International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 12075-12082

A New Categorical Data Clustering Technique Based on Genetic Algorithm

Irani Hazarika[1], Anjana Kakoti Mahanta[2] and Dulumoni Das[3]

[1]Research Scholar & Assistant Professor (Contractual), [2]Professor, [3]Research Scholar

[1,2,3] Department of Computer Science, Gauhati University, GNB Nagar, Jalukbari, Guwahati, Assam, India.

[1]Orcid: 0000-0003-3468-946X

Abstract

This paper presents a categorical clustering algorithm based on genetic algorithm. Clusters are represented using summary vectors. Genetic algorithm is used to search for appropriate cluster summary vectors. Chromosomes are of variable lengths and encode the cluster summaries. This algorithm is applied on a number of real life data sets. Experimental results show that this method gives pure clusters on most of the datasets with no errors present. For the other data sets also very small amount of errors have been noticed.

Keywords: clustering; categorical data; genetic algorithm; cluster summary

INTRODUCTION

Clustering is a kind of unsupervised learning technique, used to partition large data sets into small homogeneous groups in such a way that cohesion and coupling are maximized and minimized respectively in these groups. One of the classical but most popular numeric data clustering method is K-Means [1] algorithm, where clusters are represented by means of the clusters. However in many real life databases other types of data are found. Here the attribute values are categorical in nature and have no natural ordering in the attribute domain. Due to this, algorithms like K-Means are prohibited from being used with categorical data clustering. Categorical data clustering algorithm K-Modes [2] replaces the means of the clusters with clusters modes and modes are updated through a frequency based method. A fast summary based categorical clustering algorithm has been proposed in [3]. In this agglomerative clustering algorithm, for each cluster instead of keeping all data points in the cluster a distribution summary vector i.e. summary of all points in the cluster is kept together with the number of points in the cluster i.e. its size. Other efficient categorical data clustering algorithms in literature are ROCK (Guha et al.1999) [4], QROCK (M Dutta et al. 2005) [5] etc. To measure similarity between a pair of data points ROCK algorithm uses concept of links instead of distances. In QROCK it is proved that if the input data points are considered as vertices of a graph then the final clusters obtained by ROCK algorithm are some connected components of the graph. Hence, in QROCK clusters are computed by determining connected components of the graph, which reduces the computing time of ROCK algorithm.

Genetic algorithm (GA) is a stochastic evolutionary algorithm guided by the principle of biological evolution and natural selection. Genetic algorithm was proposed by Holland. GA's have been applied for finding optimal solutions in many optimization problems. For optimization, GA uses the concept of survival of fittest and genetic operation such as selection, crossover and mutation. Genetic algorithm based clustering proposed in [6][13], encodes the chromosomes with the cluster center of the traditional K-Means algorithm for numeric data clustering. G-ANMI [7] is mutual information based genetic algorithms for categorical data clustering. G-ANMI uses average normalized mutual information (ANMI) as objective function of the genetic algorithm. In [8], the authors proposed an information theoretic approach (ALG-RAND) using genetic algorithm to categorical data clustering. Cluster ensemble has recently emerged as a powerful alternative to standard cluster analysis. Various cluster ensemble methods based on mutual information [9] [10], entropy [11] etc are proposed for clustering categorical data.

The aim of this paper is to develop a summary based categorical data clustering technique using genetic algorithm, which minimizes clustering error. Here, instead of cluster centre [6], each categorical cluster is represented by a cluster summary [3]. In traditional GA [6] [7] initial population is created by some randomly generated cluster centers. Unlike traditional GA method, in this paper, different runs of summary based clustering algorithm [3] is used on the data set, to find some initial sets of cluster summaries. These cluster summaries are used to create the initial population set in GA. The advantage of this initialization over random initialization is that, it increases the chances of obtaining a lower clustering error in the initial cluster sets than random initialization method. Thus, initial population holds some sets of locally optimize cluster summaries which are better than random initialization. Now, from these locally optimal cluster summaries, GA will try to find a globally optimal solution set. This globally optimal solution set holds the ultimate cluster summaries for the categorical data. This method increases the accuracy of data clustering, because good local optima always increase the chances of obtaining better global optima. One

advantage of this algorithm is that number of clusters (k) is not needed to specify. Experiments have been carried out on some real life datasets. Experimental results show that the proposed method produces higher accuracy than the previous works.

