Journal of Universal Computer Science, vol. 18, no. 4 (2012), 507-531
submitted: 29/9/11, accepted: 14/12/11, appeared: 28/2/12 © J.UCS

The Unification and Assessment of Multi-Objective Clustering Results of Categorical Datasets with H-Confidence Metric

Onur C. Sert, Kayhan Dursun, Tansel Özyer
(TOBB Economics and Technology University, Computer Engineering, Ankara, Turkey
{ocsert, kdursun, ozyer}@etu.edu.tr)

Jamal Jida
(Department of Informatics, Lebanese University, Tripoli, Lebanon
Jamal.Jida@ul.edu.lb)

Reda Alhajj
(University of Calgary, Computer Science, AB, Canada
Global University, Computer Engineering, Lebanon
alhajj@ucalgary.ca)

Abstract: Multi objective clustering is one focused area of multi objective optimization. Multi objective optimization attracted many researchers in several areas over a decade. Utilizing multi objective clustering mainly considers multiple objectives simultaneously and results with several natural clustering solutions. Obtained result set suggests different point of views for solving the clustering problem. This paper assumes all potential solutions belong to different experts and in overall; ensemble of solutions finally has been utilized for finding the final natural clustering. We have tested on categorical datasets and compared them against single objective clustering result in terms of purity and distance measure of k-modes clustering. Our clustering results have been assessed to find the most natural clustering. Our results get hold of existing classes decided by human experts.

Keywords: Multi-Objective Clustering, NSGA-II, h-confidence
Categories: I.5.1, I.5.3, I.5.4, I.5.5, J.4

1 Introduction

Clustering is partitioning data into well separated and compact groups. It is an unsupervised classification; the number of classes is not given a priori. It has been extensively studied in literature. e.g., k-means [Huang, 97; Huang, 98] is widely known clustering method for numerical data. K-modes clustering method is the variation of k-means for categorical data. It replaces the notion of means with modes according to frequency measurement of attributes [Huang, 97; Huang, 98]. ROCK clustering [Guha et al., 00] is a hierarchical approach for clustering categorical data. Initially, each cluster is assigned a separate cluster and clusters are merged with closeness metric. Algorithm applies a threshold and closeness metric calculates the sum of the number of links between pairs and those links depend on the number of neighbors in common for instances. Squeezer is another method that initially constructs stand-alone clusters with one instance and next coming instances are

determined according to threshold criteria specified by the user [He et al., 02]. LIMBO uses information bottleneck mechanism and stores all the clusters in tree structure and remaining data is gathered under the closest cluster[Periklis et al., 04]. CACTUS [Ganti et al., 99] clustering uses fast summarization techniques and performs clustering with two time scan of the dataset. COOLCAT[Barbara et al., 02] algorithm relies on entropy estimation and finds initial cluster centers that are most dissimilar and rest are assigned with the most similar one. STIRR[Gibson et al., 98] algorithm works in iterative fashion in dynamic environment. CLOPE [Yang et al., 02] algorithm uses histogram based width and height information for slope estimation. Another variation is using expected weighted coverage density estimation of histogram [Hua et al., 08].

All these clustering algorithms use one single objective and it tries to find the most natural clustering. Idea of employing multi-objective for clustering emerged recently. Several algorithms computing with single objective for clustering may need more than one objective needs for both modeling real world cases because, in real life, it is more likely to have more than one objective and conflicting objectives are of more sense to human decision making; and uncertainty in optimality plays an important role. More than one objective deduces alternative and potential solutions and leave the decision to human decision makers, multi-objective approach [Coello, 98] is much better because of optimizing all the objectives simultaneously in a partially ordered search space. During the optimization, conflicts are likely to occur as well as consensus.

Multi-objective evolutionary approaches have been adopted and studied. The simplest approach is to assign weight values for each objective and weighted sum of the objectives will give the final score. This approach can make sense however, assigned weight values are subjective and this makes the results biased with the weight values. Some others are non aggregating approaches not pareto-based (VEGA) [Schaffer, 98] and pareto based approaches such as MOGA [Zitzler et al., 00], NSGA [Srinivas and Deb, 95], NPGA[Horn et al., 94], and SPGA[Zitzler and Thiele, 99; Zitzler, 99].

Genetic algorithms use an evolutionary approach [Goldberg, 89]. A genetic algorithm uses a population as a set of potential solutions and each solution has a fitness value and the population evolves at every generation Genetic algorithms have been successfully used for the search and optimization problems.

Recently, Multi-objective genetic algorithms have been applied for clustering numerical data. Variations of it have been used for clustering data[Ozyer and Alhajj, 09; Ozyer and Alhajj, 08; Ozyer and Alhajj, 06a; Ozyer and Alhajj, 06b; Ozyer et al., 06]. NSGA [Srinivas and Deb, 95] has been used for finding dominating set as set of alternative solutions. In these algorithms, given number of clusters value k, multi-objective clustering solutions have been given and this value has been used for clustering the same data with k+1 clusters value. Algorithm worked in this manner and finally, a complete set of clustering results have been used for cluster validation. Results are interpreted with majority voting criteria according to relative metrics. Statistical analysis with sampling and divide and conquer approaches have been performed on results. This work mainly concerns on numerical data. Our contributions in this paper are:

- Our algorithm has been extended to work on categorical and mixed datasets. Categorical dataset related objectives and k-means objective have been used and clustering results have been compared.
- Different from iterative approach clustering with bit representation has been used. Clustering results have been obtained independent from number of clusters value.
- We assume a pool of co-occurrence results between pair of instances are drawn as the result of clustering. These results are produced as the natural solution of running different objectives at the same time and alternative clustering solutions are found by NSGA[Srinivas and Deb, 95] algorithm. A matrix of co-occurrence has been used further for both validation and merging clusters. We have used h-confidence information for clique finding that has been studied in[Xiong et al, 06].
- Clustering results have been validated by the metric proposed in[Chen and Liu, 03]

The outline of the paper is as follows: Section 1 is the introduction; section 2 includes the multi-objective clustering with genetic algorithm and validation part; section 3 has the experimental results with discussion. Section 4 includes the conclusion and future work part.

2 Multi-Objective Clustering With Genetic Algorithm

2.1 Overview

Categorical data clustering with multi-objective genetic algorithm works in the same way traditional genetic algorithm works. The algorithm has been given in Figure 1. According to our algorithm, number of chromosomes value is set to P, We first start with a predefined number of chromosomes. Number of chromosomes value is set to P with chromosome and mutation values. Our algorithm works as the following, Chromosomes have shuffle crossover and mutation operators followed by the k-mode operator. K-mode operator has been identified for quick convergence at each generation. K-mode operator simply reassigns the instances to the closest clusters in term of their frequencies. It has almost the same idea as the k-means but works for categorical data. For the mixed dataset both operators can be applied.

Thus, we have 2xP chromosomes and best P chromosomes are picked according to NSGA (Non-Dominated Sorting) algorithm. It basically works with domination rules. In general a multi-objective optimization problem can be described as[Zitzler, 99]:

According to formulation, optimization of a y vector is a vector of functions in order to show multi objective optimization. e(x) vector represents the constraints in order to determine the feasible solutions where x and y vectors represent the domain and range of the problem. NSGA algorithm has been used for pareto ranking of the alternative solutions. Let's assume for any two solutions, y_1 and y_2, y_1 dominates y_2 if all the objective scores of y_1 are better than y_2, otherwise y_1 is dominated by y_2 or incomparable with y_2. y_1 and y_2 are incomparable when neither y_1 nor y_2 has the situation that all objective scores are superior to one another. In this case, they take up the same pareto layer.

$$\text{Max} \setminus \text{Min} \quad y = f(x) = (f_1(x), f_2(x), \ldots, f_k(x))$$
$$e(x) = (e_1(x), e_2(x), \ldots, e_m(x)) \geq 0$$
$$x = (x_1, x_2, \ldots, x_n) \in X$$
$$y = (y_1, y_2, \ldots, y_k) \in Y$$

Our purpose is to get the weights of objectives as by product of our algorithm without any prior assumption. NSGA ranking gives results in layers. At each layer, incomparable results are located. Results at the next layer are inferior to results at the previous layers.

