Genetic Algorithm Based Recommender System – IGC Plus: A Novel Clustering Dependant Recommender System

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 2, Issue No. 1, 24–40
©The Author(s) 2013
https://journals.mirlabs.in/index.php/ijcsnt
10.18486/ijcsnt/2.1.018

Mohd Abdul Hameed[1], S. Ramachandram[2], Omar Al Jadaan[3]

Abstract

Information Gain through Clustering Plus (IGC+) is a novel clustering dependant recommender system applied in both cold start and non-cold start problems. For comparison purposes, a number of Clustering Dependant Recommender Systems (CD_RSs) have been consider and explored in this paper including IGCRGA (Information Gain Clustering through Rank Based Genetic Algorithm), IGCEGA (Information Gain Through Clustering Elitist Genetic Algorithm), among others. The two evaluation metric, Mean Absolute error (MAE) and Expected Unity (EU) used in this work show that, IGC+ emerged vector in terms of providing quality recommendation to the end user amongst the CD_RSs considered.

Keywords

Expected Utility (EU), Mean Absolute Error (MAE), Clustering Dependant Recommendation System (CD_RS), Collaborative Filtering (CF), Popularity, Entropy, Bisecting K-mean Algorithm, Genetic Algorithm (GA), Elitist Genetic Algorithm (EGA), Rank Based Genetic Algorithm (RGA) and Clustering Plus.

Received: 12 December 2012; **Revised:** 31 March 2013; **Accepted:** 15 April 2013;
Published: 30 April 2013;

[1, 2]Dept. of CSE, University College of Engg, Osmania University, Hyderabad, Telangana , India.
[3]Medical and Health Sciences University, University College of Engg, Ras Al-Khaimah, United Arab Emirates.

Corresponding author:
Email: researcher.hameed@gmail.com

© 2013 by **Mohd Abdul Hameed, S. Ramachandram and Omar Al Jadaan** Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license,(http://creativecommons.org/licenses/by/4.0/). This work is licensed under a Creative Commons Attribution 4.0 International License

Introduction

Information Gain through Clustering Plus (IGC+) is a novel and proposed clustering dependant recommender system applied in both cold start and non-cold start problems. The motivation behind developing IGC+ is highlighted under problem statement in section II and a detailed explanation of IGC+ is given in section III.

Al Mamunur Rashid et al[2] proposed IGCN (Information Gain Clustering Neighbor), a recommender system which use IG (Information Gain) and clustering in a bid to generate recommendations. This coined the beginning of CD_RSs (clustering dependant recommender systems).

Mohd Abdul Hameed et al[1,10,11] developed this concept to a higher level by proposing and developing a number of CD_RSs – namely, IGCGA, IGCEGA, IGCRGA, among others – briefly explained in section IV of this paper. Other name coined for these breeds of RS include IG dependant heuristics and Hybrid systems. The latter name was suggested because these systems use both IG (Information Gain) and CF in the process of providing a recommendation to a user.

These various CD_RSs were implemented and experimented upon and the results are discussed in section VII of this paper, and thereafter the conclusion follows, section VIII.

Motivation / Problem Statement

First and foremost, there is lack of information and knowledge on the performance, behavior and effects of Clustering Plus, a proposed and novel clustering algorithm, in personalization problems – both cold start and non cold start problems; and secondly, there is the urge and zeal to improve the quality of recommendation provided by the recommender systems. In view of this, a new approach, Information Gain through Clustering Plus (IGC+) is proposed, developed and experimented upon in a bid to address this information gap and also provide an improvement in terms of better recommendation.

Information Gain through Clustering Plus (IGC+)

Al Mamunur Rashid et al[2] argues that the problem with information theoretic measures like entropy, popularity, among others, is that they are static by nature, i.e. they are unable to adapt to the rating history of the users. As such, informativeness of items not rated so far is the same for all users of the system regardless of their rating history; however, perfect personalization of a user needs a dynamic algorithm, which has the ability to adapt to continuously changing user rating pattern / style, which lead to better selection of the best neighbor.

To achieve this objective in IGC+, the computation of IG of each item is repeated for each iteration. Based on previous rating users are clustered using Clustering Plus approach. In IGC+, bisecting k-mean is replaced by Clustering Plus to eliminate the local minima sensitivity of k-mean algorithm – the clustering algorithm used in IGCN – and focus on global minima.

IGC+ consists of two sessions, namely, the clustering session in which Clustering Plus is applied, and the recommendation or profiling session which is further subdivided into non-personalization and personalization stages. The Non personalization stage is used to address the cold start problem, in other words, new users, while the personalization stage solves problems associated with users whose profiles are known.

Clustering Session: Clustering using Clustering Plus

The basic feature of clustering is to group the users such that similarity is maximized in intra cluster and minimized in inter cluster users. Based on the above similarity function, the clusters are regarded as chosen user neighborhood, the required neighbors of a user may join from any cluster.

