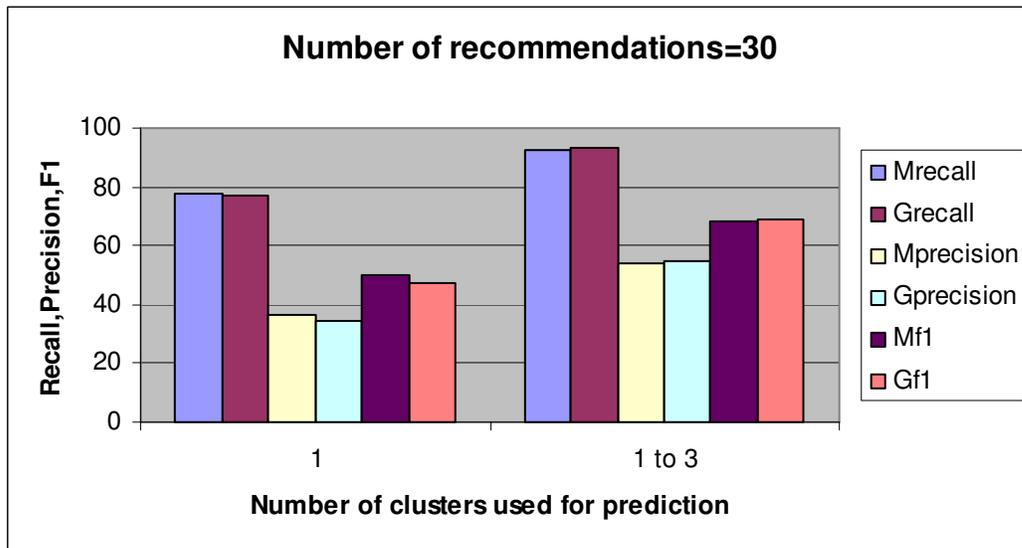


Fig14: Precision, Recall and F1 for C=20, R=30

However results also revealed that both algorithms exhibited comparable performance for all decision support metrics. To validate these results mean absolute error (MAE) was computed as shown in Fig15. Results demonstrate that in both the cases (predictions made using 1 cluster and more than one clusters) predictive accuracy of memetic algorithm is significantly better than the

genetic algorithm.

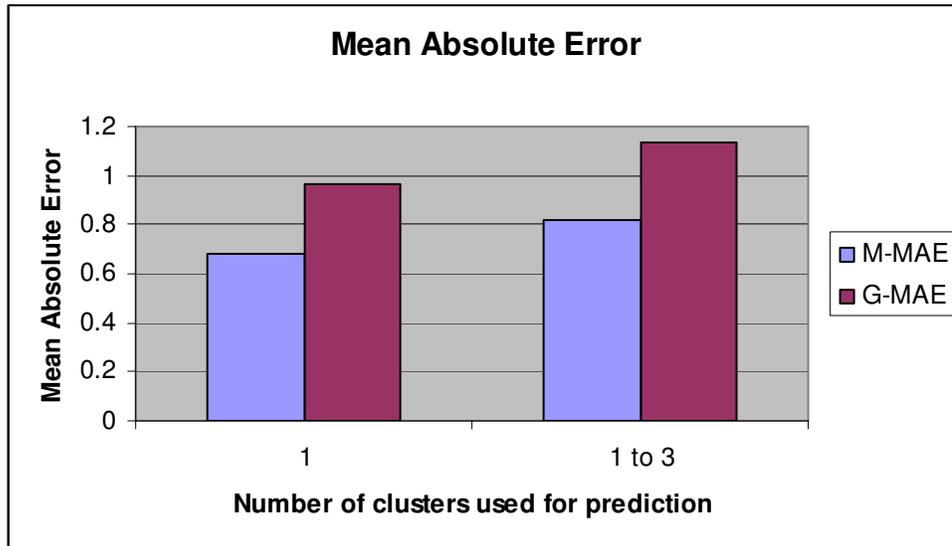


Fig 15: Mean Absolute error over number of clusters used for predicting ratings.

