IJICT International Journal of Information & Communication Technology

ITRC

Volume 1- Number 1- March 2009 (pp. 39-47)

A New Recommender System based on Cooperative Co-evolution Algorithm

Mohammad Reza Ahmadi
Iran Telecom. Research Center
Tehran, Iran
m.ahmadi@itrc.ac.ir

Abstract— **Expansion of global networks and storing the extensive amount of information in various websites, create a serious needs for filtering the irrelevant information in a personalized way. Collaborative filtering or recommender system is a filtering technique that allows incorporation of the profiles which can be implicitly learned from previous activities [1]. We have proposed the CoCo-CF[1] as an effective method suitable for collaborating filtering running in a Jini-grid computing[2] platform and operational in a distributed environment. The CoCo-CF generates representative records from stored preferences and seeks for the answer with the best fitness in the recommender system. We have considered the user satisfaction rate, feasibility of available results, user familiarity and average response time as the evaluation factors. Also we have focused on mean absolute deviation, mean square error and ranked evaluation as the performance evaluation parameters. The obtained results confirm that the CoCo-CF is a successful method for collaborative filtering.**

Keywords- Genetic Algorithm, Cooperative co-evolutionary algorithm, Collaborative Filtering, Recommender systems.

I. INTRODUCTION

Internet is a new technology which provides a global communication network. Demand for this network has exploded in the last few years into an exponential growth and the available knowledge penetrated to all aspect of the daily life. In this distributed environment, finding specific information is becoming a complicated task. Any searching process using the World Wide Web turns out thousands of results and a high percentage of this information is not effective and more often than not irrelevant as well. Internet users are overwhelmed by the flow of online information, hence the need for adequate systems that will help them manage such situations. Collaborative filtering systems recommend users with items that people with similar preferences liked in the past. Recommendation systems are implemented in web sites to allow users to locate the preferable items quickly and to avoid the possible information overloads. Collaborative filtering techniques have been successful in enabling the prediction of user preferences in the recommendation systems. There are three major processes in the recommendation systems: object data collections, similarity decisions, and recommendation computations [2]-[5]. A number of methods have been developed for the "collaborative filtering". If we need to choose between varieties of options which we do not have any experience, we will often rely on the opinions of others who do have such experience. However, when there are thousands or millions of options, like in the Web, it becomes practically impossible for an individual to locate reliable experts that can give advice about each of the options. As a simple solution, instead of asking opinions of each individual, you might try to determine an "average

[1] Cooperative co-evolutionary collaborative filtering.

[2] A parallel processing approach in large scale computational problems over a network of multiple distributed computers.

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

opinion" of the group. This, however, ignores your particular interests, which may be different from those of the "average person". You would rather like to hear the opinions of those people who have interests similar to your own, that is to say, you would prefer a specialized domain. If the similarity metric has indeed selected the people with similar tastes, chances are great that the options evaluated by that group will also be appreciated by the advice-seeker. A typical application is the recommendation for selection of books, movies, services or products of any kind [6]-[9]. For this recommendation procedure, we introduce a new method of collaborative filtering based on genetic algorithm with cooperative co-evolutionary model using grid computational technique in distributed environment [10]. The motivation for this approach lies on integrating the immune system and cooperative co-evolutionary concept using an agent based grid computing in a distributed environment as an intelligent solution for selecting a recommended option in a distributed data networks. In our approach, we have considered the previous successful experiences as the target for recommender system. We have index all the targets in a pre-defined structure categorize in different classes which are accessible from distributed points. We model all the incoming request based on its key parameters and assign a specific coding record for each item. A distributed model of genetic algorithm tries to find the best fitness solution for all the requests in a distributed environment. The solutions with acceptable fitness are the recommended answers for incoming requests. The rest of paper is organized as follows: Section II explains the concept of collaborate filtering. Section III presents the CoCo-CF system. Section IV focuses on the CoCo-CF operational procedure. Section V shows the performance evaluation. Finally, we conclude the paper in section VI

II. CONCEPT OF COLLABORATIVE FILTERNG

Collaborates filtering (CF) is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources etc. This method brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data. This technique is based on making automatic predictions (filtering) about the interests of a user by collecting test information from many users (collaborating) [11],[12]. The underlying assumption of CF approach is that those who agreed in the past tend to agree again in the future. There are varieties of methods and technique for accessing to the right requests in the webs such as active filtering, passive filtering and Item based filtering [13]. On the other hand, collaborative filtering is characterized by concept which may change in the real world. To make time-critical predictions, we should consider that the target users' recent ratings reflect the future preferences more than older ratings. Therefore, for a more realistic prediction, we need knowledge about the present, future and the past with an intelligent capability.