The objective of Clustering Plus is to partition a data set $X = \{x_i \mid x_i \in R^d, i = 1, 2, \ldots, n\}$ into K clusters so as to minimize the Total Within Cluster Variation (TWCV),

$$f(M) = \sum_{j=1}^{k} \sum_{x \in c_j} \|x - m_j\|^2 \tag{1}$$

where $M = \{m_1, m_2, \ldots, m_k\} \subset R^d$ is the parameter set of $f(M)$ and $m_j = [m_{j1}, m_{j2}, \ldots, m_{jd}]$ is the prototype of cluster C_j.

The basic steps of clustering by Clustering Plus are described in details below:

1. String representation: Each string is a sequence of real numbers representing the K cluster centers. For an N-dimensional space, the length of a string is $N * K$, where the first N positions (or, genes) represent the N dimensions of the first cluster center, the next N positions represent those of the second cluster center, and so on.

Example for Representation/encoding of chromosomes: The data of a user is set of demographic variables as discussed under IGCN. If we represent it using **string of encoding method** a sample chromosome for centroids C1 = (M, 52, Doc) and C2 = (M, 60, Engg) will be:

M	52	Doc	M	60	Engg

Figure 1. Chromosome representation in string-of-encoding method.

Here Length of a gene = dimension (d) = 3 and No of genes = No of clusters (N) = 2. Total length of the chromosome = $N * d = 2 * 3 = 6$.

Prototype Embedded representation: Here each gene is not just a centroid instead a prototype of the clusters i.e. Length of each gene = dimension* no of clusters. A sample chromosome in this case will be:

	Gene1						Gene2				
M	52	D oc	M	30	Eng g	F	28	Eng g	M	6 0	Doc

Figure 2. Chromosome representation in prototype embedded method.

Here dimension = 3 but length of gene = $3 * 2 = 6$. Therefore length of chromosome = no of cluster*gene length = $2 * 6 = 12$.

2. Population initialization: K cluster centers encoded in each string are initialized to K randomly chosen points from the dataset. This process is repeated for each of the P chromosomes in the population, where P is the size of the population.
3. Fitness computation: The fitness computation process consists of two phases. In the first phase, the clusters are formed according to the centers encoded in the string under consideration. This is done by assigning each point X_i, $i = 1, 2 \ldots n$, to one of the clusters C_j with center Z_j such that: $\|X_i - Z_j\| < \|X_i - Z_p\|$, where $p = 1, 2 \ldots k$ and $j \neq p$. All ties are resolved arbitrarily. After the clustering, the cluster centers encoded in the string are replaced by the mean points of the respective clusters. In other words, for cluster C_i, the new center Z_i^* is computed by equation (1). These Z_i^*s replace the previous Z_is in the string. Subsequently, the clustering metric M is computed by equation (2),

$$z_i^* = \frac{1}{n_i} \sum_{x_j \in c_i} x_j, \quad j = 1, 2 \ldots k \tag{2}$$

$$M = \sum_{i=1}^{K} m_i M_i, \quad \text{where } M_i = \sum_{x_j \in c_i} \|x_j - z_i\| \tag{3}$$

   The fitness function is defined as $f = 1/M$, so that maximization of the fitness function leads to the minimization of M.
4. Selection: The chromosomes are selected from the mating pool according to their ranks[6]. In the proportional selection method, a string is assigned a number of copies, which are proportional to their rank in the population, and are sent to the mating pool for further genetic operation. Roulette wheel selection[6,7] is one common technique that implements the proportional selection method[3].
5. Crossover: The N-1 point crossover is used where each crossover point is set on the boundary between two prototypes in the chromosome. The two chromosomes involved in the crossover exchange their prototypes with each other with a predefined crossover probability. When $N \geq 2$, the crossover probability must be less than 1 to avoid the situation that the offspring are just duplicates of their parents.

Example: One point crossover: In this type, a crossover point is chosen and both the parents split into two parts at that point. The right part of the first chromosome is appended to left part of second chromosome. This produces first offspring. Then right part of the second chromosome is appended to left part of the first chromosome to produce second offspring. This is demonstrated via following example:
N-point crossover: In this type, instead of one, N crossover points are chosen and the parents split into N+1 parts. The parents swap their alternative parts to produce two child chromosomes[4]. See example below:

6. Mutation: In most cases, the mutation operator is to make a random change to the alleles of the chromosomes. The K-means operator (KMO)[9,15] is used for mutation since it guarantees to improve the clustering. KMO performs one step of the K-means algorithm using the prototypes in the chromosome as the initial prototypes, and then the resulting prototypes are concatenated

