DDG-CLUSTERING: A NOVEL TECHNIQUE FOR HIGHLY ACCURATE RESULTS

Zahraa Said Ammar and Mohamed Medhat Gaber
Centre for Computer Science and Software Engineering, Monash University,
900 Dandenong Rd, Caulfield East, VIC3145, Australia

ABSTRACT

A key to the success of any clustering algorithm is the similarity measure applied. The similarity among different instances is defined according to a particular criterion. State-of-the-art clustering techniques have used distance, density and gravity measures. Some have used a combination of two. Distance, Density and Gravity clustering algorithm "*DDG-Clustering*" is our novel clustering technique based on the integration of three different similarity measures. The basic principle is to combine distance, density and gravitational perspectives for clustering purpose. Experimental results illustrate that the proposed method is very efficient for data clustering with acceptable running time.

KEYWORDS

Data mining, data clustering, cluster density, cluster gravity, k-means clustering.

1. INTRODUCTION

Clustering is the process of grouping similar objects together to extract meaningful information from data repositories. It is done in such a way that the objects assigned to the same cluster have high similarity, while the similarity among objects assigned to different clusters is low [6]. Clustering analysis is a subject of active research in numerous fields. Many similarity measures and clustering techniques have been proposed in the literature [4]. Similarity between objects is assessed according to different measures. These techniques can be divided into several categories including *distance-based*, *density-based* and *gravitational* clustering techniques. An example of *distance-based* algorithm is *K-means* [1], one for *density-based* is *DBSCAN* [9], and for *gravitational clustering* is the one developed in [5].

The majority of data clustering techniques are *distance-based*. One of the most used *distance-based* techniques is *K-means*. *K-means* is a clustering algorithm that uses iterative approach to find K-Clusters based on distance for the similarity measure [1]. *K-means* is highly used due its ease of implementation. However, the number of clusters needs to be specified in advance, is unable to handle noisy data and outliers, and is unsuitable to discover clusters of non-convex shape [2].There are different algorithms proposed to cluster datasets. Partition-based and density-based algorithms are commonly seen as fundamentally and technically distinct. Work on combining both methods has focused on an applied rather than a fundamental level and without considering the gravitational force of clusters [8]. Although these algorithms are shown to be efficient, they may easily lose very important information about the distributions of clusters, which are important to match the similarity among clusters. In [3] a proposed hybrid clustering algorithm has been proposed to combine representative based clustering and agglomerative clustering methods. However they employ different merging criteria and perform a narrow search without considering the gravity and density of clusters. In this paper we propose a novel clustering technique based on the three measures of density, gravity and distance to get more accurate clustering results. We coined our technique *DDG-Clustering*.

The rest of the paper is organized as follows. In Section 2, we describe our proposed approach. Section 3 presents the experiment performed over both synthetic and real data sets, along with their results and analysis. Finally we conclude the paper and present our future work in Section 4.

2. DDG-CLUSTERING

We have developed a new clustering technique based on the gravitational, distance and density based clustering approaches. The basic ideas behind applying this algorithm are:

1. The combination between distance, gravity and density approaches will generate more accurate clusters.

2. Depending on the type of application, we may need different approaches to get more accurate clustering results. The *DDG-Clustering* algorithm combines the different approaches and gives the user the decision of which approach is more important depends on the nature of the dataset.

3. Number of clusters will be known in advance using the one-pass *LWC* algorithm [10] before implementing *k-means* algorithm.

The process starts by applying the *LWC* algorithm on the dataset. The output of the *LWC* represents the number of clusters in the dataset. Using *LWC* algorithm in *preparation step* helps to know the number of clusters in advance before applying the *K-means*. Then, the *K-means* is applied for only one or two iterations and stops before maintaining the maximum stability status. This will produce the initially shaped clusters to be ready for the *formation step*.

After completing the first component of preparation, the second component is performed on the initially shaped clusters until maintaining the stability in all clusters. The main idea is to examine the data point from different perspectives and then make voting among the different approaches' to choose the best candidate cluster. The algorithm is repeated until convergence. A Pseudo code of the formation component is given in Figure 1, where the *DP[i]* is the data point of index *i* in initial clusters denoted as *initialClusters*.

```
1.  Do
2.  Foreach DP(i) is the data Point belongs to initialClusters(j)
3.       checkDistance( DP(i), initialClusters)
4.       checkGravity( DP(i), initialClusters)
5.       checkDensity( DP(i), j, initialClusters)
6.       Vote( DistanceCandidate, DensityCandidates, gravityCandidates)
7.       Assign DP(i) to the best candidate cluster
8.  End for
9.  While( Stability Criteria)
```