A. *Collaborative Filtering Task*

The task of CF is predicting the preference of a user assigns to items based on preference data of that users and preference data of other users. One way for ranking the preferences is encoding the items by a numeric score such as one, two and so on. A different setting is that of implicit preference data. In this case, the users do not explicitly enter their preference; but, their actions are recorded and interpreted as preference assertions. The CF system usually reports its output in two different forms. If the user reports his preferences using numeric values, the system may try to predict the numeric value associated with the predicted items. This may be reported as a probability vector for each predicted item or summarized by statistics such as the mean and variance of the probability vector. Alternatively, the predicted items may be sorted as a list and presented to the active user without the predicted numeric scores. In the first case, performance is often measured by computing the distance between the mean of the predicted numeric scores and the actual preference values. In the second case, the ranked list may be evaluated using expected utility of a user selecting items from that list with probability exponentially decreasing in the rank of the item. Traditionally, CF systems completely ignore any content information about the items. For example in the case of movie recommendation, movie features such as length, language etc. are ignored. The predictions for the active user are made only on the basis of his/her and other users' predictions. The system is ignorant as to whether the items being recommended are movies, web pages, songs or anything else. Recently, some attempts have been made at incorporating item features into CF system. This is an interesting prospect that seems to increase the recommendation performance [16],[17]. In the following section we have proposed a new method based on cooperative co-evolutionary immune system called CoCo-CF. This technique is based on biological system and expands the concept of collaborative filtering for distributed environment.

III. THE COCO-CF SYSTEM

In this section, we introduce architecture of our cooperative co-evolutionary immune system using for collaborative filtering. In continue the system structure and evaluation parameters are discussed.

A. *The CoCo-CF Architecture*

A genetic algorithm is a search technique used in computing to find exact or approximate solutions to search and optimization problems. The GA [3] algorithm repeatedly modifies with genetic operators and seeking for the answer with the best fitness. On the other hand, the co-evolution algorithm is an

[3] Genetic Algorithm

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

extended version of GA with multiple groups of populations. In addition, the Cooperative Co-evolutionary method includes several genetically isolated groups that evolve in a parallel model. The individual member from each group collaborates with other members through a representative population[4] and improves their fitness according to a specific objective function [10]-[13]. The core of this technique is applicable to most of the optimization problem where each problem needs specific approach [14]. The structure of procedures, system parameters, fields of information and progress strategy specialized for each application system. Here, we have applied the Co-Co method to find the exact or approximate recommendation solution as a collaborative filtering technique This approach focuses on considering the previous experiences as the target solution and seeking for the possible answer with the best fitness. We have implemented the CoCo-CF algorithm in Jini-Grid environment running in a distributed model. Figure 1 introduces the architecture of the CoCo-CF algorithm.

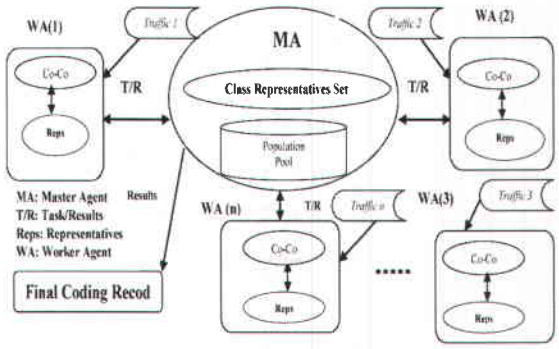


Figure 1. The Architecture of CoCo-CF in a distributed environment.

