## An evolutionary approach for combining results of Recommender Systems techniques based on Collaborative Filtering

Edjalma Q. da Silva Institute of Informatics Federal University of Goiás F Goiás, Brazil Email: edjalmasilva@inf.ufg.br

Celso G. Camilo Júnior Institute of Informatics Federal University of Goiás Goiás, Brazil Email: celso@inf.ufg.br

Abstract-Recommendation systems work as a counselor, behaving in such a way to guide people in the discovery of products of interest. There are various techniques and approaches in the literature that enable generating recommendations. This is interesting because it emphasizes the diversity of options; on the other hand, it can cause doubt to the system designer about which is the best technique to use. Each of these approaches has particularities and depends on the context to be applied. Thus, the decision to choose among techniques become complex to be done manually. This article proposes an evolutionary approach for combining results of recommendation techniques in order to automate the choice of techniques and get fewer errors in recommendations. To evaluate the proposal, experiments were performed with a dataset from MovieLens and some of Collaborative Filtering techniques. The results show that the combining methodology proposed in this paper performs better than any one of collaborative filtering technique separately in the context addressed. The improvement varies from 9.02% to 48.21% depending on the technique and the experiment executed.

#### I. INTRODUCTION

N order to eliminate doubts in situations where we have to choose among products or items we are faced with, we usually rely on recommendations that are passed by others. These recommendations are informed to us directly ("word of mouth") [1] or through recommendation texts, opinions of movies and book reviewers, printed newspapers, among others. A recommender system helps to increase the capacity and effectiveness of transmitting and receiving suggestions, a well known process in the social relationship among human beings [2]. In a typical system, people provide evaluations to items they have bought or used. These evaluations are usually represented as ratings.<sup>1</sup>. The recommender system uses these gradings of items to suggest the best n items to other users. One of the major challenges of such systems is to perform the appropriate combination between user expectations and products, services and people to be recommended, i.e. discovering this relationship of interest is a major problem [3].

Adomavicius [4] classifies recommender systems into three major categories regarding the approach used to generate the recommendations: (i) *content-based* approach, in which similar

Luiz Mário L. Pascoal	Thierson C. Rosa
Institute of Informatics	Institute of Informatics
Federal University of Goiás	Federal University of Goiás
Goiás, Brazil	Goiás, Brazil
Email: luizpascoal@inf.ufg.br	Email: thierson@inf.ufg.br

items to those the user showed preference in the past are recommended; (ii) *collaborative filtering* which recommends items chosen by people with similar preferences to the user and; (iii) *hybrid* approaches that combine techniques of both previous approaches to attempt to solve some problems inherent to each of them in isolation.

The approach of collaborative filtering has become popular in the fields of academia and industry with great speed. Companies like Google, Amazon and Netflix make greate use of this approach because of its significant competitive advantage. Until today, the development of Collaborative Filtering algorithms has focused mainly on how to provide accurate recommendations [5]. This approach basically follows four steps [3]:

- 1) Calculate the similarity of each user to the target-user (similarity metrics).
- 2) Select a subset of *h* neighbors, i.e., users with highest similarity to the target-user, in order to consider the ratings of these neighbors in the prediction.
- 3) Normalize ratings and compute the predictions considering the evaluations of neighbors with their weights. The weight in this case is the value of similarity between the neighbor and the target-user.
- 4) Sort items in decreasing order of predicted scores and present the best *n* items to the target-user.

However, Collaborative filtering algorithms can be classified into two types: *memory-based* algorithms and *model-base* algorithms. They differ in how they process the matrix of ratings (User  $\chi$  Item). The memory-based collaborative filtering uses the entire matrix to calculate its prediction. Generally make use of similarity measures to select users (or items) that are similar to the target-user. Then, the prediction is calculated from the ratings of these neighbors. These algorithms are also known as user-based algorithms, where the focus of obtaining neighbors is on the user [1] or algorithms based on items (itembased), where the focus on obtaining neighbors is on item [6]. The model-based algorithms, first build a model that represents the matrix of ratings (off-line), i.e. that represents the behavior of users, and therefore, predict (online) their ratings [1], [3], [7].

In the literature various techniques (Euclidean, Tanimoto,

<sup>&</sup>lt;sup>1</sup>These ratings are commonly represented as a grade in the range [1, 5] or as a number of "stars" in the same range.

Pearson correlation, etc.) are presented to calculate the similarity between users as alternative measures of similarities to be used on memory-based algorithms. This is good because it emphasizes the diversity of options; on the other hand, this may cause doubt as to which technique to choose to recommend n items. Each of these approaches have particularities and depend on the context to be applied, therefore each case must be analyzed before choosing which technique to adopt.