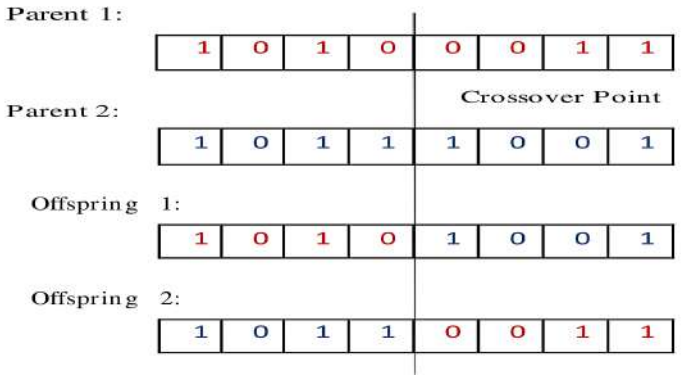

Figure 3. One point crossover

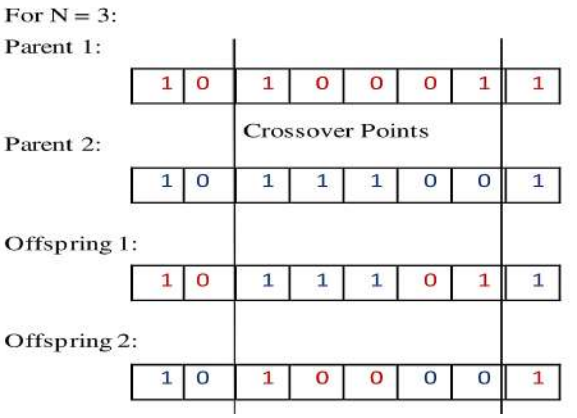

Figure 4. N-point crossover

to produce the new chromosome. In GKA, both mutation and KMO operators might yield chromosomes with some empty clusters containing no data samples. In[8], these chromosomes were termed as illegal strings and were eliminated by several time-consuming schemes. FGKA[15] improved GKA by avoiding this overhead. In PGKA, the illegal prototypes corresponding to empty clusters will not be updated by the one-step K-means clustering in the KMO mutation. Since the chromosomes with illegal prototypes usually have larger TWCV values, they are knocked out during the evolution process of GA[10].

7. Elitism: Elitism, a new operation, has been added to improve the quality of GA results and guarantee the convergence to global solution, where the good solutions are not lost during the other genetic operations / phases. In this phase, the two populations, parent and children population, are put together, then sorted based on their fitness and the best N solutions are selected and incorporated in the new generation, where N is the population size.

8. Termination criterion: In this phase, the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of iterations. The best strings identified up to the last generation provide the solution to the clustering problem.

Algorithm 1 shows the pseudo codes for Clustering Plus.

Algorithm 1: Clustering Plus

```
Generate the Initial Population Randomly.
Evaluate the Individuals in the Population and
    assign a fitness value to each individual.
repeat
    Selection Operation.
    Crossover Operation.
    Mutation Operation
    Elitism Operation
Until Stopping Criteria is met
```

Profiling Session

The created user clusters are used for profiling, the best approach applied in this situation is ID3 algorithm, which presents results in the form of a decision tree that holds cluster numbers at leaf nodes and each internal node represents a test on an item indicating the possible way the item can be evaluated by the user[5]. The item which holds the maximum IG is taken as the root node. The IG of an item a_t is evaluated using equation below.

$$IG(a_t) = H(C) - \sum_r \frac{c_{at}^r}{|c|} H(c_{at}^r) \tag{4}$$

The evaluation of Information Gain (IG) is the initialization process for the Profiling Session. The pseudo code for IGC+ are shown in Algorithm 2.

Algorithm 2: IGC+

```
// Clustering Session
Create c user clusters using Clustering Plus
// Profiling Session
Compute information gain (IG) of the items
// Non-personalized step: The first few ratings to
   build an initial profile
repeat
    Present next top n items ordered by their IG value
    Add items the user is able to rate into his profile
until the user has rated at least i items
//Personalized step: Toward creating a richer profile
repeat
    Find best l neighbors based on the profile so far
```

```
    Re-compute IG based on the l users' ratings only
    Present next top n items ordered by their IG values
    Add items the user is able to rate into his profile
until best l neighbors do not change.
```

Other Clustering Dependant Recommender Systems (CD_RSs)

Table 1. Comparison of IG Dependant Heuristics (CD_RSS)

yellow **Features**	**IGCN**	**IGCGA**	**IGCEGA**	**IGCRGA**	**IGC+**
Global Minima	Sometimes attained	Always attended	Always attended	Always attended	Always attended
Initialization value	Dependant	Independent	Independent	Independent	Independent
Elitism	Not applicable	Absent	Present	Present	Present
Selection type	Not applicable	Proportional	Proportional	Rank based	Rank based
Cross over type	Not applicable	Single point	Single point	Single point	N-1 point
Mutation operator	Random number	Random number	Random number	Random number	K-mean
Clustering algorithm	Bisecting K-mean	GA	EGA	RGA	Clustering Plus

Table I shows a comparison of CD_RSs along the various dimensions. The CD_RSs explored in this section in chronological order include IGCN, IGCGA, IGCEGA, and IGCRGA.

IGCN: Information Gain through Clustered Neighbors

As an information theoretic measure, one advantage of IGCN over popularity, entropy and its variants is that it takes into account the user's historical ratings and thereby, more adaptive to user's rating history.

IGCN works by repeatedly computing information gain of items, where the necessary ratings data is considered only from those users who match best with the target user's profile[4,12]. Users are considered to have labels corresponding to the clusters they belong to; and the role of the most informative item is treated as the most useful to the target user in terms of reaching the appropriate representative cluster.

