A New Online Clustering Approach for Data in Arbitrary Shaped Clusters

Richard Hyde, Plamen Angelov
Data Science Group, School of Computing and Communications
Lancaster University
Lancaster, LA1 4WA, UK
Email: r.hyde1@lancaster.ac.uk p.angelov@lancaster.ac.uk

Abstract—

In this paper we demonstrate a new density based clustering technique, CODSAS, for online clustering of streaming data into arbitrary shaped clusters. CODAS is a two stage process using a simple local density to initiate micro-clusters which are then combined into clusters. Memory efficiency is gained by not storing or re-using any data. Computational efficiency is gained by using hyper-spherical micro-clusters to achieve a micro-cluster joining technique that is dimensionally independent for speed. The micro-clusters divide the data space in to sub-spaces with a core region and a non-core region. Core regions which intersect define the clusters. A threshold value is used to identify outlier micro-clusters separately from small clusters of unusual data. The cluster information is fully maintained on-line.

In this paper we compare CODAS with ELM, DEC, Chameleon, DBScan and Denstream and demonstrate that CODAS achieves comparable results but in a fully on-line and dimensionally scale-able manner.

I. Introduction

In modern times we have seen an ever increasing number of situations providing streams of data. The need to make sense of the data in real time and in an adaptable real time time environment requires new techniques in data analysis. Not only are offline methods unsuitable for data streams, storage of the large volumes of data created by these streams is impractical.

Here we address these concerns by evolving the micro-clusters as new data is presented and by removing the need to store the data. The technique is named Clustering Online Data-streams into Arbitrary Shapes (CODAS).

The technique presented here has two main stages. The first creates micro-clusters when the number of data samples within a given initial radius of any data sample reaches a specified value. These values are set by the user and may vary between applications. Unlike many traditional density based clustering techniques which use a fixed radius for the micro-clusters in CODAS we adapt the micro-cluster radius in two ways. Each cluster consists of a centre point, an outer cluster radius and an inner core cluster radius which is a fixed proportion of the outer radius. This value is known as the feather value and using 0.5 allows for a new micro-cluster to be created exactly between two close neighbours that touch, but do not overlap. Data samples are considered to be members of a cluster if they lie within the outer cluster radius. The radius of the cluster is adjusted according to the mean distance of the data samples from the cluster centre. The mean distance is updated recursively so does not require the data samples to be retained. Data samples which have a low local density do not form clusters but remain as outliers.

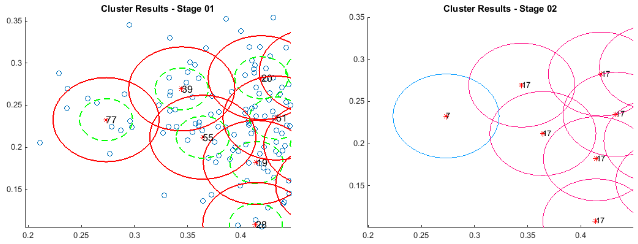
The second stage combines any of these micro-clusters that overlap into global clusters. In this way arbitrary shapes, including traditionally difficult shapes such as concave clusters, can be produced. To simplify the calculations required for joining the micro-clusters they are limited to hyper-spheres. Thus they overlap if the sum of the radii is greater than the distance between the centres and cluster connections are found with computationally efficient logical operations. Data samples which do not have the required local density remain as single outlier samples.

II. State of the Art

Alternative online data stream clustering techniques such as ELM [1], DEC [2] provide real time clustering of data streams. Both of these techniques operate on data streams in real time but are limited to hyper-ellipsoidal cluster shapes. The basis for ELM is to store the local mean as a cluster centre and to adjust the cluster centre and radii as more data arrives. DEC maintains a list of core and non-core clusters defined by the weight of the cluster. The weight decays over time or is increased as new data samples join the cluster. In this way core clusters may decay to non-core, non-core clusters my disappear or increase their weight to become core clusters or new, non-core, clusters may be created. In both techniques the clusters created are hype-ellipsoidal. In the case of convex cluster shapes DEC may create many smaller hyper-ellipsoidal clusters or one large cluster encapsulating all the data.

