

Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm

Behzad Soleimani Neysiani^{1*}, Nasim Soltani², Reza Mofidi³, Mohammad Hossein Nadimi-Shahraki⁴

¹Faculty of Electrical & Computer Engineering, University of Kashan, Kashan, Isfahan, Iran

^{2,3}Department of Computer Engineering, Allame Naeini Higher Education Institute, Naein, Isfahan, Iran

⁴Faculty of Computer Engineering, Najafabad branch, Islamic Azad University, Najafabad, Isfahan, Iran

*Corresponding author E-mail: b.soleimani@grad.kashanu.ac.ir, Tell: 0989139606471

E-mail: rezamofidi66@naeini.ac.ir³, n.soltani@naeini.ac.ir², nadimi@iaun.ac.ir⁴

Received: 07 October 2018; Accepted: 18 November 2018; Published: 08 February 2019

Abstract—Recommender systems that possess adequate information about users and analyze their information, are capable of offering appropriate items to customers. Collaborative filtering method is one of the popular recommender system approaches that produces the best suggestions by identifying similar users or items based on their previous transactions. The low accuracy of suggestions is one of the major concerns in the collaborative filtering method. Several methods have been introduced to enhance the accuracy of this method through the discovering association rules and using evolutionary algorithms such as particle swarm optimization. However, their runtime performance does not satisfy this need, thus this article proposes an efficient method of producing cred associations rules with higher performances based on a genetic algorithm. Evaluations were performed on the data set of MovieLens. The parameters of the assessment are: run time, the average of quality rules, recall, precision, accuracy and F1-measurement. The experimental evaluation of a system based on our algorithm outperforms show than the performance of the multi-objective particle swarm optimization association rule mining algorithm, finally runtime has dropped by around 10%.

Index Terms—Recommender system, Collaborative filtering, Association rule mining, Genetic algorithm, Multi-objective optimization.

I. INTRODUCTION

The success of online business generally depends on the ability to deliver personalized products to the potential customers [1, 2]. There are three main types of recommender systems: Content-based [3] which is a method to analyze the content of items and try to understand the regularities among them. Knowledge-based [4] which is a method keeping the knowledge of users, items, and the needs of a particular user.

Collaborative Filtering [2, 5] which is a method make recommendations based on user with similar preferences. Collaborative filtering methods have many challenges, for example, dealing with sparsity in the dataset, to improve scalability, and to improve the efficiency of recommendations. Many efforts have been investigated to overcome the problems of collaborative filtering to produce high quality recommendations.

Some of these approaches include: The traditional neighborhood-based techniques [5] which are methods, collaborative filtering based on item, make recommendations through neighbors history. Clustering methods [6] which are methods, clustering algorithms increase the scalability of collaborative filtering by clustering items. Rule-based reasoning [7] which is a method This method has the ability to learn from past experiences to deal with new problems. Association Rule Mining (ARM) [8, 9] which is a method for discovering interesting relations between items or users in a transactional data set.

Sparsity is one of the problems that limits the effectiveness of collaborative filtering. When a transaction data set is sparse, the collaborative filtering become inefficient to identify similarities in consumer interests. Huang et al, presented one way to deal with the sparseness problem by Applying associative retrieval techniques among consumers through past transactions and feedback [10]. Leung et al, introduced a collaborative filtering framework based on Fuzzy Association Rules and Multiple-level Similarity (FARAMS) [11]. This method improves both recommendation accuracy and coverage. Further, researchers are investigating the use of association rule mining for collaborative filtering. Lin et al, developed an efficient Adaptive-Support Association Rule Mining (ASARM) for recommender systems [12, 13]. This algorithm took advantage that the mining task is specific by fixing the head of the rule for the target user or item. Further, the algorithm does not need the minimum support to association rule mining. Tyagi et al, proposed enhancing collaborative filtering recommendations by utilizing Multi-Objective Particle

Swarm Optimization (MOPSO) embedded association rule mining [14]. In this algorithm, association rules discovered by particle swarm optimization in the framework of collaborative filtering to improve accuracy and performance. However, their performance does not fulfill.

