Pertanika J. Sci. & Technol. 27 (1): 123 - 141 (2019)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Adaptive Genetic Algorithm for Feature Weighting in Multi-Criteria Recommender Systems

Gursimarpreet Kaur[1]* and Saroj Ratnoo[2]

[1]*Space Applications Centre, Indian Space Research Organization, Ahmedabad 380015, India*
[2]*Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar 125001, India*

ABSTRACT

Recommender Systems (RS) have proven to be a successful personalization technique in this era of ever increasing information overload. Among many available recommendation techniques, Collaborative Filtering (CF) is the most popularly used. However, most of the CF applications use single ratings for recommending items and the use of multi-criteria ratings in the recommendation process is still under-explored. This paper proposes multi-criteria RS based on Adaptive Genetic Algorithm (AGA). The AGA design, which updates the crossover and mutation rates dynamically, is employed to model the users' preferences for multi-criteria ratings on different attributes of items. The AGA optimizes a user's preferences for different attributes in the form of a weight vector. Thus, the AGA finds an individual optimal weight vector in relation to each user. The weight vector is used to recommend items to the respective user. The experiments are conducted on Yahoo movies, a well known multi-criteria rating dataset. The experimental results confirm that the AGA based multi-criteria RS outperforms the traditional single criteria based Collaborative Filtering RS and the simple GA based multi-criteria RS.

Keywords: Collaborative filtering, Genetic Algorithms, multi-criteria, Recommender Systems

ARTICLE INFO

Article history:
Received:18 January 2018
Accepted:12 September 2018
Published: 24 January 2019

E-mail addresses:
gursimarpreet8029@gmail.com (Gursimarpreet Kaur)
ratnoo.saroj@gmail.com (Saroj Ratnoo)
* Corresponding author

INTRODUCTION

The exponential growth of information on the internet has led to the problem of information overload. Although, it has become convenient for users to access a wide range of information, at the same time

ISSN: 0128-7680
e-ISSN: 2231-8526

it is all the more possible now to get astray while searching for some specific information of interest on the internet. More often, the users have to pass through many links before reaching the information that they need. In this scenario, Recommender Systems (RS) have emerged as an important tool to provide users only the selective choices. Recommender Systems are personalized information filtering techniques that suggest only a limited number of items that are most likely to be of interest or to be suitable to one's needs (Ricci, Rokach, Shapira, & Kantor, 2011). RS have been widely implemented in the application domains like recommending music, movies, online courses, learning material, books and video. These are categorized into five basic classes i.e. content-based, Collaborative Filtering (CF) based, demography-based, knowledge-based and community-based (Adomavicius & Tuzhilin, 2005). Though, each technique has its own pros and cons; the CF technique is the most popular one among all of these. One of the major limitations of the existing RS is that these are based on overall rating value as the sole criterion for evaluating users' preferences. Users might express their opinion based on different features or attributes of an item, so even if two users agree on global ratings, they may have completely diverse preferences on different features of that item (Sanchez-Vilas, Ismoilov, Lousame, Sanchez, & Lama, 2011). A user may prefer a movie because of its story-line whereas another user may like the movie due to acting or extraordinary visual effects. Both of these users rate the particular movie high, but for different reasons. Hence, in such cases, it is inappropriate to find similarities between such users only on the basis of overall rating as the single criteria. Nothing is more annoying than getting recommendations for the items in which a user is least interested. By using multi-criteria ratings, we can gather information about the specific preferences of users based on the different attributes of items to be recommended and avail the opportunity to generate more accurate recommendations (Adomavicius & Kwon, 2007; Wang & Geng, 2008; Adomavicius, Manouselis, & Kwon, 2011; Zarrinkalam & Kahani, 2012).

A Genetic Algorithm (GA) is a stochastic search technique based on the mechanism of natural selection and genetic evolution to solve complex optimization problems. Genetic Algorithms (GAs) have been widely and effectively used in the field of recommender systems (Fong, Ho, & Hang, 2008; Bobadilla, Ortega, Hernando, & Alcalá, 2011; Sohrabi, Mahmoudian, & Raeesi, 2011). Since a user gives different importance or priority to each feature in multi-criteria RS, GAs have been mainly used for optimizing weights given by users to different features of an item (Fong, Ho, & Hang, 2008; Hwang & Hwang, 2010; Salehi, Pourzaferani, & Razavi, 2013; Parveen, Kant, Dwivedi, & Jaiswal, 2015). Although all these authors have reported the effectiveness of their research as compared to the traditional recommender systems, all these works use a static set of the parameters that are fixed at the beginning of a GA run. Setting appropriate values for crossover and mutation rates, the two GA main operators, is of significant importance for the success of a GA. Optimal values of these operators are problem specific and most often these are determined by hit and trial method. A GA with such a parameter setting may get stuck in local convergence and does not guarantee optimal results. To alleviate this problem, an Adaptive Genetic Algorithm (AGA) can be used. In AGA, parameter values for GA operators get updated dynamically according to the fitness values of the solutions at that particular generation (Srinivas & Patnaik, 1994). Since an AGA is capable of maintaining a better balance between exploration and exploitation, it avoids premature convergence which is essential for finding global optimal solutions.