IGCN consists of two sessions, namely, the clustering session in which bisecting k-mean algorithm is applied, and the recommendation or profiling session which proceeds with evaluation of IG (Information Gain) and is further subdivided into non-personalization and personalization stages. The Non personalization stage is used to address the cold start problem, in other words, new users, while the personalization stage solves problems associated with users whose profiles are known.

Algorithm 3 shows the pseudo codes for IGCN. This algorithm is similar to IGC+, except in IGC+, Clustering Plus is used as a clustering algorithm.

Algorithm 3: IGCN Algorithm

```
- Create c user clusters using bisecting k-mean
- Compute information gain (IG) of the items
```

```
// Non-personalized step: The first few ratings to
   build an initial profile
-Repeat
    - Present next top n items ordered by their IG scores
    - Add items the user is able to rate into her profile
-Until the user has rated at least i items
// Personalized step: Toward creating a richer profile
-Repeat
    - Find best l neighbors based on the profile so far
    - Re-compute IG based on the l users' ratings only
    - Present next top n items ordered by their IG scores
    - Add items the user is able to rate into her profile
-Until best l neighbors do not change
```

Example of how IGCN works on a data set:

Step 1: Clustering using k-means. The clustering of users in IGCN is done based on user's demographic data viz. gender, age and occupation. In this example each user record is set of these three items. For simplicity we are using following abbreviations: Gender M – Male, F – Female; Occupation Engg – Engineering, Doc – Doctor. If (M, 36, Doc) is a user record then it means this is data of a Male of 36 years age whose occupation is Doctor. Let table-1 be an example data set that is to be clustered using k-means algorithm.

Table 2. Example dataset to be clustered.

User record No	Point
1	(M, 25, Engg)
2	(F, 50, Doc)
3	(M, 39, Engg)
4	(M, 22, Doc)
5	(F, 27, Engg)
6	(M, 40, Engg)
7	(F, 30, Engg)
8	(M, 60, Engg)
9	(F, 44, Doc)
10	(m, 54, Doc)
11	(F, 37, Doc)
12	(F, 47, Doc)

Initially some random cluster centers are chosen. Here we choose user record 4 and 8 of the data set. This means there will be 2 clusters such that first cluster have got (M, 22, Doc) as its center and second have got (M, 60, Engg). Using similarity metric we have to find similarity of all the user records with these initial cluster centers. The similarity metric used here is jaccard's coefficient defined below [16,17]:

$$\text{Jaccard co-efficient} = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

Beginning with User record1, similarity between User record1 (M, 25, Engg) and C1 (M, 22, Doc) $= \frac{|A \cap B|}{|A \cup B|} = \frac{1}{5}$. Similarity between User record1 and C2 (M, 60, Engg) $= \frac{|A \cap B|}{|A \cup B|} = \frac{2}{4}$. User record1 is got more similarity with C2 than with C1 so it falls in cluster 2. Similarly the similarity of all the user records with respect to both the centroids is tabulated below:

Table 3. K-means clustering iteration-1.

User record No	Point	Similarity with C1 (M,22,Doc)	Similarity with C2 (M,60,Engg) → Cluster
1	(M,25,Engg)	1/5	2/4 – C2
2	(F, 50, Doc)	1/5	0/6 – C1
3	(M,39,Engg)	1/5	2/4 – C2
4	(M, 22,Doc)	3/3	1/5 – C1
5	(F, 27,Engg)	0/6	1/5 – C2
6	(M,40,Engg)	1/5	2/4 – C2
7	(F, 30,Engg)	0/6	1/5 – C2
8	(M,60,Engg)	1/5	3/3 – C2
9	(F, 44, Doc)	1/5	0/6 – C1
10	(M,54, Doc)	2/4	1/5 – C1
11	(F, 37, Doc)	1/5	0/6 – C1
12	(F, 47, Doc)	1/5	0/6 – C1

Next find the new cluster centers for both the clusters C1 and C2 by taking similarity of all the records falling in cluster1 and cluster2 respectively. Records in C1 have got majority females so the gender of centroid will be female. Likewise occupation will be doctor and age will be mean of all the ages in cluster1. This gives us the new centroid C1 = (F, 42, doc) (mean of ages is 42.3 which is rounded off to 42). For centroid 2 we have majority gender as males, occupation as Engineering and mean of ages = 36.8. So C2 = (M, 37, Engg). Find similarity of all the above values using jaccard's coefficient with respect to the new cluster centers and re-categorize the values.

Table 4. K-means clustering iteration-2.