We do not know about any previous work that combines the results of distinct techniques in order to produce a final ranking of n items composed of the best items suggested by each technique in isolation. Furthermore, we know that no technique would be the best for all contexts. Therefore, we propose the hypothesis that the combination of rankings of recommendations resulting from the memory-based techniques would have a better result than any of them alone. As the act of discovering a good combination manually is a difficult task, it is desired that the combination be automated. For such matter, the paper proposes a Genetic Algorithm (GA) [8], [9] able to automate the combination of results of different memorybased similarity techniques. The GA was chosen because it is widely used in the literature. Moreover GAs are known for their flexibility, easiness of implementation, and effectiveness in performing global search in adverse environments.

In this approach, the GA should be able to generate a list (L) of n items to be recommended. These items are selected from the ranking of techniques used in the combination. Therefore, the list formed by the GA depends on the performance of each technique. The techniques that achieve lower error (RMSE) will have more items among the n finals. An example of the composition of this list, in case of ||L|| = 10, would be: 3 items coming from the rank of technique B and 4 items arising from the rank of technique C, totaling 10 items in the final list proposed by the GA.

Moreover, a scenario was created to test the proposal performance in a multi-objective context, i.e. the GA in addition to seeking a solution with low error (RMSE), it attempts to maximize the removal of user's ratings prior to recommendation. This scenario aims at verifying the behavior of the techniques and the GA when we have a reduced number of ratings from the target-user. To make this possible, we created two variables w1 and w2, where w1 is the weight given to the representativeness of a technique in the composition of the final list L and  $w^2$  is the weight given to the amount of ratings that the target user has on the database. The purpose of these variables is to create the possibility of simulating multiple database scenarios. As an example, imagine a database which has few ratings, just to simulate this scenario we could set the value for the variables w1 and w2 respectively, with 0.2 and 0.8. With this configuration the GA is forced to find a good solution where one has a few ratings from the target-user.

The rest of the paper is organized as follows: In Section II we briefly review some of the research literature related to our work. In Section III we present the main theoretical concepts needed to develop this paper. In Section IV we present our proposal. In Section V we present experiments and results. In section VI, conclusions and future work are shown.

#### II. RELATED WORK

The first system created using the Collaborative Filtering (CF) approach was the *Tapestry* [2], [5], which was a system with complete capabilities of filtering electronic documents. In this system a user can create filtering rules of *e-mail* such as "Show me all documents answered by other members of my research group". This system required the users to determine the relevant predictive relationships. As a result, these systems were only valuable in small closed communities where everyone was aware of interests and duties of other users.

There are studies that make comparisons between traditional CF methods and proposal of hybrid schemes [10]. Dellarocas in [11] predicts the use of CF techniques combined with basic (field value) and advanced (frequency domain) mechanisms to generate estimates of personalized reputation, and to better protect the system against manipulation of malicious entities. Herlocker et al. in [12] propose adjustments in the CF technique with the application of a weight on the similarity coefficient given to each rating, based on the amount of items in common previously evaluated with the target-user. In the work in [13], the authors propose a hybrid model to improve CF algorithms; thus, the proposed algorithm makes use of a content-based approach to get users interested in the collection of multi-dimensional vector model, and then they start from the recommended strategy for collaborative filtering starting to find their target customers interested in the most similar "neighbors".

Other work [14], proposes a cascade hybrid recommendation to the combination of One-Class classification and CF. The article breaks down the problem of recommendation into a cascade recommendation schema of two levels where the goal is to get benefits of the CF methodologies and contentbased approach. The first level makes use of the content-based approach by applying the One-Class classification paradigm to incorporate the user preferences individually (subjective) in the recommendation process approach. The second level has the purpose of assigning specific scores to items to classify them.

#### III. BASIC CONCEPTS

#### A. Recommender Systems

Recommender systems (RS) have become an important area of research since the emergence of the first articles on collaborative filtering in the mid-1990s. There has been much work performed in industry and academia with regard to developing of new approaches to RS over the last decade.

Over the years RS were classified in different ways by several authors; however there is not a classification completely accepted by the user's community, professionals and researchers. Schafer [15] modeled the architecture of a recommender system and produced a fairly comprehensive taxonomy that considers various characteristics of a RS.

The taxonomy architecture (Figure 1) proposed by Schafer [15] involves three distinct modules, each of which can be modeled and implemented in different ways. This division into blocks facilitates the understanding of these systems. The blocks are separated as follows: *Target-User* is the module

responsible for collecting information about the target-user; *Community* is information about interactions of the target-user and other users with the system. These interactions occur at the time when the user evaluates a product for example; And *Output* is represents the system response as a suggestion of product or service. The flow of interaction between these three modules may be observed in Figure 1.



Fig. 1. Taxonomy of recommendation systems proposed by Schafer [15] for e-commerce.