The PSO algorithm is useful for continuous search space even though it can use discrete operators too, but the genetic algorithm is more appropriate for discrete search space and because the rules contains of items and need substitution operators for searching in this space, genetic algorithm seems be better. In this paper, the genetic algorithm is used to identify association rules without minimum support. Genetic algorithm is efficient for search space, when the search space is too large. In this approach does not need users to specify the minimum support threshold. Instead of discovering a large number of interesting rules in traditional mining models, only the most interesting rules are discovered according to the fitness function. In this research, produced an efficient collaborative filtering based on association rules.

The rest of this paper is organized as follows. Section 2 describes other researches in this subject, section 3 a review of the association rule mining, genetic algorithms and multi-objective optimization. Section 4 describes the proposed method. Section 5 presents the recommended strategy and research methodology. Section 6, briefly introduces then used datasets and discusses the experimental. Finally, conclusions and are given in section 7.

II. RELATED WORKS

Generally, the association rule mining (ARM) considered two steps:

- First: finding all frequent items set
- Second: generating strong association rules frequent item set found in step one

There are several heuristic and meta-heuristic algorithms which are used for ARM especially by genetic algorithm, but Adaptive Support Association Rule Mining (ASARM) used for ARM in recommendation system first time which has some limitation based on this application like single left item set and time limitation in real-time systems. In this algorithm, rule generation has done by a CBA-RG algorithm and its aim was about recommender system based on observation. This algorithm implemented by C++ which used EachMovie dataset but it was slow[14].

In [23], researchers show improving the accuracy of collaborative filtering recommendations with clustering and ARM. In this research, they preprocessed data to discover similar pattern among users and after that used cluster techniques to reduce the size of data. Then similarities between them were computed.

In [24], they study online interactive collaborative filtering with dependencies between items. They

computed the item dependencies as the clusters on arms and also they provided an online algorithm based on particle learning.

III. LITERATURE REVIEW

In this section, the backgrounds of association rule mining, Genetic Algorithm, and multi-objective optimization are briefly described.

A. Association Rule Mining (ARM)

Association rule mining has been introduced by Aggarwal [15]. Association rule mining is one of the most important data mining techniques. It aims to discover relative relationships from commercial transaction data set [15]. Therefore, association rule mining can help in the business decision making process by letting the decision makers understand customers buying habits [15, 16].

The rules are in the form of $A \rightarrow B$, where A and B are item sets and $A \cap B = \emptyset$. If A happens, B is most possible to happen. The standard Association rule mining algorithms return those rules that meet the threshold values set for support and confidence measures which are the measures of interestingness [15]. Threshold of support and confidence is denoted by minimum support and minimum confidence. These parameters are usually set by users. The support and confidence of an association rule ($A \rightarrow B$) are defined as follows:

$$\text{Sup}(A \rightarrow B) = \frac{\text{Number of transactions which contain } A \text{ and } B}{\text{Number of transactions in the database}} \quad (1)$$

$$\text{Conf}(A \rightarrow B) = \frac{\text{Number of transactions which contain } A \text{ and } B}{\text{Number of transactions which contain } A} \quad (2)$$

The Apriori algorithm is one of the association rule mining algorithm, which was designed by Agrawal et al. In 1993 [15]. This algorithm can generally be considered as a process having two steps. The first step is to find all frequent item set, and the second step is to generate strong association rules by the using of frequent item sets.

B. Genetic Algorithm (GA)

Genetic Algorithm was first described by John Holland in the 1960s and further developed by Holland [17]. GA is one of evolutionary algorithms. The basic idea of the algorithm is that hereditary characteristics are transmitted by the genes. These characteristics are transferred to the next generation by the chromosomes [18, 19]. Each gene in the chromosome represents a property. Chromosomes are reproduced by the crossover and the mutation and create a new generation. There are many methods how to select the best chromosomes, for example roulette wheel selection, tournament selection, rank selection and some others[20-22].