This system consists of several set of separated group of agents where each set manages an individual group and they are coordinating by a master agent management. The number of computational process depends on type of information and classification of data. Each CoCo-CF class concentrates on a specific group of information. Based on figure 1, the initial data is stored in a population pool. This population of data is divided into several sub-populations, where each group evolves in a specific category. In the first level of process, a particular co-evaluation algorithm in each group cooperates with other groups under a master agent management. Each group generates a set of representatives which has the best fitness with all group members. The final set of representative will generate from the entire representative groups in different classes. This method is an effective technique which will suggest the best matching record by using the existing information in a distributed environment.

[4] A set of the best individuals in a group.

B. *The CoCo-CF Process Modes*

In this part we introduce the functionality of CoCo-CF algorithm. At first, we create a data pool based on the observed items and user preferences. As we add more data to the database, the recommendation becomes more realistic. On the other hand, these preferences may change in time from user perspective and technology point of view. We classified all the data in different categories and assigned a digital coding record for each item. Also, we consider a string of states for each record. Thus, the database includes a set of digital records with related state process for all data set. Among all states in each record, the CoCo-CF selects the best fitness representative member. Moreover, among all representative members in each class, the algorithm selects the best set of class representative members. As a result, we have a set of class representative members which are stored in the system. This procedure completes the training phase and prepares the system for recommendation phase. In recommendation process, the system investigates to find the best matching member in the class representative record for each new entry. Then select the best class category and set of related states for each process. The result distinguishes the sequence of coding records which are the best candidate for recommender system.

C. *The CoCo-CF Mathematical Analysiss*

In CoCo-CF system, we have proposed a multi segment string schema for creation of coding records [9]-[11]. A creator consists of 128 bits threshold, 256 bits pattern field, 256 bits mask field and 128 bits classification field. Any state code includes 128 bits for main IP address and 128 bits extra for details of information. The format of coding record is shown in figure 2. The mask filed controls the data classification and the setting parameters. In order to create a state records, the mask field applies to the pattern field where the value of "1" generates a corresponding bit and the value of "0" generates a don't care (*X*) bit. As a result, we obtain the state records with three fields including the threshold field, 256 bits pattern field and finally the classification field. This structure will model a state record which binds a family of similar information with common characteristics. We consider an event pattern as a string of 256 bits with the following specification:

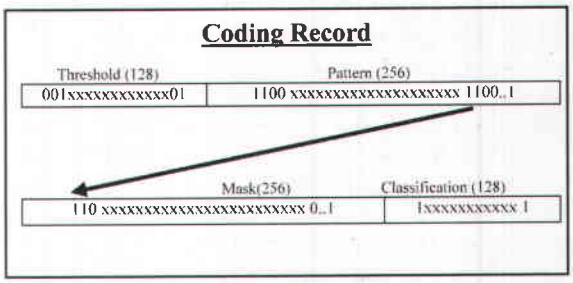


Figure 2. Coding record format

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

Record: A sequence of states (s_i) which starts and ends for any inquiry.
State record: A digital coding which is assigned to a specific state.
String record: A set of state records.

$$\text{Re } cord\ (1) : \{s^1_1, s^1_2, s^1_3, \ldots\ldots\ s^1_m\}$$
$$\text{Re } cord\ (2) : \{s^2_1, s^2_2, s^2_3, \ldots\ldots\ s^2_m\}$$
$$\ldots\ldots$$
$$\text{Re } cord\ (n) : \{s^n_1, s^n_2, s^n_3, \ldots\ldots\ .s^n_m\}$$
(1)

Where s^n_m refers to *m*-state *of n-th* record. It should be noted that the value of m may not be the same in different records. We consider a set of digital coding corresponding to all states as follows:

$$\text{Re } cord\ (i) = \{[xx\ldots x],[xx\ldots x],\ldots.[\ xx\ldots x]\}$$
(2)

where x has value of (0/1/don't care). On the other hand, r_i defines the representative record in string record i with m members.

$$CR_i = \{[\ r_1\ ],[\ r_2\ ],\ldots.[\ r_k\ ]\}$$
(3)

Also CR_i refers to class representatives and the TCR defines total class representative in the whole database.

$$TCR = \{[CR_1],[CR_2],\ldots.[CR_c]\}$$
(4)