Overall, work contributes notably towards the collaborative filtering research. Rigorous experiments performed from multiple perspectives demonstrate that for varied number of users or clusters both memetic and genetic algorithms may show similar performance for decision support metrics but statistically memetic algorithm provides more predictive accuracy as compared to genetic algorithm. Studies also established that if more than one clusters are used for making rating predictions, values of recall, precision and f1 amplify but along with that mean absolute error also escalates. However mean absolute error of memetic algorithm lies within the acceptable range. On the whole, empirical results precisely corroborate that Memetic collaborative filtering based recommender system perform significantly better as compared to Genetic collaborative filtering algorithm based recommender system.

5. CONCLUSION

This paper presented multi-perspective comparative evaluation of memetic and genetic clustering algorithms for model based collaborative filtering recommender system. Memetic algorithms are genetic algorithms that incorporate local search in the evolutionary concept. Performance of both the algorithms was measured from different perspectives like variation in number of clusters, effect of increasing the number of users, varying number of recommendations and using either one or more than one cluster for computing ratings of the unrated items. Painstaking experiments were performed on MovieLens dataset to compare the decision support and statistical accuracy of memetic and genetic algorithms. Results establish that both the memetic and genetic algorithms may depict comparative performance for the systems displaying ranked list of items along with predictions for how much users would like them. However recommender system may not succeed if the predicted ratings it displays to the user are incorrect even if a recommender system was able to correctly rank users movie recommendations. Therefore memetic algorithm is a better choice for recommender system applications where the goal is to display predicted ratings along with the recommendations and to present the ranked list of items.

REFERENCES

- [1] Adomavicius G., Tuzhilin A. (2008). Toward the Next generation of Recommender systems: A survey of the State-of-the-art and Possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, vol 7, No. 6, pp. 734-749.
- [2] Arabie L. J. ,Hubert G. , and DeSoete P. (1999), *Clustering and Classification*. Singapore: World Scientific.
- [3] Aranha C.,Iba H.(2008) Application of a Memetic Algorithm to the Portfolio Optimization Problem.AI, LNAI 5360, pp. 512–521.
- [4] Banati, H., Mehta S. (2010), Memetic Collaborative filtering based Recommender System, 2nd Vaagdevi International Conference on Information Technology for Real World Problems, accepted for publication, In IEEE press.
- [5] Bedi P., Sharma R., Kaur H.(2009), Recommender System based on Collaborative behavior of Ants. *Journal of Artificial Intelligence*, Vol 2, No. 2, pp. 40-55.
- [6] Bennett J., Elkan C., Liu B., Smyth P., and Tikk D.(2007) Kdd cup and workshop 2007. *SIGKDD Explorations*, vol 9, no. 2, pp. 51–52.
- [7] Breese, J.S., Heckerman, D., Kadie, C. (1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Madison, Wisconsin. Morgan Kaufmann pp. 43-52
- [8] Casillas A. , Gonzalez de Lena M. T. , and Martinez R. (2003), Document Clustering into an unknown number of clusters using a Genetic Algorithm, LNAI 2807, pp.43-49.
- [9] Dawkins, R. (1976), *The Selfish Gene*. Oxford University Press.
- [10] Dubes A. K. and Dubes R. C.(1988), *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice- Hall.
- [11] Eiben AE, Smith JE (2003) *Introduction to evolutionary computing*. Springer, Berlin.
- [12] Elbeltagi E, Hegazy T. and Grierson D. (2005), “Comparison among five evolutionary-based optimization algorithms,” *Advanced Engineering Informatics*, Vol 19, No.1, pp.43–53.
- [13] Everitt B. S. , Landau S. , and Leese M. (2001), *Cluster Analysis*. London, U.K.: Arnold.
- [14] Garg P.(2009), A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm *International Journal of Network Security & Its Applications (IJNSA)*, Vol.1, No 1.
- [15] Goldberg D, Nichols, D., Oki, B.M., Terry, D. (1992), Using Collaborative Filtering To Weave an Information Tapestry. *Using Collaborative Filtering To Weave an Information Tapestry*. *Communications of the ACM*, 35(12): pp. 61–70
- [16] Goldberg K.,Roeder T., Gupta D. and Perkins C.,(2001), Eigentaste: A Constant Time Collaborative Filtering Algorithm ,*Information retrieval journal*, Vol 4, pp133-151.