Also, according to Schafer [15], applications aimed at generating "recommendations" to users in e-commerce systems combine information about the target-user with the communities where products and the user are located. Thus, the websites use decisions on the level of customization and delivery method to transform them into specific recommendation packages, ensuring personalized recommendations. Comments and ratings of a user about the recommendation received or even about a specific product can generate additional inputs for future recommendations.

1) Recommender Systems based on Collaborative Filtering: A system based on CF assumes that if two users have similar interests, then users will demonstrate interest for the same products. In general, consider a list of users  $U = \{u_1, u_2, \ldots, u_{||U||}\}$  and a list of items  $I = \{i_1, i_2, \ldots, i_{||I||}\}$ .

Each user  $u_i$  has a list of items m for which he has expressed interest. Thus, if  $m \subset I$  (it is possible that m is a null set), there is a distinguishable user  $U_a \in U$ , called targetuser a, for which it is task of collaborative filter to find an item of interest, in particular seeking recommendations. Thus, there will be a list of n items,  $n \subset I$ , for which the target-user will be interested more. The recommended list should be of items not evaluated by the target-user, sorted in decreasing order of values of predicted scores by the collaborative filter. This interface of Collaborative Filtering algorithms is also known as "Top-N" Recommendation [16].

Taking Table I as a model, we can show how to apply in practice the collaborative filtering. The first step of the CF system is based on searching users with similar habits of consumption, i.e. calculating the similarity among users. When analyzing users 1 and 5, for the item  $(i_1)$  the difference between their ratings is 1.0, in  $(i_2)$  there is no difference and for  $(i_3)$  the difference is 0.5. Thus, we could say that users 1 and 5 are similar. By the same reasoning, users 1 and 2 would not be so similar. The calculation of similarity can occur only on items that both users have expressed preference. Table I is usually referred to as *rating matrix*.

	$Item(i_1)$	$Item(i_2)$	Item $(i_3)$
$User(u_1)$	5.0	3.0	2.5
$User(u_2)$	2.0	2.5	5.0
$User(u_3)$	2.5	-	_
$\text{User}(u_4)$	5.0	-	3.0
$User(u_5)$	4.0	3.0	2.0

To calculate the similarity between users there are several techniques, which according to the authors [3], [17]. The most common are shown in Table II, where:

- $w_{a,u}$  is the correlation of the target-user a with a given user u
- $r_{a,i}$  is the rating that the target-user gave for the item i
- $\overline{r}_a$  is the average of all ratings of the target-user (a)
- $w_a$  is the expected utility of the item *i* for user *a*
- *d* is the default rating (generally a non-committal rating, or slightly negative)
- *α* is the half-life. The half-life is the rank of the item
   on the list such that there is a 50% chance that the
   user will view that item.

TECHNIQUE	EQUATION	REFERENCE
Pearson correlation	1	[4]
Euclidean	2	[18]
Cosine	3	[4]
Spearman Rank <sup>2</sup>	1	[12]
Tanimoto	4	[19]
Loglikelihood test	5	[12], [20]
	EQUATION	
$w_{a,u} = \frac{\sum}{\sqrt{\sum_{i=1}^{m}}}$	$\frac{m}{i=1}(r_{a,i}-\overline{r}_a)(r_i)(r_i)(r_{a,i}-\overline{r}_a)^2\sum_{i=1}^m$	$\frac{u,i-\overline{r}_u)}{1^{(r_{u,i}-\overline{r}_u)^2}}  (1)$
$w_{a,u} = v$	$\sqrt{\sum_{i=1}^{m} (r_{a,i} - )}$	$\overline{r_{u,i}}^2$ (2)
$w_{a,u} = \frac{1}{\sqrt{\Sigma}}$	$\frac{\sum_{i=1}^{m} (r_{a,i} * r_{i})}{\sum_{i=1}^{m} (r_{a,i})^2} \sqrt{\sum_{i=1}^{m}}$	(1,i) $(1,i)^{2}$ (3)
$w_{a,u} = \frac{1}{ u }$	$\frac{ m_a \cap m_u }{ m_a  +  m_u ) - ( m_a )}$	$\overline{a \cap m_u }$ (4)
$w_a =$	$\sum_{i} \frac{\max(r_{a,i}-d)}{2^{(i-1)/(\alpha-d)}}$	$\frac{2,0)}{-1)}$ (5)

The second step of the CF-based system is to select a subset of users with higher similarity. After that, the next step is to

<sup>&</sup>lt;sup>2</sup>The Spearman rank correlation coefficient is similar to Pearson, but rather than compute a correlation based on the original preference values (Eq.1), it computes a correlation based on the relative rank of preference values. [12].

calculate the predictions. The prediction is the act of inferring what appraised value would give the user a product that he has not assessed yet. An example of this calculation, noting the table I, would be filling in the gaps left by the user  $u_3$ on items  $i_2$  and  $i_3$  and user  $u_4$  on the item  $i_2$ . Therefore, the prediction is made independent of the technique used, because it is generated by a weighted average of ratings of neighbors that have an acceptable coefficient of similarity. According to the authors [3], [4], [21], the prediction can be calculated by Eq. (6).