SPARCL [3], Chameleon [4] and DBScan [5] are all techniques for clustering arbitrary shapes offline. Sparcl utilises a two layer approach whereby k-means [6] clustering is used to create a large number of micro-cluster centres. These micro-cluster centres are then further clustered using a hierarchical approach to join these micro-clusters. Chameleon and DBScan are techniques that successfully cluster arbitrary shapes however both work offline and so require the full data set. An incremental version of DBScan [5] was proposed which allows for incremental modification of the dataset. However

This work was supported by the National Environmental Research Council (Grant number NE/J006262/1)



(a) Core micro-cluster radii

(b) Combined micro-clusters

Fig. 1. Illustration of core micro-cluster regions showing (a) micro-cluster radius in red and, micro-cluster core radius in green (b) micro-clusters combined to the global clusters

after each increment the micro-cluster connections are made or broken according to the changes and so the whole dataset is required to be available for each increment.

A method known as DenStream was proposed in [5]. A set of core- and potential-micro clusters are maintained. Each micro-cluster is created from a stored set of data with a decaying weight. By decaying the data samples those with a weight below a threshold are discarded and the memory requirement is limited somewhat. The technique has an initialisation phase, using DBScan, to create an initial set of micro-clusters. Additionally, while the micro-clusters are maintained in an on-line fashion the process of combining the micro-clusters into final clusters is an off-line approach carried out on demand.

A. CODAS Approach

Traditional clustering techniques for arbitrary shapes designate data samples as 'core' or 'non-core'. However, this requires storage of the data samples and so ever increasing storage capacity which is to be avoided in on-line clustering. CODAS stores only the information related to the micro-clusters and each micro-cluster has a 'core' and 'non-core' region.

The following terminology is used for CODAS:

1) sample: any data point in n dimensions
2) threshold: the minimum number of sample within a micro-cluster radius for it to become a core-micro-cluster
3) non-core-micro-cluster: a micro cluster with local density below the threshold core-micro-cluster: a micro-cluster with a local density above the threshold
4) global cluster: a cluster consisting of a number of intersecting micro-clusters

In general CODAS is a data driven approach to divide the data space in to core and non-core regions. Each micro-cluster consist of a non-core region of radius r_0 and a core region being $0.5r_0$. Any micro-cluster above a given density threshold is considered for global cluster membership. Micro-clusters with no intersections form global clusters. Micro-clusters with core regions that intersect another micro-cluster non-core region form a single, larger global cluster. Non-core regions are considered to be edges of global clusters.

New data from the data stream will fall in to one of 3 regions:

1) empty space where it will form a new, non-core-micro-cluster
2) micro-cluster non-core region where it will be assigned to the cluster, the cluster count updated and the micro-cluster centre recursively updated to the mean of it's samples.
3) micro-cluster core region where it will be assigned to the micro-cluster and the cluster count updated

The micro-cluster that has been modified, or created, by this process is then checked to see if the local density is above the threshold. If it is then it is checked for new intersections with other micro-clusters. If new intersections have been made then all the linked micro-clusters are assigned to the same global cluster. This maintains arbitrarily shaped data space regions of global clusters online.

With this approach at any given time a data sample can be checked for it's global cluster membership, any new sample is immediately clustered and outliers are identified as members of non-core-micro-clusters.

Figure 1 shows a subset of a plot of test data. Figure 1(a) shows the core and non-core radii of the micro-clusters. Data sample without a micro-cluster radii are in non-core micro-clusters which are not displayed. Where the core radius of any cluster intersects a non-core radius of any other the clusters combine as shown in figure 1(b).

III. CODAS Algorithm

A. CODAS Equations

Here we will describe the CODAS algorithm including the variables, equations and pseudo code.