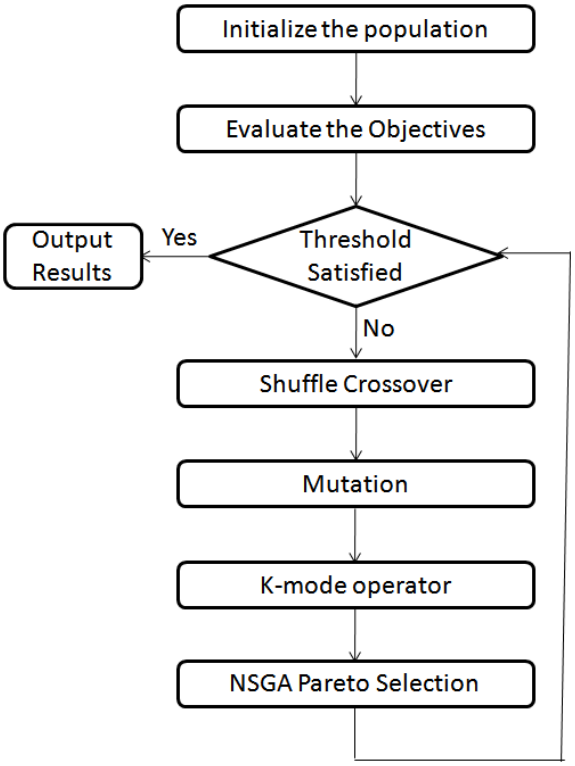

Figure 1: Flowchart of the Multi-objective Genetic algorithm for Clustering Categorical Data

Here, there are many approaches for determining the condition for termination. One choice would be to run for a determined number of generations. This would stop before convergence. Another choice would be to terminate when there is no further improvement for the optimization of objectives. Although either of them can be used, we have decided to use the second one to terminate the algorithm automatically when it converges.

After outputting the results of multi-objective genetic algorithm, we then acquire a matrix M of size NxN where N is the number of instances:

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \end{bmatrix}$$

Each x_i value gives the number of co-occurrences for instances i and j. Results of the multi-objective genetic algorithm are the overall summary of how many times pairs of instances are in the same cluster. We obtain the ensemble of clustering solutions that utilize alternative solutions without sacrificing any objective. We can figure this out as there are several experts and they performed clustering in their point of view and their solutions are gathered in matrix M containing co-occurrences. In other words, it is a complete graph having association strength between instances.

At this point, a final clustering based on experts' suggestions should to be done. This clustering will benefit from the scope of all solutions.

2.2 Description of the algorithm

2.2.1 Chromosome Encoding

We have arranged chromosomes in bit representation. Each bit has been arranged randomly. For the dataset D with n transactions, each chromosome includes n times k bits (Figure 2). Each reserved k bits give us the cluster number of the corresponding instance.

1	**0**	**0**	1	1	1	…	…	…	**1**	**0**	**1**

Figure 2: Chromosome Representation

In our system, number of clusters value's arranged automatically. We assume that at most, √n clusters may be formed and number of bits value is determined to cap this value. In our evolutionary approach, the number of clusters value will be subject to change during iterations. There is no fixed value assignment for the number of clusters value, Given D = x_1, x_2, ..., $x_{n,,}$ The value, √n is calculated, and log $\left|\sqrt{n}\right|$ bits is reserved. That is the value k.

2.2.2 Shuffle Crossover

Shuffle crossover for the bit representation works as follows: a new child chromosome is produced from two different chromosomes; a random crossover point is specified initially. Then, at each generation, chromosomes are combined around this point.

Shuffle Crossover includes a mixing phase before parent chromosomes are coupled which set apart this procedure from the classic crossover phase.

Each chromosome includes some number of bits as to specify cluster numbers of the transactions in the dataset, as stated before. During the mixing phase, for each chromosome, two bits are selected randomly, and they are switched. This mixing process is repeated with the number of total transactions in the dataset. Every switching process is kept in an array for to be used later.

After the mixing phase is completed, chromosomes are coupled and after each coupling two new chromosomes are generated.

When all new chromosomes are generated, a process called unshuffle is applied on all the resulting chromosomes. This is the reverse of shuffling (mixing) phase. The array that includes all the switching processes is used here. Then, bits that are specified for each chromosome to be switched are switched again lastly. With the following illustration, this task will be clearer.

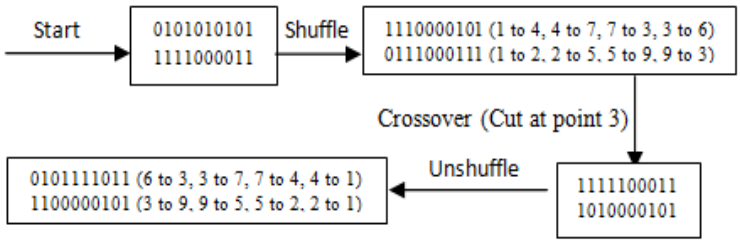

Figure 3: Shuffle Crossover

In classical crossover process, after some number of iterations there begins to be produced same generations with high probability. However, with the mixing phase of the Shuffle Crossover, the probability that new generations will be produced is increased remarkably.

The reason that number of clusters is specified with this way and cluster values are represented in binary base is to make the crossover part of the genetic algorithm to run more efficiently.

2.2.3 Objectives

For the cluster phase of transaction five different objectives is used. These

objectives are; K-Mode internal distance, K-Mode external distance, and EWCD. Assume a transactional data set D of size n has been defined as follows;

Given D = $\{x_1, x_{2\ ...xn}\}$ where x_i is the i'th transaction in dataset D. For each x_i there are m attributes and attribute set A= $\{A_1, A_2... A_m\}$ A_j is the j^{th} attribute of the dataset and A_j can take different values. Each attribute has a value set V_i and $V=V_1UV_2U..V_m$ and $V_i=\{ V_{i1}, V_{i2}, ..., V_{is} \}$. Here, V_{jk} is the k^{th} value of j^{th} attribute. The ratio $f(V_{ij})/|D|$ is the value that how many times V_{ij} occurs in dataset D. $f(V_{ij}/|C_i|)$ the value that how many times V_{ij} occurs in cluster C_i and $f(|C_i|)$ gives the total transaction number in cluster C_i .

a) ***K-Mode Internal Distance [Huang, 97; Huang, 97]:*** For each x_i and y_j pairs in D, the mode value of the attribute j is calculated.

If one of the A_j values of x_i is the same with the A_j of the mode of the x_i's cluster, then the distance value is not changed. However, if they are different the distance value is incremented by one.

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

$$\delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{if } x_j \neq y_j \end{cases}$$

For each attribute j of D, distance value d_j is obtained and the attributes sum is output as the total distance value for the dataset. The aim here is to minimize this total distance value. Since, when this value is minimized, transactions in D will get close to their cluster's center and will separate from other clusters.

b) ***K-Mode External Distance:*** This objective works in the same way k-mode internal works. Instead of calculating the distance for each x_i to the objects in cluster, distances are calculated between cluster pairs. Distance between modes of one cluster to another is calculated and the total distance is specified.

If the values of the two different modes are the same, the distance value is not changed; if they are different, it is incremented by one.

The aim here is to maximize the total distance value. Since, when this value is maximized, clusters are separated from each other properly and clustering will become more reliable.

c) ***EWCD [Hua et al., 08]:*** This objective uses an algorithm, which is designed for to satisfy partition-based clustering on transactional data. Related algorithm tries to fit as many frequent items as possible into clusters. It is usual that items in transactions can overlap between different clusters, so this algorithm takes into account all of these situations with the help of the related objective score. Topic will be clearer with the help of the following illustration.

A transactional dataset can be represented as being different shopping baskets in a supermarket. Suppose that following transaction sets (t_1, t_2, $t_3 - t_4$, t_5, t_6) are different shopping list of different customers for two supermarkets.

t_1 = {coke, milk} t_2 = {coke} t_3 = {milk, water}
t_4 = {coke, milk} t_5 = {milk} t_6 = {milk, water}

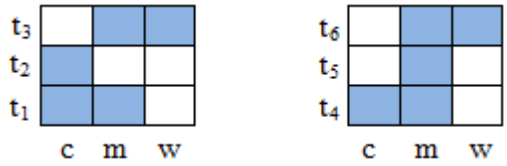

Figure 4: Two different clusters with EWCD.