This paper proposes a multi-criteria recommender system that uses an adaptive GA to optimize weights for the four criteria (acting, direction, story and visuals) given in the Yahoo movies dataset. The Adaptive GA successfully models an individual user's preferences given to different criteria in terms of weights to reason out why the particular user prefers some movies over the others. The main contribution of this paper is the design and application of an adaptive GA for optimizing weights for a Collaborative Filtering based multi-criteria RS for the movie recommendation. The experiments reveal that the recommendations made by the proposed AGA based multi-criteria recommender system are more accurate than the recommendations made by a traditional single criterion based recommendation technique. Moreover, as the proposed AGA sets the probability of crossover and mutation dynamically depending on the state of the GA population (converging or diverging); it achieves a significant performance improvement as compared to the simple GA based RS.

The rest of the paper is organized as follows: After the introduction in section I, section II presents the essential background and related work on multi-criteria RS. It reviews the status of research in applying GA and AGA to model users' preferences. Section III proposes the adaptive GA design. Section IV illustrates the overall design of RS using AGA with the help of a block diagram. The experimental design and results are described in section V. Section VI concludes the paper and points to future research directions.

BACKGROUND DETAILS AND RELATED WORK

Recommender Systems

Recommender systems suggest interesting items in the cases where the range of choices exceeds a user's ability to view them to reach a proper decision. This narrowing down of items aids in improving browsing and consumption experience of customers and thus increases customer loyalty and sales provided that recommendations made are correct according to the tastes and interests of diverse users. Most of the popular RS are based on Collaborative Filtering (CF) technique for making recommendations. A Collaborative Filter based RS presumes that the users who have similar preferences in past are likely to have same preferences in future too. Recommender Systems (RS) use some form of user feedback which is generally in the form of item ratings. At present, most of the RS use only the overall rating values of items for gaining access to users' opinions. RS based on a single criterion as overall ratings consider two users similar if they have similar overall ratings. However, an agreement between two users on overall rating does not necessarily mean that these users have similar preferences for the various aspects of the item. A RS will be more effective if it not only finds what people like but also captures the essence why they like it, i.e., it should recognize preferences not just patterns. Hence, the focus of research has recently shifted from single criteria RS (SCRS) based on overall ratings to multi-criteria RS (MCRS) that account for preferences of users for different attributes of items to make more valuable recommendations (Adomavicius & Kwon, 2007; Teng & Lee, 2007; Hassan & Hamada, 2016, 2017).

Multi-criteria Recommender Systems (MCRS)

A Multi-criteria RS (MCRS) is relatively a new technique that takes into account user's ratings on many attributes in addition to the overall ratings. In MCRS, the overall rating is predicted quite differently as compared to that of single criterion based recommendation techniques. The overall rating is resolved based on a number of ratings given to the attributes of items. Hence, MCRS need to capture the degree or the weight of users' preferences for different facets of items of their interest. The main objective of MCRS is to model a user's preferences from the values of multi-criteria ratings assigned by that user to the various items' attributes. This amounts to searching for a vector of optimal weights for reflecting an individual user's preferences over multiple criteria which is an optimization problem. (Adomavicius & Kwon, 2007; Manouselis & Costopoulou, 2007; Lakiotaki, Tsafarakis, & Matsatsinis, 2008; Adomavicius, Manouselis, & Kwon, 2011; Hassan & Hamada, 2016). The foremost contribution in developing MCRS came from (Adomavicius & Kwon, 2007). The authors analyzed the MCRS framework on Yahoo movie dataset and their results confirmed the superiority of MCRS over SCRS with respect to error rates, precision, recall and f-measure etc. Since their inception, MCRS have proved their merit in several application domains (Li, Wang & Geng, 2008; Adomavicius, Manouselis & Kwon, 2011; Sanchez-Vilas, Ismoilov, Lousame, Sanchez, & Lama, 2011; Sohrabi, Mahmoudian & Raeesi, 2011; Jannach, Karakaya & Gedikli, 2012; Rodriguez, Posse & Zhang, 2012; Salehi, Pourzaferani, & Razavi, 2013; Parveen, Kant, Dwivedi, & Jaiswal, 2015). Overall, there is still a scope to further explore MCRS techniques for making the recommendation process more accurate and effective.

Genetic Algorithms in MCRS

A GA follows meta-heuristic technique motivated by the principle of natural genetics and evolution. GAs have been consistently used to solve difficult optimization and search problems. For the operation of a GA, an initial population of solutions is created randomly in which each solution is a finite length string known as a chromosome. At every evolutionary step, the fitness function is applied to compute the fitness of each individual solution. A fitness function is an application dependent predefined quality criterion. A fitness proportionate selection is carried out to create a new population by probabilistically taking fittest individuals from the previous population. These solutions then reproduce to form new individuals on the application of genetic operators, i.e., crossover and mutation. This whole process is repeated until a stopping criterion is reached (Goldberg, 1989; Michalewicz, 1996).