C_i - co-ordinates of micro-cluster centre i
d_i - distances from new sample to micro-cluster centre i
G_i - global cluster of micro-cluster i
I_i - list of micro-clusters intersecting micro-cluster i
m - value of N_i when a non-core micro-cluster to become a core-micro-cluster, user input
N_i - number of samples in each sub-cluster i
n - number of micro-clusters
r_0 - micro-cluster radius, user input
S_n - sample n

$$d_i = \|S_n - C_i\| \tag{1}$$

$$C_j = \frac{(N_i - 1) \times C_j + S_n}{N_i} \tag{2}$$

$$r_j = \sqrt{\alpha r_0^2 + (1-\alpha)\tilde{d}_j} \tag{3}$$

B. CODAS Pseudo-Code

The pseudo-code for the CODAS algorithm is as follows:

```
1) load new data sample
2) calculate $d_i$ for all micro-cluster centres $C_i$
3) if $d_{i(min)} < r_0$
    a) increment $N_i$
    b) if $d_{i(min)} > \frac{r_0}{2}$ update cluster centre using equation 2
4) else create new micro-cluster in it's own global cluster
    a) $C_i = S_n$
    b) $G_i = \max(G) + 1$
5) end if
6) calculate distance $D$ to all micro-cluster centres
7) find new list of intersections $I_{i(new)} = D < 1.5r_0$
8) if the intersection list has changed $I_{i(new)} \neq I_i$
    a) update intersection list $I_i = I_{i(new)}$
    b) update global cluster for all intersecting micro-clusters $G(I_i) = G_i$
9) end if
10) while data stream continues, go to 1
```

IV. Test Data

The algorithm has been tested on 3 artificial sets of data designed to test different extremes of cluster shapes and separations. Plots of each dataset are shown in figure 2. Further we also test the technique on datasets used by offline techniques Chameleon [4] and SPARCL [3] where they are referred to as DS1, DS2 and DS3.

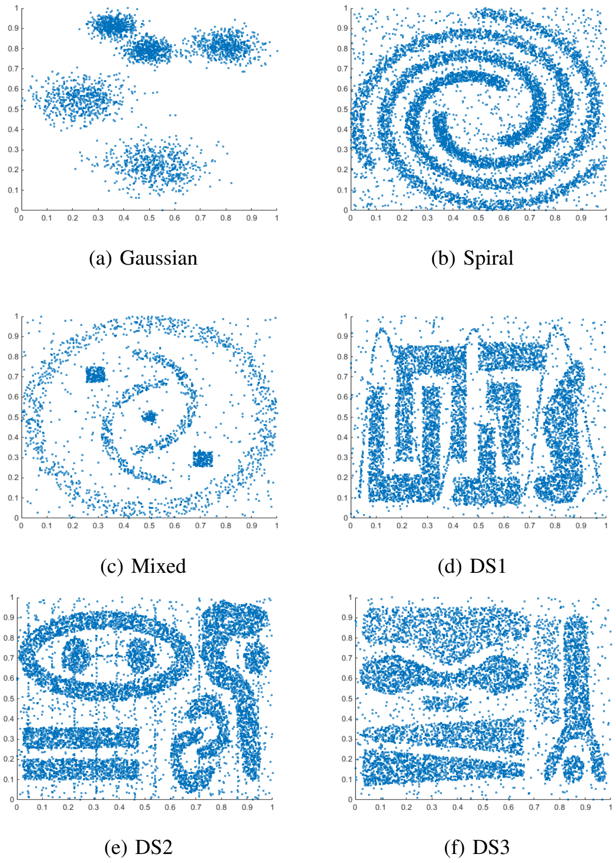
1) Gaussian Clouds Dataset (fig. 2a). This dataset contains 5 clusters with data generated on a gaussian distribution in each.
2) Spiral Dataset (fig. 2b). This data set contains data generated with a random spread about 3 spirals. Each spiral set contains 2,500 samples for a total of 7,500. An additional 1,500 random samples are added for noise.
3) Mixed Dataset (fig. 2c). This dataset test the ability of the dataset to cope with a complex array of cluster shapes and locations. The dataset is constructed of:
    a) A gaussian data cloud in the centre containing 100 samples
    b) Two convex clusters which both surround the central data cloud have intersecting convex hulls containing 150 samples each
    c) 2 rectangular clusters containing 200 samples each
    d) A ring cluster surrounding all of the above data containing 1,000 samples.
    e) an additional 500 random samples in the data space representing noise.
4) DS1 (fig. 2d) 6 natural clusters of 8,000 x 2D samples including 10% noise
5) DS2 (fig. 2e) 9 natural clusters 0f 10,000 samples including 10% noise.
6) DS3 (fig. 2f) 8 natural clusters of 8,000 samples including 10% noise.