C. Multi-Objective Optimization

In this article, support and confidence are considered as two objective functions. A rule is optimal if it possesses high quality and the quality of the rules is assessed by taking the weighted average of support and confidence of the rule. Thus, for an association rule $A \rightarrow B$, the quality of the rule is defined as:

$$\text{quality}(A \rightarrow B) = w_1 \times \text{sup}(A \rightarrow B) + w_2 \times \text{conf}(A \rightarrow B) \quad (3)$$

Where, w_1 and w_2 are the weights assigned to support and confidence measures respectively so that $w_1 + w_2 = 1$ and $w_1, w_2 \in [0,1]$ [14].

IV. GENETIC ALGORITHM-BASED ASSOCIATION RULE MINING (GARM)

Association rules are discovered with genetic algorithm. The recommender system produces recommendations using these rules. Association rules are produced for target objects. Target objects can be items or users.

A. Representation Scheme

Association rule $A \rightarrow B$ has two disjoint item sets A and B, where A is called the body of the rule and B is head of the rule [25] (Fig 1). In this paper, recommender system requires rules with fixed head and there is no need to update the head of the rule. These types of rules could

be discovered more efficiently than the rules with arbitrary heads. Therefore, Algorithm GARM discovers rules for a single target object. Each of the chromosomes in the search space is represented by the body of the rules ($A = \{A_1, A_2, \dots, A_d\}$). The set of $\{A_1, A_2, \dots, A_d\}$ is binary encoded for Item A. For example, the rule $I_5 \rightarrow I_8$ is represented (0101).

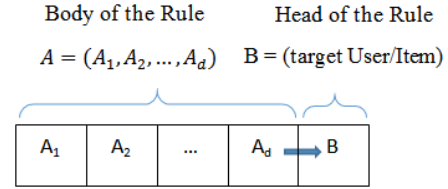



Fig.1. Chromosome encoding.

B. Transformation Schema

In order to association rule mining from the transaction dataset, a transformation scheme is needed, which can accelerate the process of data scanning [14]. In the left table of Fig 2, 12 ratings (1-5) from 5 users on 4 Items. Transformation scheme has been shown in the right table below. To transform the dataset, the average votes were calculated for each user, and then, the votes lower than the average were considered zero, and the votes higher than the average were considered one. For example, the average votes for User 4 is $3 (\frac{1+5+2+4}{4})$, $I_2 = 5$ and $I_4 = 4$ are higher than the average, therefore I_2 and I_4 were considered one. By this way, the transaction dataset into binary form, support and confidence can be calculated more quickly [26].

User Id	Item Id			
	I ₁	I ₂	I ₃	I ₄
User 1	3	0	1	2
User 2	0	1	4	0
User 3	0	0	0	2
User 4	1	5	2	4
User 5	4	0	0	2



User Id	Item Id			
	I ₁	I ₂	I ₃	I ₄
User 1	1	0	0	1
User 2	0	0	1	0
User 3	0	0	0	1
User 4	0	1	0	1
User 5	1	0	0	0

Fig.2. Transformation of transaction database

C. Garm Algorithm Schema

The steps of GARM Algorithm as follows:

Step1: In the first generation, a number of the chromosomes are produced. These chromosomes are produced randomly.

Step2: Pairs of the chromosomes are selected by selecting the function. Then, crossover and mutation operations are performed on them. Crossover and mutation probabilities were taken respectively as 0.8 and 0.05. This process continues until the new generation is not improved. The quality of the rules is measured by Equation 3.

Step3: Rules in each generation are saved with their quality in a database.

Step 4: This database will be the input of the recommender system.