Counting on GAs' promising history in the domain of optimization, researchers have used GAs to search for an optimal weight vector for a user's preferences in case of MCRS. Fong, Ho & Hang, (2008) proposed a novel MCRS method by taking the input data from both *Movielens* and *IMDB* movies datasets. A total of 37 features were taken from both datasets in order to prevent any bias from one set of data. Then the weights of these criteria were optimized using a GA. Hwang & Hwang, (2010) proposed a framework for integrating CF technique with GA, wherein GA was used for criteria weighting. Some more GA based MCRS proposals were suggested in (Jannach, Karakaya & Gedikli, 2012; Rodriguez, Posse & Zhang, 2012; Geng, Li, Jiao, Gong, Cai & Wu, 2015). Recently, Parveen, Kant, Dwivedi and Jaiswal (2015) treated the problem of n-MCRS as n single criteria problems. They solved these individual problems and then the overall rating was taken as the aggregation of these ratings. A GA was used to optimize priorities of users on these criteria. In addition, some multi-objective pareto-efficient approaches have also been promulgated in the domain of MCRS (Ribeiro, Lacerda, Veloso & Ziviani, 2012; Ribeiro, Ziviani, Moura, Hata, Lacerda, & Veloso, 2014).

This is an established fact that setting of GA parameters (most importantly crossover probability (p_c) and mutation probability (p_m)) is one of the common factors that contribute to the success or failure of a GA to search for the globally optimal solution. Most of the GA implementations for optimizing users' preferences over different attributes set the parameters by hit and trial method and used a static parameter setting. In these implementations, the parameters are usually tuned experimentally by trying many combinations of values of probability of crossover (p_c), probability of mutation (p_m), and population size and then their effect is analyzed on the final solution. The combination of parameters that appears best somehow is fixed prior to running GA and the parameter setting remains the same for the whole GA process. Crossover and mutation probabilities mainly control the extent of exploitation and exploration during the lifetime of a GA. A GA is successful in realizing its full potential only if it is able to achieve an appropriate balance between exploitation and exploration. An *ad hoc* static parameter setting directs the evolution towards local convergence. Adaptive GAs, instead of relying on constant values of crossover and mutation rates fixed at the beginning of GA, are able to determine these parameters adaptively by using the information contained in the state of the current GA population. AGAs can achieve a better balance in exploitation and exploration by changing the crossover and mutation probabilities dynamically to match the diverging or converging status of the GA population at hand and hence, these can maintain diversity in addition to preserving the convergence capacity (Pellerin, Pigeon, & Delisle, 2004; Srinivas & Patnaik, 1994).

We have come across only one attempt of using adaptive genetic algorithms for improving prediction accuracy of multi-criteria recommender systems by (Hassan and Hamada 2017). The proposed approach integrates a SlopeOne algorithm (SoA) with adaptive GA to determine the level of significance in improving the prediction accuracy of AGA based MCRS as compared to single rating based SoA. In this paper, we have taken up the task of applying an adaptive GA framework to arrive at better optimal weights for users' preferences that are subsequently input to a collaborative filtering based RS for making more accurate recommendations.

THE PROPOSED AGA DESIGN

The multi-criteria RS proposed in this paper is based on Adaptive Genetic Algorithm to improve the accuracy of recommendations. We have implemented two versions of AGA to find out the optimal weights of four feature ratings, i.e., Acting, Direction, Story and Visuals given in Yahoo movies dataset. In the first version, the crossover and mutation probabilities are updated from one generation to the next. The change in probabilities depends on the state of the GA population. These probabilities, once updated for a generation remains same for all the individuals in the GA population at that generation. In the second version of AGA within each generation, crossover and mutation probabilities vary from individual to individual chromosome in the GA population depending on their fitness. This section illustrates the design of the AGA in detail.

Chromosome Encoding and Population Initialization

Since the aim of AGA is to find the optimal weights of ratings of four features (Acting, Direction, Story and Visuals of Yahoo movies dataset), each of these features is treated as a gene and together these four features form a chromosome. Each gene in this string represents an explicit attribute weight and it can take any value between 0 and 1. Value of a gene suggests the amount of preference a user has for the particular feature. A value near 0 is treated as dislike, 0.5 is average liking and a value near 1 indicates that the user likes this particular feature of the movie. Hence, chromosomes are real-encoded as they can take any real value between 0 and 1. The value of each gene is further normalized by dividing it by the sum of the overall initial chromosome weight vector as shown in the second row of Figure 1. This way the sum of all weights is always 1. The population is initialized randomly the same way as the individual chromosome.

	Acting	Direction	Story	Visuals
Weights	W1	W2	W3	W4
Initial random weights	0.8	0.4	0.3	0.2
Normalized weights $= \frac{w_i}{\sum_{i=1}^{n} w_i} = 1.7$	0.470	0.235	0.176	0.117

Figure 1. Chromosome encoding

Fitness Function

Fitness function is also known as objective function or evaluation function as it evaluates the goodness or worth of each individual solution in the population of a GA. Fitness is evaluated at each generation right from the initialization of population. Selection of fitness function is very crucial in the functioning of a Genetic Algorithm. Convergence to an optimal solution is dependent on the type of fitness function selected. Our AGA based multi-criteria RS uses fitness function which minimizes the difference between overall ratings and the aggregate of individual ratings of different criteria's, as given in equation 1.

$$fitness\ function := \sum_{i=1}^{N} \left| r_{overall} - \frac{w_1 r_1 + w_2 r_{2+} \ldots + w_k r_k}{w_1 + w_2 \ \ldots + w_k} \right| \quad (1)$$

Where $w_1, w_2, \ldots, w_k$ are the weights assigned by AGA and $r_1, r_2, \ldots, r_k$ are the ratings given by a user to the k criteria. The overall score given by a user to a movie is represented by '$r_{overall}$' and N denotes the number of items (or say movies) rated by that particular user (Parveen, Kant, Dwivedi, & Jaiswal, 2015).

