Volume 4, Issue 8, August 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Implementation and Decoding of Novel Genetic Algorithm for Constraint Based Associative Clustering

**B. Kranthi Kiran***
[1]Assistant Professor, Department of Computer Science and Engineering, JNTUHCEJ, Karimnagar, Telangana, India

**A Vinaya Babu**
Professor, Department of Computer Science and Engineering, JNT University Hyderabad, Telangana, India

***Abstract:*** ***We are introducing a novel Genetic Algorithm for constraint based associative clustering. Subset formation of discretized data set is done. The data points are fitted to appropriate cluster using fitness calculation. Using Bayes factor, crossover computation, mutation and contingency table genetic algorithm is implemented. The performance was tested on UCI database using the evaluation index clustering accuracy. For Big data clustering our algorithm shows better accuracy (75.7%). The paper describes the entire implementation process.***

***Keywords: Discretization, subset Formation, Fitness Calculation, Bayes Factor, Crossover, Mutation, Contingency Table, Genetic Algorithm, Constraint, Associative Clustering.***

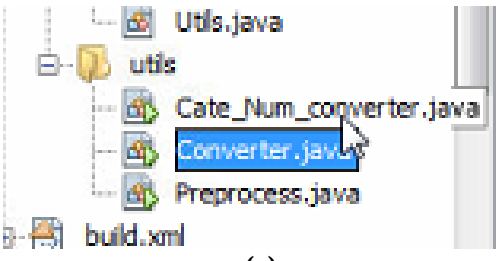
## I. INTRODUCTION

Clustering analysis by mapping real world census data to gene expressions is the key concept [24] of our proposal. Each record of the data set is treated as a chromosome or solution. The chromosomes are discretized first. The discretized chromosomes are partitioned in to subsets using subset formation [30]. Need of dimensions is evaluated and using fitness calculation the data points are put in appropriate clusters. Contingency table is generated for the same. Maximized Bayes factor is used for fitness selection. Crossover operator is applied for data exchange between different chromosomes. Mutiation operator is applied to overcome local optimum. The optimum chromosome obtained is received as the optimum solution. Overview of our proposal is given in the remaining sections with the help of code snippets generated in Java under netbeans environment and the pseudocodes of our proposed two novel algorithms.

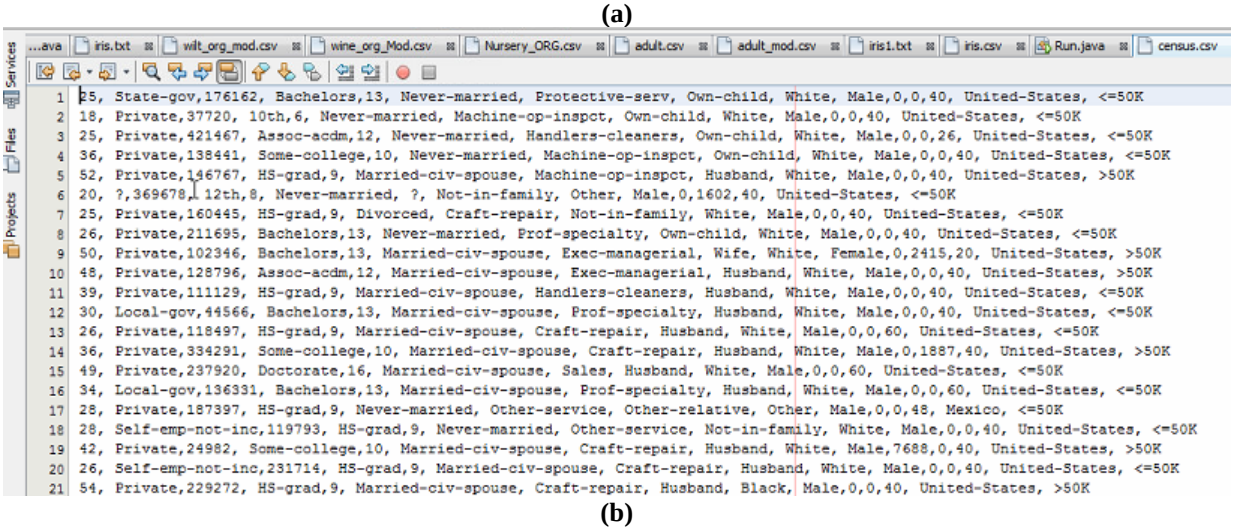
## II. NOVEL CONSTRAINT BASED ASSOCIATIVE ALGORITHM

It involves subset formation of multidimensional data and applying constraints for associative clustering for getting clustered output.