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{h} (r_{u,i} - \overline{r}_u) * w_{a,u}}{\sum_{u=1}^{h} |w_{a,u}|}$$
(6)

Where h is the amount of best neighbors and it is at the discretion of each system that uses collaborative filtering.

Finally, the sorting is performed in decreasing order of values of the predictions and returned the best n items as recommendations.

2) Evaluation of Recommending Systems: The input to a recommender system is a rating matrix M, similar to that presented in Table I. To evaluate a recommender algorithm A, another matrix  $M_t$  is obtained from M by removing k ratings. Matrix  $M_t$  is used as input to the recommending algorithm to be evaluated. The objective of algorithm A is to predict correctly the values of the ratings absent from the matrix. Let  $R = \{r_1, r_2, \ldots, r_x\}$  be the set of ratings absent from  $M_t$ . Algorithm A produces a set of predictions  $P = \{p_1, p_2, \ldots, p_x\}$  when trying to guess the corresponding values in R. The evaluation of A is done by computing the accumulated error produced by A in its predictions.

There are different metrics used to compute the error of a recommender algorithm. In this paper we use the *Root Mean Squared Error* (RMSE), which became extremely popular in recent years, after being used in the Netflix Prize competition [3]. For a given algorithm A the RMSE is computed as described in Eq. (7).

$$RMSE_{A} = \sqrt{\frac{\sum_{i=1}^{x} (p_{i} - r_{i})^{2}}{x}}$$
(7)

Where x is the amount of items that were recommended,  $p_i$  the prediction of the algorithm and  $r_i$  the corresponding true rating. The most accurate algorithm is the one with the slowest RMSE value for a given matrix  $M_t$ .

#### B. Genetic Algorithm - GA

The first step when using GA is to represent each possible solution  $\chi$  in the solution-space as a sequence of symbols chosen from a finite alphabet A. which varies according to the target problem. Each sequence s represents an individual (solution) and can be seen (metaphorically) as a chromosome. Each symbol in s is considered a gene. Most GA solutions use a constant-size population of chromosomes and each chromosome has also a fixed size [22], [23]. After definition of each chromosome for the specific problem to be solved by an GA, an initial population  $P_0$  of candidate chromosomes (solutions for the problem) is created.

GAs are iterative algorithms and in each iteration a new population is derived from the population existing at the begin

of the iteration. The control flow of a GA corresponds to the following steps [24]:

- 1) Create the initial population  $P_0$  of chromosomes.
- 2) Evaluate each chromosome in the current population.
- Chose parent chromosomes to generate new chromosomes.
- 4) Apply genetic operators to the chosen parents in order to generate new chromosomes which will compose the next population (the next generation).
- 5) Kill the old population.
- 6) Evaluate each new chromosome. If time is over or the the best chromosome has been found stop, otherwise, go to step 3.

In this paper we use GA to automatically combine memorybase collaborative filtering techniques derived from distinct similarity measures, with the objective to demonstrate our proposed hypothesis. The next section discuss how we use an GA to perform this combination.

#### IV. PROPOSAL

In this section we describe a GA that produces a list (L) of items to be recommended. L is composed by picking the best items from the rankings produced by each individual collaborative filtering technique used (Pearson, Euclidean distance, Spearman, Tanimoto an Loglikelihood). The GA was then used as an optimizer of results of each technique recommender.

Thus, the aim of the GA is to obtain a good solution (chromosome) to the problem of choosing the appropriate number of elements on the top of each individual ranking produced by the used techniques. The chosen elements are used to compose the final ranking. Consequently, each candidate solution (chromosome) must contain a sequence of numbers of elements from each ranking to be used in the final ranking L.

However, we are also interested in obtaining a good solution even in hard conditions which occur whenever there are few ratings of the the target-user in the rating matrix. To this end, we introduce another information in each chromosome that is the number of ratings given by the target-users that are removed from the rating matrix.

#### A. Chromosome Representation

Consider the set of techniques  $T = \{t_1, t_2, t_3, \dots, t_{||T||}\}$ , where each technique t in T uses a different similarity measure. A chromosome corresponding to each individual is composed of ||T|| genes, where each gene corresponds to a pair as shown in Figure 2, where  $x_i$  represents the quantity of



Fig. 2. The structure of a chromosome used by the proposed GA.

items obtained from the top of the list (ranking) produced by technique  $t_i \in T$  that is used to compose the final ranking. The second element of the pair  $(k_i)$  corresponds to the number of

ratings that are removed from the rating matrix M to generate the matrix  $M_t$ .