User record No	Point	Similarity with C1 (F,42,doc)	Similarity with C2 (M,37,Engg) → Cluster
1	(M, 25, Engg)	0/6	2/4 – C2
2	(F, 50, Doc)	2/4	0/6 – C1
3	(M, 39, Engg)	0/6	2/4 – C2
4	(M, 22, Doc)	1/5	1/5 – C2
5	(F, 27, Engg)	1/5	1/5 – C2
6	(M, 40, Engg)	0/6	2/4 – C2
7	(F, 30, Engg)	1/5	1/5 – C2
8	(M, 60, Engg)	0/6	2/4 – C2
9	(F, 44, Doc)	2/4	0/6 – C1
10	(m, 54, Doc)	1/5	1/5 – C1
11	(F, 37, Doc)	2/4	1/5 – C1
12	(F, 47, Doc)	2/4	0/6 – C1

See tuples 4, 5, 7, 10, their similarity with both C1 and C2 is same in such a case decide their cluster based on their age difference with respect to the centroids. Using this we have modified tuple 4 cluster

to be C2. Now again find the next new cluster centers in the same way as done above. The resulting centroids are: C1 = (F, 46, Doc), C2 = (M, 34, Engg). Take these new centroids and re-categorize the records until there is no change in cluster values.

Observe that the algorithm depends on cluster centers chosen initially. For different initial cluster centers different solutions are possible. The found solution might not be the optimal one. This is because the algorithm depends on initial chosen centers and terminates as soon as a solution is found out irrespective of whether this is an optimal solution or not.

Step 2: calculate Information gain of items using the above formula.

Step 3: Making a non-Personalize Profile. Present these items to user for rating in descending order of their IGs. If user rates item add it to his/her profile. This is done as follows:

Table 5. Example profiles of some users.

User Record No	Item1	Item2	Item3	Item4	Cluster
1	3	2	3	0	C1
2	3	0	4	2	C2
3	2	3	4	2	C1
4	2	3	4	5	C2
5	3	0	3	4	C2
6	3	0	1	4	C1
7	3	2	1	4	C1
8	2	3	1	0	C2

To the table of users we add their ratings and the cluster they fall in. Each tuple here is profile of some user. Like the first tuple is profile of user 1, second of user2 and so on. When a new user comes in a tuple is created for this user and ratings are added. If a user does not rate a particular item the respective item will have a null rating.

Step 4: repeat step 3 till some i items are rated. To make a non-personalized profile some i items are to be rated by this user. Later a personalized profile is made as shown in next step.

Step 5: Making a Personalized Profile. Here first a set of best l-neighbors for the target user are found. This is done using jaccard's coefficient. Let this be profiles of two users:

Table 6. Profile of user1.

user id	Item id	Rating
4	1	3
	12	1
	26	2
	30	4
	35	2

Jaccard co-efficient $= \frac{|A \cap B|}{|A \cup B|} = \frac{2}{8} = \frac{1}{4}$. Next re-compute IGs and present items to user in descending order of their IG scores. When any item is rated by the user it is added to the profile. Repeat current step until best l-neighbors do not change.

Table 7. Profile of user2.

user id	Item id	Rating
5	1	3
	13	4
	26	2
	30	3
	40	2

IGCGA: Information Gain – Clustering through Genetic Algorithm

In normal k-means clustering, centers are randomly selected as initial seeds and clustering proceeds in steps / Phases. In each step, points are reassigned to the nearest cluster. This process has the inherent ability to keep the cluster centers generated in each step to be very close to the initial chosen random centers. As such friend centers of this clustering technique are heavily dependent upon initial choice of centers, which is random. Due to this uncertainty attributed to the random initialization, it is desirable to introduce some heuristic to make sure that the clustering finally reflects optional clusters (as measured by some clustering metric). GA and K-means based GA are such technique introduced to target and optimize the aforementioned fallback.

The searching capability of GAs is used in IGCGA for purposes of appropriately determining a fixed number K of cluster centers in R^N, thereby, appropriately clustering the set of n unlabeled points. The clustering metric that is adopted is the sum of the Euclidean distances of the points from their respective cluster centers. Mathematically, the clustering metric M for the k clusters $C_1, C_2 \ldots C_K$ is given by

$$M(C_1, C_2 \ldots C_K) = \sum_{i=1} \sum_{x_j \in C_i} \|x_j - z_i\| \tag{6}$$

The task of the GA is to search for the appropriate cluster centers $z_1, z_2 \ldots z_k$ such that the clustering metric M is minimized.

In view of this, Abdula Hameed et al[1], proposed IGCGA, which uses GA in the clustering process instead of K-mean bisecting algorithm used in IGCN. K-mean Bisecting Algorithm, as explained above, does not guarantee the attainment of global minima, in other words, the algorithm sometimes locks up in local minima depending on the initial random centre values chosen. However, on the other hand GA ensues that global minima is attained. The effect of this has produced good result in terms of better recommendation for IGCGA as compared to IGCN. This fact is supported by the two evaluation metrics discussed in section VI.

In GA, randomly generated solutions (centers) are populated and in each step of the process, are evaluated for their fitness weight giving greater emphasis to solutions offering greater fitness; and by so doing, there is surety that only good solutions are influenced in the final clusters[17]. In addition, the crossover and mutation phases ensure production of better solution based on previous solution. Generally, the technique, iterated over several generations, ensures that most of the points in the solution space become randomly selected potential initial centers and are evaluated in the next steps. This leaves no room for any uncertainty raised due to the initial selection. The whole solution space is traversed in search of a potential center, and hence the possibility of ensuring a global maxima is high. This is the

main advantage of using GA over normal k-mean algorithm and this benefit has been fully exploited in IGCGA.