Rest of the paper is organized as- in section 2 a review of related works is given. In section 3 details of the summary based agglomerative clustering algorithm [3] and some definitions related to this paper are given. In section 4 our proposed genetic algorithm based clustering method for categorical data is elaborated. In section 5 experimental results of the proposed method are compared with the G-ANMI [7], ALG-RAND [8] methods. In section 6 conclusion of the work is given.

SUMMARY BASED CATEGORICAL DATA CLUSTERING ALGORITHM

Some terminologies that have been used in the summary based algorithm [3] have been given below and it is followed by the complete algorithm. First the categorical data set is converted to numeric data set. The categorical attributes of each input data are first converted to 0/1 attributes by considering one column for each distinct value that an attribute can have.

DISTRIBUTION SUMMARY OF A CLUSTER

Let x_i ($1 \leq i \leq n$) are data points in m dimensional space R^m. Let C be a cluster containing the n points x_1 , x_2 , x_3 ,……, x_n. The distribution summary d_sC in m dimensional space R^m of the cluster C is-

$$d_sC = \frac{\sum_{i=1}^{n} x_i}{n}$$

EXAMPLE1:

Let a categorical dataset with two attributes A and B. Here, A has two values a1, a2 and B has three values b1, b2, b3. Suppose a cluster C contains three data points x_1, x_2, x_3 with the following values for A and B.

Data	A	B
x_1	a1	b1
x_2	a1	b2
x_3	a2	b2

After converting the data set into 0/1 values, the values of x_1, x_2, x_3 with 5 dimensions (one dimension for each attribute value) will be as follows-

Data Point	a1	a2	b1	b2	b3
x_1	1	0	1	0	0
x_2	1	0	0	1	0
x_3	0	1	0	1	0

From these above data, distribution summary of cluster C (d_sC) can be computed. In computing the distribution summary of a cluster, the mean is taken for each component that corresponds to the value of some attribute in the original dataset, this value indicates the membership value (in the fuzzy set sense) of this attribute value in the corresponding cluster. Hence,

$$d_sC = \left[\frac{(x_1^1 + x_2^1 + x_3^1)}{3}, \frac{(x_1^2 + x_2^2 + x_3^2)}{3}, \ldots\ldots, \frac{(x_1^5 + x_2^5 + x_3^5)}{3}\right]$$

$$= \left[\frac{(1+1+0)}{3}, \frac{(0+0+1)}{3}, \ldots\ldots, \frac{(0+0+0)}{3}\right]$$

$$= [0.67, 0.33, 0.33, 0.67, 0]$$

If all the objects in a cluster have the same value for a particular attribute then the corresponding component in the summary vector with has value 1 and if none of objects have that attribute value then the corresponding value will be 0.

It is to be noted that the sum of the components for an attribute within a distribution summary is always 1.

In the above example 1st and 2nd dimensions contain the values for attribute A. Hence,

$$dsC^1 + dsC^2$$

$$= 0.67 + 0.33$$

$$= 1$$

Now the attribute values for B are in 3rd , 4th and 5th dimensions. Hence,

$$dsC^3 + dsC^4 + dsC^5$$

$$= 0.33 + 0.67 + 0$$

$$= 1$$

FUZZY SIMILARITY MEASURE

Let, d_sC^y denote the y^{th} dimension of distribution summary vector d_sC for any cluster C in m dimensional space R^m. Similarity between two clusters C_1 and C_2 is computed using fuzzy similarity, since the distribution summary vectors d_sC_1 and d_sC_2 can be treated as fuzzy sets.

