A Modified Brainstorm Optimization for Clustering Using Hard c-Means

Conference Paper · November 2015

DOI: 10.1109/ICRCICN.2015.7434236

Reetika Roy
VIT University
1 PUBLICATION **1** CITATION
SEE PROFILE

J. Anuradha
VIT University
27 PUBLICATIONS **68** CITATIONS
SEE PROFILE

A Modified Brainstorm Optimization for Clustering Using Hard c-Means

Reetika Roy
School Of Computing Science and Engineering
VIT University, Vellore
Vellore, India
reetika.roy2012@vit.ac.in

Anuradha J
School Of Computing Science and Engineering
VIT University, Vellore
Vellore, India
januradha@vit.ac.in

***Abstract*— The preeminent intention of the proposed study is exploring the performance of the Brainstorm Optimization algorithm in Hard c-means clustering of data. The rationale behind this analysis is to generate a random solution set of centroids and then modify the centroids so as to refine the clusters. As we are using Brainstorm Optimization which is a form of evolutionary algorithm this refinement of centroid happens through competition and cooperation with existing centroid values. This algorithm incorporates both exploitation and exploration of the search space to generate the new centroids. The algorithm has been implemented with the Iris data set and its validity and effectiveness is tested with the help of commonly used internal evaluation measures for clustering like Davies Boudlin Index and Dunn Index.**

Keywords— Clustering, optimization, Brainstorm Optimization

I. Introduction

Clustering of data is the process of dividing multidimensional data vectors into different clusters or bins based on similarities [1]. Intuitively, data within a valid cluster have more similarities between themselves than they have with data belonging to a disparate cluster. Clustering has found its use in a variety of applications including domains like image processing, exploratory data analytics, pattern classification, data mining, and in solving a variety of mathematical problems. In most situations there is very little prior information available regarding the data and too many assumptions should not be made while dealing with the data. It is under these strict constraints that we can see the novelty of clustering in examining the interrelationships between data in order to assess their structure.

While in supervised learning, the algorithm has an external teacher to indicate the target class to which the data belongs like in case of discriminant analysis, clustering is a process of unsupervised learning where the given collection of data is not labelled and finding meaningful cluster solely depends on the data [2].

On the other hand, population based algorithms in recent times have shown greater promise than traditional single point based algorithms in solving problems that consist of a set of points (population) through competition and cooperation with each other [3]. Brainstorm optimization is one such population based optimization inspired from the brainstorming process enforced by human beings- the most intelligent of animals, in deliberating over solutions to problems. It's implementation can be seen in various optimization problems like optimal location and setting of FACTS devices [4], optimization of DC Brushless Motor [5], reactive power dispatch problem [6] and other optimization problems across various domains.

The first part of the paper deals with the introduction of clustering techniques and the brain storm optimization and the existing study that has been done on them. In the next section we introduce our modified implementation of it in refining the centroid values in the clustering process. And finally we give a proof of the effectiveness of our algorithm in light of existing k-means algorithm and evaluate the validity of the clusters [7] formed by this algorithm.

II. BACKGROUND STUDY

Techniques like clustering and hard c-means are being used for a long time now. While evolutionary computation [8] is a recent upcoming field that when coupled with the existing techniques in different ways have proven to be efficient and effective.

A. Cluster Analysis

Clustering is a method of placing elements from a data universe X consisting of n data samples into c classes having identified the number of subclasses of c clusters such that $2 \leq c < n$. Here $c = 1$ would mean there are no clusters at all and $c = n$ would mean that each data point itself represents a unique cluster. The core underlying principle in clustering is the premise that numerical data members belonging to any particular cluster exhibit more mathematical closeness among each other than they do with data members belonging to different clusters [1].

This similarity between members of clusters can be measured in a number of ways. The simplest method is calculating distance between pairs of features within the feature space. Using a suitable distance measure we can see that distance between same clusters is much less than that between points in two different clusters. In this paper, we do this by optimizing the weighted sum of squares between the data points and the cluster centers in the feature space.

