Volume 5, Issue 12, December 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Survey for Optimization Techniques in Data Mining

Nancy Bhardwaj, Gurwinder Kaur
GGSCMT Kharar, Punjab,
India

Abstract*— There are number of techniques proposed by several researchers to analyze the performance of clustering algorithms in data mining. All these techniques are not suggesting good results for the chosen data sets and for the algorithms in particular. Some of the clustering algorithms are suit for some kind of input data. This research work uses arbitrarily distributed input data points to evaluate the clustering quality and performance of two of the partition based clustering algorithms namely k- Means and k-Medoids. To evaluate the clustering quality, the distance between two data points are taken for analysis. The computational time is calculated for each algorithm in order to measure the performance of the algorithms

***Keywords*— DBSCAN, PSO, SGF**

I. INTRODUCTION

This study is aimed to give a comparative review of two of the various partitioning based clustering methods. Clustering is a division of data objects into groups of similar objects. Such groups are called clusters. Objects possessed by same cluster tend to be similar, while dissimilar objects are possessed by different clusters. These clusters represent groups of data and provide simplification by representing many data objects by fewer clusters. And, this helps to model data by its clusters. Clustering is a method of unsupervised learning and a well known technique for statistical data analysis. It is used in many fields such as machine learning, image analysis, pattern recognition, outlier detection, and bioinformatics to name a few. Various researchers have proposed different methods to achieve clustering. Along with managing a very large dataset, a robust clustering method must satisfy some requirements such as scalability, dealing different types of attributes, discovering clusters of arbitrary shape, high dimensionality, ability to deal with noise and outliers, interpretability and usability. With clustering, time complexity increases with dealing large number of dimensions and large set of data objects. Also the effectiveness depends upon the definition of similarity (or dissimilarity) among objects. Along with this, the output of clustering can be interpreted in different ways [1]. Different clustering methods can be classified into various categories such as partitioning based methods, hierarchical methods, grid-based methods, density-based methods, model-based methods, methods for high dimensional data and constraint-based clustering [2]. Among all these methods, this paper is aimed to explore two methods – k-means and k-medoids – which are partitioning based clustering methods. These methods are discussed along with their algorithms, strength and limitations.

II. LITERATURE SURVEY

DBSCAN: Density Based Spatial Clustering of Applications with Noise. In this section, we present the algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) which is designed to discover the clusters and the noise in a spatial database. Ideally, we would have to know the appropriate parameters Eps and MinPts of each cluster and at least one point from the respective cluster. Then, we could retrieve all points that are density-reachable from the given point using the correct parameters. But there is no easy way to get this information in advance for all clusters of the database.

However, there is a simple and effective heuristic to determine the parameters Eps and MinPts of the "thinnest", i.e. least dense, cluster in the database. Therefore, DBSCAN uses global values for Eps and MinPts, i.e. the same values for all clusters. The density parameters of the “thinnest” cluster are good candidates for these global parameter values specifying the lowest density which is not considered to be noise.

The idea of it was:

1. ε-neighbor: the neighbors in ε semi diameter of an object
2. Kernel object: certain number (MinP) of neighbors in ε semi diameter.
3. To a object set D, if object p is the ε-neighbor of q, and q is kernel object, then p can get “direct density reachable” from q.
4. To a ε, p can get “direct density reachable” from q; D contains Minp objects; if a series object p1, p2,….. pn, p1= q. pn = q. then pi+1 can get “direct density reachable” from pi., Pi D ,1≤ i≤ n.
5. To ε and MinP, if there exist an object o(o<D) p and q can get “direct density reachable” from o,p and q are density connected. Density Reachability and Density Connectivity: Density reachability is the first building block in dbscan. It defines whether two distance close points belong to the same cluster. Points p1 is density reachable from p2 if two conditions are satisfied: (i) the points are close enough to each other: distance (p1, p2) <e, (ii) there are enough of points in is neighborhood: |{r: distance(r, p2)}|>m, where r is a database point.

Density connectivity is the last building step of dbscan. Points p0 and pn are density connected, if there is a sequence of density reachable points p1,i2,...,i(n-1) from p0 to pn such that p(i+1) is density reachable from pi. A dbscan cluster is a set of all density connected points.

Explanation of DBSCAN Steps

DBScan requires two parameters: epsilon (eps) and minimum points (minPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.

If the number of neighbors is greater than or equal to minPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the evaluation process for all the neighbours recursively.

If the number of neighbors is less than minPts, the point is marked as noise.