The pseudo codes for IGCGA are similar to those of IGCN, however in IGCGA, GA is used as a clustering algorithm. A comparison between GA and Clustering Plus shows that the selection, crossover, mutation are performed differently and furthermore, GA lacks the Elitism stage.

Since four stages – namely, string representation, population initialization, fitness computing and termination – are the same as those of Clustering Plus and already discussed in section III, subsection A, our discussion is proceeding with and highlighting only those stages which are performed differently in GA including Selection, Crossover and Mutation.

Selection: chromosomes are selected from the mating pool directed by using the survival of the fittest concept of natural genetic systems. In this selection, also called proportional selection method, a string is assigned a number of copies proportional to its fitness in the population, that are picked to be incorporated into the mating pool for further genetic operations. Roulette wheel selection[6,7] is one common technique that implements the proportional selection method.

Table 8. Mutation parameters

Population Size	Max. Gen	Mutation Prob.	Crossover Prob.
25	100	0.09	0.9

Crossover: Crossover is a probabilistic process that exchanges information between two parent / initial chromosomes for generating two children / resultant chromosomes. In GA, single point crossover with a fixed crossover probability is used. For chromosomes of length l, a random integer, called the crossover point, is generated in the range $[l, l-1]$. The portions of the chromosomes lying to the right of the crossover point are exchanged / swapped to produce two offspring.

Mutation: Each string undergoes mutation with a fixed probability. For binary representation of chromosomes, a bit position (or gene) is mutated by simply flipping its value. Since floating representation is considered, mutation is effected by a number in the range $[0, 1]$ generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$$\begin{aligned} &v \pm 2 \times \delta \times v, \quad v \neq 0 \\ &v \pm 2 \times \delta, \quad v = 0 \end{aligned} \tag{7}$$

The $+$ or $-$ sign occurs with equal probability. Note that mutation could have been implemented as $v \pm \delta \times v$. However, one problem with this form is that if the values at a particular position in all the chromosomes of a population become positive (or negative), then a new string having a negative (or positive) value at that position cannot be generated. In order to overcome this limitation, a factor of 2 is incorporated while implementing mutation. Other forms like $v \pm \delta \times E \times v$ where $0 < E < 1$ would also have satisfied the purpose. One may note in this context that similar sort of mutation operators for real encoding have been used mostly in the realm of evolutionary strategies ([8], Chapter 8). Other parameter values used in the mutation procedure are listed in table VIII above.

IGCEGA: Information Gain – Clustering through Elitist Genetic Algorithm

IGCEGA is a CD_RS which uses EGA as its clustering algorithm. A comparison of EGA with GA shows that EGA has an Elitism stage which GA does not incorporate. The Elitist stage, which has already been discussed in detail in section III subsection A, has given EGA an advantage over GA in that it tends to preserve good solution which would have been lost during mutation stage, the preceding stage. Because of this inherent advantage of EGA, IGCEGA has outperformed IGCGA – which uses GA during clustering – in term of producing better recommendation to the end user.

The pseudo codes for IGCEGA are no different from those of IGCGA and IGCN except in IGCEGA, EGA is used as the clustering algorithm.

IGCRGA (Information Gain Clustering through Rank Based Genetic Algorithm)

IGCRGA is a novel CD_RS for solving personalization problems. In a bid to improve the quality of recommendation of RS and to alleviate the problem associated with personalization heuristics, which use fitness value in the selection stage of the clustering process, Abdula Hameed et al[11] proposed IGCRGA.

IGCRGA using RGA a derivative of GA still resolves the problem associated with IGCN (Information Gain Clustering Neighbor) which sometime traps the algorithm in local clustering centroids. Although this problem was alleviated by both IGCGA and IGCEGA, IGCRGA solves the problem even better and this fact is supported by Mean Absolute Error (MAE) and Expected utility (EU), the two evaluation metric used in this work.

Personalization heuristics (for example IGCGA, IGCEGA, among others), which use fitness value in the selection phase of clustering process, are associated with the problem of lack of diversity in the selection due to biasness evaluated as the absolute difference between two fitness values.

Both RGA and EGA are GA used in clustering and therefore have a lot in common including the basic stages involved in their algorithms. However the difference between the two is implicit and lies in the selection stage[?]. RGA uses rank based selection while EGA uses proportional selection in reference to fitness value. This difference has transformed into a tremendous boost in terms of better recommendation for IGCRGA. Rank based selection is discussed under Clustering Plus in section III, subsection A.

Experimentation

The CD_RSs discussed in section III and IV above, were implemented through the use of Java JDK 6, a java programming language flavour. For purposes of experimentation, these CD_RSs interacted with the dataset obtained from Movie Lens database, and particularly, with the base table – containing 100,000 records and 3 attributes namely user_Id, movie_id and rating.

The MAE (Mean Absolute Error) and EU (Expected Utility) – used as evaluation metrics – corresponding to each CD_RS and to the size of the recommendation was computed and tabulated as shown in tables XI and X respectively. The result of the tables is represented in graph form in figure 5 and 6.