Selection

Selection is an operator which allows better fit individuals to get selected for the successive generations. Selection has no relation to the type of problem or fitness function and thus it is an independent portion of a GA. This RS uses roulette wheel selection which is also known as fitness proportionate selection. In this, a probability of selection is assigned to each individual by dividing its fitness by total fitness of the population which results in a normalized value between 0 and 1. By this technique, highly fit chromosomes have more chances of survival to the successive generations as compared to the weaker ones that keep getting eliminated as a GA progresses.

Adaptive Crossover and Mutation

Crossover or recombination operator takes two better fit parent solutions and produces offspring which are likely to have high fitness as compared to the parents. Many types of crossover operators can be found in the literature, for example, single point, double point, uniform and heuristic etc. Our AGA based RS uses adaptive heuristic crossover which is a good option for real coded GAs. It helps to maintain diversity as it not only repositions genetic material but also introduces new one. Expression for the heuristic crossover for two parents *Q* and *P*, out of which *Q* is better fit parent, is given below.

$$Offspring1 = r(Q - p) + Q \tag{2}$$
$$Offspring2 = r(Q) + (1 - r)P \tag{3}$$

In equations 2 and 3, *r* is a random number between 0 and 1. A larger value of r makes the crossover more exploratory.

The crossover and mutation probabilities are dynamically updated as given in (Srinivas & Patnaik, 1994). Crossover probability, p_c for the two versions of adaptive GA is calculated adaptively by using equations 4 and 5.

Crossover probability for the first version which is to be updated from generation to generation but remains fixed for all the individuals in the GA population.

$$p_c = \begin{cases} \dfrac{k_1}{f_{max} - f_{avg}} & for\ f_{max} > f_{avg} \\ k_1 & otherwise \end{cases} \tag{4}$$

Crossover probability for the second version that is to be updated for every individual separately in the GA population.

$$p_c = \begin{cases} \dfrac{k_1 \times (f_{max} - f_{best\_par})}{f_{max} - f_{avg}} & for\ f_{max} > f_{avg}\ and\ \ f_{best_{par}} > f_{avg} \\ k_1 & otherwise \end{cases} \tag{5}$$

The values f_{max}, f_{avg} f_{best_par} respectively represent the maximum fitness, average fitness of the generation and fitness value of the better fit parent out of the two parents to be crossed.

Mutation is a genetic operator which alters one or more gene values, resulting in a different chromosome. Purpose of applying mutation operator is to recover the lost genetic material and thus introduce some amount of diversity in the population. There are many types of mutation operators and we have applied an adaptive random mutation, i.e., some genes are randomly changed by a value between 0 and 1. The way mutation probabilities are computed dynamically for the two respective AGAs are given in the equations 6 and 7.

Mutation probability for the first version which is to be updated from generation to generation but remains fixed for all the individuals in the GA population.

$$p_m = \begin{cases} \dfrac{k_2}{f_{max} - f_{avg}} & for\ f_{max} > f_{avg} \\ k_2 & otherwise \end{cases} \quad (6)$$

Mutation probability for the second version that is to be updated for individuals in the GA population.

$$p_m = \begin{cases} \dfrac{k_2 \times (f_{max} - f_{ind})}{f_{max} - f_{avg}} & for\ f_{max} > f_{avg} and\ \ f_{ind} > f_{avg} \\ k_2 & otherwise \end{cases} \quad (7)$$

The value f_{ind} in equation 7 is the fitness of the individual to be mutated.

When f_{max} is closer to f_{avg} then the status of GA is closer to convergence and hence the value of mutation and crossover probabilities increase to avoid local convergence. Further, if the f_{max} is very close to f_{ind} or f_{best_par}, the crossover and mutation probabilities acquire low values, i.e., the better individuals with fitness near to maximum fitness do not require to be crossed or mutated with high probabilities and vice-versa. This preserves individual with fitness closer to the max fitness into the successive generation and disrupts the individuals with low fitness by applying crossover and mutation with higher probabilities. The crossover and mutation probabilities become literally zero when $f_{max} = f_{best\_par}$ and $f_{max} = f_{ind}$ respectively. The probabilities become k_1 and k_2 when $f_{best\_par=f_{avg}}$ and $f_{ind} = f_{avg}$ respectively. The individuals with average fitness are also disrupted with high probabilities. The values for k_1 and k_2 are kept to be 1.0 and 0.5. The high values of k_1 and k_2 are there to produce chaotic conditions when GA is about to converge to take it out of local convergence.

Stopping Criteria

All the above steps of a GA are repeated until the stopping criterion is reached. For our GA, stopping criteria is stall generation limit. In this approach, GA keeps on going until there is no change in the best fitness value over some predefined number of generations. On reaching this criterion, individual with best fitness value is returned, which is a weight vector corresponding to the four criteria's.