B. Hard c-means Clustering (HCM)

Hard c-means Clustering is a method of separating data in a crisp sense, that is, each data point can belong to only one data cluster or partitions of data proposed by Bezdek [1981]. If we create a hard c-partition of X as a family of sets {S_i, i = 1, 2, 3, … , c}, then rules given below should apply:

$$\bigcup_{j=1}^{c} S_i = X \qquad (1)$$

$$S_i \cap S_j = \emptyset, \qquad \text{all } i \neq j$$

$$\emptyset \subset S_i \subset X, \qquad \text{all } i$$

Where, we represent the universe of data samples as a set space X = {x_1, x_2, x_3, ... , x_n}, and the number of class partitions, or clusters, into which we want to classify this is given by c. The characteristic function is defined as:

$$\chi_{ik} = \begin{cases} 1 \, , x_k \in S_i \\ 0 \, , x_k \notin S_i \end{cases}$$

For easiness in representation, we define our membership assignment of the jth data point in the ith cluster, or class, as

$$x_{ij} \equiv x S_i \left(x^j\right) \qquad (5)$$

Now the most reasonable c-partition is selected on the basis of an objective function. One such proposed criteria is the within-class sum of squared errors approach using a Euclidean norm to compute distance.

This algorithm expresses F (U, v) as our objective function given by:

$$F\,(U, v) = \chi_{ik}\,(d_{ik})^2 \qquad (6)$$

Where, U = the partition matrix,

and the parameter, v = a vector of cluster centers.

In this objective function equation, d_{ik} = a Euclidean distance measure between the kth data sample x_k and ith cluster center v_i, given by

$$d_{ik} = \left| x_{kj} - v_{ij} \right| = \left[\sum_{j=1}^{m} \left(x_{kj} - v_{ij} \right)^2 \right]^{1/2} \qquad (7)$$

There are a number of search algorithms that aim at improving the centroid values to move the objective function to an optimum value, resulting in valid clusters.

C. Brainstorm Optimization

As introduced in [2] and [9], brainstorm is a form of evolutionary computation that draws inspiration from the creative process followed by human beings during problem solving. When human beings face a problem they cannot solve individually, they form a group of people (who are usually from different backgrounds) and they get together to brainstorm. This has been seen to solve the problem with a much higher probability. The process includes a facilitator who ensures the generation of new ideas by administering the group to follow Osborn's pioneering set of rules for the process of idea generation [SMITH2002], namely

TABLE I. IDEA GENERATION BY OSBORN'S RULES

Rule 1. Hold back on judgement or criticism: No idea is a bad idea

Rule 2. Anything goes: Any idea coming to the mind is worth recording and sharing

Rule 3. Cross-fertilize (Piggy-back): Generated new idea by the amalgamation of existing ideas

Rule 4. Go for quantity: As many ideas as possible should be generated

Following these guidelines, we can produce a plethora of ideas and keep people involved in the process as perceptive as possible. To sum up, the process as seen from [2] and [9] goes as follows.

TABLE II. STEPS IN THE BRAINSTORMING PROCESS

Step I. For the brainstorming process, get a group of people from divergent backgrounds;

Step II. Use the rules given in table I for engendering new ideas;

Step III. Elect a number of ideas as better ideas by having a set of clients who pose as the owner of the problem select them;

Step IV. Breed new ideas by the rules in Table I, using the ideas picked in Step III with higher probability as clues;

Step V. Repeat Step III to have more better ideas to be picked out by the owners;

Step VI. In any respect, pick an idea and come up with more ideas using its functions and characteristics based on rules in Table I;

Step VII. Analyzing the ideas obtained and creating new ones from them will eventually generate a good enough solution. Otherwise go back to steps III and VI till we have the best possible idea.

During the process, the BSO algorithm uses grouping, replacing and creating operations mainly to create new idea from existing ideas to improve the idea generation process. We start with a randomly generated initial set of ideas, X_i=[x_{i1}, x_{i2}, $x_{i3, \dots}$, x_{iD}] where $1 \leq i \leq n$, n being the population size and D the dimensions of the problem, that is the number of factors relevant to the problem. BSO creates more new ideas based on the existing ones. These may be based on either a singular selected cluster or two selected clusters. A cluster is selected usually by the roulette strategy method, that is, more the idea in the cluster the greater the probability of it getting selected.

$$P_j = \frac{|M_j|}{N} \qquad (8)$$

where P_j is the probability of cluster j being selected, M_j is the number of ideas in cluster j. Meanwhile, when two clusters are selected it is done at random.