### *A. Discretization*

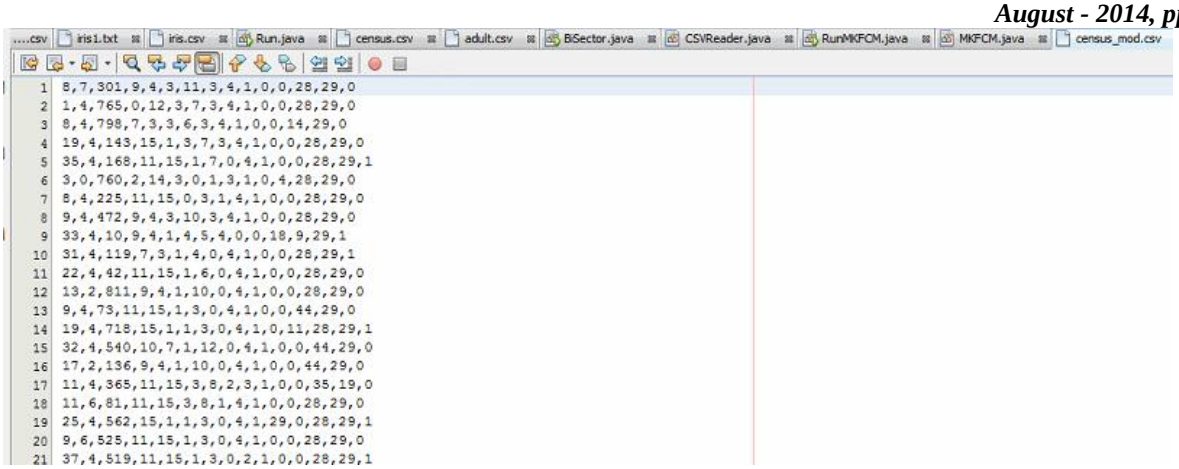



**(a)**




**(b)**




**(c)**

```
void createDiscretedInput() {

    discrted_input = new ArrayList<String>();

    for (String s : input) {
        String str[] = s.split(",");
        double[] v = new double[arr_size];
        for (int a = 0; a < arr_size; a++) {
            v[a] = Double.parseDouble(str[a]);

            for (int x = 0; x < div; x++) {

                if ((v[a] < ((x + 1) * range[a])) && (v[a] >= ((x) * range[a]))) {

                    v[a] = x;
```

(d)

Fig.1 Discretization (a. discretization b. census data c. discretized census data d. discretization process)

Discretization of the data is done as per the algorithm 1. Census data showing population information is represented in the form of numbers using discretization. The code snippet createDiscretedInput() is employed for the same.

### *B. Subset Formation*

```
void initializeSubsets() {
    subsets = new ArrayList[s_count];
    for (int a = 0; a < s_count; a++) {
        subsets[a] = new ArrayList<String>();
    }
}

void createSubsets() {
    ArrayList<String> dis_clone = new ArrayList<String>();
    ArrayList<Integer> added = new ArrayList<Integer>();
    dis_clone = (ArrayList<String>) discrted_input.clone();

    int cnt = 0;
    r = new Random();
    for (int a = 0; a < s_count; a++) {
        int x = 0;
```

Fig.2 Subset formation

Subset formation is achieved by putting l number of data points in m subsets.

## III. NOVEL GENETIC ALGORITHM

Genetic Algorithm is employed as Big data is under consideration [27]. In genetic algorithm, a chromosome or solution representation is used to explain each chromosome in the population. Each chromosome is made up of a sequence of genes from a particular alphabet. An alphabet can contain binary digits, floating-point numbers, integers, symbols (i.e., A, B, C, D), etc. therefore; binary value representation is exploited to explain the chromosome in this document. In this depiction, two task of associative clustering process is programmed by the chromosome. Particularly,

each chromosome is explained by a sequence of $M = 1 \times [n + k]$ binary digit numbers, where $n$ is the dimension of the data space. $k$ is the number of initial subsets.

In the proposed GS algorithm, an initial population of $p$ contain binary digits can be arbitrarily generated. The length of the particular chromosome $c$ is $L_c$, it contains two main functions (i) first $n$ value signifies the dimension whether need or not and (ii) $[n+1]$ to $[n+k]$ signifies that data points of subset have to go whether cluster 1 or cluster 2.