From the illustrations, there are six different shopping lists and they include different items in different order. With the following WCD formula, it can be seen which clustering is better.

$$WCD(C_k) = \frac{\sum_{j=1}^{M_k} occur(I_{kj})^2}{S_k \times N_k}$$

Where C_k is cluster k, I_{kj} is j'th attribute in cluster k, S_k is the sum of occurrences of all attributes in cluster k, N_k is the number of transactions in cluster k.

Figure 4 stores the overall shopping lists in two clusters. According to Figure 4, in the first cluster (left one), coke and milk occur 2 times and water occurs 1 time. So S_k is 5 for this cluster. The summation part in the formula is 9. Also there are 3 lists. With this inferences score is 9 / 15.

In the second cluster (right one), coke and water occur 1 time and milk occurs 3 times in all of the shopping lists. So S_k is also 5 for this cluster. The summation part in the formula is 11 this time. Also again there are 3 lists. With this inferences score is 11 / 15.

This is an expected result, since for the first clustering there isn't any item that all the customers get in the same time, but for the second, milk is a common item for all of them. So, instead of clustering three distinct transactions without any common property, clustering with at least one is more appropriate.

This is a WCD – based clustering criterion. To evaluate and quantify a clustering result like $C^K = C_1, C_2, \ldots\ldots, C_K,$ EWCD objective function is used. Within the system this objective result is tried to be maximized by the genetic algorithm.

EWCD Initialization:

Input:

C: Cluster set ($C_1,C_2,\ldots,C_n$) for n clusters (this set contains just numbers from 1 to n as to represent cluster numbers , e.g. cluster 1 is represented with value 1 in this set)

S: Chromosome sequence which contains cluster numbers of each transaction in dataset (initially all are assigned to null cluster 0) (e.g. for a 2 cluster and a 3 transaction chromosome this may be a sequence like {1,1,2})

for each transaction t_i in S do
  take one cluster number c_j from C and assign that
  cluster to t_j that maximizes EWCD of S
end for

For genetic algorithm to run initially, we need to generate some chromosomes sequences in some way. In classical method, these sequences are generated randomly. However, if this method is used, objective functions' score over these kinds of chromosomes are far from to be high enough to provide expected results. For this reason, a method called EWCD initialization is used to provide higher results initially.

With this method, for all transactions in a chromosome, they are assigned to null cluster at first. Then, for the sequence, an iteration is applied over the chromosome. For each transaction t_i, a cluster number C_j is assigned to t_i. Then EWCD value is calculated for each assignment. For all C_j, which makes the EWCD maximum is assigned to transaction t_i as the cluster number.

To find the appropriate cluster number, we don't need to make EWCD calculation from the rough each time. Since, there is a EWCD score initially with the null cluster assignment. With this information, we only try one cluster and can see how much it increases or decreases EWCD estimation. Then which cluster makes the highest positive difference, it will be the cluster that is assigned to the concerned transaction. This method is called *WCD.deltaAdd()* and is illustrated below.

For a clustering result $C_K = C_1,C_2, \ldots ,C_K$ where $K < N$, *Expected Weighted Coverage Density (EWCD)* is calculated with the following formula.

$$EWCD(C^K) = \frac{1}{N} \sum_{k=1}^{K} \frac{\sum_{j=1}^{M_k} occur(I_{kj})^2}{S_k}$$

Where M_k is number of distinct attributes in cluster k; I_{kj} is j'th attribute in cluster k; S_k is the sum of occurrences of all attributes in cluster k.

2.3 Unification of Clusters with h-confidence

Multi-objective clustering results give us population number of solutions. Assume that a dataset which includes n transactions is defined as; D = x_1, x_2, ..., x_n ; and $num(x_1, x_2, \dots, x_n)$ is defined to be the amount that how many times transactions x_1 to x_n are clustered together.

Then h-confidence value can be calculated with the following formula[Xiong et al., 06];

$$h(x_1, x_2, \dots, x_n) = \frac{num(x_1, x_2, \dots, x_n)}{\max\left(num(x_1), num(x_2), \dots num(x_n)\right)}$$

We have used h-confidence information to combine the cliques in bottom up fashion until entire dataset belongs to one cluster.

For the large-scale data it may take too much time since the level of cluster tree will increase and genetic algorithm can be used again as the basis until a pre-specified threshold and then obtained cliques can be used for merging.

The main logic with this part is setting up a tree from up to bottom, and interpret results while each level of this tree representing specific cluster numbers.

3 Cluster Validation

After finding results for different number of clusters, we try to predict the real number of clusters among based on our results. For this prediction we used IEE Delta Square metric [Chen and Liu, 03].

This metric is based on the difference of the expected entropy among different results. We expected the peek points of IEE Delta Square to give the best candidates for true number of clusters of the used datasets.

Let A be a set of attributes which includes different attributes and these attributes have different values. We can calculate the entropy as;

$$E(\tau) = \frac{1}{d}\sum_{i=1}^{d} E(A_i)$$

$$E(A_i) = -\frac{1}{\log_2 D_i}\sum_{a_{ij} \in A_i} P(a_{ij}) \log_2 P(a_{ij})$$

We can calculate expected entropy for datasets which contains N elements and partitioning clusters;

$$EE(C^k) = \sum_{i=1}^{k} \frac{N_i}{N} E(C_i) = \frac{1}{N}\sum_{i=1}^{k} N_i E(C_i)$$

Then we can calculate the increasing rate of expected-entropy *(IEE)*, the differential order of IEE curve *(*ΔIEE*)* and the 2nd differential order of IEE curve $\Delta^2 IEE$.

$$IEE(k) = EE(C^k) - EE(C^{k+1}) \quad \text{and} \quad \Delta IEE(k) = IEE(k) - IEE(k+1)$$

$$\text{so,} \quad \Delta^2 IEE(k) = \Delta IEE(k-1) - \Delta IEE(k)$$

4 Experiments and Discussion

4.1 Description of Datasets

We used 7 different datasets. These datasets are obtained from UCI data repository. Details of the datasets are shown at Table 1.

Name of the Dataset	Number of Clusters	Number of Elements
Zoo	7	101
Soybean (Small)	4	47
Hayes – Roth	3	160
Heart (Categorical)	5	303
Congressional Voting	2	435
Tic – Tac – Toe Endgame	2	958
Balance Scale	3	625

Table 1: Details of the Datasets

4.2 Comparison of our results to k-modes with respect to k-modes distance formulation

We ran both k – modes algorithm and our multi objective algorithm for these 7 datasets and calculate the purity(pur.) value of the results for different number of clusters with k-modes and multi-objective clustering by using objective pairs a,b; a,c ; and b,c (Section 2.2.3). The results are shown in the tables below:

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
7	0,71	316	0,78	215	0,68	256	0,90	154
8	0,72	302	0,82	203	0,76	188	0,92	140
9	0,76	300	0,84	178	0,80	186	0,93	132
10	0,79	258	0,88	149	0,85	147	0,96	126
11	0,80	268	0,88	136	0,88	130	0,97	123
12	0,83	246	0,92	120	0,92	112	0,97	121
13	0,84	232	0,94	111	0,95	97	0,98	118
14	0,85	204	0,95	103	0,96	89	0,98	116
15	0,86	207	0,96	95	0,97	83	0,98	105
16	0,87	209	0,97	91	0,97	83	0,98	102

Table 2: Purity Measure and Total Distance for Zoo Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
4	0,75	284	1	199	1	199	0,98	205
5	0,83	249	1	176	1	176	0,98	191
6	0,81	273	1	165	1	164	0,98	176
7	0,85	269	1	160	1	149	1,00	161
8	0,94	230	1	148	1	133	1,00	153