The fuzzy similarity measure is defined as-

$$sim(C_1, C_2) = \frac{|dsC_1 \cap dsC_2|}{|dsC_1 U dsC_2|}$$

$$= \frac{\sum_{y=1}^{m} min\,(dsC_1^y, dsC_2^y)}{\sum_{y=1}^{m} max\,(dsC_1^y, dsC_2^y)}$$

This is the fuzzy set version of Jaccard's coefficient [14]. Here union, intersection and cardinality of a fuzzy set are as defined in the literature [3]

EXAMPLE 2:

Suppose cluster C_1 contains 2 data points x_1= [1, 0, 1], x_2= [0, 1, 1] and cluster C_2 contains 1 data point x_3= [1, 0, 1].Then,

d_sC_1= [0.5, 0.5, 1]

d_sC_2 = [1, 0, 1]

Now, fuzzy similarity between C_1 and C_2 is

$$sim\,(C_1, C_2) = \frac{(0.5+0+1)}{(1+0.5+1)}$$

$$= 0.6$$

MERGING OF TWO CLUSTERS

Let cluster C_1 and C_2 contain n1 and n2 points respectively in m dimensional space R^m. When similarity between C_1 and C_2 are more than a user defined threshold Θ, then the two clusters can be merged into a single cluster C_{new}. The distribution summary d_sC_{new} of the new cluster C_{new} in m dimensional space R^m is given by-

$$d_sC_{new} = \frac{(n1 * dsC_1 + n2 * dsC_2)}{n1+n2}$$

EXAMPLE3:

If we merge the two clusters C_1 and C_2 given in example 2, to a single cluster C_{new} then

$$d_sC_{new} = \left[\frac{(2*0.5+1*1)}{(2+1)}, \frac{(2*0.5+1*0)}{(2+1)}, \frac{(2*1+1*1)}{(2+1)}\right]$$

$$= [0.67, 0.33, 1]$$

Algorithm 3.1: The clustering algorithm is given below:

```
begin
  set SET=Φ
   input n,Θ
     for  i=1 to n  do
        begin
  input a data point d
            compute dsC, the distribution
            summary of the cluster C
  consisting of the data point d
            only
  while there is C1 Є SET  with
                    sim (C, C1)≥Θ
                  begin
  C2 =merge(C, C1)
   remove C1 from SET
  delete cluster C
   C‹-C2
            end
          add C to SET
      end
    end
```

Here, SET is the final set of cluster summaries given as output by the algorithm. This algorithm is an agglomerative clustering algorithm. The algorithm keeps a distribution summary vector i.e. summary of all points in a cluster and number of points in each cluster. Initially each data point is considered as a cluster and distribution summary of each cluster is the data point itself. After that depending upon a similarity value (Θ) similar clusters are merged together. This is continued till the clusters keep on merging with one another. Each time when two clusters are merged into a single one then the distribution summaries of the clusters are used to compute the distribution summary of the new cluster.

In our implementation of this paper, in merging phase, instead of one similarity threshold Θ, two similarity thresholds Θ1 and Θ2 are used. If similarity value between an input data point and a cluster is higher than an input threshold Θ1 then the data point is added to that cluster. If similarity value between two clusters is higher than an input threshold Θ2 then both the clusters are merged.

PROPOSED METHOD

An elaboration on the terminologies used by the genetic algorithm and description of some computational steps has been given below followed by the complete algorithm.

CHROMOSOME REPRESENTATION

Each chromosome string is encoded with distributed summaries of clusters as in [3]. If there are k cluster summaries and dimension of the data points is *m* then length of a chromosome T is m*k i.e chromosome T is represented as-

T= [$d_sC_1^1$, $d_sC_1^2$,…..,$d_sC_1^m$; $d_sC_2^1$, $d_sC_2^2$,….., $d_sC_2^m$; .....; $d_sC_k^1$, $d_sC_k^2$,….., $d_sC_k^m$]. Here, first m values represent cluster summary d_sC_1 for first cluster C_1 and so on for k clusters.