V. Methodology

The CODAS algorithm has been implemented in Matlab R14b and the tests run on a PC with an Intel Core i7 processor and 8GB of memory. The code has not been parallelized to take advantage of the processor cores.

CODAS is run across each dataset and the run time recorded together with the mean time per sample. The data is discarded after processing. The results are a set of micro-clusters and their global cluster assignment. To illustrate the accuracy of these clusters we re-analyse the data to check it's global cluster assignment and display the results, coloured by global cluster.

We further test the effect of dimensionality on CODAS. We took the spiral dataset and divided each natural cluster through a varying number of additional dimensions. The original (x,y) dimensions were retained so the results can be projected back on to this plain. In this way all other variables remain as



(a) Gaussian

(b) Spiral

(c) Mixed

(d) DS1

(e) DS2

(f) DS3

Fig. 2. Plots of the datasets used for testing CODAS

TABLE I
CODAS Dimension Test Example

Sub Cluster	x	y	Dim3	Dim4	Dim5	Dim6	Dim7	Dim8
1	9.6447	-3.5968	0.6677	0.3340	0.3332	0.3332	0.3331	0.3333
1	9.6782	-4.1302	0.6674	0.3333	0.3333	0.3337	0.3337	0.3340
2	-1.4015	6.6578	0.3331	0.6673	0.3335	0.3338	0.3339	0.3339
2	-2.1020	6.1781	0.3330	0.6677	0.3339	0.3338	0.3330	0.3337

TABLE II
CODAS Cluster Accuracy Results

Dataset	Accuracy (%) Min	Max	Average	Average Purity (%)	Average Assigned (%)
Gaussian	86.21	88.45	87.83	99.62	87.83
Spiral	87	100	99.33	100	99.33
Mixed	98.3	100	98.9	99.9	99.28

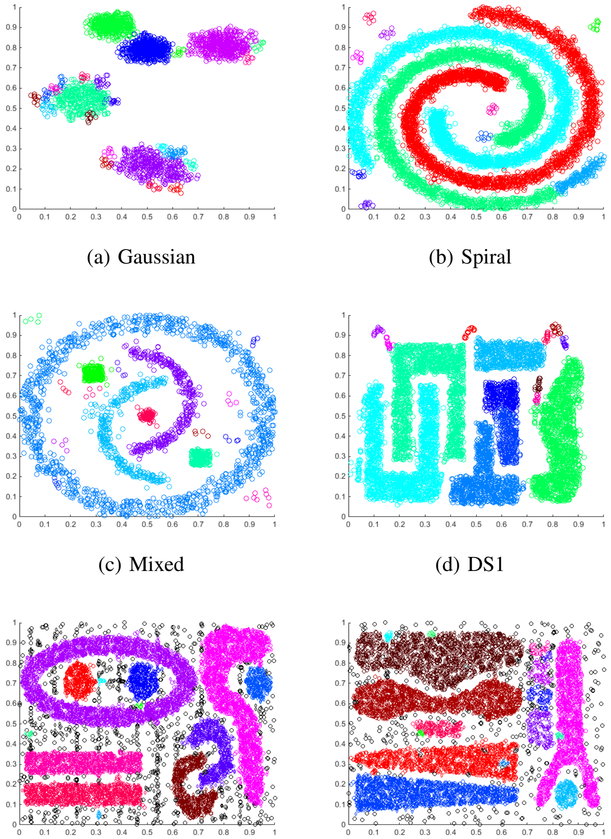
constant as possible and only additional dimensionality is introduced. An example of how the data is generated across these additional dimensions is given in Table I.