Table 3: Purity Measure and Total Distance for Soybean (Small) Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
3	0,43	210	0,46	248	0,41	246	0,46	261
4	0,43	190	0,46	236	0,45	233	0,46	218
5	0,42	191	0,46	206	0,45	219	0,46	200
6	0,46	171	0,49	204	0,49	210	0,48	185
7	0,47	166	0,49	178	0,49	186	0,49	162
8	0,48	171	0,53	165	0,49	163	0,55	150
9	0,49	142	0,58	151	0,53	147	0,55	133
10	0,48	136	0,59	143	0,55	138	0,55	131

Table 4: Purity Measure and Total Distance for Hayes – Roth Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
5	0,54	581	0,58	566	0,57	606	0,59	564
6	0,54	591	0,59	559	0,57	601	0,59	542
7	0,56	567	0,59	529	0,57	572	0,59	539
8	0,54	542	0,59	508	0,60	532	0,59	538

Table 5: Purity Measure and Total Distance for Heart (Categorical) Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
2	0,61	2581	0,67	2425	0,61	2684	0,70	1787
3	0,69	2476	0,85	1698	0,68	2510	0,82	1568
4	0,78	2232	0,85	1588	0,68	2402	0,82	1501
5	0,78	2278	0,85	1538	0,69	2386	0,84	1373
6	0,81	2119	0,85	1483	0,70	2308	0,84	1311
7	0,82	2027	0,85	1459	0,71	2275	0,84	1303
8	0,81	2082	0,85	1411	0,77	2227	0,84	1261

Table 6: Purity Measure and Total Distance for 1984 United States Congressional Voting Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
2	0,65	4980	0,65	4847	0,68	4553	0,65	4689
3	0,65	4650	0,65	4712	0,68	4550	0,65	4675
4	0,65	4406	0,66	4581	0,68	4478	0,66	4532
5	0,65	4249	0,66	4413	0,68	4473	0,66	4409
6	0,65	4094	0,66	4177	0,69	4165	0,66	4387
7	0,65	3949	0,69	4082	0,69	4050	0,68	4107
8	0,66	3905	0,69	3970	0,69	3992	0,68	3985

Table 7: Purity Measure and Total Distance for Tic – Tac – Toe Endgame Dataset

#of clust.	K–Modes		Obj. a and b		Obj. a and c		Obj. b and c	
	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.	pur.	obj. a.
3	0,48	1542	0,55	1745	0,57	1790	0,51	1755
4	0,51	1388	0,55	1664	0,59	1728	0,51	1669
5	0,52	1330	0,56	1622	0,59	1673	0,54	1635
6	0,54	1302	0,56	1608	0,62	1589	0,54	1570
7	0,53	1243	0,56	1575	0,63	1553	0,55	1485
8	0,55	1238	0,56	1495	0,63	1466	0,55	1421

Table 8: Purity Measure and Total Distance for Balance Scale Dataset

As can be seen from Tables 2 to 8, for all the fitness function pairs' results (Internal Distance – External Distance, Internal Distance – EWCD, External Distance - EWCD) of the multi objective algorithm that has been implemented outweigh the results of the k-modes algorithm. Also in every step, k-modes distance is decreasing due to the fact of a better clustering. However, sometimes purity results can be bigger than k-modes distance. This is because of the fact that, the implemented algorithm tries to improve more than one objective's results at the same time instead of just one objective.

4.2.1 Validation of clustering results by using Δ^1IEE

We examined this metric for Zoo, Hayes – Roth and Soybean (Small) datasets. Results are showed at from Figure 5 to Figure 7. In these figures y-axis indicates the value of Δ^2IEE metric;

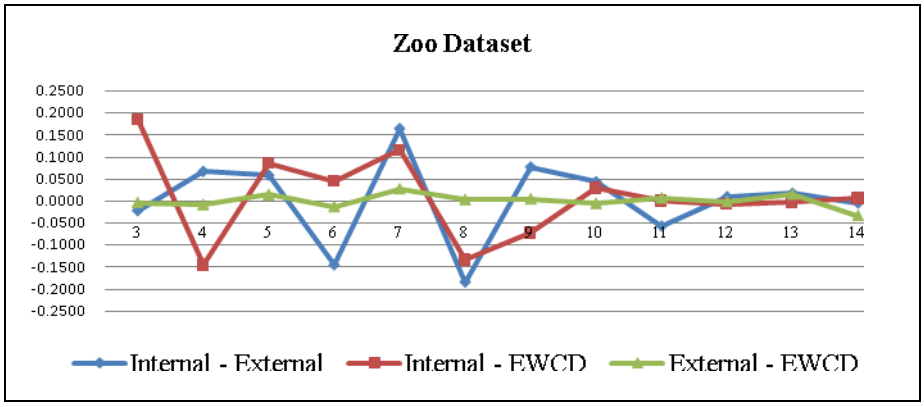

Figure 5: Number of Clusters Prediction of Zoo Dataset

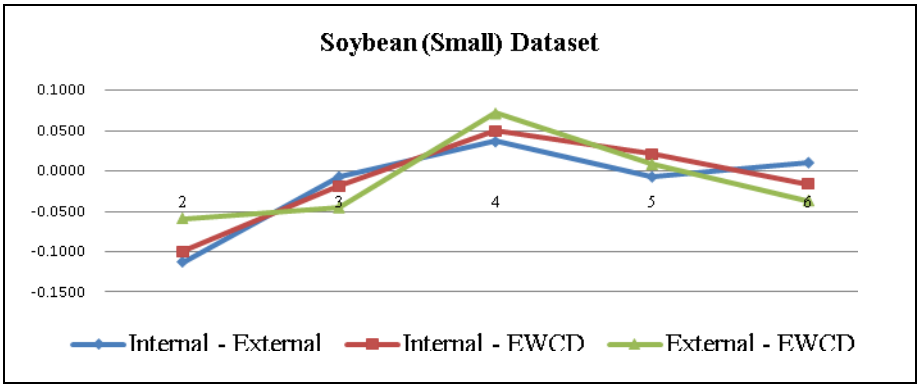

Figure 6: Number of Clusters Prediction of Soybean (Small) Dataset

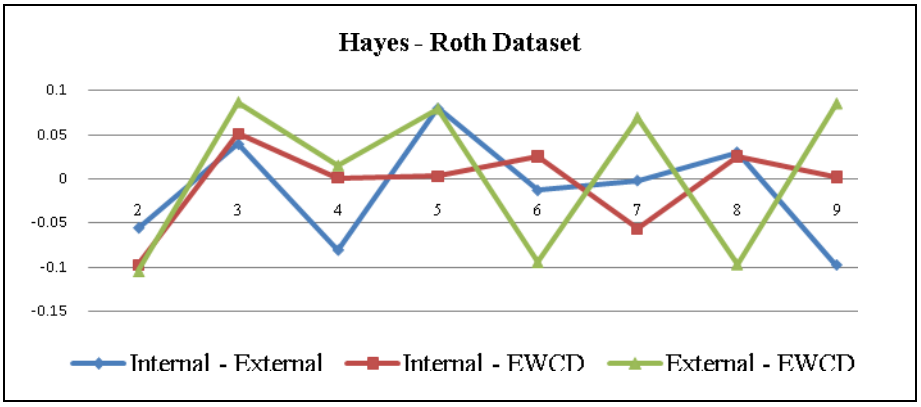

Figure 7: Number of Clusters Prediction of Hayes – Roth Dataset

In the figures from 5 to 7, the peek points of each line indicate us the proper number of best clustering numbers for the data sets. For the Zoo data set, for 7 clusters, $\Delta^2 IEE$ reaches its maximum value, and 5 and 9 clusters present its second and third maximum values. From these results, we can conclude that to cluster the Zoo data set, 7,5 and 9 clusters will be a logical choice. As a support for our conclusion, from the original data set it can be seen that 7 clusters is the correct clustering number. In the same manner, $\Delta^2 IEE$ reaches its maximum with the 4 cluster numbers for the Soybean (small) dataset, which is also the same number for its original cluster number. Lastly, ideal clustering numbers for Hayes-Roth dataset is found to be 5 and then 3. The original dataset is clustered into 3 clusters. All these results indicate the accurateness of our approach.

We have also examined results for the data sets hearth-categorical, congressional voting, balance scale and tic-tac-toe. However, because of the small numbers of their original clustering, there are some discrepancies with the $\Delta^2 IEE$ metric, these are presented in the figures from 8 to 11 and described below.