INITIAL POPULATION

For creating the initial population set (P_0), chromosomes are generated using summary based clustering algorithm 3.1. The output k summaries of the clusters (i.e SET) are encoded as a

new chromosome. If the size of the initial population is z then the algorithm is repeated z times to generate that many number of chromosomes. Each time the algorithm is applied to the data set, a different ordering of the data points is taken. Here chromosomes are variable length. This is because depending upon the ordering of the data set the number of clusters i.e. the value of k given as output by the algorithm 3.1 may vary.

The steps for generating initial population are as follows-

for t=1 to z

Generate SET_t by executing algorithm 3.1

Encode SET_t into t^{th} chromosome of P_0

end for

FITNESS COMPUTATION

The fitness function is used to assign a fitness value to each chromosome. For each t^{th} chromosome T_t in a population, fitness value F_t is computed using fitness function as follows-

Let x Є C_j is a data point in cluster C_j , $|C_j|$ denotes number of data points in cluster C_j and C_x denote the cluster consisting of data point x only.

for t=1 to z

k= number of cluster summaries in t^{th}

chromosome

$$F_t = \frac{\sum_{j=1}^{k} \sum_{\forall x \epsilon C_j} sim(C_j, C_x)}{k * |C_j|}$$

Assign F_t as fitness value for the t^{th} chromosome

end for

Here sim (C_j, C_x) is fuzzy similarity measure between C_j and C_x as defined in section 3. The maximization of fitness value F_t during evaluation leads to maximization of intra cluster similarity.

REASSIGNMENT OF DATA POINTS TO CLUSTERS

For each chromosome (T_t) in a population, each data point x in the dataset is reassigned to any one of the clusters present in T_t by using the similarity measure described above. If T_t has k clusters then x is assigned to any one of the clusters C_j Є T_t, j=1, 2…..k as follows-

Let, n denotes total number of data points in the dataset and z denotes size of current population.

for t=1 to z do

k= number of cluster summaries in t^{th}
Chromosome
for i=1 to n do
C_x= cluster containing data point x_i
for j=1 to k do
M_j =sim (C_x, C_j)
end for
M_p= max (M_j) , pЄj
Assign x_i to cluster C_p
end for
end for

RE-COMPUTATION OF CLUSTER SUMMARY

For all chromosomes in a population, cluster summaries are recomputed after reassigning data points to the clusters. For a cluster C with n data points the y^{th} component d_sC^y of the cluster summary d_sC is recomputed as

$$d_sC^y = \frac{\sum_{i=1}^{n} x_i^y}{n}$$

where x_i, i=1,..., n are the data points in C and x_i^y represents the y^{th} component of x_i

SELECTION

In the selection phase, chromosomes are selected from the current population set depending upon their fitness to generate the next generation population set. One popular selection technique is roulette wheel selection in which higher fitness chromosomes get a higher probability for selection. Each chromosome assigns a number of copies and chromosomes with higher fitness get more number of copies. Size of the population set remains unchanged. This population set goes for crossover and mutation operations.

CROSSOVER

In the crossover operation two parent chromosomes exchange information to produce two offspring's. Each pair of chromosome goes to crossover with a fixed probability p_c. For a variable length chromosome pair, crossover points are chosen such that after crossover total length of the each offspring chromosome remains same as the corresponding parent chromosome and the summation of the components for an attribute within a distribution summary remains 1.