### *A. Fitness Computation*

Fitness computation involves subset formation, contingency table formation, bayes factor calculation and selection of maximum bayes factor values for fitting in appropriate cluster.

```
    System.out.println("=============================================
    double BF1 = findFitness(zero_cluster);
    double BF2 = findFitness(one_cluster);
.println("BF1="+BF1+"\tBF2:"+BF2);
    double BF = (BF1 < BF2) ? BF1 : BF2;
    cl_arr = new ArrayList[2];
    cl_arr[0] = (ArrayList<String>) zero_cluster.clone();
    cl_arr[1] = (ArrayList<String>) one_cluster.clone();
    double ac = findAccuracy(cl_arr.clone());

    info.add(new Cluster_Info(in, zero_cluster, one_cluster, BF, ac));
}
```

Fig. 3 Fitness Calculation

```
double calculateBayesFactor(ArrayList<ContingencyTable> ctable) {
    double n_d = 0.5;
    double n_x = 0.5;
    double n_y = 0.5;

    ArrayList<ContingencyTable> ct = ctable;
    double num_val = findNumValue(ct, n_d);

    double din_val = findDinValue(ct, n_x, n_y);
    //System.out.println(num_val + "\t" + din_val);
    double BF = (num_val / din_val);
    return BF;

}
```

Fig.4 Bayes Factor Calculation

```
    createDiscretedInput();
    initializeSubsets();
    createSubsets();
    init_population = new int[10][s_count + (int) arr_size];
    createInitialPopulation();
    int iterate=2;
    for(int x=0;x<iterate;x++)
    {
      applyCrossOver();
      applyMutation();
    }
      show();
    //applyClustering();
}
```

***B. Cross-over Operator***

```
void applyCrossOver()
{
     int cross_population[][] = new int[10][s_count + (int) arr_size];
      for (int x = 0; x < init_population.length; x++) {
        for (int y = 0; y < init_population[x].length; y++) {
           cross_population[x][y] = init_population[x][y];
        }
     }
      int pos=init_population[0].length/2;
       for (int x = 0; x < init_population.length; x++) {
           for(int y=0;y<pos;y++)
             cross_population[x][pos+y]=  init_population[x][y];
           for(int y=0;y<pos;y++)
               cross_population[x][y]=  init_population[x][pos+y];
       }
```

Fig. 5 Crossover Computation

The cross-over point is arbitrarily chosen and after that the two segments of parents are replaced to form offsprings. The mutation operator works on a single individual and forms an offspring by mutating that individual. On the basis of the fitness function and form the novel generation the recently generated individuals are assessed. The chromosome with the best fitness value is selected in every generation. The process ends after some number of generations either by the user or vigorously by the program itself, where the best chromosome acquired will be taken as the best solution. The best string of the last generation provides the solution to our clustering problem.

## IV. CLUSTERING ACCURACY

$$Clustering\ Accuracy(\Omega, C) = \frac{1}{N}\sum_{k}\max_{j}\left|\omega_k \cap c_j\right|. \qquad (1)$$

Clustering Accuracy is computed using (1). The accuracy is 75.7% which is more accurate than the state of the art.

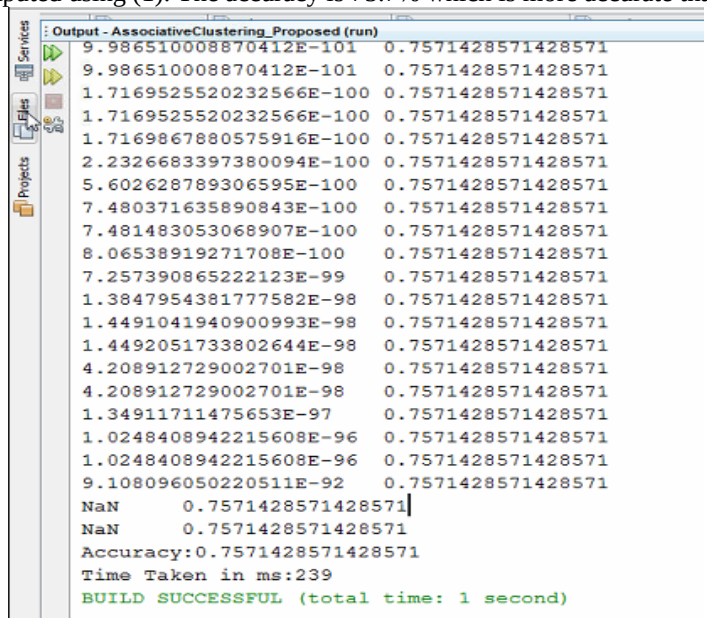



Fig.6 Accuracy of Proposed Algorithm

## V. PROPOSED OPTIMAL CONSTRAINT BASED ALGORITHM

**Input** : Multidimensional Data D
**Output** : Clustered Output
Intermediate processes: Subset formation and optimal Associative Constraint based data Clustering
**Parameters** :