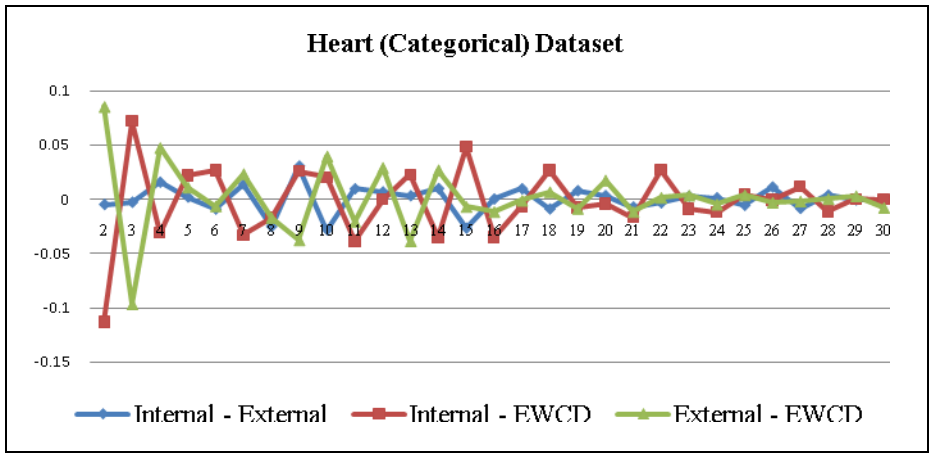

Figure 8: Number of Clusters Prediction of Heart (Categorical) Dataset

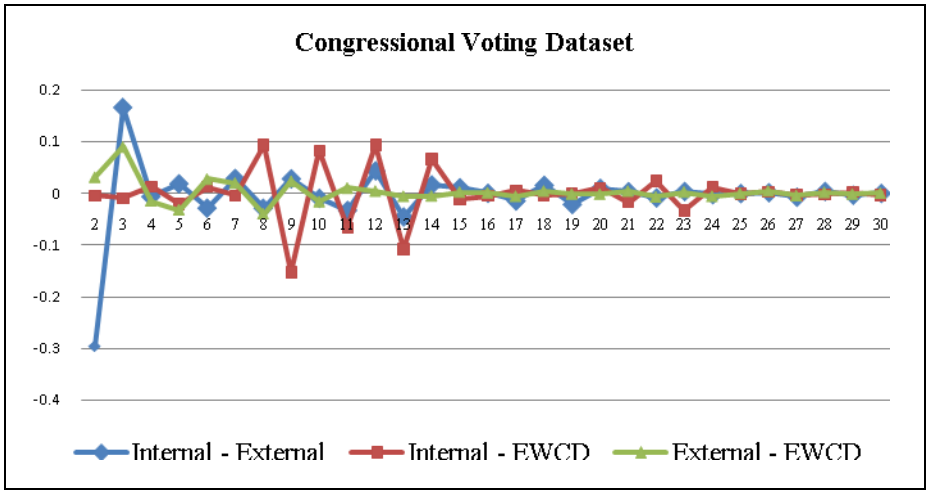

Figure 9: Number of Clusters Prediction of Congressional Voting Dataset

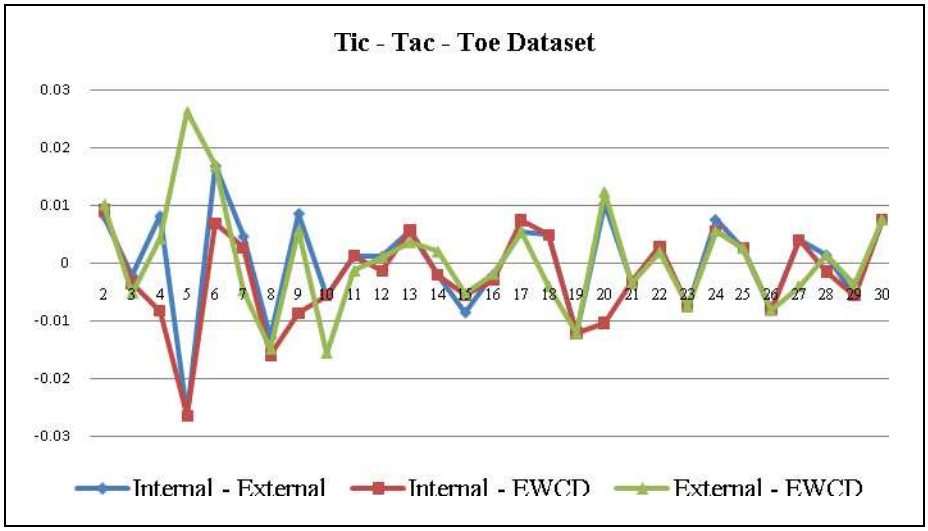

Figure 10: Number of Clusters Prediction of Tic-Tac-Toe Dataset

While calculating the $\Delta^2 JEE$ value, due to the results of the clustering processes, previous, current and next cluster values are being used. Because of this, $\Delta^2 JEE$ values of the first two and last two clusters are taking different values from expected. Because of this characteristic of the $\Delta^2 JEE$ metric, $\Delta^2 JEE$ value for the data sets that have originally clustered into two clusters, is not giving accurate estimates. Figure 8 – 11 gives examples of datasets.

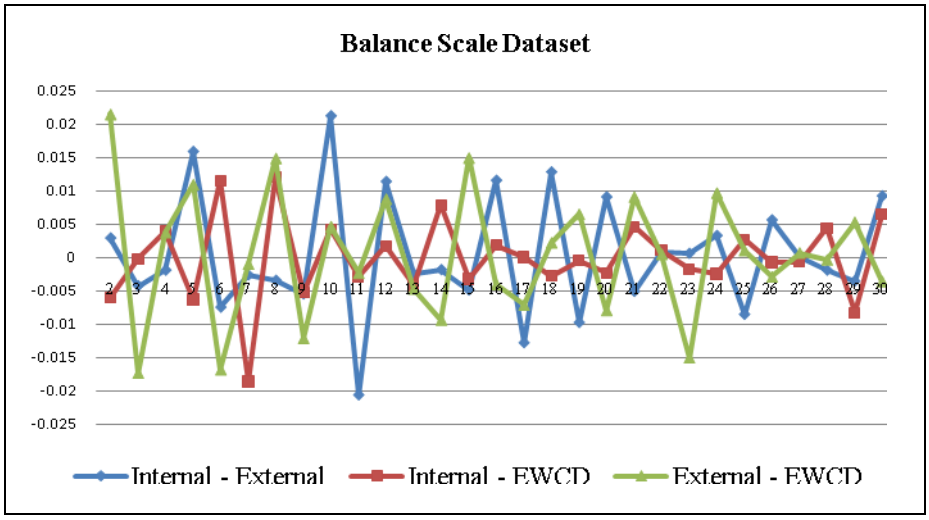

Figure 11: Number of Clusters Prediction of Balance Scale Dataset

4.2.2 Quick Convergence and Purity

Also we have taken experiments on computer with i5 processor, 4MB RAM configuration and application has been implemented with python programming language. We have used four different data sets from UCI machine learning repository[1]. They are zoo, hayes, soybean and credit card datasets. Number of chromosomes value is assigned to 100 and p_c=0.9 and p_m=0.05. We have applied k-modes operator for quick convergence at each iteration and termination criteria has been set to 100 iterations. Another option would be to stop when the best results repeat at the next generation. At the experiments, we have tried different objective pairs for these datasets such as (K-modes internal, EWCD); (K-modes internal and K-modes external; (K-modes external and EWCD) pairs and analyze purity scores; illustrate how the result of the objective scores quickly converge for only zoo dataset because all experiments converged before 18 iterations at maximum(Figure 12). It can be observed that results converge quickly with k-mode operator.

Figures 13, 14 and 15 give the corresponding purity scores of the zoo dataset and purity of the data. We have taken the same experiments for k-modes algorithm and we found the purity score 0.612 for 7 clusters. All the results got better score than k-modes and k-modes internal with k-modes exetrnal pair was slightly better.