Let, z denotes size of current population and k_t denotes number of cluster summaries in t^{th} chromosome of the current population. Now, the steps for crossover are as follows-

for t=1 to z-1 do
Generate a random number r Є[0, 1]
if r<p_c do

r_1=min(k_t, k_{t+1})
Generate a random point r_2 Є[0, r_1]
Exchange the substrings of t^{th} and $t+1^{th}$ chromosomes both inclusive r_1^{th} and r_2^{th} cluster summaries.
end if
t=t+1;
end for

This procedure will ensure that after crossover the modified chromosomes still represent cluster summaries.

MUTATION

After crossover mutation operation is done. Each chromosome goes for mutation with a probability p.

Let, m denotes total number of components in a cluster summary and z denotes size of current population. N_a denotes total number of attributes in the dataset. Now, the steps for mutation are as follows-

for i=1 to z do
Generate a random number rЄ[0, 1]
if r<p
k=number of cluster summaries in i^{th} chromosome
Generate a random number j Є [0, k]
Generate a random number r_a Є [0, N_a]
d_a= total number of possible values for r_a^{th} attribute
Generate random numbers y_1, y_2Є [0, d_a]
v_1= value for y_1^{th} component of r_a^{th} attribute in j^{th} cluster summary
v_2=value for y_2^{th} component of r_a^{th} attribute in j^{th} cluster summary
c= index of starting position for r_a^{th} attribute in j^{th} cluster summary
v= min{ v_1, 1-v_1, v_2, 1-v_2}
Replace (v_1 with $v_1 - v/2$ and v_2 with $v_2 + v/2$)

(This will ensure that $\sum_{y=1}^{d_a-1} d_s C_j^{c+y=1}$)

end if
end for

TERMINATION CONDITION

This algorithm is terminated after N^{th} generation, where N is taken as input. The best chromosome among all generations population set is selected as the solution chromosome for the clustering problem. Data are clustered according to the cluster summaries encoded in the solution chromosome.

The proposed algorithm is given below-

Suppose P_i denotes i^{th} generation population set, N is the total number of generations and z is the population size. Let T_t denote t^{th} chromosome in the current population and F_t denote fitness of T_t. Then P_0 is created as-

P_0= T_1, T_2, ……., T_z

Where, T_t is encoded with the SET_t i.e the t^{th} partition set from the initial partition sets of summary based clustering method.

1. i=0, B=Φ, P_i=Φ
2. input z, N
3. input P_0
4. for t=1 to z
   Compute F_t for T_t of P_0
5. for i=1 to N
i. Apply Selection operation on P_{i-1} to create P_i
ii. Apply Crossover operation on P_i
iii. Apply Mutation operation on P_i
iv. for t=1 to z
a. Reassign the data points to cluster summaries in T_t
b. Recompute the cluster summaries in T_t
c. Compute fitness F_t again
v. select a chromosome T_t with maximum F_t from P_i
vi. add T_t to set B
6. Select best chromosome T_{best} with maximum fitness value F_{best} from set B as final cluster summaries.
7. Cluster all the data points using T_{best}
8. Display the final set of clusters.

TIME COMPLEXITY OF THE ALGORITHM

Suppose there are n numbers of data points in the data set. Let the dimension of the data points be m and population size be z. The data can be clustered in at most n numbers of cluster summaries. The complexity of the algorithm 3.1 is $O(mn^2)$ [6]. To initialize the chromosomes in the initial population set algorithm 3.1 is executed z number of times and so it takes $O(zmn^2)$ time. Computing the fitness values of the initial z number of chromosomes require O(mnkz) time. Computing the similarity value between a data point and its corresponding cluster summary requires O(m) time. Select operation takes O(z), crossover operations takes O(zmn) and mutation operation requires O(zm) time. Reassigning data points to clusters requires computation of similarity values between each data point and the k cluster summaries and so it requires O(mnk) time. After reassignment, re-computation of the cluster summaries require O(mn) time. Computing the fitness of a chromosome requires O(mn) time after the re-computation of cluster summaries. For one iteration of the for

loop in statement 4, reassignment of data points to clusters require O(mnkz) time, re-computation of clusters require O(mnz) time, computation of fitness values of the chromosomes require O(mnz) time and hence the overall time requires is O (z + zmn + zm + mnkz + mnz + mnz) = O(mnkz) time. If N is the total number of generations then to compute the best fit chromosome it takes O (N) time. The overall time complexity of the proposed algorithm is therefore $O(zmn^2 + mnkz + Nmnkz)$. Now k = O(n) and if the number of iterations is kept fixed then the complexity is $O(zmn^2)$.