TABLE III. PSEUDO CODE FOR PROPOSED ALGORITHM

```
Algorithm:

01. Begin
02. Randomly generate N solution sets of cluster centroids ($X_i$, $1 \leq i \leq N$) and evaluate their fitness (in this case,
    objective function)
03. While (no convergence) do
        a. Cluster the N solution set into M clusters
        b. Record the solution set with least objective function as cluster center for each cluster
           //Probability of selecting a randomly selected cluster, 0.2
        c. If (random(0,1) < probabilityReplace)
               i. Replace the centroid of a randomly selected cluster with an idea generated at random
        d. End If
        e. Loop N times
                   //Probability of generating a new centroid based on one cluster, 0.8
               i. If (random(0,1) < probabilityOne)
                      1. With a probability $p_m$, select a cluster m
                         //Probability of selecting cluster center, 0.4
                      2. If (random(0,1) < probabilityOneCenter)
                             a. Enumerate a new idea $Y_i$ from a selected cluster center by adding a random value
                                to it.
                      3. Else
                             a. Enumerate new idea $Y_i$ from any haphazard solution set in the same way.
                      4. End If
              ii. Else //Generating new centroid based on two clusters
                      1. Randomly select two clusters $j_1$ and $j_2$
                         //Probability of using the cluster centers, 0.5
                      2. If (random(0,1) < probabilityTwoCenter)
                             a. Combine the two selected centers to get a new centroid $Y_i$
                      3. Else
                             a. Combine any two random centroids from the selected two clusters to get a new
                                centroid $Y_i$
                      4. End of If
             iii. End of If
              iv. Evaluate the new objective function and replace older value if better
        f. End of For
04. End of While
05. End
```

Moreover after cluster selection the algorithm decides whether to use cluster centers to create new ideas or any random value within the cluster.

III. Proposed Algorithm

The proposed algorithm makes use of properties from the above discussed techniques with slight modifications to create new cluster centers.

We initialize our algorithm the same way as the conventional BSO [2] and use the same concepts of grouping, replacing and creating operations. In step 02, we generate a solution set of size N consisting of arbitrary data points from the feature space and consider them our initial set of centroids. For each of these we evaluate the objective function using the formula 7. We then cluster the N solutions into M clusters of varied sizes.

In step 03.(c), we randomly replace a cluster's centroid with a certain probability for it. While all other steps ensure exploitation within the search space, this gives the possibility of exploration which diversifies the solution and increases possibility of refining the centroid values by probing in a much larger search space. We can do this replacement as,

$$Y_i = random(L_d, H_d) \qquad (9)\ ,$$

where L_d and H_d are lower and upper bounds respectively to limit the search within the possible values for centroids so that no unnecessary large or small value gets assigned to it and it loses its relevance to the problem.

The probability of selected a single cluster or two clusters is the same as been described in the BSO algorithm in formula 8. When a single cluster has been selected, whether we create a new ide based on cluster center or any random value, the new idea is created as:

$$Y_i = X_i + \mathrm{random}(0,1) \times (X_a - X_b) \qquad (10)\ ,$$

where X_a and X_b are any two random set of data points within the cluster.

When the new idea is created based on two ideas we use a method that has been borrowed from genetic algorithms where the new child is better than the parent [10]. We use crossover algorithms for the same. For this we could use Arithmetic crossover to create Y_i, that is:

$$Y = N \times X_1 + (1 - N) \times X_2 \qquad (11)$$

$$Y' = (1 - N) \times X_1 + N \times X_2 \qquad (12)\ ,$$

where N is a random value between 0 and 1, for all values of X where X_1 and X_2 are the two selected solution sets from clusters j_1 and j_2, respectively.

Or we can use heuristic crossover [11], that gives a more informed solution taking the best parent to create the child using the formula:

$$Y = best\ parent + R * (best\ parent - worst\ parent) \qquad (13)$$

$$Y' = best\ parent \qquad (14)\ .$$

where we determine best parent as the one with the lesser objective function and R is again a random value between 0 and 1.

After the Y_i has been created we evaluate its objective function and compare it to the existing objective function of X_i. If better we replace the old centroid by the new centroid values.

If convergence criteria is met the algorithm terminates and gives us the best centroid result, otherwise continues generation by the process of grouping, replacing and creating until we reach convergence.

IV. Experimental Studies

In the course of this paper, we have implemented the Brainstorm algorithm to create new centroid values in a clustering algorithm to obtain better clusters. It can be set up in a number of ways just by changing the values of the different parameters used in the algorithm:

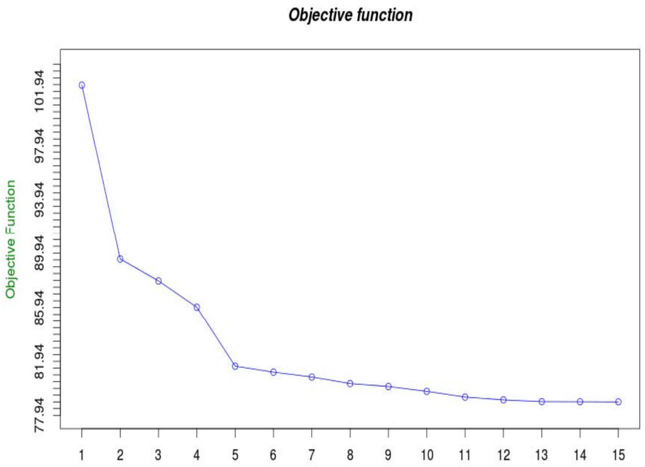

Fig. 1. Change in objective function value

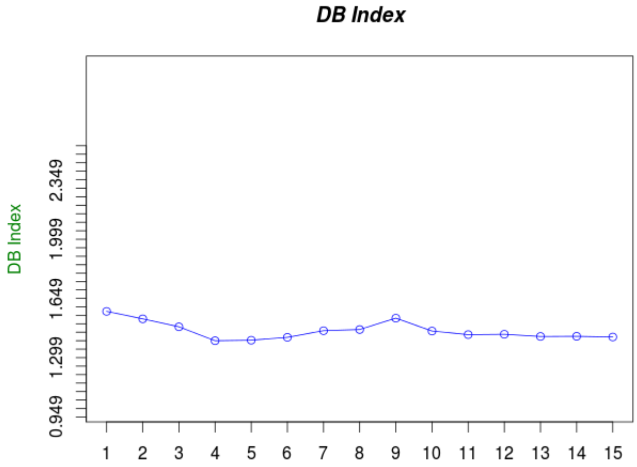

Fig. 2. Change in Davies Boudlin Index value

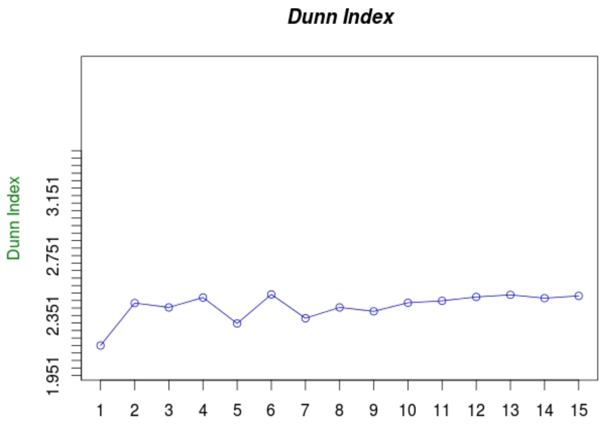

Fig. 3. Change in Dunn Index

TABLE IV. ALGORITHM PARAMETERS

Algorithm	Parameters				Objective Function	DB Index	D Index
	probabilityReplace	*probabilityOne*	*probabilityOneCenter*	*probabilityTwoCenter*			
BSO *k=3*	0.3	0.8	0.4	0.5	78.95	1.43	2.482
BSO *k=3*	0.4	0.8	0.5	0.5	78.94	1.42	2.480
BSO *k=3*	0.3	0.6	0.4	0.5	79.99	1.40	2.494
BSO *k=3*	0.2	0.8	0.4	0.5	81.09	1.43	2.361

The plot in Fig 1, Fig 2 and Fig 3 shows us the change in the objective function value, DB Index and Dunn Index over the iterations of our algorithm. All of these values are from the implementation of this algorithm on the Iris dataset.

The algorithm has been tested on the Iris Dataset which has 150 instances with four attributes for each- sepal length, sepal width, petal length and petal width. We cluster these 150 instances into three clusters here taking *k=3* in all the data that has been presented. The objective function, Davies Boudlin Index (DB Index) and Dunn Index have been plotted for the same in Fig I, II and III over a number of iterations. As we can see from the Fig I, even if we start out with a very high objective function, the algorithm can drastically bring the value down and then it slowly moves towards its convergence. The same can be seen in Fig II and III as well. Although some intermediate value generated may show a sudden optimum value, eventually we reach an overall best solution for the problem.

The objective function has already been discussed earlier. It optimizes intra-cluster similarity. Another measure is the DB Index value introduced in [12], which is based on the proportionality of within cluster and between cluster differences. The Davies Boudlin Index is mathematically represented as:

$$DB = \frac{1}{k}\sum_{m=1}^{k} \max_{m\neq n}\{D_{mn}\} \qquad (15)$$

where $D_{m,n}$ = the value for within-to-between cluster distance ratio for the m^{th} and n^{th} clusters.