The pseudo code is as follows:

GARM algorithm

Input: Transformed Dataset D_t , Target Object t

Output: Top-k association rules

Step1: set_rules \leftarrow Randomly initializing

Step2: while (Not converge)

(par1, par2)

= Selection Function (Set_rules)

(CH1, CH2) \leftarrow

Crossover (par1, par2, probability = 0.8)

(ch1, ch2)

\leftarrow Mutation (CH1, CH2, probability = 0.05)

Fitness by $\text{quality}(\text{ch1} \rightarrow t) = W_1 * \text{sup}(\text{ch1} \rightarrow t) + W_2 * \text{conf}(\text{ch1} \rightarrow t)$

$$\begin{aligned} \text{quality (ch2} \rightarrow \text{t)} \\ = W_1 * \text{sup(ch2} \rightarrow \text{t)} + W_2 \\ * \text{conf (ch2} \rightarrow \text{t)} \end{aligned}$$

Update Set_rules

End

Step3: Find Top-k rules in set_rules

Step4: send Top-k rules for Recommender System

The algorithm is explained by an example. Table 1 shows an example of a dataset. It is assumed that I_2 is the target object.

Step1: Two rules $9 \rightarrow 2$, $7 \rightarrow 2$ are produced randomly. The body of rules is encoded (Fig 3).

Step2: Two rules $3 \rightarrow 2$, $1 \rightarrow 2$ are produced in a new generation (Fig 5, 5).

Step3: The quality of the rules is measured by the Equation 3 and saved in the database. It is assumed that $w_1=0.6$ and $w_2=0.4$ Table 2 shows quality of two rules.

Table 1. Example of dataset.

User Id	Item Id								
	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈	I ₉
U ₁	0	1	0	1	1	0	0	1	1
U ₂	1	0	1	0	0	1	1	1	1
U ₃	0	0	1	0	0	1	1	0	0
U ₄	0	1	1	1	1	0	0	0	0
U ₅	1	0	1	0	1	0	0	1	1
U ₆	1	1	1	0	0	0	1	1	1
U ₇	0	0	0	1	0	0	1	0	0

1	0	0	1
0	1	1	1

Fig.3. Chromosomes encoding of 9 and 7.

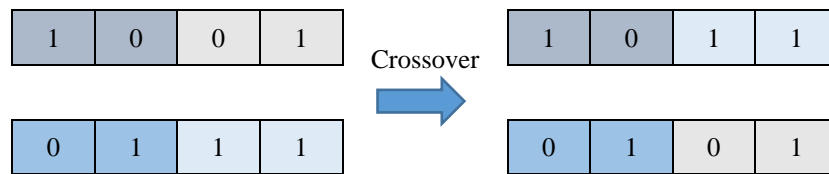


Fig.4. Crossover.

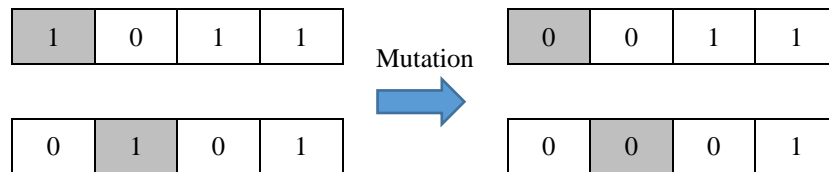


Fig.5. Mutation.

Table 2. Quality of the rules.

Rules	Sup	Conf	Quality
$3 \rightarrow 2$	0.28	0.4	0.328
$1 \rightarrow 2$	0.14	0.33	0.216

V. RECOMMENDATION STRATEGY

The recommender system produces the suggestions based on association rule mining. The association rules are discovered in the form of $(I_1, I_2) \rightarrow I_t$. If the target user likes items I_1 and I_2 , that user would also like item I_t . These types of rules could be discovered offline and can recommend a target item it to a target user u_i if the user likes both items i_1 and i_2 [14]. The proposed model in this article has been shown in Fig 6.