VI. Results

A. Cluster Validity

For the first three datasets, spiral, gaussian and mixed we had a priori knowledge of the natural cluster assignment of the data. In this case we could measure the quality of the clusters. We use two measures, average cluster purity and assignment accuracy. Average cluster purity is a widely used metric and is given by:

$$purity = \frac{\sum_{i=1}^{n} \frac{S_i^d}{S_i}}{n} \qquad (4)$$



(a) Gaussian

(b) Spiral

(c) Mixed

(d) DS1

(e) DS2

(f) DS3

Fig. 3. Images for results of the CODAS algorithm on various test datasets

TABLE III
CODAS Speed Test Results

Data Set	Mean Time / Sample (ms) CODAS	ELM	DEC	DBScan
Gaussian	0.461	0.085	0.163	0.048
Spiral	0.474	0.189	1.29	0.12
Mixed	0.421	0.130	~	0.13
DS1	0.488	0.232	~	0.114
DS2	0.476	~	~	0.130
DS3	0.483	~	~	0.103

Where S_i^d is the number of samples in cluster i in dominant cluster d and n is the number of clusters.

A large number of small clusters with high purity can disguise a low number of large clusters with low purity however and so we also measure cluster accuracy, which is the number of samples in a cluster that belong in that cluster and no other and is given by:

$$accuracy = \frac{\sum_{i=1}^{n} \frac{S_i^d}{S_i}}{N_i} \qquad (5)$$

where S_i^d is the number of samples in dominant class d, S_i is the number of samples in cluster i and N_i is the total number of samples. This gives a measure of the likelihood that a samples placed in a cluster is correctly placed.

With any technique that allows samples to be outliers it is possible to create pure and accurate cluster from few samples and so we also measure the percentage of samples that have been assigned as follows:

$$assigned = \frac{\sum_{i=1}^{n} S_i}{N_i} \qquad (6)$$

The CODAS algorithm was run 10 times on each dataset and the maximum, minimum and mean accuracy recorded. The results for these are given in table II.

B. Speed

The speed of CODAS has been compared to ELM, DEC and DBScan. DEC and ELM are both techniques that provide fully maintained cluster information online, however both produces hyper-ellipsoid shaped clusters resulting in split natural clusters as shown in figure 4. DBScan is offline but is capable of producing arbitrary shapes and has been included here to compare the cluster results with CODAS. Denstream requires an offline component and an initialisation which are both based on DBScan so will be related to the DBScan results for multi dimensionality. CODAS has a speed penalty due to the extra calculations required to create the arbitrary shapes. However we can see that it still compares favourably with ELM and outperforms DEC and DBScan at higher dimensions.

C. Dimensionality

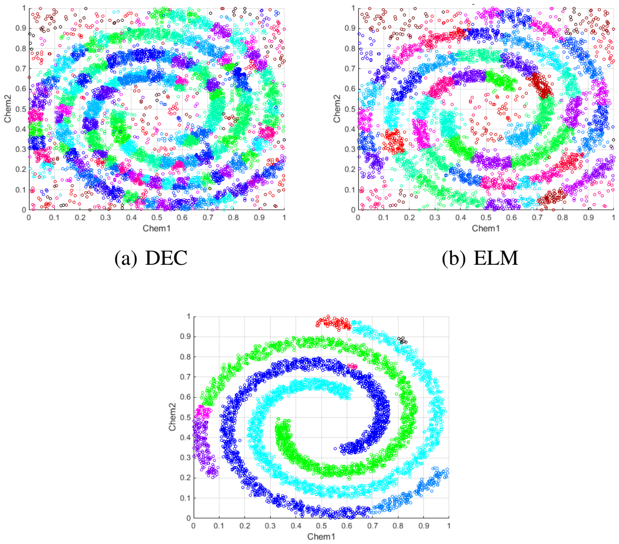
By utilising hyper-spheres for micro-clusters the cluster joining technique is largely dimensionally independent. Micro-clusters are joined if the edges of their hyper-spheres overlap.