In mathematical terms,

$$D_{mn} = \frac{\overline{d_m}+\overline{d_n}}{d_{mn}} \qquad (16)$$

where $\overline{d_m}$ = the mean distance between each point in the m^{th} cluster and the centroid of the m^{th} cluster.

$\overline{d_n}$ = the mean distance between each point in the m^{th} cluster and the centroid of the m^{th} cluster.

$d_{m,n}$ = the Euclidean distance between the centroids of the m^{th} and n^{th} clusters.

Thus the optimal clustering solution should have a less Davies Boudlin Index. The worst case solution will have a very high value for this measure.

The third measure used in this case is the Dunn Index [13]. It is also an internal evaluation scheme and this is based only on the data in the particular cluster. It aims as maximizing inter- Change in Dunn Index value cluster distance and minimizing intra-cluster distance like all other algorithms. It can be defined as:

$$DU_k = \frac{\min\limits_{1\le i<j\le k} \delta(C_i,C_j)}{\max\limits_{1\le m\le k}\Delta_m}, \qquad (17)$$

where $\delta(C_i, C_j)$ = the inter-cluster distance between cluster C_i and C_j [14].

Meanwhile, the numerator that is the diameter of the cluster can be calculated by a number of ways, like taking the distance between the farthest two points belonging to the same cluster, mean of pairwise distance between data points within the cluster or the men of distance of all data points from the centroid. We have used the later in our experiment. In case of Dunn Index, a higher value refers to compact and well separated clusters.

We have provided the comparison of these three parameters for the Iris dataset using the conventional k-means clustering as well as our proposed algorithm in the Table 4.

TABLE V. EVALUATION OF ALGORITHM

Parameter	Algorithms	
	HCM	***HCM using BSO***
Objective Function	142.89	**78.94**
DB Index	3.294	**1.421**
Dunn Index	0.911	**2.482**

V. CONCLUSION

In this paper we have proposed a technique of refining cluster centroids in the conventional hard c-means clustering using the Brainstorm Optimization algorithm instead of the conventional iterative process. Having compared it to the conventional process using some common benchmarks like objective function, Davies Boudlin Index and Dunn Index we see our algorithm performing significantly better than the HCM algorithm without BSO. Our algorithm is successful in reducing the intra-cluster distances and forming separate compact clusters. This algorithm is still in its inception and further research and study can be conducted on its implementation, its parameters or its performance with respect to other optimization techniques to improve it more.

REFERENCES

[1] Jain, A.K., M. Narasimha Murthy, Flynn, P.J.: "Data clustering: a review", ACM Computing Surveys (1999), page 264-323

[2] Zhi-hui Zhan, Jun Zhang, Yu-hui Shi, and Hai-lin Liu: "A Modified Brain Storm Optimization" , WCCI 2012 IEEE World Congress on Computational Intelligence, June, 10-15, 2012 - Brisbane, Australia.

[3] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest and Dusan Fister: "A Brief Review of Nature-Inspired Algorithms for Optimization"

[4] A. Rezaee Jordehi, "Brainstorm optimisation algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems", International Journal of Electrical Power & Energy Systems Volume 69, July 2015, page 48–57

[5] Haibin Duan, Shuangtian Li, Yuhui Shi: "Predator-Prey Brainstorm Optimization for DC Brushless Motor", Magnetics, IEEE Transactions on (Volume:49 , Issue 10), 2013, page 5336 – 5340

[6] K.Lenin, Dr. B.Ravindhranath Reddy, Dr. M.Surya Kalavathi: "Brain Storm Optimization Algorithm for Solving Optimal Reactive Power Dispatch Problem", International Journal of Research in Electronics and Communication Technology (IJRECT 2014), Vol. 1, Issue 3 July - Sept 2014

[7] Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems (2001) page 107-145

[8] Yang, X.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)

[9] Y. Shi, "Brain storm optimization algorithm," in Proc. 2nd Int. Conf. on Swarm Intelligence, 2011, page 303-309

[10] Tatsuya Nomura, "An Analysis on Crossovers for Real Numbe r Chromosomes in an Infinite Population Size"

[11] Zbigniew Michalewicz, "Heuristic Methods for Evolutionary Computation Techniques"

[12] Davies, D., Bouldin, W.: A cluster separation measure. IEEE PAMI 1 (1979), page 224-227

[13] Dunn, J.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics (1974), page 95-104

[14] Sandro Saitta, Benny Raphael, and Ian F.C. Smith, A Bounded Index for Cluster Validity" in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, page 271- 350