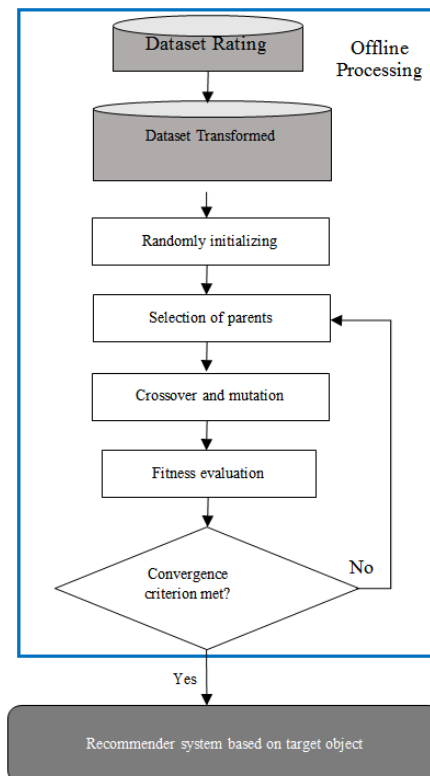


Fig.6. Recommendation strategy model

VI. EXPERIMENTAL EVALUATION

In this section, the genetic algorithm-Association Rule Mining (GARM) has been evaluated. The GARM algorithm has been compared to the Multi-Objective Particle Swarm Optimization Association Rule Mining (MOPSO) algorithm [14]. Experiments show that the GARM algorithm outperformed the MOPSO algorithm. These experiments were conducted using Movielens 1M dataset.

A. Dataset

The experiment uses Movielens 1M dataset. MovieLens datasets were collected by the GroupLens research project at the University of Minnesota. The data were collected through the MovieLens website (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. Movielens 1M contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 users, each user has rated at least 20 movies.

Two subsets of Movielens 1M were considered. In the first subset, 1500 users were selected randomly from the users who had given less than 50 votes (called sparse dataset). In the second subset, 1500 users were selected randomly from the users who had given more than 800 votes (called density dataset). Table 3 shows the properties of the datasets that have been used.

Table 3. Datasets used in the experimental analysis.

Dataset	S/D	#Users	#Items	Min. #Voted	Max. #Voted
Sparse dataset	S	1500	3952	20	50
Density dataset	D	1500	3952	800	1932

B. Evaluation Metrics And Impact Of Parameters

The GARM algorithm is converge when there is no significant improvement in the values of fitness of the population from one generation to the next. Converge is stopping criteria. There are two evaluation criteria. The first one is the average quality of the rules in each iteration. The other one is executed time. To evaluate the parameters effect, 100 target items were selected randomly. The quality of the rules are measured by Equation 3. The experiments were executed on CPU Doul-core E5400@2.70 GHz and 2G RAM machine, and software was MATLAB (2013).

In the first pretest, the initial population of 4, 10 and 16 were evaluated. The GRAM algorithm for Iterations 30, 40, 50, ... were performed. Selection function is Non-selective. Non-selection means that the new generation is replaced with the previous generation. Fig 7 shows the convergence of initial population 4 is achieved in the 190th iteration and 167th minutes, the convergence of initial population 10 in the 80th iteration and 163th minutes, and the convergence of initial population 16 in the 50th iteration and 173th minutes. Therefore, the best solution is achieved with initial population 10. The convergence of initial populations are shown in Table 4.

In the second pretest, the GARM algorithm for different selection functions have been evaluated. Four selective functions are evaluated: Roulette wheel, Rank, Non-selective, and Elitism. Fig 8 shows the average convergence of selection functions for different iterations. The convergence of Roulette wheel is achieved in 50th iteration and 102th minutes, the convergence of Rank in 60th iteration and 120th minutes, the convergence of Non-selection in 80th iteration and 163th minutes, the convergence of Elitism in 60th iteration and 122th minutes. Therefore, the best solution is achieved with Roulette wheel. The convergence of selection functions is shown Table 5.

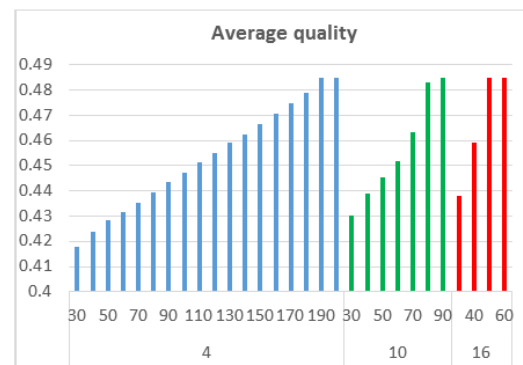


Fig.7. Average quality in each iteration for different populations (Horizontal axis: first row Iteration, second row population)

Table 4. The convergence of initial populations.