- [17] Gong S., Ye H., Tan H., (2009), Combining Memory-Based and Model based Collaborative filtering in recommendation systems. Pacific-Asia conference on Circuits, Communications and system.
- [18] Herlocker J. L , Konstan A. J., Borchers A. Riedl J.(1999), An Algorithmic framework for performing Collaborative filtering. 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp.230-237.
- [19] Herlocker J. L.,(2004) Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, Vol 22, No. 1, pp. 5-53.
- [20] Hill, W., Stead, L., Rosenstein, M., Furnas, G. (1995) Recommending and Evaluating Choices in a Virtual Community of Use. In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. ACM Press pp. 194-201.
- [21] Hsu C., Shyr W., Wu C. and Chou P. (2007) Downlink Spatial Division Multiple Access (SDMA) Optimization of Smart Antennas by Phase-Amplitude Perturbations Based on Memetic Algorithms. Second International Conference on Innovative Computing, Information and Control .
- [22] Jain A. K. , Murty M. N. , and Flynn P. J.(1999) ,Data clustering: A review, ACM Computing Survey, vol. 31, pp. 264–323.
- [23] Jiao L., Gong M., Wang S., Hou B., Zheng Z. and Wu Q. (2010) Natural and Remote Sensing Image segmentation using Memetic computing. IEEE computational Intelligence magazine.
- [24] Jin R., Si L. (2004) A Bayesian Approach toward Active Learning for Collaborative Filtering, Proceedings of the 20th conference on Uncertainty in artificial intelligence, pp.278 – 285.
- [25] Kaufman L. and Rousseeuw P. J. (1990), Finding Groups in Data—An Introduction to Cluster Analysis. Series in Probability and Mathematical Statistics .New York: Wiley.
- [26] Kennedy J., Eberhart R., and Shi Y. (2001) Swarm Intelligence, Morgan Kaufmann.
- [27] Kim K., Ahn H.,(2008) A recommender system using GA K-means clustering in an online shopping market, Expert Systems with Applications: An International Journal, Vol 34, No. 2, pp.1200-1209.
- [28] Ko S., Lee J., (2002), Discovery of User Preference through Genetic Algorithm and Bayesian Categorization for Recommendation, ER Workshop. LNCS 2465, pp. 471-484
- [29] Kohrs A. and Merialdo B. (1999) Clustering for Collaborative Filtering Applications. In Proceedings of CIMCA'99. IOS Press.
- [30] Konstan, J.A., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J. (1997), GroupLens: Applying Collaborative Filtering To Usenet News. Communications of the ACM, 40(3), pp. 77–87.
- [31] Kowalski G., (1997) Information Retrieval Systems—Theory and Implementation. Norwell, MA: Kluwer Academic.
- [32] Lu T., Li T., (2007), Predicting Customer's Preference in E-Commerce Recommendation System: A Genetic Algorithm Approach .ICICIC.
- [33] Maltz D, Ehrlich, E. (1995), Pointing The Way: Active Collaborative Filtering. In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, ACM, pp. 202–209