#### B. The proposed GA

The initial population  $P_0$  used by the GA is created randomly, with  $x_i$  and  $k_i$  in each chromosome *i*, receiving any random values since these values satisfy the restriction specified in Eq. 8.

$$\sum_{t=1}^{\|T\|} x_t = \|L\| \tag{8}$$

Where ||L|| is the size of the final ranking produced by GA, which is a value known in advance.

Table III shows a sample of chromosomes that satisfy the restriction in Eq. (8) for ||L|| = 10.

 
 TABLE III.
 EXAMPLES OF CHROMOSOMES (EACH LINE CORRESPONDS TO A DISTINCT CHROMOSOME)

Р	ΈA	EU	UC SPE		PE	TAN		LOG	
X	k	X	k	X	k	X	k	X	k
3	0	2	0	2	0	1	0	2	0
1	0	6	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0	10	0

For each chromosome in a given population P (initially  $P = P_0$ ) the GA computes the accumulated RMSE (RMSE<sub>t</sub>) for each technique t. The accumulated values of RMSE for techniques are used to compose the fitness function that is used to evaluate each chromosome as will be explained later in this section.

Figure 3 shows how the GA computes the RMSE for each technique and how these partial RMSE computations are totalized to generate the RMSE for a chromosome.



Fig. 3. Computation of the RMSE for each technique and for the chromosome that corresponds to a candidate solution.

Where:

- t is one of the technique in T;
- u is one user in M;

- $M_{full}$  is the complete matrix containing all ratings of users to items as exemplified in Table I. In addition to giving result in matrix M, the matrix  $M_{full}$  aims to provide the value of the actual rating  $(r_i)$  that the user entered for the item and this value is then used to the calculation of  $RMSE_u$ .
- M is the matrix composed by the ratings of 4.7% of the users contained in  $M_{full}$  -283 the first users found.
- k is the number of ratings made by the target-user u, removed from M to obtain the matrix  $M_t$ .
- $M_t$  is the resulting reduced matrix obtained from the rating matrix M by eliminating k of the ratings of user u.
- x is the number of recommendations to be generated for the computation of RMSE user u in the technique t.
- RMSE<sub>u</sub> is the value of RMSE for the predictions of a technique t to a given user u.
- RMSE<sub>t</sub> is the accumulated values of RMSE<sub>u</sub> due to the predictions made by technique t for each user u.

To compute the accumulated  $RMSE_t$  for each technique t the GA produces a matrix  $M_t$  for each user u. Matrix  $M_t$  is obtained by first copying the ratings of M and next removing  $k_t$  ratings of user u, where  $k_t$  corresponds to the second component of the t-th gene in the chromosome. Then, predictions  $(p_i)$  are generated for user u using technique t applied over the matrix  $M_t$ . The real values  $(r_i)$  of each item are obtained by performing queries on matrix  $M_{full}$ . Next, the value of  $RMSE_u$  is computed for user u using Eq. (7) but substituting x in the equation for the value of  $x_t$  in the chromosome. The value of  $RMSE_u$  is then accumulated in  $RMSE_t$ . The above processing is repeated for each user.

The accumulated values  $\text{RMSE}_t$  are computed for each technique *t* as just explained and these values are also summed up to obtain the grand total  $\text{RMSE}_{total}$  which is used to evaluate the fitness of the chromosome.

After the accumulated values  $\text{RMSE}_t$  have been computed the chromosome is evaluated using the fitness function expressed on Eq. (9).

$$MIN(f(x)) = \frac{w_1 \sum_{t=1}^{\|T\|} RMSE_t + w_2 \sum_{t=1}^{\|T\|} (1-q)}{\|T\|}$$
(9)

where:

- $RMSE_t$  is the accumulated value of RMSE for technique t.
- $w_1$  and  $w_2$  are input parameters for technique t for the GA, with values in the range [0, 1].  $w_1$  is used as a weight of the importance of the technique in the composition of the final ranking produced by GA.  $w_2$  corresponds to the weight given to the quantity of ratings that are removed for the target-user when generating the matrix  $(M_t)$  as shown in Figure 3.

