IGCEGA: A NOVEL HEURISTIC APPROACH FOR PERSONALISATION OF COLD START PROBLEM

Mohd Abdul Hameed, S. Ramachandram
Dept. of CSE
University College of Engg
Osmania University
Hyderabad 500007, AP, India
researcher.hameed@gmail.com
schandram@gmail.com

Omar Al Jadaan
Medical and Health Sciences University
Ras Al-Khaimah
United Arab Emirates
o_jadaan@yahoo.com

***Abstract* - IGCEGA, an acronym for Information Gain Clustering through Elitizt Genetic Algorithm, is a novel heuristic used in Recommender System (RS) for solving personalization problems.**

In comparison with IGCGA (Information Gain Clustering through Genetic Algorithm), IGCEGA is not associated with the inherent problem of increasing the possibility of losing good solution during the crossover phase, which translates into increasing the guarantee of converging to a global minima and consequently, enhancing the accuracy of the recommendation.

Besides, IGCEGA using the technique of global minima still resolves the problem associated with IGCN (Information Gain through Clustered Neighbor), which traps the algorithm in local clustering centroids. Although this problem was alleviated by IGCGA, IGCEGA solves the problem even better because IGCEGA assumes the lowest Mean Absolute Error (MAE), the evaluation matrix used in this work.

Results of the experimentation of the various heuristics / techniques in RS used in personalization for cold start problems – for instance Popularity, Entropy, IGCN, IGCGA - showed that IGCEGA is associated with the lowest MAE, therefore, best clustering, which in turn results into best recommendation.

Keywords: personalization, web personalization, mean absolute error, recommendation system, collaborative filtering, popularity, entropy, bisecting k-mean algorithm, genetic algorithm(GA), and elitist genetic algorithm(EGA).

I. INTRODUCTION

In a bid to alleviate the problem of IGCN (Information Gain through Clustered Neighbor), a heuristic based on the application of K-mean algorithm in Recommendation System (RS), Hameed [1] proposed IGCGA (Information Gain Clustering through Genetic Algorithm), which is based on the application of Genetic Algorithm (GA) in Collaborative Filtering (CF). Although the proposed algorithm improved the quality of recommendation, it is associated with the inherent problem of increasing the possibility of losing good solution during the crossover phase, which translates into lowering the guarantee of converging to a global minima and consequently, affecting the accuracy of the recommendation. In view of this, a new approach, IGCEGA is proposed in this work, which alleviate this problem and hence elevates the accuracy of RS.

Rashid [2] proposed and explored the use of information theoretic approaches, namely popularity, entropy0, HELF and IGCN, in web personalization however, the use of entropy was not implemented and therefore not experimented upon. In this work, entropy, popularity as well as hybrid systems (namely, IGCN, IGCGA, and IGCEGA) have been implemented and explored in a bid to solve cold start problem in relation to personalization. The hybrid systems experimented upon use both IG (Information Gain) and CF in the process of providing a recommendation to a new user.

Experimentation of the various heuristics / techniques in RS used in personalization for cold start problems was conducted and a comparison of their respective MAE was performed. The various heuristics / techniques explored include: Popularity, Entropy, IGCN, IGCGA and IGCEGA. The results are discussed in sections V.

A. *Problem statement*

First, IGCGA is associated with the inherent problem of increasing the possibility of losing good solution during the crossover phase of GA, which translates into lowering the guarantee of converging to a global minima and consequently, affecting the accuracy of the recommendation.

Second, there are few algorithms, whose solutions converge to a global minima and therefore, necessitating the need to explore other avenues in a bid to develop more algorithms.

In view of this, a new approach, IGCEGA is proposed in this work, to address these problems and challenges.

978-0-7695-4437-3/11 $26.00 © 2011 IEEE
DOI 10.1109/CSNT.2011.83

II. PERSONALIZATION HEURISTICS FOR COLD START PROBLEMS

This section presents a brief explanation of the various personalization heuristics for cold start problems use in this work. Table I shows a comparison of IG dependant heuristics.

A. Popularity

Popularity of an item indicates how frequently users rated the item. Popular items may be good at connecting people with each other as co-raters, since many people are likely to rate popular items. [2]

One disadvantage of using Popularity measure to elicit preferences, as pointed out by Al Mamunur Rashid et al [2], is the possibility of worsening the prefix bias - that is, popular items garnering even more evaluations. Unpopular items, lacking enough user opinions, may be hard to recommend. This situation would not improve if the system keeps asking opinions on popular items.