[1] http://archive.ics.uci.edu/ml/

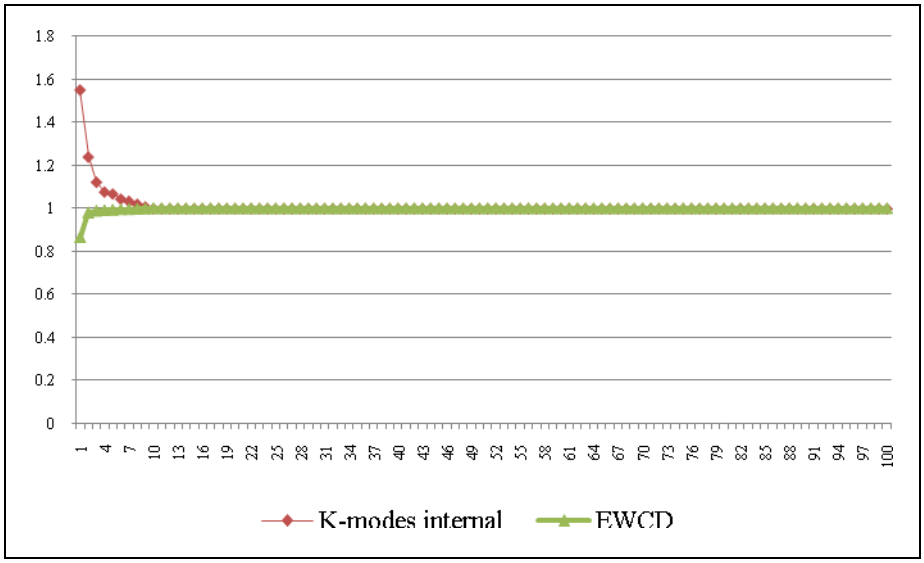

Figure 12: Convergence of K-modes internal and EWCD for zoo

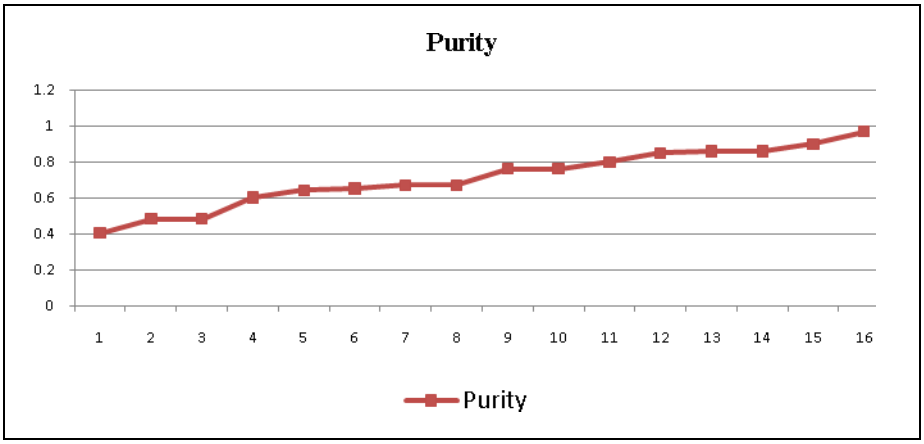

Figure 13: Purity results of K-modes internal and EWCD for zoo

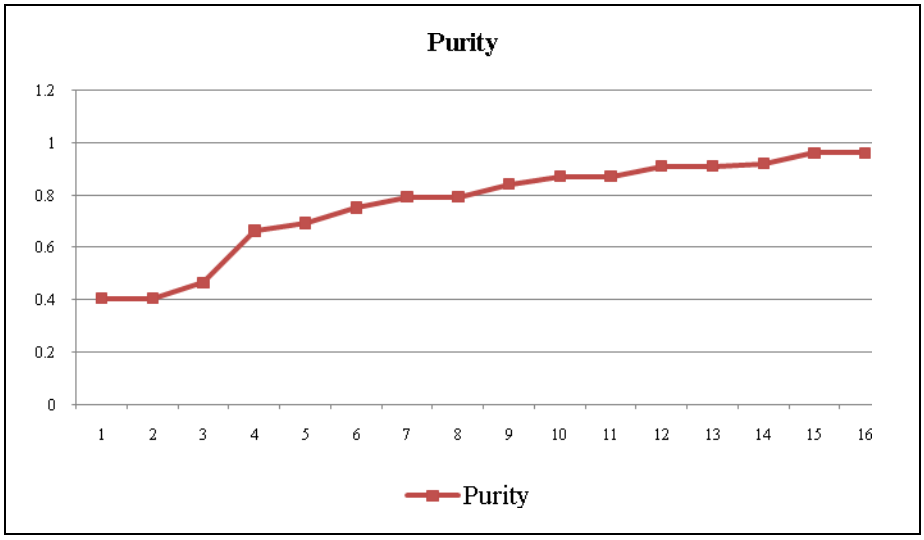

Figure 14: Purity results of K-modes internal and K-modes external for zoo

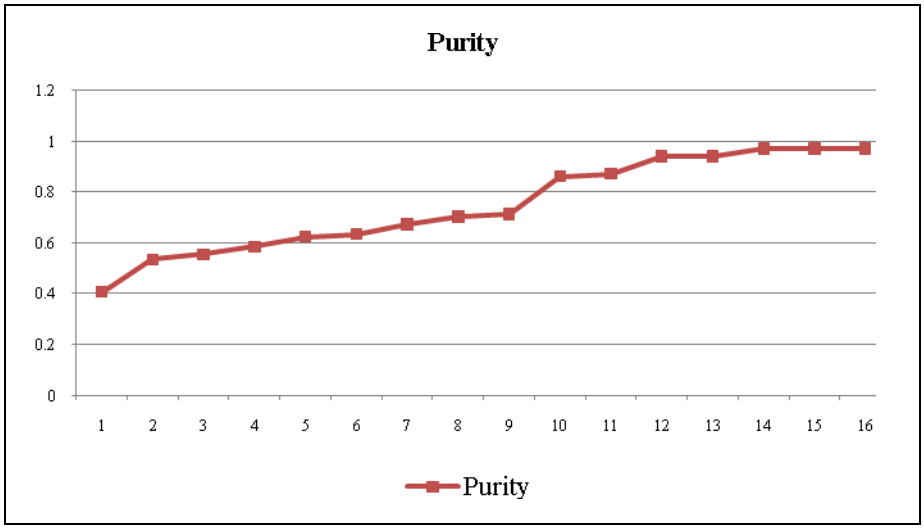

Figure 15: Purity results of K-modes external and EWCD for zoo

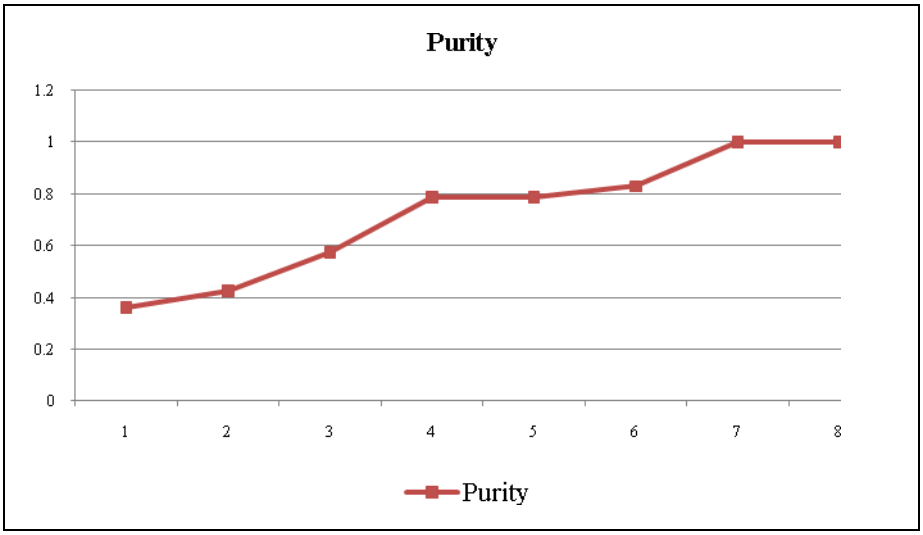

Figure 16: Purity results of K-modes internal and EWCD for soybean

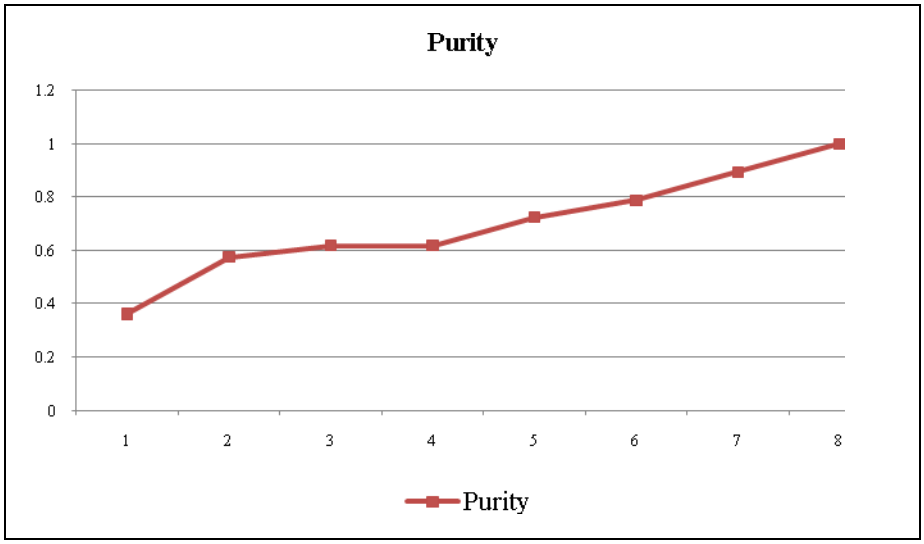

Figure 17: Purity results of K-modes internal and K-modes external for soybean

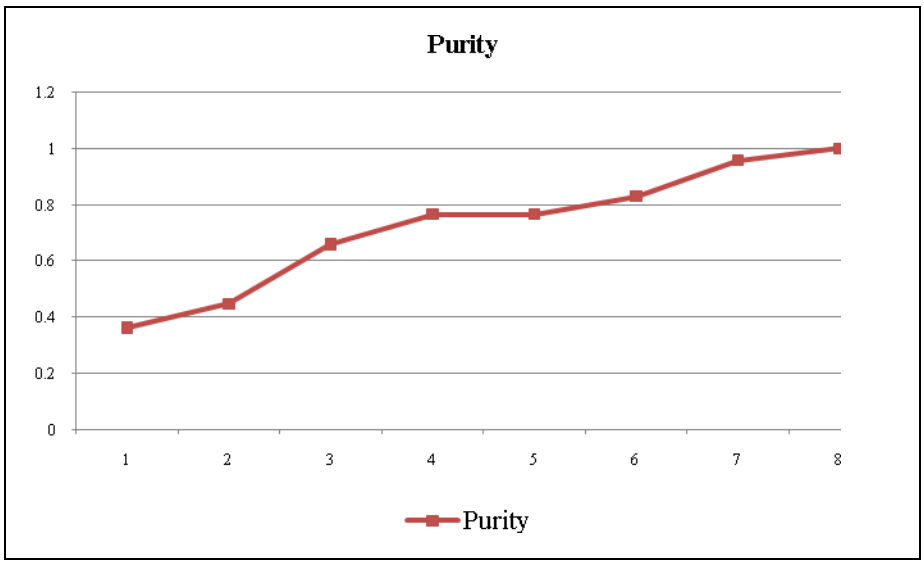

Figure 18: Purity results of K-modes external and EWCD for soybean