• q is the value computed by Eq. (10), where R is the quantity of ratings of each user in the rating Matrix M.

$$q = \frac{k}{R-1} \tag{10}$$

The fitness function (Eq. 9) is composed of two components. The first component  $\left(\frac{w_1 \sum_{t=1}^{\|T\|} RMSE_t}{\|T\|}\right)$  is responsible for the totalization of the error in the prediction process. To do this, the RMSE was chosen as the evaluation metric, because it has been used in many studies that measures the accuracy of predictions [3]. The second component  $\left(\frac{w_2 \sum_{t=1}^{\|T\|} (1-q)}{\|T\|}\right)$  represents the total performance of each technique taking into account the complexity of the scenario in which it was inserted. The complexity to generate predictions is related to the amount of the target-user ratings. Thus, the sum of these components guide the GA in finding solutions. The parameters  $w_1$  and  $w_2$  are variables used by the project designer to determine which problem he wants to give priority to, so they are important components of the fitness function.

The whole process described above is repeated to each chromosome of population P. Then the chromosomes with better scores computed by the fitness function are operated by genetic operators (crossover and mutations) to produce a new population P'. Then, the GA repeats the whole process with the new population P'. The generation of new populations is repeated until the stoppage condition is achieved. In our GA we used the genetic convergence and a pre-defined number of iterations (100 genarations) as stop conditions. The GA terminates whenever one of these criteria occurs.

#### V. EXPERIMENTS AND RESULTS

We want to evaluate the RMSE of our recommender based on GA, comparing it with traditional methods of CF using Pearson's correlation, Euclidean distance, Spearman Correlation, Tanimoto coefficient and Loglikelihood. Importantly, the algorithm created is able combine any type of recommendation approach. In this paper we used the approaches of collaborative filtering most widely adopted as a proof of concept. We made use of the movieLens dataset (MovieLens 1M Data Set  $(.zip)^{3}$  to perform the experiments. These files contain 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000. To this database was given the name 1M Date Set. For implementation of the experiments was used 4.7% of 1M Data Set as M, which corresponds to 283 users where each user has 10 ratings. To implement the proposed model we made use of two "frameworks" enshrined in their respective fields, Apache Mahout<sup>4</sup> [17] as recommender and Jenes  $2.0^5$  as the GA engine.

#### A. Experiment One - GA Calibration

The GA calibration is performed so that one could get the best setting of variables that will be used by the GA in comparison scenario with the individual techniques. The GAs is a stochastic process, so we performed many embryonal experiments in order to identify the best set of configuration parameters. The algorithm was executed 30 times for each configuration presented in table IV. However, no significant impacts were identified by altering the parameters in these experiments.

TABLE IV. SETTING VARIABLE VALUES FOR EXECUTION OF GA

PROPERTY	VALUE
w1	1
w2	0*
Probability of Crossover	60%, 70%, 80%, 90%
Method Crossover	One point, Two points ([22], [23])
Elitism	1 Individual (Randomly), Worst ([22], [23])
Selection	Tournament ( [22], [23])
Mutation	2%, 5%, 10%, 30%
Population	100
Stopping criterion	Genetic convergence or limit 100 generations.
Number of running	30

\*When performing calculating the fitness of the individual, the GA

does not take into account the withdrawals of the target-user ratings.

Thus, the following parameters to control the GA were chosen: Probability of Crossover of 80%, One Point for the Crossover Method, Elitism of 1 Individual (Randomly replaced), Tournament as Selection criteria and Mutation rate of 2%. For all experiments were considered the value of  $k_t$  equal to zero, in order to give the combination of techniques and each individual techniques the same experimental conditions.

#### B. Experiment two - GA vs CF in isolation techniques

After GA calibration, new battery of tests was carried out. The purpose of performing this experiment is to verify if performance (lowest error) of combining techniques is better than any technique in isolation. In this experiment we fixed

TABLE V. RMSE OF THE CF TECHNIQUES

$(L_n)$	PEA	EUC	SPE	TAN	LOG
10	0.425	0.267	0.3836	0.244	0.242
9	0.428	0.274	0.3834	0.246	0.243
8	0.434	0.280	0.3817	0.252	0.244
7	0.434	0.290	0.3817	0.249	0.243
6	0.434	0.295	0.3908	0.249	0.245
5	0.437	0.289	0.3897	0.236	0.242
4	0.438	0.290	0.3935	0.228	0.244
3	0.451	0.265	0.3897	0.196	0.220
2	0.432	0.280	0.3843	0.220	0.236
1	0.374	0.269	0.3589	0.230	0.252

the values of every  $k_t$  in the cromosomes of the populations to zero. This was done because the intent of this experiment

<sup>&</sup>lt;sup>3</sup>The database may be obtained through the page link: http://www.grouplens.org/node/73

<sup>&</sup>lt;sup>4</sup>http://mahout.apache.org

<sup>&</sup>lt;sup>5</sup>http://jenes.intelligentia.it/

was to compare the accuracy of the combination of results produced by the GA to the accuracy of each technique alone.