AGA BASED MULTI-CRITERIA RECOMMENDER SYSTEM ARCHITECTURE

In this section, we discuss the proposed RS which is based on AGA. In this RS, YAHOO movie dataset is used which has individual ratings for different criteria and an overall criterion for each movie. Here, we describe how to incorporate these multi-criteria ratings into the CF approach. The overall rating is not an independent rating and it is some aggregation function *f*, of different criteria ratings.

$$r_{overall} = f(r_1, r_2 \dots r_k)$$

Where $r_{overall}$ is the overall rating of a movie and $r_1, r_2 \cdots r_k$ are the individual ratings for the k criteria; the value of k is 4 in this case (i.e. for Acting, Direction, Story and Visuals). The aggregation function models a user's preferences and to figure out the aggregation function *f*, some technique is required, which in our case is AGA. We have used AGA to optimize weights of these four criteria as different users place different priorities on these movie attributes. The block diagram for the overall architecture for the proposed recommender system is shown in Figure 2.

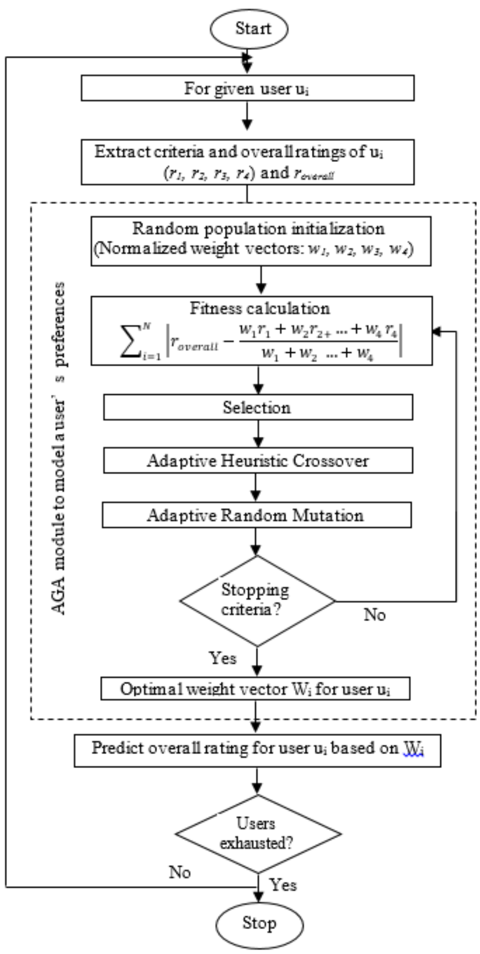

Figure 2. The architecture of AGA based multi-criteria recommender system

EXPERIMENTAL DESIGN AND RESULTS

There are two objectives of this experimental evaluation- i) to compare the performance of a single criterion traditional Item-based Collaborative Filtering RS with a GA based multi-criteria RS. ii) To compare the performance of the AGA against the simple GA in the domain of multi-criteria RS. All the genetic algorithms used for the experimentation in this paper have been implemented on Windows 7 platform using R studio. We have used 'recommenderlab' R package for making collaborative filtering based recommendations. To evaluate the performance of the proposed system, a fraction of randomly selected ratings given by a user have been used for building the model and the rest of the ratings have been used for the test purpose.

The Dataset

We have taken data on Yahoo Movies from Jannach, Karakaya and Gedikli (2012) in ready to use form who extracted it from the website (http://movies.yahoo.com). In this dataset, each movie has 5 ratings, i.e., 4 criteria ratings (acting, direction, story and visual effects) and 1 overall rating. The dataset contains 976 movies from 6078 users. Each rating has a value in the range 1 to 5. We have considered ratings by the users who have rated more than 10 movies and those movies which have been rated at least by 10 users. This way the rating data comes from overall 50 users.

Recommendation Algorithms

To compare and evaluate the proposed RS, we have used 5 variations of algorithms i) Traditional Item-based Collaborative Filtering (TICF) which finds out similarity between the items rated by different users, based on the similarity it chooses a neighbor set for the current user and then it presents recommendation. ii) Genetic algorithm based CF (GA_CF), which applies a GA to find out optimal weights for the criteria and then based on these weights, it calculates the overall ratings for making recommendations. iii) Mean_CF algorithm which simply takes the mean of the ratings for the four criteria as the overall rating. iv) Adapt_CF which uses an adaptive version of the GA for optimizing weights. This algorithm computes the probabilities of crossover and mutation (p_c and p_m) for every new generation adaptively, however the probabilities remain constant over a generation. v) Adaptive_CF which computes the probabilities of GA operators (P_c and P_m) for each individual separately as defined in section 4. This adaptive version of GA automatically adjusts these probabilities according to the fitness of the individuals participating in the reproduction process of the GA.

Evaluation Metrics

We have used four metrics to compare the performance of these 5 algorithms, i.e. Mean Absolute Error (MAE), Precision, Recall and F1 measure. The performance metrics are defined below.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|r_{overall} - r'_{overall}|$$

$$precision = \frac{correctly\ recommended\ movies}{total\ movies\ recommended}$$

$$recall = \frac{correctly\ recommended\ movies}{total\ no\ of\ relevant\ items}$$

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Where N is the number of movies rated by the ith user, $r_{overall}$ is overall rating predicted by the RS under consideration and $r'_{overall}$ is the actual overall rating given by the user to that movie.