The goal is to find the best total class representatives (TCR) where the CR_i members have the best fitness with s^n_1 members in its related record. To select the members with the best fitness, we should calculate the match-strength factor in the members. We define S as the match-strength factor in Record (i) with size of 256 bits. The value of S can be simply obtained by comparison of similar position bits in records and the requested information based on Eq. 5, where x is a digital value of the requested item and y is the string of state records:

$$S(x,y) = \sum_{i=1}^{l} \begin{cases} 1 & if\ [x_i]=[y_i] \quad or \quad x_i \neq X \\ 0 & else \end{cases}$$
(5)

here l=256 and X refers to don't care values. In order to find the maximum fitness for Record (i), we obtain a member which has the maximum match-strength based on Eq.6.

$$S_{\max_i}(s_1,s_i)=S(s_1,s_i) \Big| S(s_1,s_j) \geq S(s_1,s_k)$$
$$for \quad i=1\ldots m-1 \qquad for \quad j,k=1\ldots m \quad k \neq j$$
(6)

In set of representative members which are generated in existing records, representative members give a set of members which have the best fitness for each record. We assume CR_i has the maximum match-strength (best fitness) in set of recording members where:

$$TCR = \sum_{i=1}^{n} CR_i$$
(7)

The prototype system is implemented using this procedure and the system is capable for recommendation process.

IV. THE COCO-CF OPERATIONAL PROCEDURE

In CoCo-CF model, we have implemented the prototype system in a Jini-Grid platform running in a distributed environment. We have considered three phases in system processing. In initialization phase, we load the system with previous user preferences records. In training phase, the system generates a set of digital codes for all members (states) in each record. Then separate the members in different classes. Among all members in each class, the algorithm generates a set of representative member(s). Then, the system creates a set of class representative member(s) for all representative records. The class representative member(s) are the best reference members in the related class. A class representative member has the following characteristics: First, any request which has the best fitness with one class representative member(s) has the highest priority for becoming a qualified answer. Second, the best matching representative member distinguishes the best state record for an incoming request.

CoCo-CF Operational Procedure

Step 1: Create a data pool for user preferences
Step 2: Classify the data based on subjects
Step 3: Create the digital coding records
Step 4: Distinguish all the state records
For all class population
{
Create the representative record in each state process,
Create the representative record in each class,
Distinguish the class representative members,
For any incoming request,
{
Select the best fitness member in class representative,
Select the best fitness member in representative record,
}
Recommend the best record for an entry.
}
End.

Figure 3. The recommender system algorithm

In operational phase, the system creates a digital coding record for each entry. Then, the algorithm seeks for the class representative members with the best fitness. As a result, the class representative members which have the best fitness with the incoming record is the best qualified option for

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

recommendation. Figure 3 shows the main procedure for recommender system.

A. Convergence of the Algorithm

CoCo-CF system has the responsibility to generate the best coding records based on an archived data and recommending the best possible options for the new requests. In this case, cooperation of the members for the best fitness value is an important procedure. In order to evaluate the contribution of members in the algorithm, we have implemented a prototype system and evaluate the system response in consecutive generations. The system is operational in a Jini-grid platform with Microsoft anonymous web data [15]. The details of system architecture and technical specifications are based on section 3. To generate the representative records, the process starts from worker agents in one group where the agents initiate their activity by generating a random preliminary population in the system. During the process, cooperation of the members should converge for the best fitness. Otherwise, the matching rate will stagnate. In this situation, it is necessary to add new members to the system and any unproductive members should be eliminated from population. Once the desired numbers of generations have completed or the pre-defined conditions are achieved, the process will terminate.

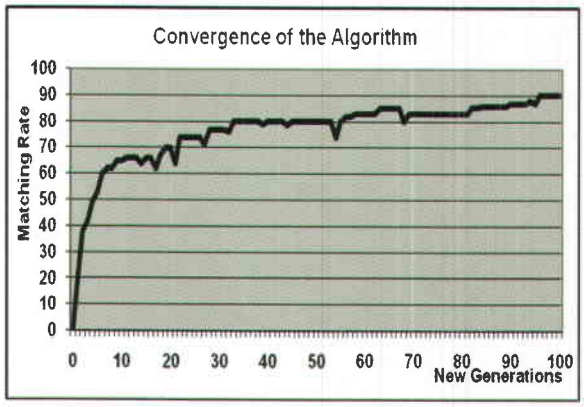


Figure 4. CoCo-CF system response.