EVALUATION PROCEDURE

ACCURACY

If there are k number of clusters then the accuracy R of the clusters are measured as [9] –

$$R = \frac{\sum_{i=1}^{k} a_i}{n}$$

Where n is the number of data points in the classified dataset and a_i is the total number of data points from the dominating class in i^{th} cluster. In a cluster the dominating class is that class to which maximum data points in the cluster belong to.

ERROR

Error (E) that occurs in the clustering is measured as [9]-

E= 1-R

EXPERIMENTAL RESULTS

To test the effectiveness of the proposed algorithm, computer programs were developed using C++. Several arts of state datasets used in [9] such as Mushroom dataset, Congressional vote dataset, Zoo dataset obtained from UCI Machine Learning Repository has been used. The quality of clusters obtained has been calculated using average intra cluster similarity of the clusters and clustering error present in the clusters. In all cases it has been observed that the quality of clusters obtained by our proposed method is better than those reported in refs. [7][9]. Experiments and the results are included in this section. Description of the Real Life Datasets Used:

In our experiments, four real-life datasets are used. These are Mushroom dataset, Soybean dataset, Zoo dataset and Congressional Vote dataset. Datasets are taken from the UCI Machine Learning Repository.

MUSHROOM DATASET

This dataset contains the record of 8124 instances of mushroom. Number of attributes is 22. Each record is either from edible class or from poisonous class. Out of 8124 species 4208(51.8%) are edible and 3916(48.2%) are poisonous.

ZOO DATASET

In this dataset the total number of records (animals) is 101, number of attribute is 17 and total number of class is 7. The first attribute is animal name which is unique for each instance. Out of 17 attributes 15 attributes are of Boolean type and the remaining two are of numeric type. The distribution of number of instances in 7 classes are given below-

Table1: Class distribution of zoo data

Class	1	2	3	4	5	6	7
Instances	41	20	5	13	4	8	10

- *Congressional Vote Dataset*

This data set includes votes for each of the U.S. House of Representatives Congressmen in 1984. The dataset contains 435 numbers of instances with 17 attributes. The two class labels are democrat and republican. 267 instances are of class democrat and 168 instances are of class republican.

EXPERIMENTAL RESULTS ON THE DATASETS

The proposed method is applied on the real life datasets mentioned in section IV. Performance of the proposed method is compared with the results of K-Modes[2], G-ANMI[7], ALG-RAND[8], Squeezer[11], ccdByensemble[9], K-ANMI[10], TCSOM[12], which are obtained from paper G-ANMI[7], ccdByensemble[9]. This comparison is done in terms of clustering error as defined in section III.-

CREATION OF INITIAL PARTITIONS

The initial sets of partition (SETs) are created using summary based categorical clustering algorithm [6]. In this proposed method two input thresholds Θ1 and Θ2 are used as given in section III. The value of Θ2 is taken to be larger than that of Θ1. This is done so hoping that after the merging of two clusters the similarity values of the data points in the merged cluster from the new cluster representative will still remain greater than or equal to Θ1. The output of summary based categorical clustering algorithm is depended on these threshold values. The choosing of the appropriate thresholds leads to obtaining better local optima in the initial partition sets.

In table2, it is shown that how these input thresholds affect the clustering errors of the resultant cluster set. By means it can be said that in this global optimization problem, a better

global optima can be achieved by finding a good local optima in the initial partition sets.