- [34] Merlo, G., Caram, F; Fernández V., Britos, P. Rossi, B. & García-Martínez R. (1999) Genetic-Algorithm Based Image Compression, SBAI-Simpósio Brasileiro de Automação Inteligente, São Paulo, SP, 08-10 de Setembro de.
- [35] Merz P., Freisleben B. (2000), Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem. IEEE transactions on evolutionary computation vol. 4 no. 4.
- [36] Merz P., Zell A.,(2002),Clustering Gene expression Profiles with Memetic Algorithms. Parallel Problem Solving from Nature, LNCS, Vol VII, pp. 811-820.
- [37] Mitchell M.,(1998) An Introduction to Genetic Algorithms. Cambridge,MA: MIT Press, 1998.
- [38] Moscato, P., Cotta C., (1999), A Gentle Introduction to Memetic Algorithms. In: Handbook of Meta-heuristics, pp. 1–56. Kluwer, Dordrecht.
- [39] Muruganandam A., Prabhakaran G., Asokan P., Baskaran V. (2005), A memetic algorithm approach to the cell formation problem. International Journal of Advance Manufacturing Technology vol. 25, pp. 988–997
- [40] Nathan N. Liu, Min Zhao, Evan Xiang, Qiang Yang. (2010), Online Evolutionary Collaborative Filtering,, RecSys2010, ACM.
- [41] Park H. and Cho S. (2006),Evolutionary Fuzzy Clustering for Gene Expression Profile Analysis”, SCIS&ISIS2006.
- [42] Park H., Yoo S., and Cho S.(2005) Evolutionary Fuzzy Clustering Algorithm with Knowledge-Based Evaluation and Applications for Gene Expression, Journal of Computational and Theoretical Nanoscience Vol.2, pp. 1–10.
- [43] Rao K. N., (2008), Application Domain and Functional Classification of Recommender Systems-a Survey, DESIDOC Journal of Library and Information technology, Vol 28,No. 3,pp.17-35
- [44] Razib M. O., Safaai D., Rosli M. I., Zalmiyah Z. and Saberi M. M.,(2005), Automatic Clustering of Gene Ontology by Genetic Algorithm, International Journal of Information Technology vol 3 no. 1.
- [45] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.(1994) GroupLens: An Open Architecture For Collaborative Filtering Of Netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM Press pp. 175-186
- [46] Russell S., Norvig P., Artificial Intelligence: A Modern Approach, 2nd Edition
- [47] Schafer J. Ben , Frankowski D. , Herlocker J., and Sen S.,(2007),Collaborative Filtering Recommender Systems, The Adaptive Web, LNCS 4321, pp. 291 – 324.
- [48] Shardanand U., Maes P.,(2005) Social Information Filtering: Algorithms for Automating “Word of Mouth”. Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210-217
- [49] Terveen L., Hill W., Amento B., McDonald D and Creter J., (1997), PHOAKS: a system for sharing recommendations. Communications of the ACM, March 1997 v40 n3 p59(4)
- [50] Turgut D., Das S. K., Elmasri R. and Begumhan, (2002), Optimizing Clustering Algorithm in Mobile Ad hoc Networks Using Genetic Algorithmic Approach”, Global Telecommunications Conference,GLOBECOM '02. IEEE.

- [51] Ungar L. H. and Foster D. P. (1998), Clustering Methods for Collaborative Filtering. In Proc. Workshop on Recommendation Systems at the 15th National Conf. on Artificial Intelligence. AAAI Press.
- [52] Vitorino R., Fernando M., (2000) Image Colour Segmentation by Genetic Algorithms, RecPad 2000 - 11th Portuguese Conf. on Pattern Recognition, pp. 125-129.
- [53] Volk J., Herrmann T., Wuthrich K. (2008), Automated sequence-specific protein NMR assignment using the memetic algorithm MATCH. Journal of Biomolecular NMR vol. 41, pp.127–138.
- [54] Wang Y., Qin J.A.,(2007), Memetic-Clustering-Based Evolution Strategy for Traveling Salesman Problems. RSKT , LNAI 4481, pp. 260-266.
- [55] Wei S. and Soon C. P., (2006) Genetic Algorithm-based Text Clustering Technique: Automatic Evolution of Clusters with High Efficiency”, Proceedings of the Seventh International Conference on Web-Age Information Management Workshops (WAIMW'06), IEEE.
- [56] Xue G., Lin, C., & Yang, Q., et al.(2005), Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the ACM SIGIR Conference, pp.114–121.
- [58] Zhang F., Chang H.,(2006) A Collaborative Filtering Algorithm Employing Genetic Clustering to Ameliorate the Scalability Issue, IEEE International Conference on e-Business Engineering .
- [59] <http://www.grouplens.org/node/73.org/node/73>

Dr. Hema Banati is an associate professor of Computer Science in Dyal Singh College at University of Delhi. Her research interests include Web engineering, software engineering and Human Computer Interfaces, multiagent systems and E-learning. She has many international publications to her credit. She can be reached at banatihema@hotmail.com



Shikha Mehta is a research scholar in Computer Science Department at University of Delhi. Her research interests include Web applications engineering, Distributed Artificial Intelligence, Data Mining, Soft Computing, evolutionary algorithms and MultiAgent systems. She can be reached at mehshikha@gmail.com