$D = \{d_{ij}; 0 \le i \le m \text{ and } 0 \le j \le n\}$ → Dataset having n attributes

$D_d$ = Discretized data

$F\max$ →Maxumum value of every feature

$F\min$ → Minimum value of every feature

N → dimensions

P → mapping = $p:\quad R^N \rightarrow \bigcup_j R^{n_j}, \qquad n_j \le N$

m, n → subsets
i, j → subspaces
*I* → Interval
d→ Euclidean Distance
l→ first l no of shortest distances chosen for $m^{th}$ subset
p→ point in a high dimensional data space

$Dev(k) = \frac{Max(d_k) - Min(d_k)}{2}$ → Deviation for every k

**Start**
for all $d_{ij} \in D$
do
call *Discretization*
end

Discretization
Compute $F\max$ , $F\min$
Normalize as:

$$\left.\begin{array}{ll} 0, & input < 1(Dev(k)) \\ 1, & input < 2(Dev(k)) \\ 2, & input < 3(Dev(k)) \\ 4, & input < 4(Dev(k)) \end{array}\right\}$$

$D_d$

Call Subset_formation
End
Subset_formation
Select P in space randomly
for all p
compute d
select first l for m
repeat
End

## VI. PROPOSED GENETIC ALGORITHM FOR OPTIMAL ASSOCIATIVE CLUSTERING

n → dimension of data space
k→ no. of initial subsets
P →initial Population
C→ Chromosome
$L_c$ → length of chromosome
m, n → subsets
i, j → subspaces
*I* → Interval
d→ Euclidean Distance
l→ first l no of shortest distances chosen for $m^{th}$ subset
p→ point in a high dimensional data space

**Start**
for all C
do
call *Encoding*

call *fitness_Computation*
call *Genetic_Algorithm*

end

Compute *Encoding {*
dimension needed or not for first n
data points of the subset have to go in cluster 1 or cluster 2 }

Compute *fitness_Computation{*
Initial Data set selection
Subset formation
Contingency table formation
Bayes factor computation
Maximize bayes factor selection for fitness}

*Compute Genetic_Algorithm{*
Selection Operator
Crossover Operator
Mutation Operator }

## VII. CONCLUSION

We introduce a novel Genetic Algorithm for constraint based associative clustering. The performance was tested on UCI database using the evaluation index clustering accuracy. For Big data clustering our algorithm shows better accuracy (75.7%). The paper describes the entire implementation process.