If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset.

Advantages of DBSCAN

DBScan requires two parameters: epsilon (eps) and minimum points (minPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.

If the number of neighbors is greater than or equal to minPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the evaluation process for all the neighbours recursively.

If the number of neighbors is less than minPts, the point is marked as noise.

If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset. Disadvantages of DBSCAN

DBScan requires two parameters: epsilon (eps) and minimum points (minPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.

DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-epsilon combination cannot be chosen appropriately for all clusters then

III. CLUSTERING ALGORITHM

The k-Means Algorithm

The k-Means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori [10, 11]. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

where is a chosen distance measure between a data point and the cluster centre , is an indicator of the distance of the *n* data points from their respective cluster centers.

1. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the k centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The algorithm is significantly sensitive to the initial randomly selected cluster centers. The k- Means algorithm can be run multiple times to reduce this effect. k-Means is a simple algorithm that has been adapted to many problem domains and it is a good candidate to work for a randomly generated data points. One of the most popular heuristics for solving the k-Means problem is based on a simple iterative scheme for finding a locally minimal solution [3, 4, 10]. This algorithm is often called the k-Means algorithm.

Strengths:

Relatively scalable and efficient in processing large data sets; complexity is *O (i k n),* where i is the total number of iterations, *k* is the total number of clusters, and n is the total number of objects. Normally, *k<<n* and *i<<n.*

Easy to understand and implement.

Weaknesses:

Applicable only when the mean of a cluster is defined;
not applicable to categorical data.
Need to specify *k*, the total number of clusters in
advance.
Not suitable to discover clusters with non-convex shape,
or clusters of very different size.
Unable to handle noisy data and outliers.
May terminate at local optimum.
Result and total run time depends upon initial partition.

The k-Medoids Algorithm

The k-Means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data [11]. Instead of taking the mean value of the objects in a cluster as a reference point, a medoid can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the k-Medoids method. The basic strategy of k- Medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoids) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. The k-Medoids method uses representative objects as reference points instead of taking the mean value of theobjects in each cluster is the key point of this method. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects.
The strengths and weaknesses of this algorithm are mentioned as below.

Strengths:

More robust than k-means in the presence of noise and outliers; because a medoid is less influenced by outliers or other extreme values than a mean.

Weaknesses:

Relatively more costly; complexity is *O(i k (n-k)2),* where i is the total number of iterations, is the total number of clusters, and *n* is the total number of objects. Relatively not so much efficient.
Need to specify *k*, the total number of clusters in advance.
Result and total run time depends upon initial partition.

Distance Measure

An important step in most clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point ($x = 1$, $y = 0$) and the origin ($x = 0$, $y = 0$) is always 1 according to the usual norms, but the distance between the point ($x = 1$, $y = 1$) and the origin can be 2, or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance. Another important distinction is whether the clustering uses symmetric or asymmetric distances. Many of the distance functions listed above have the property that distances are symmetric (the distance from object A to B is the same as the distance from B to A). In other applications, this is not the case. A true metric gives symmetric measures of distance. The symmetric and 2-norm distance measure is used in this research work. In the Euclidean space **R**n, the distance between two points is usually given by the Euclidean distance (2-norm distance). The formula for 2-norm distance is:
The 2-norm distance is the Euclidean distance, a generalization of the Pythagorean Theorem to more than two coordinates. It is what would be obtained if the distance between two points were measured with a ruler: the "intuitive" idea of distance.

IV. PSO (PARTICLE SWARM OPTIMIZATION)

Particle swarm optimization (PSO) is a new evolutionary computing method that was developed by Kennedy and Eberhart in 1995 through the simulation of simplified social models of bird flocks. Due to its excellent performance, PSO has become one of the hotspots in evolutionary computing research and has been used in many applications such as function optimization, neural network training, and fuzzy control systems in recent years. The algorithm of PSO emulates from behavior of animals societies that don't have any leader in their group or swarm, such as bird flocking and fish schooling.[22] Typically, a flock of animals that have no leaders will find food by random, follow one of the members of the group that has the closest position with a food source (potential solution). The flocks achieve their best condition simultaneously through communication among members who already have a better situation. The process of PSO algorithm in finding optimal values follows the work of this animal society. Particle swarm optimization consists of a swarm of particles, where particle represent a potential solution. The algorithm of PSO is initialized with a particles obtained from SGF texture feature and then searches for optima by updating generations.