Figure 1. Formation Component

According to density approach, within a threshold we check whether the data point will increase or decrease the density of the clusters when it joined or left clusters respectively. The density of cluster is simply considered as the distribution of the data points into the cluster as in the formula (1). Where (m) is the number of points in the cluster, (p) is the data point and (C) is the cluster centre.

$$ClusterDensity = \frac{SizeOfCluster}{AverageDist.} \quad (1)$$

$$AverageDist. = \frac{\sum_{i=0}^{m} |p_i - C|}{m} \quad (2)$$

The current density of the clusters collection is compared to the expected density if the data point is moved to other cluster. The density algorithm examines the clusters within fixed threshold. The clusters which will cause global density gain, if the data points joined it, are chosen as a candidate clusters from the density perspective. The algorithm of density is illustrated in Figure 2.

```
1. Procedure CheckDensity( DP[i],current cluster, IntialClusters)
2.      Current global density=0
3.      Expected global density=0
4.      For each (Cluster N within density threshold)
5.         Calculate the current density of clusterN
6.         Add the Current densit y to global density
7.      End For
8.      For each (Cluster N within density threshold)
9.         Add DP[i] to cluster N and remove it from current cluster
10.        Calculate the Expected global density
11.        Compare Current and Expected  global density
12.        If ( Expected global Density > current global Density)
13.           Calculate the densityGain = ExDensity- currentDensity
14.           Add Cluster N to the set of density candidate clusters DensityCandidates
15.        End if
16.     End For
17.  Return DensityCandidates;
```

Figure 2. Density Measure in *DDG-Clustering*

The second perspective is to examine the cluster according to its gravitational force. There exists a kind of force between any two objects in the universe and this force is called gravitation force [7]. According to Newton universal law of gravity, the strength of gravitation between two objects is in direct ratio to the product of the masses of the two objects, but in inverse ratio to the square of distance between them. The law can be described as follows:

$$F_g = G\frac{m_1 m_2}{r^2} \qquad (3)$$

Where F is the gravitation force between two objects; G is the constant of universal gravitation; m_1 is the mass of object 1; m_2 the mass of object 2 and r the distance between the two objects.

Each cluster generates its own gravitational force created from its weight. The bigger the weight of the cluster the stronger the gravitational force produced from it. And therefore, the more points that cluster can attract. If the data point location is within the gravitational field of a cluster, then the data point will be attracted by this cluster's gravitational force. Therefore, that cluster will be considered as a candidate cluster from the gravitational point of view. The gravitational threshold controls the size of the gravitational field surrounding the cluster. A pseudo code for the gravity measure is given in Figure 3.

```
1. Procedure CheckGravity( DP[i], intialClusters)
2.   For each (Cluster N in the intialClusters and)
3.      Calculate the distance between the centre of cluster N and DP[i]
4.      Calculate the gravitational force between DP[i] and cluster N
5.      If (gravitational force is within the gravitational threshold)
6.        Add Cluster N to the set of gravity candidate clusters GravityCandidates
7.      End If
8.   End For
9.  Return GravityCandidates;
```

Figure 3. Gravity Measure in *DDG-Clustering*

The data points are lastly checked from the distance based prospective. After Applying the density, gravity and distance examination, we get a set of candidate clusters from each approach. The voting system is used to decide which cluster is the best candidate to allocate the data point in. The pseudo code for the voting system is illustrated in Figure 4.

```
1.   Procedure Vote( DistanceCandidate, DensityCandidates, gravityCandidates)
2.      For each (Cluster N in the candidate clusters)
3.        If( Cluster N is the DistanceCandidate  and Cluster N is member in DensityCandidates and
          Cluster N is member in the GravityCandidates
4.          Set Cluster N as the best candidate cluster
5.        Else If( Cluster N is the DistanceCandidate and Cluster N is member in the DensityCandidates)
6.          Calculate DensityGain as Cluster N DensityGain
7.          For each (Cluster J in the DensityCandidates and Cluster J is member in the GravityCandidates)
8.            Calculate DensityGain as Cluster J DensityGain.
9.          End For
10.         If( cluster J DensityGain> Cluster N Density Gain)
11.           Set Cluster J as the best candidate Cluster
12.         Else
13.           Set Cluster N as the best candidate cluster.
14.         End If
15.       Else If (Cluster N in the DensityCandidates and Cluster N is member in the GravityCandidates)
16.         Set Cluster N as the best candidate cluster.
17.       Else If ( Cluster N is the DistanceCandidate and Cluster N is member in the GravityCandidates)
18.         Set Cluster N as the best candidate cluster.
19.       Else
20.         Set DistanceCandidate as the best candidate cluster.
21.       End If
22.     End For
23.  Return best candidate cluster;
```