CLUSTERING ERROR

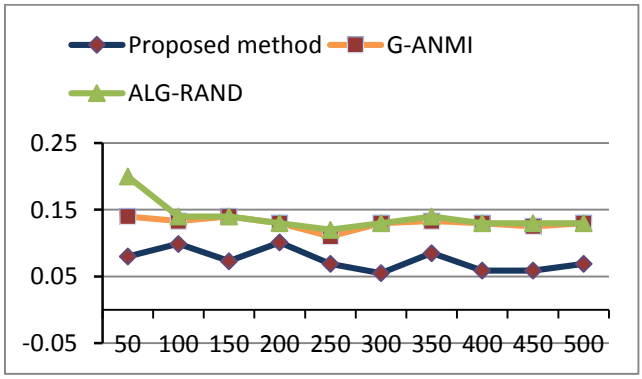
In fig2, the clustering errors corresponds to size of the populations are plotted for our proposed method, G-ANMI and ALG-RAND (as given in [7]). The population size is increased from 50 to 500. For a fair comparison same population size is considered as given in G-ANMI[7]. In all the experiments number of iteration is kept fixed as 10, crossover probability is 1.0 and mutation rate is 0.01. It is seen that in the mushroom and zoo datasets, our proposed method minimizes the clustering error to almost zero, which is better than G-ANMI and ALG-RAND. Also in case of vote dataset, proposed method gives better result than G-ANMI and ALG-RAND.

Table 2: Clustering error for different threshold values

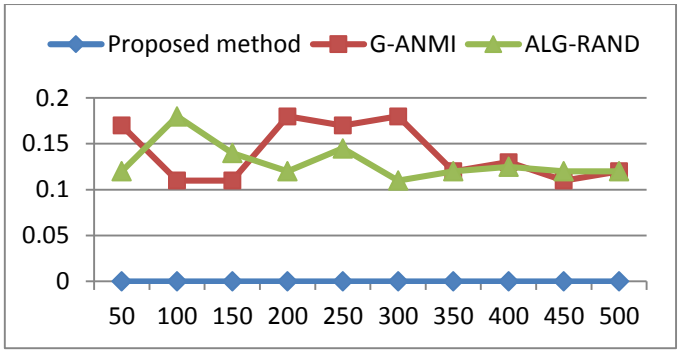
Dataset	**Input Threshold**		**Minimum clustering Error**
	Θ1	**Θ2**	
Vote	0.29	0.43	0.05
	0.2	0.35	0.112
Zoo	0.42	0.53	0
	0.25	0.40	0.04
Mushroom	0.50	0.59	0
	0.40	0.45	0.0002

Table 3: Threshold value used by our proposed method on different dataset in these experiments.

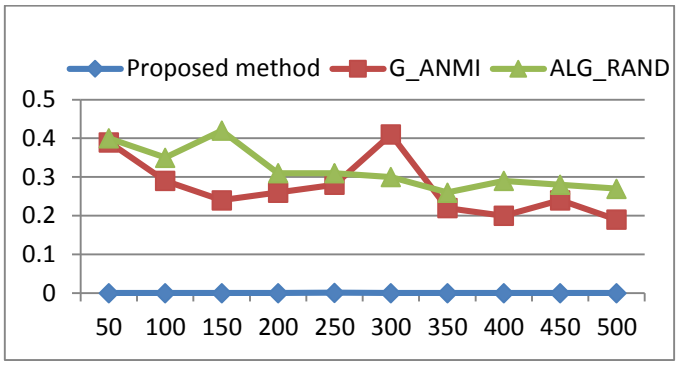
Dataset	**Input Threshold**		**Average clustering Error**
	Θ1	**Θ2**	
Vote	0.43	0.29	0.074
Zoo	0.53	0.42	0
Mushroom	0.59	0.50	0.00009