Initial population	Iteration of convergence	Moment of convergence
4	190	167 min
10	80	163 min
16	50	173 min

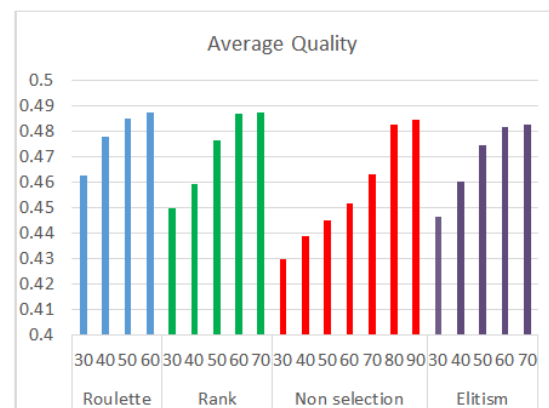


Fig.8. Average quality in each iteration for different selection functions (Horizontal axis: first row Iteration, second row selection functions)

Table 5. The convergence of selection functions.

Selection function	Iteration of convergence	Moment of convergence
Roulette wheel	50	102 min
Rank	60	120 min
Non selection	80	163 min
Elitism	60	122 min

The third pretest is designed to assess the effects of weights w_1 and w_2 on recommendation quality. Fig 9 shows the tradeoff between the precision and recall becomes evident for higher and lower values of w_2 . From these observations, the best result sets $w_1=0.6$ and $w_2=0.4$. The best solution of these three pretests are shown Table 6.

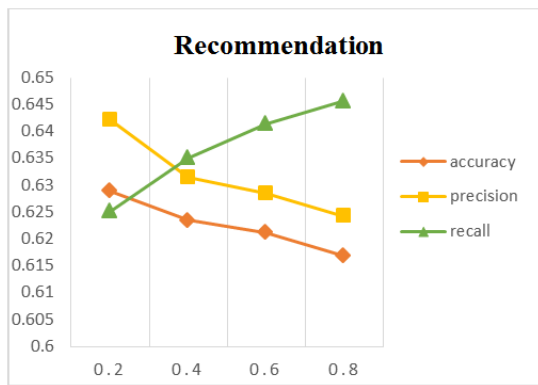


Fig.9. Impact of weight w_2 on recommendations quality ($w_1+w_2=1$)

Table 6. The parameter settings.

parameters	values
Initial population	10
Selection function	Roulette wheel
Iteration convergent	50
w_1	0.6
w_2	0.4

C. Experiments

The proposed technique was implemented on different datasets. The execute time are measured at the moment of convergence for GARM and MOPSO algorithms. The effectiveness of collaborative filtering algorithms has

been measured by accuracy, precision, recall and F1-measure. To evaluate the recommender system the four-fold cross-validation method has been used. Then, the dataset was randomly split into training and test data with a ratio of 75%|25%. The parameters are measured, defined as:

$$\text{Accuracy} = \frac{\text{number of correctly classified items}}{\text{total items classified}} \quad (4)$$

$$\text{Precision} = \frac{\text{number of correctly recommended items}}{\text{total recommended items}} \quad (5)$$

$$\text{Recall} = \frac{\text{number of correctly recommended items}}{\text{total items liked by user}} \quad (6)$$

$$\text{F1 - measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (7)$$

Table 7 and Table 8, the results obtained from GARM algorithm are compared with results from MOPSO algorithm. The results show that the execute time to make recommendations when using GARM algorithm is 12% lower than the execute time when using MOPSO algorithm, while the quality measures (accuracy, precision, recall and F1-measure) are almost equal. Fig 10 shows comparison of GARM and MOPSO based on execute time.