In practice, each class includes several group of populations which cooperate together to include the whole data classes. The progress of matching rate should improve continuously; otherwise, the system will stagnate. In this case, we need to generate more groups of data in the system. The system response shows that by increasing the number of groups, contribution of members may downfall during several steps. In this case, the fitness will reduce and the system response may deviate from expected direction. This temporary behavior will recover quickly by improving the cooperation among existing and new members as it is shown in Fig.4. The result of this evaluation confirms that in the proposed system, the algorithm encourages the majority of population for cooperation and the system does not stagnate or deviate during the process.

V. Performance Evaluation

In this section we have evaluated the capability of CoCo-CF system in recommendation process.

A. System Evaluation

The metrics that we have discussed so far involve defining the variables that we believe will affect the utility of a recommender system for the users and affect the reaction of the users to the system. In this section we face the question of how directly evaluate user "reaction" to a recommender system. For this reason, we have considered several scenarios which are important for better evaluation and more clear judgment about the proposed algorithm.

a) Satisfaction rate evaluation

In this model, we have evaluated the user satisfaction rate for CoCo-CF algorithm in the recommender system. We have implemented the prototype system based on sec.3 in a Matlab environment using C programming language. We have applied the user previous exercises in the database for system initialization [15]. The algorithm generates representative records and class representative records for all class of information in the database. Then, we have applied the new requests to the system and encourage the users to use the recommended options as well as the self trial procedure.

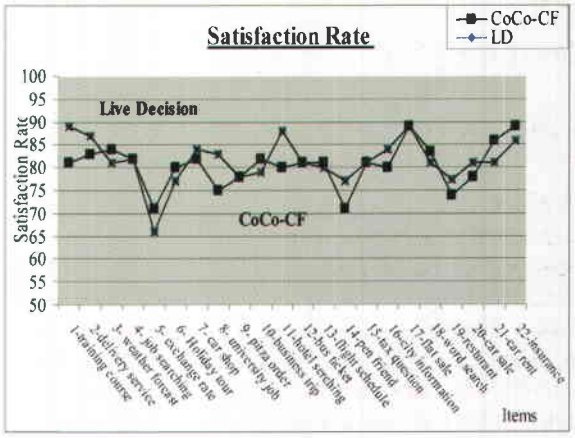


Figure 5. Comparison between live searching and recommender system.

Figure 5 compares the results of the recommender system and live user investigation. This figure compares the satisfaction rate for recommended options and live decisions in 22 different items. The obtained results show that the average satisfaction rate in CoCo-CF is higher than live trials because of variety of available options and complexity of the tasks in the live systems. On the other hand, improving the satisfaction level is very time consuming in the live process and the user should investigate in many options to find an expected result. The level of satisfaction rate can improve in recommender system by increasing the number of trials in the reference database. Moreover, the live satisfaction level depends on many functions such as,

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

selected item, available options and power of search engine. On the other hand, in a recommender system, it is very important to measure how often the system leads its users to a wrong recommendation. As it is shown, the gap between predicted and actual rating is not great concern as they move in a reasonable boundary. As a result, the evaluation of satisfaction rate certifies that the CoCo-CF algorithm is an effective and successful method for filtering the irrelevant information and improving the satisfaction rate compare to live searching procedures.

b) Results availability evaluation

In this model, we consider the possibility that an available result exist in a live trial or in a recommender system. Technical specification and evaluation model are the same as previous scenarios. Figure 6 compares the possibility of successful available results using CoCo-CF recommendation algorithm and live searching procedure. The results for the two searching methods show that the recorded answer has higher level of availability compare to the live searching in the global website.

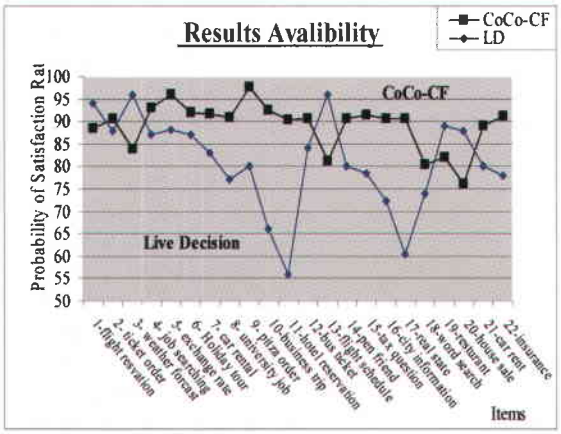


Figure 6. Results availability.