B. Entropy

Entropy of an item represents the dispersion of opinions of users on the item. Considering a discrete rating category, entropy of an item a_t, is given by equation: $H(a_t) = -\sum_{i=1}^{n} pi \log_2 \mathrm{pi}$

where p_i denotes the fraction of a_t's ratings that is equal to i. Notably, the logarithm to base 2 is used because entropy is a measure of the expected encoding length expressed in bits.

One limitation of entropy is that it often selects very obscure items leading to senseless information on items which are rated by a very small number of people, in which situation the rating frequencies or popularities cannot be inferred.

In general, it is not possible to infer the rating frequencies or popularities of items from their entropy scores. A plot (graph) between entropy and popularity (rating frequency, to be exact) of items, shows that entropy and popularity are only slightly correlated (correlation coefficient is only 0.13) [2].

TABLE I A COMPARISON OF IG DEPENDANT PERSONALISATION HEURISTICS

Features	IGCN	IGCGA	IGCEGA
Global Minima	Results do not attain	Results attain	Results attain
Initialization value	Dependant	Independent	Independent
Time complexity	Low	High	High
Gene / Chromosome Population	Not applicable	Reduce gradually with time	Preserved

A few other researchers who employed entropy as a measure for informativeness on other domains also report mixed results. For example, in their work on using information theoretic measures such as entropy to find informative patterns in data, Al Mamunur et al, observed that in addition to picking informative patterns, entropy suggests “garbage” (meaningless or not useful) patterns to be useful as well [3] .

Other variants of entropy, such as Entropy0, are not discussed and considered in this work.

C. 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]. 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.

The few design decisions taken into consideration while developing IGCN include:

- Goal of building profiles is to find right neighborhoods
- Neighborhoods correspond to user clusters

Algorithm 1 shows IGCN algorithm implemented in this work with a view of comparison with other personalization heuristics for cold start problems.

Algorithm 1: 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
```

D. 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 attributes 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 is one such technique introduced in IGCGA to target and optimize the aforementioned fallback.

The searching capability of GAs is used 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 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

$$M\,(C_1, C_2 \ldots C_K) = \sum_{i=1} \sum_{xj \in Ci} \| x_j - z_i \| \qquad (3)$$

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.

The algorithms for IGCGA and GA are shown in Algorithm 2 and 3 respectively.

Algorithm 2: IGCGA

```
- Create c user clusters using GA
- 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
```

Algorithm 3: GA

```
1: Generate the Initial Population Randomly.
2: Evaluate the Individuals in the Population and
   Assign a fitness value to each individual.
3: repeat
4:    Selection Operation.
5:    Crossover Operation.
6:    Mutation Operation.
7: Until Stopping Criteria is met
```

In IGCGA, 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. Moreover, 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 bisecting k-mean algorithm.

E. IGCEGA: Information Gain - Clustering through Elitist Genetic Algorithm

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. In IGCN, a previous approach, the computation of IG of each item is repeated in each iteration. Based on previous rating users are clustered using bisecting k-mean approach. In IGCEGA, bisecting k-mean is replaced by EGA to eliminate the local minima sensitivity of k-mean algorithm and focus on global minima. 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 formed 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. 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.

$$\mathbf{IG\ (a\ _t) = H(C) -}\ \textstyle\sum_r \frac{C^r_{at}}{|C|}\ \mathbf{H}\ (C^{\ r}_{at})$$

The basic steps of clustering by EGA 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.
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 … 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 …k and j ≠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 (2). These z_i^* s replace the previous z_i s in the string. Subsequently, the clustering metric M is computed by equation (3)

   $$z_i^* = \frac{1}{ni}\sum_{x\,j\in\,Ci} xj, \quad j=1, 2\ldots k \qquad (2)$$

   $$M = \sum_{i=1}^{k} mi\text{, where } m_i = \sum_{x\,j\in\,Ci} ||xj - zi|| \qquad (3)$$

   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 directed by using the survival of the fittest concept for natural genetic systems. In the proportional selection method adopted in this paper, a string is assigned a number of copies, which are proportional to their fitness in the population, and are sent to the mating pool for further genetic operation. Roulette wheel selection [5], [6] is one common technique that implements the proportional selection method.
5) Crossover: Crossover is a probabilistic process that exchanges information between two initial chromosomes for generating two resultant chromosomes. In this paper a 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 pair of chromosomes is broken at the crossover point and the four resultant pieces are interchanged.
6) 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 in this paper, mutation is effected by a number δ in the range [0, 1] generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

   $$v \pm 2 * \delta * v, \quad v \neq 0 \qquad (4)$$

   $$v \pm 2 * \delta\ , \quad v = 0 \qquad (5)$$

   The + or - sign occurs with equal probability. Note that mutation can be implemented as:

   $$v \pm \delta * v \qquad (6)$$

   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 it is impossible to generate a new string having a negative (or positive) value at that position. In order to overcome this limitation, a factor of 2 is incorporated while implementing mutation. Other forms like

   $$v \pm (\ \delta * \varepsilon) * v \qquad (8)$$

   where $0 < \varepsilon < 1$ would also have been satisfied. 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 [7], Chapter 8. The remaining parameter values are listed in table II.
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. The IGCEGA and EGA algorithm are depicted in algorithm 4 and 5 respectively.

TABLE 11 EGA PARAMETERS

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

Algorithm 4: IGCEGA