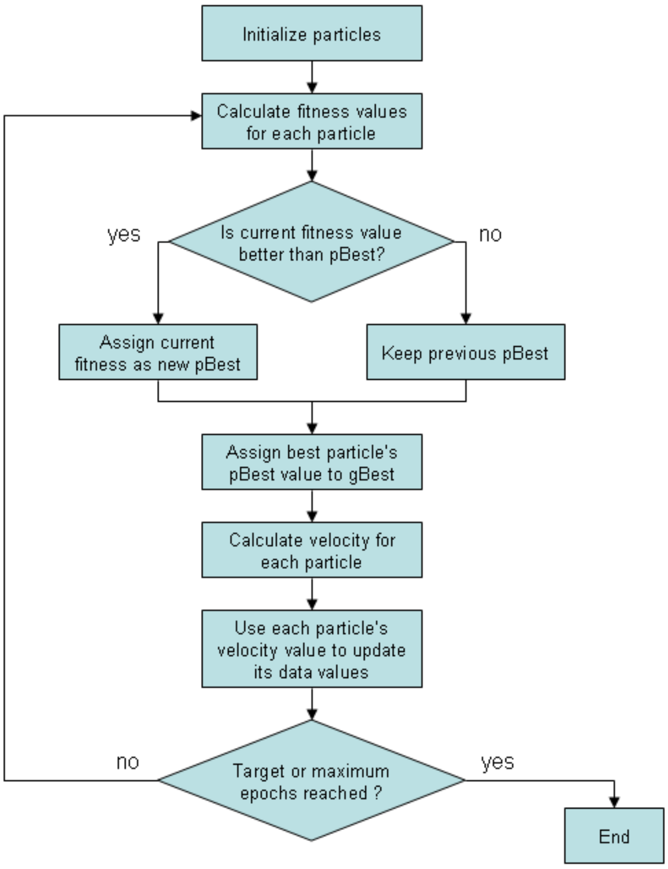
Initialize particles
Calculate fitness values for each particle
Is current fitness value better than pBest?
yes
no
Assign current fitness as new pBest
Keep previous pBest
Assign best particle's pBest value to gBest
Calculate velocity for each particle
Use each particle's velocity value to update its data values
Target or maximum epochs reached ?
no
yes
End

Fig 2. Flow chart for PSO

V. CONCLUSION

From the above study, it can be concluded that partitioning based clustering methods are suitable for spherical shaped clusters in small to medium sized data sets. K-means and k-medoids – both the methods find out clusters from the given database. Both the methods require to specify *k*, no of desired clusters, in advance. Result and runtime depends upon initial partition for both of these methods. The advantage of k-means is its low computation cost, while drawback is sensitivity to noisy data and outliers. Compared to this, k-medoid is not sensitive to noisy data and outliers, but it has high computation cost.

REFERENCES

[1] Manas Yetirajam, Pradeep Kumar Jena, “Enhanced Color Image Segmentation of Foreground Region using Particle Swarm Optimization” International Journal of Computer Applications (0975 – 8887) Volume 57– No.8, November 2012

[2] Salima Nebti, “Bio-Inspired Algorithms for Color Image Segmentation” International Journal of Computer Applications (0975 – 8887) Volume 73– No.18, July 2013

[3] Narinder Kumar, R P S Bedi, “NEW TECHNIQUE FOR IMAGE SEGMENTATION”, Journal of Bio-Technology and Research (JBTR) Vol.2, Issue 2 June 2012 8-16

[4] Gaganpreet Kaur, Harpreet kaur , “DCT A Review on Medical Image Segmentation Using Biogeography Based Optimization” International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-4) April 2013.

[5] Gaganpreet Kaur, Harpreet kaur: “ A Survey on Comparison between Biogeography Based Optimization and Other Optimization Method” International Journal of Advanced Research in Computer Science and Software Engineering, February 2013

[6] Majid Gholamiparvar Masooleh, and Seyyed Ali Seyyed Moosavi, “An Improved Fuzzy Algorithm for Image Segmentation”, World Academy of Science, Engineering and Technology 14 2008

[7] Linyi Li, Deren Li: “Fuzzy entropy image segmentation based on particle swarm optimization”

[8] K.M.MURUGESAN, DR.S.PALANISWAMI, “EFFICIENT COLOUR IMAGE SEGMENTATION USING MULTI-ELITIST- EXPONENTIAL PARTICLE SWARM OPTIMIZATION”, Journal of Theoretical and Applied Information Technology