Soybean dataset has 307 instances with 35 categorical attributes. Soybean datase has 19 classes and we managed to find the purity value as one in Figure 18 and very close to 1 for Figures 16 and 17 in 8 clusters. We have obtained 0.88 for the same number of clusters value

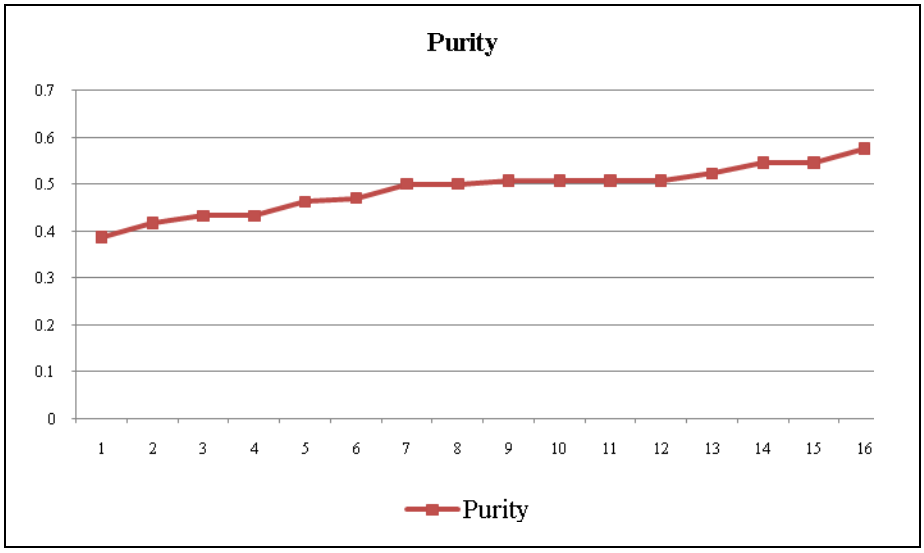

Figure 19: Purity results of K-modes internal and EWCD for hayes

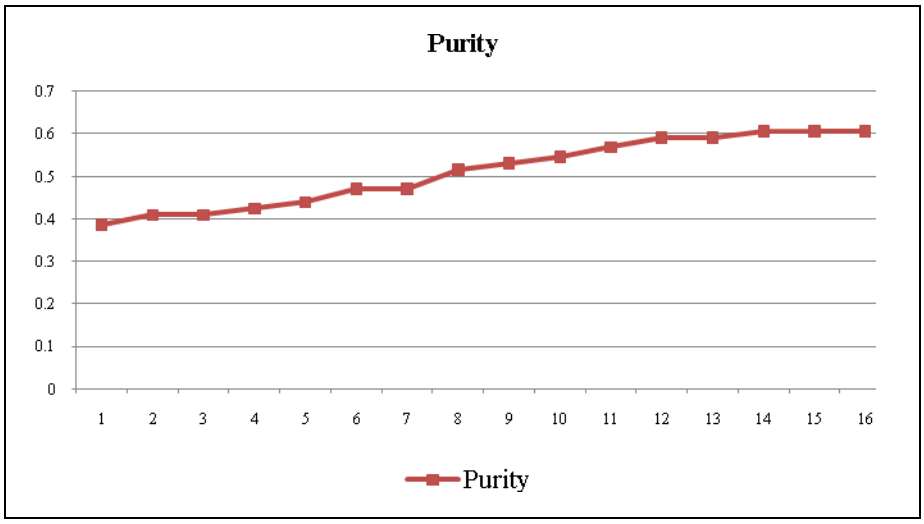

Figure 20: Purity results of K-modes internal and K-modes external for hayes

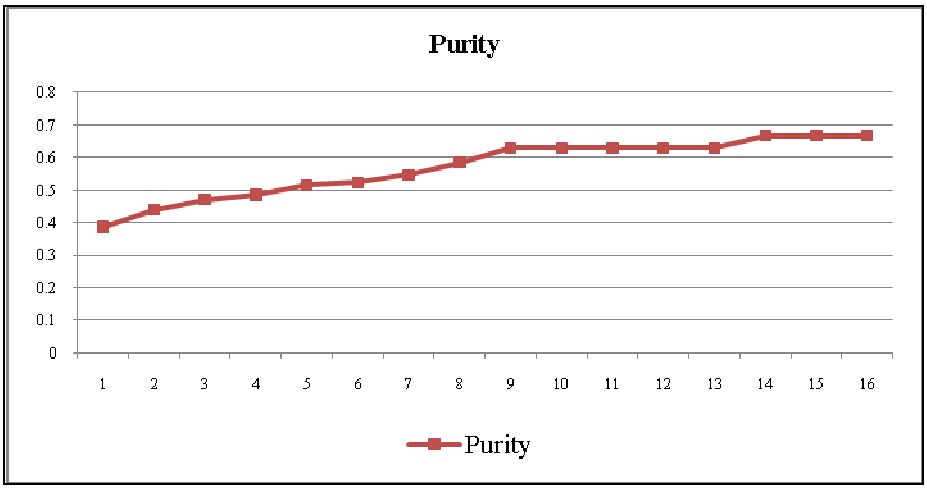

Figure 21: Purity results of K-modes external and EWCD for hayes

Hayes dataset contains 160 instances with five attributes and dataset originally contains three classes. It has been tested and purity results for three classes are .43, .40 and .47 in Figures 19, 20 and 21 respectively whereas purity result for k-modes for three clusters has been found as .38.