It is obvious that in the recommending mode, the expected results are available in the database where seeking a new answer in an environment with huge similar items is a very difficult task and may not easily meet the user satisfaction. On the other hand, many un-wanted side-searching procedures miss guide the process from the main direction. Moreover, the unavailability of expected information or long accessing delay will reduce the speed of search process and increase the number of fly-back steps. As a result, probability of available results and satisfaction rate in CoCo-CF algorithm is higher than live search method because of guarantee for the stored options in the system where in the live searching method; there is not any guarantee for finding an expected result.

c) User experiences evaluation

This scenario certifies that the user's familiarity with searching procedures is an important factor for satisfaction rate. Practically, the ability of a user in selecting the suitable options and ignoring the irrelevant information will affect the searching time and number of fly-back process. We implement the algorithm similar to previous scenarios and initialized the system with data from previous user experiences and trained the system for operation. In practice, we have compared several groups of users who had different level of familiarity with 22 subjects (from novice group for the subject to the professionals). Each user based on his/her familiarity level may choose different approaches and focuses on different options. Any option may bring different results with different satisfaction level. There is possibility that a professional user proceed more successful than the stored records in the reference database. In this case, the satisfaction level exceed from recommender system.

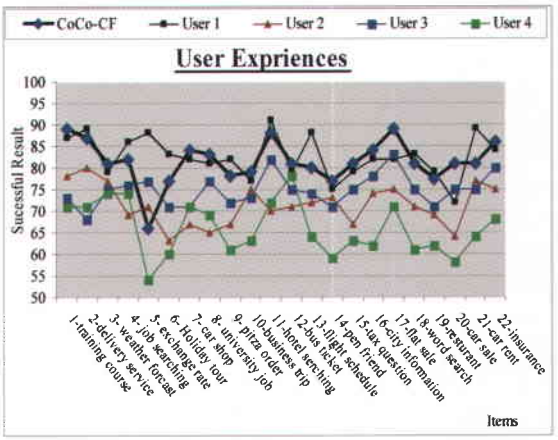


Figure 7. User experiences.

The obtain results certify that the familiarity of a user plays a key role in satisfaction level. Figure 7 compares the successful results for four groups of users which have a little familiarity, familiar, good familiarity and professionals with the subject together with the CoCo-CF algorithm. The results show that based on user familiarity with the item, the obtained results have different levels of satisfaction. For the users who have more familiarity, the results are more successful where for the professional users, the satisfaction rate may exceed from system recommendation.

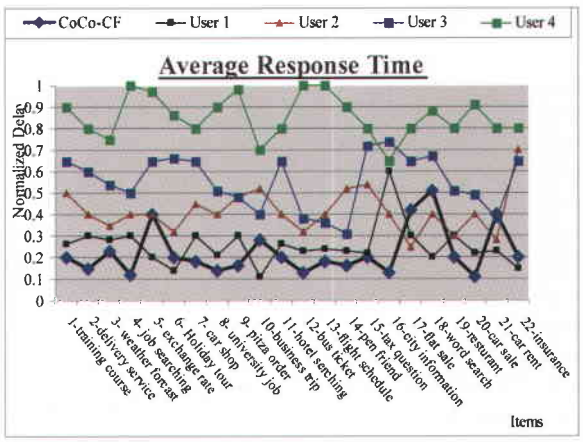


Figure 8. Average response time

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

d) Average response time evaluation

In this scenario we will evaluate the average response time which spends for obtaining the expected results where it was transparent in the previous evaluations. Period of time which spends for finding the expected results depends on several parameters such as power of search engine algorithm, user familiarity with item, wrong decision, fly-back for alternative options, user agility, application response time, and so on. Figure 8 compares five scenario using CoCo-CF algorithm and four groups of users with different familiarity levels. The average delay is normalized between zero and one. The zero value refers to minimum delay where the value of one refers to maximum delay. As depicted in figure 8, the CoCo-CF algorithm proceeds in minimum level where the novice group tolerates the maximum delay. The other groups will rank between those two groups. The results of evaluation proof that among five cases the CoCo-CF algorithm has more agility and the users consume less delay for answer.