Determining the Parameters for the Simple GA

As described earlier, the parameter setting is very important in the performance of a GA, especially the values of p_c and p_m influence the success of a GA to reach the optimal solution. We have attempted to find out an optimal parameter setting for our simple GA implementations by trying different combinations of these two values. From our experiments, we found out that GA gives maximum fitness value for the precision, recall and F-measure at p_m=0.01 and p_c=0.7-0.8 which can be observed from the line charts shown in Figure 3-5. According to these observations, we have chosen the crossover and mutation probabilities for GA_CF algorithm (simple GA) only whereas these probabilities are dynamically adjusted for the latter two versions of the AGA based recommendation algorithms (Adapt_CF and Adaptive_CF) as described in sub-section 3.4. Each chromosome in the GA population represents the weights given to four criteria by an individual user. The population size has been tuned experimentally and a population size of 50 chromosomes was found to be sufficient. The experimental parameters are summarized in Table 1.

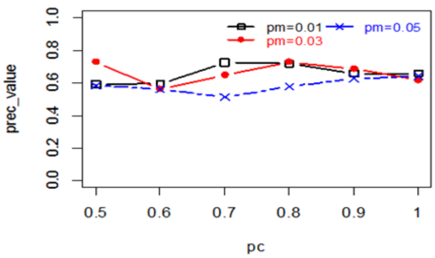

Figure 3. Precision values for different crossover and mutation rates

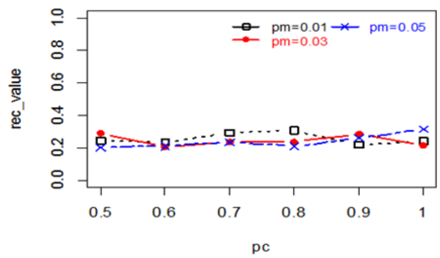

Figure 4. Recall values for different crossover and mutation rates

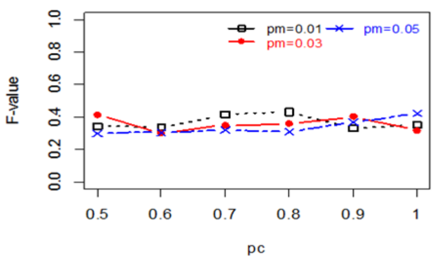

Figure 5

F-values for different crossover and mutation rates

Table 1

Parameter setting for GA

Parameter	Value
Chromosome Length	number of criteria ratings =4 (acting, direction, story and visuals in case of Yahoo movie data set)
Population Size	50
Crossover probability for simple GA, p_c (GA_CF) Crossover probability, p_c (Adapt_CF and Adaptive CF)	0.75 Dynamical adjusted (described in section 3.4)
Mutation probability, p_m (GA_CF) Mutation probability, p_m (Adapt_CF and Adaptive CF)	0.01 Dynamically adjusted (described in section 3.4)
Stopping Criteria	Stall generations

RESULTS AND DISCUSSION

Experiments were conducted on Yahoo movie dataset on 50 users using Item-based Collaborative Filtering method. We had evaluated the performance on four evaluation metrics, i.e., MAE, precision, Recall and F-measure for all the 5 algorithms used in this research. Each experiment was repeated 20 times and then average values of the evaluation metrics were recorded. The results are presented in the Table 2. For graphic illustration, these results are also portrayed in the form of bar charts given in Figures 6-9.

Table 2
Evaluation metrics

Algorithm	Error rates	Precision	Recall	F-measure
TICF	2.70645	0.4596	0.11143	0.17938
Mean_CF	1.133275	0.2968	0.17351	0.21902
GA_CF	1.01702	0.5904	0.24200	0.34289
Adapt_CF	0.697769	0.7586	0.22299	0.34467
Adaptive_CF	0.458239	0.8358	0.31325	0.45706