VII. CONCLUSIONS AND FUTURE WORK

Association rule mining is a multi-objective and is not a single objective problem. This article uses a weighted quality measure based on the support and confidence of association rules to solve the multi-objective rule mining problem. Association rules are discovered with genetic algorithm. The minimum support and minimum confidence are not specified by the user. Thus, the most interesting rules are discovered, according to the fitness function. Genetic algorithm is efficient for search space, therefore association rules are discovered faster than the traditional methods. In order to improve proposed algorithm efficiency, the initial population and selection functions were evaluated. Initial population 10 and roulette wheel was the best of results. GARM algorithm is evaluated by comparing to MOPSO algorithm. Experimentation shows that our algorithm outperforms the MOPSO algorithm.

Table 7. Comparison of GARM and MOPSO (Execute for sparse dataset)

Methods	#Items	Execute time (hours)	Accuracy	Precision	Recall	F1-measure
GARM	100	2.48	0.5633	0.5583	0.7943	0.6557
	250	6.13	0.5647	0.5597	0.7952	0.6570
	500	12.38	0.5677	0.5609	0.7973	0.6585
MOPSO	100	2.87	0.5631	0.5580	0.7943	0.6555
	250	7.15	0.5648	0.5593	0.7953	0.6567
	500	14.08	0.5673	0.5603	0.7972	0.6581

Table 8. Comparison of GARM and MOPSO (Execute for density dataset)

Methods	#Items	Execute time (hours)	Accuracy	Precision	Recall	F1-measure
GARM	100	2.25	0.5923	0.5983	0.7963	0.6832
	250	5.51	0.5934	0.5997	0.7975	0.6846
	500	11.15	0.5939	0.6021	0.7982	0.6864
MOPSO	100	2.56	0.5924	0.5980	0.7965	0.6831
	250	6.01	0.5931	0.5991	0.7978	0.6843
	500	12.79	0.5940	0.6012	0.7983	0.6858



Fig.10. Comparison of GARM and MOPSO based on execute time

In the present work, to improve the quality of recommendation, items anthology uses for identifying similar among items. To identify similar through association rules and items anthology are improved the recommendation quality. In this way, first recommendation produced by the GARM algorithm thereafter identifying similar among items by items anthology uses to make recommendations. Also in this work speed of running algorithm were low because of checking the database to evaluate each rule, take a lot of time, but with indexing transaction can have a more efficient algorithm for future work.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 734-749, 2005.
- [2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175-186.
- [3] A. Ansari, S. Essegai, and R. Kohli, "Internet recommendation systems," *Journal of Marketing research*, vol. 37, pp. 363-375, 2000.
- [4] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, pp. 331-370, 2002.
- [5] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 39-46.
- [6] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, *et al.*, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 114-121.
- [7] S. Tyagi and K. K. Bharadwaj, "A Collaborative Filtering Framework Based on Fuzzy Case-Based Reasoning," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*, 2012, pp. 279-288.
- [8] G. Prati, M. De Angelis, V. M. Puchades, F. Fraboni, and L. Pietrantoni, "Characteristics of cyclist crashes in Italy using latent class analysis and association rule mining," *PLoS one*, vol. 12, p. e0171484, 2017.
- [9] M. Grami, R. Gheibi, and F. Rahimi, "A novel association rule mining using genetic algorithm," in *Information and Knowledge Technology (IKT), 2016 Eighth International Conference on*, 2016, pp. 200-204.
- [10] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 116-142, 2004.
- [11] C. W.-k. Leung, S. C.-f. Chan, and F.-l. Chung, "A collaborative filtering framework based on fuzzy association rules and multiple-level similarity," *Knowledge and Information Systems*, vol. 10, pp. 357-381, 2006.
- [12] W. Lin, S. A. Alvarez, and C. Ruiz, "Efficient adaptive-support association rule mining for recommender systems," *Data mining and knowledge discovery*, vol. 6, pp. 83-105, 2002.
- [13] H. H. Varzaneh, B. S. Neysiani, H. Ziafat, and N. Soltani, "Recommendation Systems Based on Association Rule Mining for a Target Object by Evolutionary Algorithms," *Emerging Science Journal*, vol. 2, 2018.
- [14] S. Tyagi and K. K. Bharadwaj, "Enhancing collaborative filtering recommendations by utilizing multi-objective

particle swarm optimization embedded association rule mining," *Swarm and Evolutionary Computation*, vol. 13, pp. 1-12, 2013.