B. Performance Comparison

In addition to the evaluations in section A., we need to compare the CoCo-CF algorithm with several well-known methods [18]-[20]. In order to certify the performance of CoCo-CF system, we have investigated three parameters including mean absolute deviation (MAD), mean square error (MSE) and ranked evaluation (RE) which measures the expected true preference of the chosen item when the probability to choose a recommended item proceed exponentially with its location in a sorted list of recommendations. The main computational modules are implemented in C programming language and are called in a Matlab environment. We supply the functions for loading, handling and evaluating of three other algorithms which are proposed in the recent literatures for collaborative filtering (CF). We compared the performance of four different collaborating filtering: Pearson Correlation Coefficient (PCC), Vector Similarity (VS), Personality Diagnosis (PD) (using standard deviation parameter of 0.7) and Cooperative Co-evolution collaborative filtering (CoCo-CF).

Table 1. Performance evaluation

Parameter	PCC	VS	PD	CoCo-CF
MAD	1.0243	1.0090	1.1863	0.9872
MSE	1.7664	1.7219	2.0185	1.8234
RE	4.8426	4.8254	4.7676	5.3452

Table 1 compares three parameters for four different algorithms. It should be noted that the smaller values for mean absolute deviation and mean square error show a better performance for the systems where the big value in ranked evaluation confirm a good performance. The results show that the mean absolute deviation in CoCo-CF has the minimum value where by adding more trial data in the main database and decreasing the fitness threshold, the values will decrease accordingly. For mean square error, the CoCo-CF shows a better performance compare to PD where the PCC and VS are more successful methods. On the other hand, in ranked evaluation parameters, the CoCo-CF method shows a better performance compare to the other algorithms.

C. System Analysis

The aim of this work focuses on filtering the irrelevant information in a personalized way for the World Wide Web applications. There are verity of techniques and algorithms for this purpose. We have proposed the cooperative co-evolutionary collaborative filtering as an effective method for collaboration of distributed data in the public webs. This technique is operational in genetic algorithm and employs the grid computing for distributed computational. We have used the previous user experiences as the reference data. Then, we have classified the data and generate a coding record for each item. In the next step, we linked the state records for each individual process and generate a representative member for each process. Finally, we obtained the representative set for each class. In operation, the system investigates fitness of the incoming item with the class representative member. The algorithm tries to find the best fitness member and matching record for each entry item. As a result, the system will recommend the best matching record in the database as the best option for the requested item. In operation phase, the recommendation[5] time is an important constraint for any recommender system. In CoCo-CF system, the recommendation time is summation of training phase plus recommendation process. Training phase is a pre-operational period and prepares the system for actual process. The recommendation period should be negligible compare to the training phase. Moreover, there is a compromise between precision of the recommended item and recommendation time. It is obvious that any improvement in precision rate may influence the recommendation interval that mainly depends on threshold of the fitness. This point is more critical in the centralized processing systems. We have compared several scenarios in centralized and distributed environment and evaluated the results for several fitness levels. The results certify that the distributed capability of the Co-Co-CF reduces the recommendation time where by increasing the number of request, the CoCo-CF response time increases smoothly while the response time increases sharply in the centralized systems. Also, the threshold of fitness affects the satisfaction rate which is more tolerable in CoCo-CF compare to centralized system. It certifies that for having the same satisfaction level

[5] Time spends for decision which we call recommendation time.

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

in both systems, we need more searching time in centralized systems. Finally, for performance comparison, we have evaluated several systems and application parameters to compare the proposed method with other algorithms. It should be noted that the threshold value for fitness level and accuracy of data in the database are the key factors for satisfaction rate in a recommender system.