Figure 4. Voting Among the Three Measures in *DDG-Clustering*

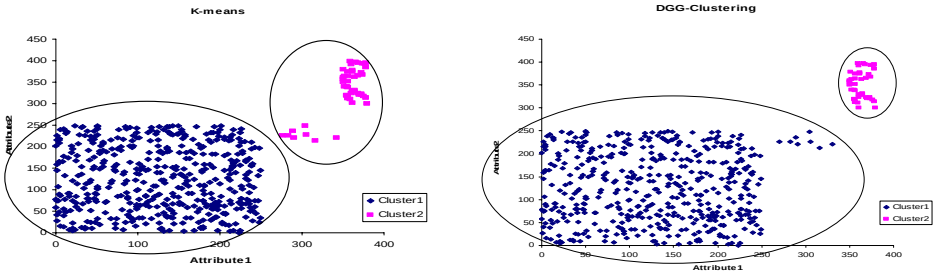
The *formation component* will be repeated till the number of points moved among clusters reduces and maintains certain percentage of accuracy. The relative importance of each approach is set by the user. In our experiments, we have assumed that all the measures have the same weight.

3. EVALUATION

To establish the practical efficiency of the proposed algorithm, we implemented it and tested its performance on a number of data sets. These included both synthetically generated data and data used in real applications. We generated data points in R2. The initial centres were chosen by taking a random sample. Then a Gaussian

distribution was then generated around each centre. We ran both K-means and DDG-Clustering algorithms. For each run, a new set of initial centre points and a new set of seed points were generated and both algorithms were run using the same data and initial centre points. The algorithms ran until convergence.

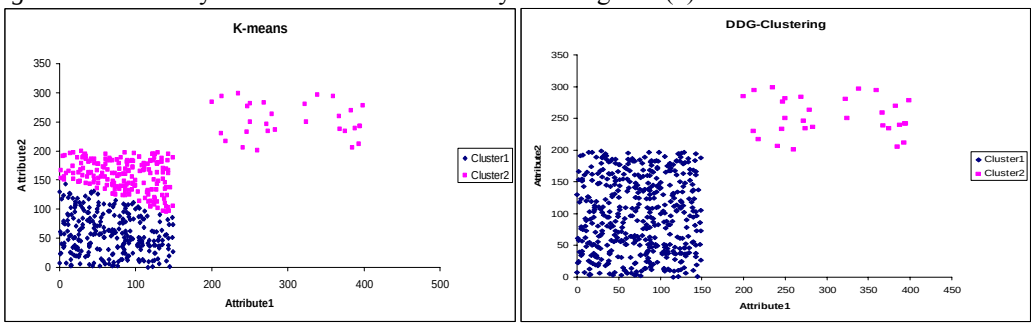
The results shown in Figure 5 illustrate how density and gravitational force could attract points effectively and allocate them to the right class. Although the points between the two clusters are assigned by the k-means to the closest cluster based on distance measure as in Figure 5(a), DDG-Clustering , as in Figure 5(b), assigned them to cluster 2 because it has a very strong gravitational force due to its weight that could attract these points toward it. On the other hand, the global density gain if these points assigned to the big cluster will be increased and therefore the points are assigned correctly to the big cluster.



(a) *K-means clustering* output (b) *DDG-Clustering* clustering output

Figure 5. Comparing *K-means* and *DDG-Clustering* on Synthetic Dataset

Figure 6(a) explains how the initial seeds in *k-means* affected the clustering output, while *DDG-Clustering* could effectively detect the clusters correctly as in Figure 6(b).



(a) *K-means clustering* output (b) *DDG-Clustering* clustering output

Figure 6. Comparing *K-means* and *DDG-Clustering* on Synthetic Dataset

Two well known real datasets from were selected for the experiments: glass and vehicle datasets are used for testing the accuracy of the proposed algorithm. Both datasets are from the UCI machine learning repository, which are available from the web site: http://www.ics.uci.edu/~mlearn/databases/. Main characteristics of these data sets are depicted in Table 1, including number of data samples, number of features (attributes) and number of data classes. Figure 7 shows the stability and fast convergence of *DDG-Clustering* when compared with *K-means*.