Table V shows the results of RMSE for each of the five selected techniques. Where PEA is the result obtained using the Pearson correlation alone, EUC for Euclidian, SPE for Spearman, TAN for Tanimoto and LOG for Loglikelihood. The  $L_n$  column represents the size of the final list L consisting of each technique alone. Note that in this scenario, the more items using Pearson the larger the RMSE and in return the more Tanimoto items or Loglikelihood the lower the RMSE.

Figure 4 shows the RMSE values obtained in each technique, if used a single technique by the recommender to a final L list of size 10. To this graphic was added the GA execution result. The GA made the combination of techniques and got at a better individual who had 3 items of Tanimoto and 7 items of Loglikelihood, totaling 10 items in the final list to be recommended.



Fig. 4. RMSE of individualized techniques and GA

The GA was the best among the methods with only 0.22022 error, which represents a decrease of 9.028% in the RMSE compared to the best result obtained using a single CF technique to generate recommendations (Loglikelihood). If we compare with Pearson's the difference increases to 48.21%. Thus, it is evident that the fact of combining the techniques outputs results in a smaller error than the use of only one in the recommendation process based on CF.

# C. Experiment Three - Analysis of the behavior of the 5 techniques of Collaborative Filtering when varying the ratings that have the target-user

The experiment in this section aims at evaluating the variation of the errors (RMSE) of the five techniques used and the variations of the errors of the combination obtained by the GA when we vary the number of items rated by the target-user. We varied the values in  $k_t$  (second component of the *t*-th gene of the chromosome) which corresponds to the number of ratings of the target-user that are removed. We conducted experiments varying  $k_t$  from 0 to 9. In the first scenario we obtain chromosomes where the values of every  $k_t$  are set to zero. In the second scenario the values of every  $k_t$  are set to 1. that is, every target-users in  $M_t$  will have one less rating. We continue this way until every  $k_t$  of each chromosome are set to 9.

Figure 5 shows each corresponding value to the solution for the technique used. Comparing scenarios with k = 0 versus



Fig. 5. Results of techniques for similarity calculation in Recommendation Systems varying the number of withdrawals

k = 9, all techniques showed an increase in RMSE (error). Spearman increased in 61.63% in RMSE, Pearson increased in 57.47%, Loglikelihood in 35.13%, Tanimoto in 34.35% and Euclidean in 27.53%. This demonstrates that the techniques of Spearman correlation and Pearson's correlation are more sensitive to the removal of items evaluated by the matrix (M) user. The techniques of Euclidean distance, Tanimoto Coefficient and Loglikelihood are able of maintain the low value of RMSE even when ratings are removed from the targetuser.

### D. Experiment Four - GA vs CF in isolation techniques in a simulated scenario with few ratings

Experiments for the multiobjective problem have been performed: obtaining a solution with low error (RMSE) and with a reduced number of ratings of the target-user. Thus, it tried to be found the best combination in scenarios with few ratings of the target-user. For such, it was necessary to use the variables w1 and w2 defined in subsection V-A with the values 0.2 and 0.8 respectively.

When assigning these values to variables w1 and w2 the GA is forced to favor individuals who had more ratings removed, while considering the error (RMSE) of the recommendations. The three best individuals are shown in Table VI. Note that in the scenario the GA tends to use more items of Euclidean, Tanimoto or Loglikelihood. This is justified by the analysis in subsection V-C.

TABLE VI. CHROMOSOME REPRESENTATION OF THE THREE BEST INDIVIDUALS OF THE EXPERIMENT

PEA		EUC		SPE		TA	TAN LOG		)G	
X	k	Х	k	Х	k	Х	k	Х	k	RMSE
0	0	8	9	0	0	1	9	1	9	0,3297
0	0	1	9	0	0	8	9	1	9	0,3316
0	0	1	9	0	0	1	9	8	9	0,3316

We can also conclude that even in environments where there are few ratings of the target-user, it is possible to get good recommendations. Comparing the results of the best individual (RMSE of 0.3297, k = 9) of Table VI and techniques of Figure 5 we can see that the GA even with 1 ratings overcomes the *PEA* technique (lowest RMSE of 0.4252, k = 0) and SPE (lowest RMSE of 0.3836, k = 0) in any quantity. Even removing 9 items from ratings of the target-user, the GA (0.3297) performed better than all the techniques, which represents a decrease of 11.65% compared to the result of the best technique (Loglikelihood). When we removed 8 ratings of all techniques, the GA (RMSE of 0.3297, even with k = 9) has a better performed, achieving a 12.10% reduction in RMSE is compared with the result of the best technique in that scenario, Tanimoto.

It is concluded that GA is able to generate good combinations of techniques in scenarios with few ratings of the target-user, i.e. the proposed GA manages to capture the particularities of the database and is able to generate good solutions.