## REFERENCES

[1] Osmar R. Z., "Introduction to Data Mining", In: Principles of Knowledge Discovery in Databases. CMPUT690, University of Alberta, Canada, 1999.
[2] Kantardzic, Mehmed. "Data Mining: Concepts, Models, Methods, and Algorithms", John Wiley and Sons, 2003.
[3] E. Wainright Martin, Carol V. Brown, Daniel W. DeHayes, Jeffrey A. Hoffer and William C. Perkins, "Managing information technology", Pearson Prentice-Hall 2005.
[4] Andrew Kusiak and Matthew Smith, "Data mining in design of products and production systems", in proceedings of Annual Reviews in control, vol. 31, no. 1, pp. 147- 156, 2007.
[5] Mahesh Motwani, J.L. Rana and R.C Jain, "Use of Domain Knowledge for Fast Mining of Association Rules", in Proceedings of the International Multi-Conference of Engineers and Computer Scientists, 2009.
[6] Souptik Datta Kanishka Bhaduri Chris Giannella Ran Wolff Hillol Kargupta "Distributed Data Mining in Peer-to-Peer Networks", Journal of internet computing, vol.10, no.4, pp.18-26. 2006.
[7] Ron Wehrens and Lutgarde M.C. Buydens, "Model-Based Clustering for Image Segmentation and Large Datasets via Sampling", Journal of Classification, Vol. 21, pp.231-253, 2004.
[8] W. Wang, J. Yang, R. Muntz, STING,"A Statistical Information Grid Approach to Spatial Data Mining", VLDB, 1997.
[9] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A multi-resolution clustering approach for very large spatial databases", VLDB, pp. 428-439, 1998.
[10] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.103-114, 1996.
[11] Ng R. T., Han J.: "Efficient and Effective Clustering Methods for Spatial Data Mining", Proceedings 20th International Conference on Very Large Data Bases, pp.144-155, 1994.
[12] Inderjit S. Dhillon and Dharmendra S. Modha, "A Data-Clustering Algorithm On Distributed Memory Multiprocessors", Proceedings of KDD Workshop High Performance Knowledge Discovery, pp. 245-260, 1999.
[13] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-bsed algorithm for discovering clusters in large spatial databases with noise", In SIGKDD, pp. 226–231, 1996.
[14] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining", In VLDB, 1994.
[15] T. Zhang, R. Ramakrishnan, and M. Livny. Birch, "An efficient data clustering method for very large databases", In SIGMOD, pp. 103–114, 1996.
[16] Jinchao Ji , Wei Pang, Chunguang Zhou, Xiao Han, Zhe Wang, "A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data", journal of Knowledge-Based Systems, vol. 30, pp. 129-135, 2012.
[17] Chen L, Chen CL, Lu M., "A multiple-kernel fuzzy C-means algorithm for image segmentation", IEEE Transaction on System Man Cybernetics: Part B, vol. 41, no. 5, pp. 1263-74, 2011.

[17] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin and Edward Y. Chang, "Parallel Spectral Clustering in Distributed Systems", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.33, No.3, pp. 568 – 586, 2011.

[18] Eshref Januzaj, Hans-Peter Kriegel and Martin Pfeifle, "Scalable Density-Based Distributed Clustering", Proceedings of 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 231-244, 2004.

[19] Josenildo Costa da Silva and Matthias Klusch, "Inference in Distributed Data Clustering", Engineering Applications of Artificial Intelligence, Vol.19, No.4, pp.363-369, 2005.

[20] Ruoming Jin, Anjan Goswami and Gagan Agrawal, "Fast and Exact Out-of-Core and Distributed K-Means Clustering", Journal of Knowledge and Information System, Vol. 10, No.1, pp. 17-40, 2006.

[21] Genlin Ji and Xiaohan Ling, "Ensemble Learning Based Distributed Clustering", Emerging Technology in Knowledge Discovery and Data Mining, Vol. 4819, pp 312-321, 2007.

[22] Olivier Beaumont, Nicolas Bonichon, Philippe Duchon, Lionel Eyraud-Dubois and Hubert Larcheveque, "A Distributed Algorithm for Resource Clustering in Large Scale Platforms", Principles of Distributed Systems, Vol.5401, pp.564-567, 2008.

[23] Samuel Kaski, Janne Nikkila¨ , Janne Sinkkonen, Leo Lahti, Juha E.A. Knuuttila, and Christophe Roos," Associative Clustering for Exploring Dependencies between Functional Genomics Data Sets", IEEE/ACM Transactions On Computational Biology And Bioinformatics, Vol. 2, No. 3,pp: 203-216, 2005.

[24] Yao Yuhui, Chen Lihui, Andrew Goh, Ankey Wong, " Clustering Gene Data Via Associative Clustering Neural Network", Proceedings of the 9th International Conference on Neural Information Processing, Vol.5, pp: 2228-2232, 2002.

[25] Hesam Izakian, Ajith Abraham, Vaclav Snasel, "Fuzzy Clustering Using Hybrid Fuzzy c-means and Fuzzy Particle Swarm Optimization", World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), India, IEEE Press, pp. 1690-1694, 2009.

[26] Swagatam Das, Ajith Abraham, Amit Konar, "Automatic Clustering Using an Improved Differential Evolution Algorithm", IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans, Vol. 38, No. 1, 2008.

[27] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, 1992.

[28] I.J. Good., "On the Application of Symmetric Dirichlet Distributions and Their Mixtures to Contingency Tables," Annals of Statistics, vol. 4, pp. 1159-1189, 1976.

[29] UCI Repository of Machine Learning databases, University of California, Irvine, Department of Information and Computer Science, http://www.ics.uci.ed/~mlearn/MLRepository.html

[30] B.Kranthikiran, M. Vinaybabu, "An Algorithm To Constraints Based Multi-Dimensional Data Clustering Aided With Associative Clustering", To Be Published