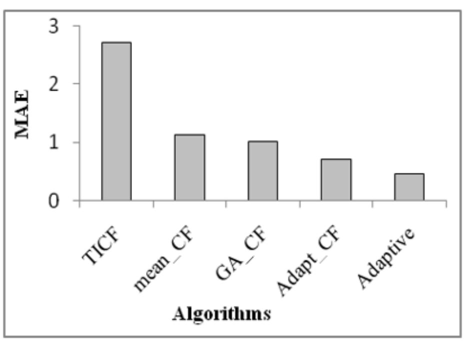

Figure 6. Comparison of error values

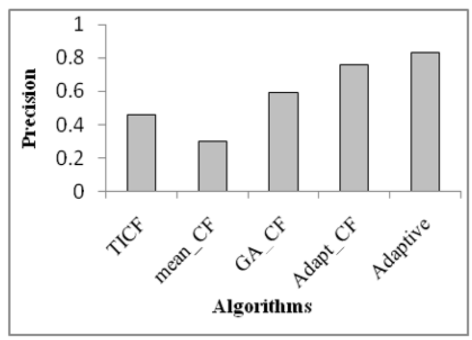

Figure 7. Comparison of precision values

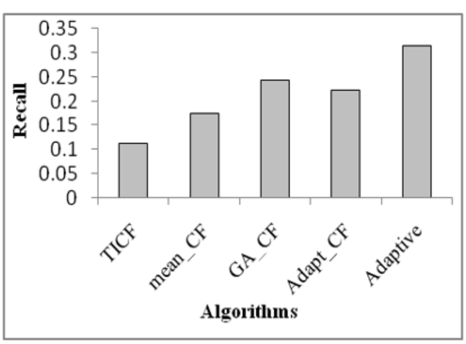

Figure 8. Comparison of recall values

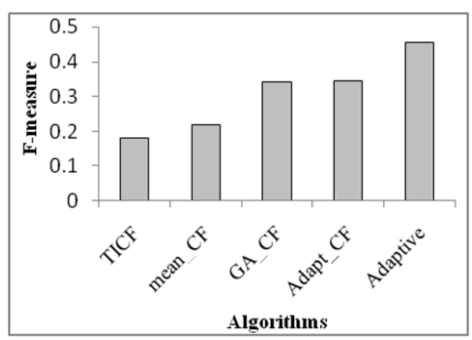

Figure 9. Comparison of F-measure values

On the whole, the results happen to support our claim that the GA based multi-criteria RS should give better performance than the single criteria traditional TICF and adaptive GA based multi-criteria RS must be a better choice than the simple GA based multi-criteria RS for optimizing weights for multiple criteria ratings.

To further validate if the proposed AGA-based multi-criteria RS is statistically significantly better than the other algorithms, we have applied the Wilcoxon signed rank test on error rates obtained by the 5 algorithms at a significance level of 5 percent (α=0.05). The error rates were recorded over 20 samples (Table 4 in Appendix I). The results of Wilcoxon signed rank test (p-values) are presented in Table 3.

Table 3

The results of Wilcoxon Signed Rank test on error rates

	TICF	GA_CF	Mean_CF	Adapt_CF	Adaptive_CF
TICF	-	-	-	-	-
GA_CF	0.000195	-	-	-	-
Mean_CF	0.000381	0.647000	-	-	-
Adapt_CF	0.000195	0.165000	0.029000	-	-
Adptive_CF	0.001953	0.001209	0.000580	0.01208	-

The null hypothesis that there is no difference between the performances of the two pairs of algorithms is rejected if the p-value at the cross-section of these algorithms in the table is less than the significance level (α=0.05). The results of Wilcoxon Signed Rank test authenticate the following:

The performance of each multi-criteria based algorithm is significantly better than the traditional item-based collaborative filtering based recommender system. Hence, multi-criteria ratings should be taken into account for recommending items.

The performance of the Adapt_CF is significantly better than the simple Mean_CF. Hence, taking mean of multi-criteria ratings is not a good choice for making recommendations.

The performance of Adaptive_CF is significantly better than all the other algorithms. Hence, it can be asserted that the adaptive GA, which adjusts the crossover and mutation rates for every individual separately in the GA population, is a better choice for optimizing the weights for the multi-criteria user ratings.

CONCLUSION

In this paper, we have proposed a novel Adaptive Genetic Algorithm (AGA) approach for the optimization of feature weights of the multi-criteria ratings. Subsequently, these weights are used in recommending movies using item-based CF. The experimental results show that additional information gathered from the various criteria's is useful in enhancing the performance of a RS. The experimental results confirm that the proposed AGA based RS outperforms the traditional item based collaborative filtering and the simple GA based

RS. In future, parallel and hybrid GAs can be applied for optimizing weights of the multi-criteria ratings for the recommendation process.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749.

Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multi-criteria rating. *IEEE Intelligent Systems, 22*(3), 48-55.

Adomavicius, G., & Kwon, Y. (2007). Multi-criteria Recommender Systems. In F. Ricci, L. Rokach & B. Shapira (Eds.), *Recommender systems handbook* (pp. 847-880). MA: Springer.

Adomavicius, G., Manouselis, N., & Kwon, Y. (2011). Multi-Criteria Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 769–803). MA: Springer.

Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems, 24*(8), 1310–1316.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *Journal of Convergence Information Technology, 12*(4), 331–370.

Fong, S., Ho, Y., & Hang, Y. (2008, September). Using Genetic Algorithm for Hybrid Modes of Collaborative Filtering in Online Recommenders. In *8th International Conference on Hybrid Intelligent Systems (HIS)* (pp. 174–179). Barcelona, Spain.

Fuchs, M., & Zanker, M. (2012). Multi-criteria Ratings for Recommender Systems: An Empirical Analysis in the Tourism Domain. In C. Huemer & P. Lops (Eds.), *E-Commerce and Web Technologies* (Vol. 123, pp. 100–111). Heidelberg: Springer.

Geng, B., Li, L., Jiao, L., Gong, M., Cai, Q., & Wu, Y. (2015). NNIA-RS: A multi-objective optimization based recommender system. *Physica A: Statistical Mechanics and its Applications, 424*, 383–397.

Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization and machine learning*. Boston: Addison-Wesley Longman Publishing Co., Inc.

Hassan, M., & Hamada, M. (2016, January). Recommending learning peers for collaborative learning through social network sites. In *7th International Conference on Intelligent Systems, Modeling and Simulations (ISMS)* (pp. 60-63). Bangkok, Thailand.

Hassan, M., & Hamada, M. (2017, September). Improving prediction accuracy of muti-criteria recommender systems using adaptive genetic algorithms. In *Proceedings of Intelligent Systems Conference* (pp. 326-330). London, UK.

Hwang, C., & Hwang, C. (2010). Genetic algorithms for feature weighting in multi-criteria recommender systems. *Journal of Convergence Information Technology, 5*(8), 126–136.