Table 1. Real Datasets Characteristics

Data Set name	No. of samples	No. of attributes	No. of classes
Glass	214	9	7
Vehicle	846	18	4

We applied *DDG-Clustering* on both vehicle and glasses datasets for 10 times and compared it with *K-means*. Both algorithms applied on the same seeds, while a new set of seed points was generated for each run. The results showed that *DDG-Clustering* attain higher accuracy in terms of number of points assigned correctly to clusters compared to *K-means*. When the number of clusters increases as in glasses dataset, *DDG-Clustering* algorithm operates more effectively as shown in Figure 8.

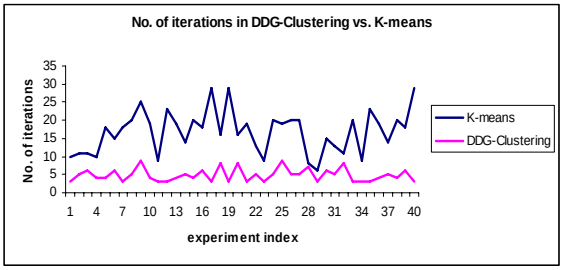
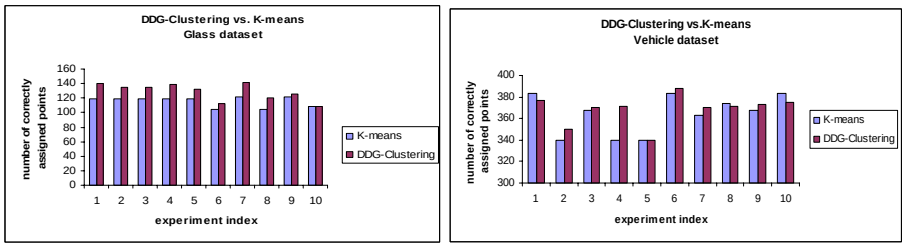


Figure 7. No of Iterations in *DDG-Clustering* vs. *K-means*



(a) glass dataset (b) vehicle dataset

Figure 8. Accuracy of *DDG-Clustering* vs. *K-means*

4. CONCLUSION

In this paper, we have proposed a novel clustering technique, *DDG-Clustering*. The technique is a two-phase algorithm using distance, gravity and density measures to cluster objects effectively. By combining the partition, gravitational and density algorithms, *DDG-Clustering* is able to maintain more accurate clustering results with acceptable running time. A series of experiments conducted on real and synthetic datasets shows the efficiency of *DDG-Clustering* and its advantage over prior clustering method.

REFERENCES

[1]A. K. Jain and R. C. Dubes, , 1988. *Algorithm for Clustering Data*, chapter Clustering Methods and Algorithms. Prentice-Hall Advanced Reference Series.

[2]A.K. Jain, M.N. Murty, and P. Flynn, 1999. *Data Clustering: A Review*, ACM Computing Surveys, vol.31, no. 3, pp. 264-323.

[3]C.R. Lin and M.S. Chen, 2002. A Robust and Efficient Clustering Algorithm Based on Cohesion Self-Merging, *Proceedings of Eighth ACM SIGKDD International Conference of Knowledge Discovery and Data Mining*.

[4]Han, J. and M. Kamber, 2000. *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco.

[5]J. Gomez, D. Dasgupta, and O. Nasraoui, 2003. A New Gravitational Clustering Algorithm, *Proceedings of the SIAM Conference on Data Mining*, San Francisco, CA.

[6]J. Han ,and M. Kamber, 2000. *Data Mining: concepts and techniques* . Morgan Kaufmann.

[7]L.Peng, B. Yang, Y.Chen, Z.Chen,2009. *Data Gravitation Based Classification,* ACM Computing Surveys, Vol. 179, No. 6, pp 809-819.

[8]M. Dash, H. Liu, and X. Xu, 2001. 1+1>2: Merging Distance and Density Based Clustering. *Proceedings of the 7th International Conference on Database Systems for Advanced Applications, Hong Kong*, pp.32-39.

[9]M. Ester, H. P. Kriegel, J. Sander, and X. Xu,1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of International Conference on Knowledge Discovery and Data Mining*. Portland, OR, 1996, pp. 226-231.

[10] M.M.Gaber, A. Zaslavsky, and S.Krishnaswamy, 2004. A Cost-Efficient Model for Ubiquitous Data Stream Mining, *Proceedings of the tenth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Perugia Italy, pp. 747-754.