VI. Conclusions

Expansion of global networks and storing the extensive amount of information in various websites, create a serous needs for filtering the irrelevant information in a personalized way. Recommendation systems can predict user behavior patterns without any knowledge of the user in advance, and can evaluate the accuracy by comparing the prediction and reality. Collaborative filtering affords the systems enough ability to provide recommendations to the users and allows incorporation of the profiles which can be implicitly learned from previous activities. The system uses databases for user preferences to suggest the same recommendation for similar topics. We have proposed the CoCo-CF method as an effective solution for collaborative filtering. The system is qualified to implement in a Jini-grid computing platform and supporting the tasks in a distributed environment. We have considered the satisfaction rate, availability of the results, user experiences and the average response time as four evaluation factors. Moreover, we suggest the absolute deviation, mean square error and ranked evaluation as the system parameters. The obtained results certify that the CoCo-CF is a successful algorithm for collaborative filtering and an effective method for recommender system especially in a distributed environment.

References

[1] P. Resnick and H.R. Varian. Recommender systems. Communications of the ACM, 40(3):56-58, 1997.

[2] Roger Hurwitz and John C. Mallory, The Open Meeting: "A Web-Based System for Conferencing and Collaboration", Proceedings of The Fourth International Conference on The World-Wide Web, Boston: MIT, December 12, 1995.

[3] Breese J.S., Heckerman D. and Kadie C, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", Proceedings 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998.

[4] G. Adomavicius, "Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749, 2005.

[5] Drachsler H., Hummel H.G.K., Koper R., "Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning", in Workshop on Social Information Retrieval for Technology-Enhanced Learning (SIRTEL'07), Crete, Greece, September 17-20, 2007.

[6] Resnick, P., Varian, H. R. "Recommender Systems". *Communications of the ACM 40*, 56-58, 1997.

[7] Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. "Analysis of Recommendation Algorithms for E-Commerce", In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00), ACM Press, New York. 285-295, 2000.

[8] H. Kautz, B. Selman, M. Shah, Referral Web: Combining Social Networks and Collaborative Filtering, Communications of the ACM, Vol 40, No 3, March 1997.

[9] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40,3, pp 77-87, 1997.

[10] Foster and C. Kesselman: The Grids: "Blueprint for a New Computing Infrastructure", Elsevier/ Morgan Kaufmann Publishers, 2004.

[11] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, C. Kadie. "Dependency Networks for Inference, Collaborative Filtering, and Data Visualization". *Journal of Machine Learning Research* 1, 2000.

[12] Adomavicius, G. and Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". IEEE Transactions on Knowledge and Data Engineering 06.2005.

[13] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the Tenth International Conference on the World Wide Web, pp. 285-295, 2001.

[14] M. R. Ahmadi and D. Maleki, "An Intrusion Detection Technique Using Co-Co Immune System for Distributed Networks (CoCo-ISD)'' International Journal of Computer Science and Network Security, pp 160-169, Vol.8, No.4, April 2008.

[15] UCI KDD Archive, Microsoft Anonymous Web Data from Microsoft web site.

[16] P. Symeonidis , A. Nanopoulos , A.N. Papadopoulos , Y. Manolopoulos, "Collaborative recommender systems: Combining effectiveness and efficiency, Expert Systems with Applications" ACM An International Journal, v.34 n.4, pp 2995-3013, May, 2008.

[17] N. Lathia , S. Hailes , L. Capra, "The effect of correlation coefficients on communities of recommenders", Proceedings of the 2008 ACM symposium on Applied computing, March 16-20, 2008.

[18] J.S.Breese, D. Heckerman and C. Kadie "Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Uncertainty in Artificial Intelligence*, 1998.

[19] D.M. Pennock, E. Horvitz, S. Lawrence and C. L. Giles. " Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based approach" *Uncertainty in Artificial Intelligence*, 2000.

[20] C.M. Kadie,C. Meek and D. Heckerman. " A Collaborative Filtering System Using Posteriors

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]

Over Weights of Evidence. " *Uncertainty in Artificial Intelligence*, 2002.



Mohammad Reza Ahmadi received the B.Sc. and M.Sc. degrees in Electrical Engineering and Communication Systems from K.N.T. University of Technology in 1986 and 1990 respectively. He received his Ph.D. in Communication Networks from Tokyo Institute of Technology, Tokyo, in 1997.

Currently he is the project manager and researcher in IT department of Iran Telecommunications Research Center (ITRC). His research interests are network security focus on intrusion detection/Prediction systems, Immune systems focus on network applications and data center design and implementation.

[Downloaded from ijict.itrc.ac.ir on 2024-03-29]