```
Create c user clusters using EGA
Compute information gain (IG) of the items
Non-personalized step:
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.
```

Algorithm 5: EGA

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

III. EXPERIMENTATION

For purposes of experimentation, the data-set obtained from Movie Lens database was used in this work. The base table - containing 100,000 records and 3 attributes namely user_Id, movie_id and rating - was used. The Systems interacting with this dataset and applying the heuristics discussed in section II above were implemented through the use of Java programming language. The MAE (used as an evaluation metric) corresponding to each heuristic and to the size of the recommendation was computed and tabulated as shown in table III. The result of the table is represented in graph form in fig.1.

Mean Absolute Error (MAE), a widely used metric in the (CF) domain, is the metric used for evaluating the quality of the recommendation, and is computed using the formula below.

MAE=$\frac{1}{n}\sum_{i=1}^{n}|fi - yi|$

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, one limitation of MAE is that only absolute differences are considered, that is to say, two pairs for example (1, 5) and (5, 1) of (actual rating, recommendation) are considered to be the same although the latter pair might be more undesirable to a visitor / user.

IV. RESULTS AND DISCUSSION

The result generated by the experimentation is shown graphically in fig. 1. It is evident from this representation that IGCEGA is associated with the lowest MAE which translates into the best recommendation, although the difference in MAE between IGCGA and IGCEGA is small. It is worthwhile noting that IGCEGA performs better than IGCGA when the recommendation size is greater than 30 items (in this case movies)

Compared to IGCN and the other two heuristics, both IGCGA and IGCEGA present a more significant performance in terms of generating accurate recommendation.

Though entropy has a lower MAE as compared to IGCN, its performance is undesirable in dynamic environment, that is to say, in environment where the historical data or the rating pattern / style of a visitor / user is continuously changing (dynamic). In contrary, Entropy works well in static environment, (that is to say, where the historical data of the visitor is not changing (static)).

As per the result of the experimentation, popularity, a static heuristic, presents the most undesirable recommendation since it has the highest MAE.

TABLE III MEAN ABSOLUTE ERROR (MAE) FOR THE PERSONALISATION HEURISTICS

Sample size	15	30	45	60	75	Mean
Popularity	0.8762	0.8039	0.7813	0.7075	0.6842	0.7706
Entropy	0.6945	0.6798	0.6703	0.6539	0.6320	0.6661
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

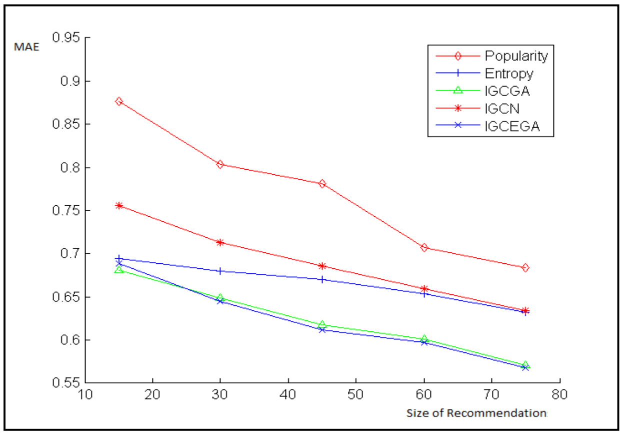

Figure 1 Graphic Representation of MAE Vs Size of Recommendation for the Personalization Heuristics

V. CONCLUSION

In reference to cold start problem, IGCEGA produces the best result in terms of generating the best recommendation for both static and dynamic environments. IGCEGA always out performs IGCGA when the recommendation size is more than 30 items.

Generally, the personalization heuristic investigated show that the quality of recommendation improves with increase in recommendation size.

IGCEGA can also be applied to solve web personalization problem.

VI. REFERENCES

[1] Mohd Abdul Hameed, OmarAl Jadaan and S. Ramachandaram. “Information Theorectic Aproach to Cold Start Problem using Elitist Genetic Algorithm”.Bhopal, Mandya Pradesh,India : 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” Minneapolis : 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” San Francisco, CA : s.n., 2002. 2002 internationalConference on Intelliganent User interfaces. pp. 127-134.

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

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

[6] 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.

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