[9] L.Sankari and Dr.C.Chandrasekar, “ SEMI SUPERVISED IMAGE SEGMENTATION USING OPTIMAL HIERARCHICAL CLUSTERING BY SELECTING INTERESTED REGION AS PRIOR INFORMATION”, Journal of Global Research in Computer Scienc, Volume 2, No. 11, November 2011

[10] Serkan Kiranyaz, Stefan Uhlmann, Turker Ince and Moncef Gabbouj, “Perceptual Dominant Color Extraction by Multi-Dimensional Particle Swarm Optimization

[11] Parag Puranik, Dr. P.R. Bajaj, Prof. P.M. Palsodkar, "Fuzzy based Color Image Segmentation using Comprehensive Learning Particle Swarm Optimization (CLPSO) – A Design Approach" International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

[12] Fahd M. A. Mohsen, Mohiy M. Hadhoud, Khalid Amin "A new Optimization-Based Image Segmentation method By Particle Swarm Optimization", International Journal of Advanced Computer Science and Applications

[13] Mohamed Sami1, Nashwa El-Bendary, Tai-hoon Kim, and Aboul Ella Hassanien1, "Using Particle Swarm Optimization for Image Regions Annotation"

[14] A. Borji M. Hamidi A. M. Eftekhari moghadam, "CLPSO-based Fuzzy Color Image Segmentation"

[15] Ritesh Srivastava, Shivani Agarwal, Ankit Goel, Vipul Gupta, "TERRIAN IDENTIFICATION USING COCLUSTERED MODEL OF THE SWARM INTELLEGENCE & SEGMENTATION TECHNIQUE"

[16] Parag Puranik, Preeti Bajaj, Ajith Abraham, Prasanna Palsodkar, and Amol Deshmukh, "Human Perception-based Color Image Segmentation Using Comprehensive Learning Particle Swarm Optimization", Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 3, July 2011

[17] Ahmed Afifi, Toshiya Nakaguchi, Norimichi Tsumura, Yoichi Miyake, "Particle Swarm Optimization Based Medical Image Segmentation Technique"

[18] Sara Saatchi and Chih-Cheng Hung, "Swarm Intelligence and Image Segmentation" Southern Polytechnic State University USA

[19] Jzau-Sheng Lin and Shou-Hung Wu, "A PSO-based Algorithm with Subswarm Using Entropy and Uniformity for Image Segmentation", International Journal of Computer, Consumer and Control (IJ3C), Vol. 1, No.2 (2012)

[20] Surbhi Gupta, Krishma Bhuchar, Parvinder S. Sandhu, "Implementing Color Image Segmentation Using Biogeography Based Optimization", 2011 International Conference on Software and Computer Applications IPCSIT vol.9 (2011)

[21] Fahd Mohsen, Mohiy Hadhoud, Kamel Mostafa, Khalid Amin, "A New Image Segmentation method based on Particle Swarm Optimization", The International Arab Journal Of Informational Technology, Vol 9, No. 5, 2012

[22] Amanpreet Kaur, M.D. Singh, "An Overview of PSO- Based Approaches in Image Segmentation", International Journal of Engineering and Technology Volume 2 No. 8, August, 2012

[23] Er.Krishma Bhuchar, ER. Rekha Rani, ER.Bharti Jyoti, "Performance Evaluation of Biogeography Based Image Segmentation"

[24] Rajwinder Kaur, Rakesh Khanna, "Medical Image Quantization using Biogeography based Optimization", International Journal of Computer Applications (0975 – 888), Volume 48– No.12, June 2012

[25] Gaganpreet Kaur, Dr. Dheerendra Singh ,Harpreet Kaur, "Detection of Abnormal Tissue Growth in MRImaging using Biogeography Based Optimization", International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 2, Issue 8, August 2013

[26] Mittu Mittal, Gagandeep, "A New Evolutionary Algorithm developed for Global Optimization (BBO)", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 2, February 2013

[27] Rajeshwar Dass, Priyanka, Swapna Devi, "Image Segmentation Techniques", IJECT Vol. 3, Issue 1, Jan. - March 2012

[28] M.R. Lohokare, S. S. Pattnaik, S. Devi, K. M. Bakwad, D. G. Jadhav, "BIOGEOGRAPHY-BASED OPTIMIZATION TECHNIQUE FOR BLOCK-BASED MOTION ESTIMATION IN VIDEO CODING", National Conference on Computational Instrumentation CSIO Chandigarh, INDIA, 19-20 March 2010

[29] "Using The ACO Algorithm in Image Segmentation for Optimal Thresholding"