#### VI. CONCLUSIONS AND FUTURE WORK

This study aimed at testing an evolutionary approach to combine results of CF-based RS-techniques. The experiments conducted have shown that when making this combination the error (RMSE) for predicting Recommender's scores decreases compared to the approach using just one technique. The results show that there is a reduction in the RMSE of at least 9.028%when compared to the best technique alone, and this difference increases to 48.21% when compared to the worst technique as can be observed in the experiments presented in the sub-section V-B. In the experiments conducted on sub-section V-C, it is demonstrated that in case of few information about the targetuser and if we has to choose a unique technique for similarity calculation, a good choice would be to use the Euclidean, Tanimoto or Loglikelihood because they were able to maintain a low value of RMSE even when some ratings are removed from the target-user.

Another point to highlight is that the GA has good performance even in the scenario where the target-user has few ratings, in which there is a reduction of 11.65% of RMSE compared to the best result (Loglikelihood) as can also be observed in the experiment presented on sub-section V-D. Considering the independence of the proposal to the recommendation techniques used, it is also suggested as future work to perform experiments with other approaches, such as content-based algorithms and hybrid approaches.

#### ACKNOWLEDGMENT

This work was partially supported by FAPEG, under grant chamada 009/2010.

#### REFERENCES

- U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," *Proceedings of the SIGCHI conference* on ..., pp. 210–217, 1995.
- [2] P. Resnick and H. R. Varian, "Recommender systems," Commun. ACM, vol. 40, pp. 56–58, March 1997. [Online]. Available: http://doi.acm.org/10.1145/245108.245121
- [3] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms," ACM Transactions on the Web, vol. 5, no. 1, pp. 1–33, Feb. 2011. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1921591.1921593
- [4] 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, no. 6, pp. 734–749, 2005.

- [5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, pp. 61–70, December 1992. [Online]. Available: http://doi.acm.org/10.1145/138859.138867
- [6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the* 2nd ACM conference on Electronic commerce, ser. EC '00. New York, NY, USA: ACM, 2000, pp. 158–167. [Online]. Available: http://doi.acm.org/10.1145/352871.352887
- [7] 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*, ser. CSCW '94. New York, NY, USA: ACM, 1994, pp. 175–186.
- [8] Adaptation in Natural and Artificial Systems. HOLLAND, J.H, 1975.
- [9] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.
- [10] S. Fong, Y. Ho, and Y. Hang, "Using Genetic Algorithm for Hybrid Modes of Collaborative Filtering in Online Recommenders," 2008 Eighth International Conference on Hybrid Intelligent Systems, pp. 174– 179, Sep. 2008.
- [11] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM, 2000, pp. 150–157. [Online]. Available: http://dl.acm.org/citation.cfm?id=352889
- [12] J. L. Herlocker, "Understanding and improving automated collaborative filtering systems," Ph.D. dissertation, University of Minnesota, 2000, aAI9983577.
- [13] L. Qingshui and Z. Meiyu, "The study of personalized recommendation technology based content and project collaborative filtering combines," in Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 5, 2010, pp. V5–506–V5–510.
- [14] A. Lampropoulos, D. Sotiropoulos, and G. Tsihrintzis, "Evaluation of a cascade hybrid recommendation as a combination of one-class classification and collaborative filtering," in *Tools with Artificial Intelligence* (ICTAI), 2012 IEEE 24th International Conference on, vol. 1, 2012, pp. 674–681.
- [15] J. Schafer, J. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining and Knowledge Discovery*, vol. 5, no. Kluwer Academic Publishers. Manufactured in The Netherlands., pp. 115–153, 2001. [Online]. Available: http://link.springer.com/chapter/10.1007/978-1-4615-1627-9\_6
- [16] G. Karypis, "Evaluation of item-based top-n recommendation algorithms," 2000, pp. 247–254.
- [17] S. Owen, R. Anil, T. Dunning, and E. Friedman, Mahout in Action. Manning Publications, 2011. [Online]. Available: http://www.mendeley.com/research/meap-edition-manningearly-access-program-mahout-action-version-6/
- [18] C. D.Manning, P. Raghavan, and H. Schütze, "An Introduction to Information Retrieval," *Cambridge University Press*, p. 581, 2009.
- [19] H. Marmanis and D. Babenko, Algorithms of the intelligent web. Manning Publications Co., 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1717301
- [20] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the Fourteenth* ..., 1998.
- [21] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609–14623, Nov. 2011.
- [22] S. Luke, *Essentials of Metaheuristics*. Lulu, 2009, available for free at http://cs.gmu.edu/~sean/book/metaheuristics/.
- [23] Computational Intelligence An Introduction. Andries P. Engelbrecht, 2007.
- [24] M. Mitchell, An introduction to genetic algorithms. MIT press, 1998.