The formula for computing Mean Absolute Error (MAE), a widely used metric in the CF domain, is shown below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{8}$$

Table 9. Mean Absolute Error (MAE) for the CD_RS

Sample size	15	30	45	60	75	Mean
IGCN	0.7561	0.7133	0.6856	0.6598	0.6341	0.6898
IGCGA	0.6811	0.6492	0.6174	0.6014	0.5713	0.6241
IGCEGA	0.6883	0.6447	0.6121	0.5972	0.5684	0.6221
IGCRGA	0.6708	0.6128	0.5812	0.5725	0.5547	0.5984
IGC+	0.6683	0.6011	0.5753	0.5643	0.5531	0.5924

Table 10. Expected Utility (EU) for the CD_RSS

Sample size	15	30	45	60	75	Mean
IGCN	3.2010	5.4128	6.0115	6.2548	6.5219	5.4804
IGCGA	5.2243	6.0410	6.1954	6.5949	6.9610	6.2033
IGCEGA	5.2924	6.0089	6.3072	6.5574	7.0868	6.2505
IGCRGA	5.8993	6.5249	6.7113	6.9018	7.0107	6.6096
IGC+	5.2366	6.6684	6.9107	7.2870	7.4936	6.7193

where n represents the number of movies recommended, f_i - expected value of rating and y_i - actual recommendation generated. In other words, MAE is a measure of deviation between the recommendation and the corresponding actual ratings.

However, a limitation of MAE is that it only considers absolute differences. MAE shows no difference between two pairs of (actual recommendation, expected rating) for a movie that are rated (1, 5) and (5, 1), although users may be more unhappy about the former pair. Therefore, another accuracy metric, Expected Utility[13] is used that penalizes false positives more than false negatives. The formula used for evaluating EU is given below:

$$EU = \sum_{i}^{1} \sum_{j}^{1} U(R_i, R_j) * P(R_i|R_j) \tag{9}$$

where $U(R_i, R_j) = R_j - 2 * |R_i - R_j|$. and $P(R_i|R_j)$ is probability of occurrence estimated using an m-estimate Cestnik smoothing[14]. The m-estimate can be expressed as follows:

$$p = \frac{r + m * P_i}{n + m} \tag{10}$$

where n is the total number of examples, r is the number of times the event for which the probability is estimated occurs, m is a constant, and P_i is the prior probability[12].

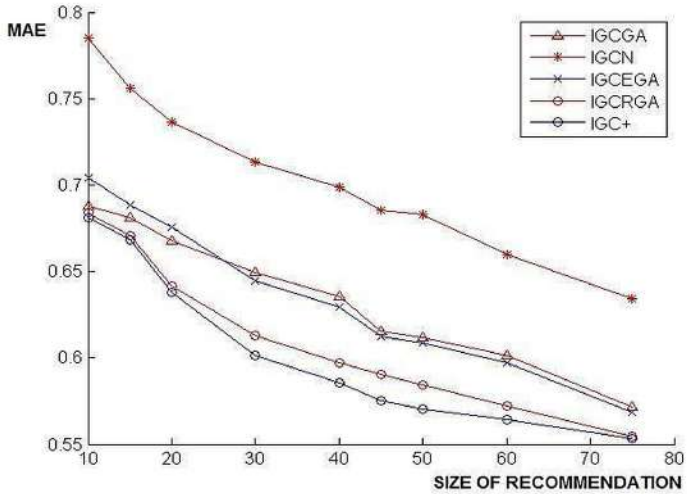

Figure 5. Graphic Representation of MAE Vs Size of Recommendation for the CD_RSs

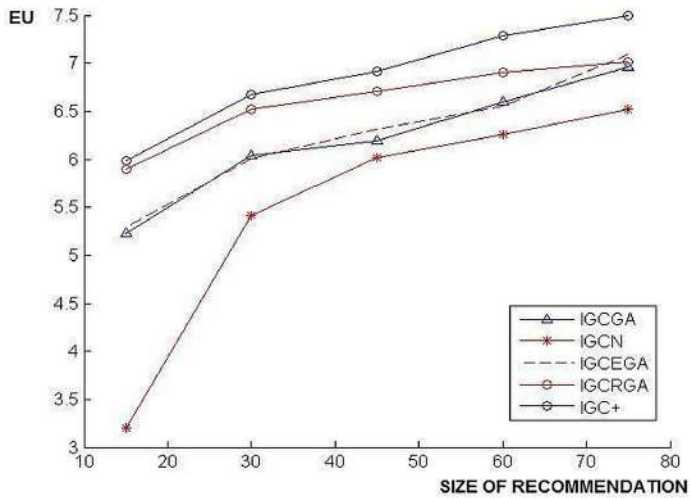

Figure 6. Graphic Representation of EU Vs Size of Recommendation for the CD_RSs

Results and Discussion

The result generated by the experimentation (figures 3 and 4) show that IGC+ provides the best recommendation; and this fact is supported by MAE and EU, the two evaluation metrics used in this work.

There is a remarkable improvement in terms of quality of recommendation between IGC+ and the pervious CD_RSs, especially ICGN the base CD_RS; that is to say, in terms of EU there is an improvement of 22.6%.