All these purity results are examined for the chosen three data sets (zoo, soybean, hayes) and convergence test has been taken only for zoo dataset to demonstrate the efficiency and effectiveness of our algorithm.

5 Conclusions And Future Work

Multi-objective genetic algorithm has been implemented before for the large variety of datasets having numeric values. We have extended further on categorical datasets but non-iteratively. We believe that clustering can have multiple objectives according to the clustering objective perspective and different results may be obtained depending on what perspective you apply it. Final clustering solutions exhibit suggestions for how the data can be clustered. Next, all suggestions are taken as multi-expert view of solutions and we have utilized clustering results to come up with a final clustering result. Combining all results leads to establish a network of instances with bonds and we have used hierarchical clustering in bottom up direction. It helped us find the purity results. We have applied the clustering validation in order to find the most natural clustering result for the categorical datasets.

Acknowledgements

This paper is part of the project sponsored by Scientific and Technical Research Council of Turkey (Tübitak EEEAG 109E241). We would like to thank for their support.

References

[Aranganayagi et al. 2009] Aranganayagi, S., Thangavel, K.: Improved K-Modes for Categorical Clustering Using Weighted Dissimilarity Measure, International Journal of Engineering and Mathematical Sciences, 5:2, 2009.

[Barbara et al. 2002] Barbará, D., Li, Y., Couto, J.: COOLCAT: An Entropy - Based Algorithm for Categorical Clustering. Conference on Information and Knowledge Management (CIKM'02), ACM, McLean, 2002, 582-589.

[Chen et al. 2003] Chen, K., Liu, L.: Towards Finding Optimal Partitions of Categorical Datasets, College of Computing, Georgia Institute of Technology, Technical Report, October, 2003.

[Coello 1998] Coello, C.C.: An Updated Survey of ga - based Multi - Objective Techniques, Technical Report Lania-RD-98-08, Laboratorio Nacional de Inform´atica Avanzada, 1998.

[Ganti et al. 1999] Ganti, V., Gehrke, J., Ramakrishnan, R.: CACTUS - Clustering Categorical Data Using Summaries, Conference on Knowledge Discovery and Data Mining (KDD'99), ACM, San Diego, 1999, 73-83.

[Gibson et al. 1998] Gibson, D., Kleiberg, J., Raghavan, P.: Clustering Categorical Data: An Approach based on Dynamical Systems, In Proceedings of 24th International Conference on Very Large Databases (VLDB'98), New York City, 1998, 311-323.

[Goldberg 1989] Goldberg, D.E.: Genetic Algorithms in Search, Addison Wesley, 1989.

[Guha et al. 2000] Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes, Inf. Syst. 25(5), 2000, 345-366.

[He et al. 2002] He, Z., Xu, X., Deng, S.: Squeezer: An Efficient Algorithm for Clustering Categorical Data, Journal of Computer Science and Technology, 17(5), 2002, 611-624.

[Horn et al. 1994] Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multi-Objective Optimization, Proceedings of IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation (ICEC'94), IEEE, Orlando, 1994, 82–87.

[Huang 1997] Huang, Z.: A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining, In Research Issues on Data Mining and Knowledge Discovery, 1997, 1-8.

[Huang 1998] Huang, Z.: Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values, Data Mining Knowledge Discovery, 2 (3), 1998, 283-304.

[Jung, 08] Jung, J.J.: Ontology-based Context Synchronization for Ad Hoc Social Collaborations, Knowledge-Based Systems, 21(7), 2008, 573-580.

[Jung, 09] Jung, J.J.: Contextualized mobile recommendation service based on interactive social network discovered from mobile users, Expert Systems with Applications, 36(9), 2009, 11950-11956.

[Jung, 10a] Jung, J.J.: Ontology Mapping Composition for Query Transformation on Distributed Environments, Expert Systems with Applications, 37(12), 2010, 8401-8405.

[Jung, 10b] Jung, J.J.: Integrating Social Networks for Context Fusion in Mobile Service Platforms, Journal of Universal Computer Science, 16(15), 2010, 2099-2110.

[Jung, 11] Jung, J.J.: Service Chain-based Business Alliance Formation in Service-oriented Architecture, Expert Systems with Applications, 38(3), 2011, 2206-2211.

[Jung, 12a] Jung, J.J.: Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB, Expert Systems with Applications, 39(4), 2012, 4049-4054.

[Jung, 12b] Jung, J.J.: Evolutionary Approach for Semantic-based Query Sampling in Large-scale Information Sources, Information Sciences, 182(1), 2012, 30-39.

[Ozyer et al. 2006] Ozyer, T., Alhajj, R.: Achieving Natural Clustering by Validating Results of Iterative Evolutionary Clustering Approach, Proceedings of IEEE Intelligent Systems, 2006.

[Ozyer et al. 2006] Ozyer, T., Alhajj, R.: Combining Validity Indexes and Multi-Objective Optimization based Clustering, Proceedings of 7th International FLINS Conference on Applied Artificial Intelligence, 2006.

[Ozyer et al. 2008] Ozyer, T., Alhajj, R.: Deciding on Number of Clusters by Multi-Objective Optimization and Validity Analysis, Journal of Multi-Valued Logic and Soft Computing 14(3), 2008, 457-474.

[Ozyer et al. 2009] Ozyer, T., Alhajj, R.: Parallel Clustering of High Dimensional Data by Integrating Multi-Objective Genetic Algorithm with Divide and Conquer, Appl. Intell. 31(3), 2009, 318-331.

[Ozyer et al. 2006] Ozyer, T., Alhajj, R., Barker, K.: Clustering by Integrating Multi-Objective Optimization with Weighted K-Means and Validity Analysis, Springer Verlag LNCS, Proceedings of IDEAL, 2006, 454-463.

[Periklis et al. 2004] Andritsos, P., Tsaparas, P., Miller, R. J., Sevcik, K.C.: LIMBO: Scalable Clustering of Categorical Data, Conference on Extending Database Technology (EDBT'04), 2004, 123-146.

[Schaffer 1998] Schaffer, J. D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. Proceedings of the International Conference on Genetic Algorithms and their Applications, Hillsdale, 1998, 93–100.

[Srinivas et al. 1995] Srinivas, N., Deb, K.: Multi-Objective Function Optimization Using Non-Dominated Sorting Genetic Algorithms, Evolutionary Computation, 2(3), 1995, 221–248.

[Xiong et al. 2006] Xiong, H., Tan, P. N., Kumar, V.: Hyperclique Pattern Discovery. Data Mining Knowledge Discovery 13 (2), 2006 , 219-242.

[Yan et al. 2008] Yan, H., Chen, K., Liu, L.: Determining the Best K for Clustering Transactional Datasets: A Coverage Density-based Approach, Journal of Data and Knowledge Engineering, 10(10), 2008, 333-343.

[Yang et al. 2002] Yang, Y., Guan, X., You, J.: CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data, In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02), ACM, New York City, 2002, 682-687.

[Zitzler et al. 1999] Zitzler, E., Thiele, L.: Multi-Objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3(4), 1999, 257–271.

[Zitzler et al. 2000] Zitzler, E., Deb, K., Thiele, L.: Comparison of Multi-Objective Evolutionary, Evolutionary Computation 8(2), 2000, 173–195.

[Zitzler 1999] Zitzler, E.: Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications, PhD thesis, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany, 1999.