- [15] M. K. Najafabadi, M. N. r. Mahrin, S. Chuprat, and H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Computers in Human Behavior*, vol. 67, pp. 113-128, 2017.
- [16] Q. Wang, C. Zeng, W. Zhou, T. Li, S. S. Iyengar, L. Schwartz, *et al.*, "Online interactive collaborative filtering using multi-armed bandit with dependent arms," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
- [17] T. I. R. Agrawal, A. Swami, "Mining association rules between sets of items in large databases," *Proceeding of the ACM International Conference on Management of Data, ACM*, pp. 207-216, 1993.
- [18] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, 1994, pp. 487-499.
- [19] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, pp. 66-72, 1992.
- [20] M. M. J. Kabir, S. Xu, B. H. Kang, and Z. Zhao, "A New Multiple Seeds Based Genetic Algorithm for Discovering a Set of Interesting Boolean Association Rules," *Expert Systems with Applications*, 2017.
- [21] N. Soltani, B. Soleimani, and B. Barekatin, "Heuristic algorithms for task scheduling in cloud computing: a survey," *International Journal of Computer Network and Information Security*, vol. 9, p. 16, 2017.
- [22] N. Soltani, B. Barekatin, and B. S. Neysiani, "Job Scheduling based on Single and Multi Objective Meta-Heuristic Algorithms in Cloud Computing: A Survey," in *Conference: International Conference on Information Technology, Communications and Telecommunications (IRICT)*, 2016.
- [23] B. S. Neysiani, N. Soltani, and S. Ghezalbash, "A framework for improving find best marketing targets using a hybrid genetic algorithm and neural networks," in *Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on*, 2015, pp. 733-738.
- [24] M. Davarpanah, "A review of methods for resource allocation and operational framework in cloud computing," *Journal of Advances in Computer Engineering and Technology*, vol. 3, pp. 51-60, 2017.
- [25] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, 1993, pp. 207-216.
- [26] S.-Y. Wur and Y. Leu, "An effective Boolean algorithm for mining association rules in large databases," in *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, 1999, pp. 179-186.

Knowledge Discovery and Data mining and Machine learning and published many papers in these fields. Now he's working on Text Mining for his PhD dissertation.



Nasim Soltani received her B.Sc. Degree in Computer engineering from Payam Noor University, and her M.Sc. degree from Allame Naeini Higher Education Institute, in 2013 and 2016 respectively. She has published several paper in Cloud Computing field in many conferences and journals. Her main research interests include Cloud Computing, Task Scheduling, Distributed Systems, Recommender system and Data Mining.



Reza Mofidi received his M.Sc. degree from Allame Naeini Higher Education Institute, in 2015. He has published several paper in Recommender system field in many conferences and journals. His main research interests include Recommender system, Collaborative filtering, Association rule mining and Data Mining.



Mohammad Hossein Nadimi-Shahraki is Assistant Professor with a demonstrated history of working in the higher education industry and research projects. Skilled in Computer Science, Databases, Data Mining, Big Data Analytics, Machine Learning, and Meta-heuristic Algorithms. He has Strong education professional with a Doctor of Philosophy (Ph.D.) focused in Artificial Intelligence - Data Mining from University Putra Malaysia.

How to cite this paper: Behzad Soleimani Neysiani, Nasim Soltani, Reza Mofidi, Mohammad Hossein Nadimi-Shahraki, "Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm", *International Journal of Information Technology and Computer Science(IJITCS)*, Vol.11, No.2, pp.48-55, 2019. DOI: 10.5815/ijitcs.2019.02.06

Authors' Profiles



Behzad Soleimani Neysiani is a lecturer at Islamic Azad University (Isfahan Branch). He received the B.Sc. and M.Sc. degrees in Computer Engineering from Islamic Azad University (Najaf Abad Branch), in 2008 and 2013 respectively. He's a PhD Candidate in University of

Kashan now. His research interests include Distributed Computing especially Clouds and P2P systems, Software engineering especially software testing, Database Management,