Jannach, D., Karakaya, Z., & Gedikli, F. (2012, June). Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce: ACM* (pp. 674–689). New York, NY, USA.

Lakiotaki, K., Tsafarakis, S., & Matsatsinis, N. (2008, October). UTA-rec: a recommender system based on multiple criteria analysis. In *Proceedings of the 2008 ACM Conference on Recommender Systems: ACM* (pp. 219–226). New York, NY, USA.

Li, Q., Wang, C., & Geng, G. (2008, April). Improving personalized services in mobile commerce by a novel multicriteria rating approach. In *Proceedings of the 17th International Conference on World Wide Web: ACM* (pp. 1235–1236). New York, NY, USA.

Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-citeria recommender systems. *World Wide Web, 10*(4), 415-441.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Heidelberg: Springer.

Parveen, R., Kant, V., Dwivedi, P., & Jaiswal, A. K. (2015, December). Enhancing recommendation quality of a multi criterion recommender system using genetic algorithm. In *International Conference on Mining Intelligence and Knowledge Exploration* (pp. 526-535). Hyderabad, India.

Pellerin, E., Pigeon, L., & Delisle, S. (2004, April). Self-adaptive parameters in genetic algorithms. In *Proceedings of Data Mining and Knowledge Discovery: Theory, Tools and Technology* (Vol. 5433, pp. 53–64). Orlando, USA.

Ribeiro, M. T., Lacerda, A., Veloso, A., & Ziviani, N. (2012, September). Pareto-efficient Hybridization for Multi-objective Recommender Systems. In *Proceedings of the 6th ACM Conference on Recommender Systems: ACM* (pp. 19–26). New York, NY, USA.

Ribeiro, M. T., Ziviani, N., Moura, E. S. D., Hata, I., Lacerda, A., & Veloso, A. (2014). Multiobjective Pareto-Efficient Approaches for Recommender Systems. *ACM Transactions on Intelligent System and Technology, 5*(4), 53:1–53:20.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.). (2011). *Recommender Systems Handbook*. MA: Springer.

Rodriguez, M., Posse, C., & Zhang, E. (2012, September). Multiple Objective Optimization in Recommender Systems. In *Proceedings of the 6th ACM Conference on Recommender Systems: ACM* (pp. 11–18). New York, NY, USA.

Salehi, M., Pourzaferani, M., & Razavi, S. A. (2013). Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model. *Egyptian Informatics Journal, 14*(1), 67–78.

Sanchez-Vilas, F., Ismoilov, J., Lousame, F. P., Sanchez, E., & Lama, M. (2011, August). Applying Multicriteria Algorithms to Restaurant Recommendation. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology: IEEE Computer Society* (Vol. 01, pp. 87–91). Washington, DC, USA.

Sohrabi, B., Mahmoudian, P., & Raeesi, I. (2011). A framework for improving e-commerce websites usability using a hybrid genetic algorithm and neural network system. *Neural Computing and Applications, 21*(5), 1017–1029.

Srinivas, M., & Patnaik, L. M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, 24*(4), 656–667.

Teng, W. G., & Lee, H. H. (2007). Collaborative recommendation with multi-criteria ratings. *Journal of Computers, 17*(4), 69-78.

Zarrinkalam, F., & Kahani, M. (2012, October). A multi-criteria hybrid citation recommendation system based on linked data. In *2012 2nd International eConference on Computer and Knowledge Engineering: ICCKE* (pp. 283–288). Ferdowsi University of Mashhad, Mashhad, Iran.

APPENDIX

Table 4

Error values of the twenty samples on the five algorithms

	TICF	Mean_CF	GA_CF	Adapt_CF	Adaptive_CF
Sample1	2.252210	0.30358791	0.9441119	0.8656271	0.4191331
Sample2	2.080407	2.311775	0.6546858	1.638706	0.8639765
Sample 3	2.454835	0.034196329	1.400590	1.009136	0.039842894
Sample 4	1.840278	1.125000	0.3163882	0.9796618	0.5076028
Sample 5	3.025597	1.596193	2.204970	0.16569957	0.9105495
Sample 6	3.274992	1.469172	0.9571358	0.4018334	0.25714517
Sample 7	2.264099	1.404465	0.6124009	1.028368	0.18682236
Sample 8	1.852795	0.7500	0.4429009	0.978819	0.2759654
Sample 9	4.562475	1.584251	0.24309682	0.20558747	0.4631856
Sample 10	3.451920	1.490113	1.776992	0.5789364	0.6911059
Sample11	1.826062	1.00000	0.4000063	0.22063870	0.27242824
Sample12	1.031181	0.7482874	0.7577126	0.6602477	0.7451005
Sample13	4.009927	1.667480	1.318044	0.7366487	0.5083829
Sample14	2.770606	1.609785	0.5218467	0.7403089	0.5770359
Sample15	1.133881	0.125000	1.685471	0.4490401	0.061920783
Sample16	2.539588	0.3132505	1.215395	0.5467240	0.6818667
Sample17	3.522435	1.615295	2.667531	0.4646946	0.3153118
Sample18	3.164317	0.50000	0.8873312	0.8245014	0.7348388
Sample19	3.996181	0.6969514	0.7485456	0.72782	0.3554881
Sample20	3.075476	2.320742	0.5853438	0.7324	0.2970926
Average	2.70645	1.133275	1.01702	0.697769	0.458239