This improvement can be accounted for by the improved crossover and mutation procedures as compared to those of IGCRGA. Before the advent of IGC+, IGCRGA was regards as the best CD_RS around; however with the advent of IGC+ this fact ceased to be true.

The improved recommendation demonstrated by IGC+ points to the fact that whenever the quality of clustering is improved so is the quality of recommendation.

In reference to table I and figure 5 and 6, It is evident that first, CD_RSs that applies GA and its derivatives as clustering algorithm produce better recommendation than those which do not for instance IGCN; second, CD_RSs which incorporate Elitism in their clustering algorithm perform better; third, CD_RSs which integrate rank based selection in their GA dependent clustering algorithm exhibit better performance; fourth, N-1 point crossover produce better results in CD_RSs than single point crossover; and finally, k-mean operator applied in the mutation procedure / stage out performs other mutation methods. All these considerations and conclusions were taken into account while developing IGC+, and as a result IGC+ emerged as a superior CD_RS.

It is worthwhile noting that the quality of recommendation is directly proportional to the size of recommendation and this fact is supported by the two evaluation metrics and is true for all CD_RSs in this work.

Conclusion

Clustering has a bearing to the quality of recommendation, and therefore Improving the quality of clustering plays a major role in improving the quality of recommendation. This fact is supported by the emergence of the superior CD_RD, namely, IGC+.

In reference to cold start and non cold start problems, IGC+ produces the best result in terms of generating the best recommendation for both static and dynamic environments.

Generally, the investigation on CD_RSs show that the quality of recommendation improves with increase in recommendation size. IGC+ can be applied in any type of personalization problem, let it be web or non web personalization.

References

[1] Mohd Abdul Hameed, Omar Al Jadaan, and S. Ramachandaram, “Information theorectic aproach to cold start problem using Genetic Algorithm”, IEEE, 2010, ISBN: 978-0-7695-4254-6.

[2] Al Mamunur Rashid, Gerge Karypis and John Riedl, “Learning Preferences of new users in Recommender System: An Information Approach.”, SIGKDD Workshop on Web Mining and Web Usage Analysis (WEBKDD), 2008.

[3] Al Mamunur Rashid, Istvan albert, Dan Cosley, Shyong K. Lam, Sean MCNee, Joseph A. Konstan, and John Riedl, “Learn New User Preferences in Recomender Systems, 2002 international Conference on Intelliganent User interfaces. pp. 127-134.

[4] Isabelle Guyon, Nada Matic, and Vladimir Vapnik, “Discovering Informative Patterns and data cleaning”, 1996.

[5] Mitchell Thomas M., “Machine Learning”, McGraw-Hill Higher Education, 1997.

[6] Omar Al Jadaan, Lakshmi Rajamani, and C. R. Rao, “Improved selection operator for Genetic Algorithm”, Journal of Theoretical and Applied Information Technology, Vol. 4, No. 4, pp. 269-277, 2008.

[7] Omar Al Jadaan, Lakshmi Rajamani, and C. R. Rao, “Parametric study to enhance genetic algorithm performance, using ranked based Roulette Wheel Selection method”, International Conference on Multidisciplinary Information Sciences and Technology (InSciT2006), volume 2, pp. 274-278, Merida, Spain, 2006.

[8] Daniel Billsus and Michael J. Pazzani, “Learning collaborative information filters”, Proc. 15th International Conference on Machine Learning, pages 46-54. Morgan Kaufmann, San Francisco, CA, 1998.

[9] K. Krishna and M. Narasimha Murty, “Genetic K-means algorithm,” IEEE Trans. Systems, Man, and Cybernetics, 29(3), June 1999.

[10] Mohd Abdul Hameed, S. Ramachandram, and Omar Al Jadaan, “IGCEGA, an acronym for Information Gain Clustering through Elitizt Genetic Algorithm”, 2011 International Conference on Communication Systems and Network Technologie, 2011.

[11] Mohd Abdul Hameed, S. Ramachandram, and Omar Al Jadaan, “IGCRGA: A Novel Heuristic Approach for Personalization of Cold Start Problem”, 2011 Fifth Asia Modelling Symposium, 2011.

[12] Thomas M. Mitchell. Machine Learning. McGraw-Hill Higher Education, 1997.

[13] Al Mamunur Rashid, Shyong K. Lam, George Karypis, and John Riedl. Clustknn: A highly scalable hybrid model- & memory-based cf algorithm. WEBKDD 2006: Web Mining and Web Usage Analysis, 2006.

[14] Cestnik, B. 1990. “Estimating probabilities: A crucial task in machine learning.” Proc. Ninth European Conference on Artificial Intelligence. 147-149.

[15] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, “FGKA: A fast genetic K-means clustering algorithm,” Proc. ACM Symposium on Applied Computing, 2004.

[16] P. Berkhin, “Survey of clustering data mining techniques,” Springer, 2002.

[17] Zahid Ansari, A. Vinaya Babu, M.F. Azeem, Waseem Ahmed “Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions” World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 5, 217-226, 2011.