A) Vote data



B) Zoo data



C) Mushroom data

Figure 2: Clustering Error vs Population Size of our proposed method with GANMI and ALG-RAND(as given in [7])

Table 4: Performance of clustering error obtained from different clustering algorithms with our proposed method (as given in [9])

Method	**Clustering error**		
	Vote data	**Zoo data**	**Mushroom data**
Proposed	0.074	0	0.00009
Squeezer	0.163	0.0190	0.206
ccdByensemble	0.115	0.234	0.315
K-ANMI	0.131	0.110	0.289
K-Modes	0.141	0.171	0.262
TCSOM	0.131	0.127	0.254

In table4, performance of our proposed method is compared with other clustering algorithms given above, on the real life dataset vote, zoo and mushroom. Here, average clustering error is used for our proposed method (on threshold values give in table 3). It can be concluded from these comparisons that our proposed method minimizes clustering error than other algorithms.

CONCLUSION AND LINES FOR FUTURE WORK

In this paper we have proposed a genetic algorithm based clustering technique for categorical data clustering. The algorithm has been applied in a number of real life datasets. In most of the cases it gives pure clusters as shown in fig2. For other datasets that are not purely clustered, it gives very small amount of clustering error. In future, more experiments with large real life data will be carried out. Analysis could be done to find an actual relationship between Θ1 and Θ2. Extension of this method to handle multimedia data is a line of future work. Feature selection techniques can be incorporated to make the clustering even better. Also outlier detection methods can be included to detect the outliers in advance or during the clustering process small size clusters may be omitted treating the points as outliers.

REFERENCES

[1] A. K. Jain, R. C. Dubes, "Data clustering: A review", ACM Computing surveys,31, 1999.

[2] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values,"Data Mining Knowledge Discovery, vol. 2(3), 1998.

[3] M. Dutta, A. K Mahanta, "A Fast Summary Based Algorithm for Clustering Large Categorical Databases", Proceeding of ICWES 12, Ottawa, Canada,12, 2002.

[4] S Guha, R Rastogi, K Shim, "ROCK: A robust clustering algorithm for categorical attributes", Information systems,25(5), pp.345-366, 2000.

[5] M. Dutta, A K Mahanta, A. K. Pujari, "QROCK: A Quick Version of the ROCK Algorithm for Clustering of Categorical Data", Pattern Recognition Letters, 26(15), pp. 2364-2373, 2005.

[6] Maulik U., Bandyopadhyay S., 2000, "Genetic algorithm- based clustering technique", Pattern Recognition, 33, pp.1455-1465

[7] S. Deng, Z. He, X. Xu, "G-ANMI: A mutual information based genetic clustering algorithm for categorical data", Knowledge Based System, 23, pp.144-149, 2010.

[8] D. Cristofor, D. Simovici, "Finding median partitions using information theoritical based genetic algorithms", Journal of universal computer science, 8(2), pp.153-172, 2002.

[9] Z. He, X. Xu, S. Deng, "A cluster ensemble method for clustering categorical data", Information Fusion, 6, pp.143-151, 2005.

[10] Z. He, X. Xu, S. Deng, "k-ANMI : a mutual information based clustering algorithm for categorical data", Information Fusion 9 (2), pp.223–233, 2008.

[11] Z. He, X. Xu, S. Deng, "Squeezer: an efficient algorithm for clustering categorical data", Journal of Computer Science and Technology, 17 (5), pp.611–624, 2002.

[12] Z. He, X. Xu, S. Deng, "TCSOM: clustering transactions using self-organizing map", Neural Processing Letters, 22 (3), pp.249–262, 2005.

[13] S. Das, A. Abraham, "Automatic Clustering Using an Improved Differential Evolution Algorithm", IEEE Transactions on Systems, man, and Cybernetics, 38(1), pp. 218-236, 2008.

[14] R O. Duda, P. E. Hard, "Pattern Classification and scene analysis", A Wiley-Interscience Publication, New York, 1973.