TABLE IV
MULTI-DIMENSIONAL SPEED TEST RESULTS

Number of	Mean Run Time (s)			
Dimensions	CODAS	ELM	DEC	DBScan
2	1.3840	0.2558	0.4877	0.1400
5	3.6688	1.2200	7.6866	2.3300
8	3.8411	1.2800	7.8800	3.0500
17	4.0803	1.3533	10.2033	4.9966
32	4.3686	1.8366	14.5333	20.5133
62	4.9401	2.6300	26.2400	54.2966
92	5.4787	3.2733	49.2633	73.9933

This is a simple comparison between the euclidean distance between cluster centres and the sum of the micro-cluster radii. Therefore the only calculation that is dimensionally dependant is the euclidean distance. With each new data sample being assigned to a single micro-cluster we only need to check the intersections for that micro-cluster. In it's current form the radii of the micro-clusters is constant and so we need only compare the euclidean distance between the changed micro-cluster and all others with 2 × r_0.

This was tested by running CODAS along with ELM, DEC and DBScan on the range of multi-dimensional versions of the spiral dataset we described in chapter V, Methodology. The cluster results are projected back on to the (x,y) plane to visually display them. Figure 5(b) shows the cluster results for the data partitioned into 99 dimensions. By retaining the same number of data samples in the same arrangement and using the same parameters we preserve all calculations. Any variation in run-time is therefore explained by the increased complexity of added dimensions alone.

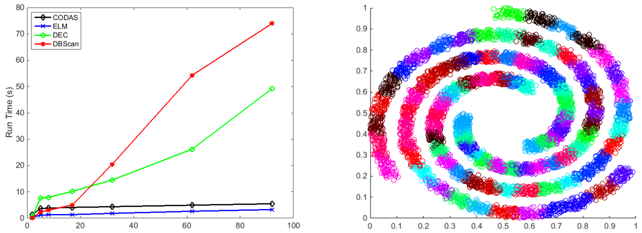


(a) DEC

(b) ELM

(c) DBScan

Fig. 4. Images for results of alternative techniques (a) DEC, (B) ELM, (c) DBScan to the spiral dataset



(a) Time vs Dimensions

(b) 99 Dimension Clusters

Fig. 5. Results for running CODAS, ELM, DEC and DBScan across multiple dimensions (a) Timings vs number of dimensions showing CODAS and ELM (b) projection of cluster in 99 dimensions on to (x,y).

VII. DISCUSSION AND CONCLUSIONS

All of the techniques under discussion here require some user parameters to be optimised. ELM has an initial radius only, DEC has an initial radius together with a γ and β parameter related to the decay and evolving nature of some data streams. DBScan uses an intial radius and minimum number of clusters within that radius to define the local density of each point. CODAS requires an initial radius r_0 and a minimum number of samples m to be within that radius similar to DBScan. These are somewhat intuitive with knowledge of the expected data stream with r_0 and m related to the relative density of the outlier regions to the cluster regions.

CODAS has been developed to manage on-line data streams that do not evolve, i.e. clusters that form will remain and continue to be of interest. Future variants of CODAS will be developed to employ an 'ageing' process for clusters that allow them to die out. CODAS has been shown to reliably cluster data streams into predictable, repeatable clusters of high purity and cluster data within these cluster regions with high accuracy. It is comparably in speed to alternative techniques, order independent and scale-able to multi-dimensional data.

REFERENCES

[1] R. Dutta Baruah and P. Angelov, "Evolving local means method for clustering of streaming data," *IEEE International Conference on Fuzzy Systems*, pp. 10–15, 2012.

[2] R. D. Baruah and P. Angelov, "DEC: Dynamically evolving clustering and its application to structure identification of evolving fuzzy model," *Transaction on Cybernetics*, pp. 1–16, 2013.

[3] V. Chaoji, M. Al Hasan, S. Salem, and M. J. Zaki, "SPARCL: Efficient and effective shape-based clustering," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 93–102, 2008.

[4] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, pp. 68–75, 1999.

[5] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," *. . . Conference on Data Mining*, no. 2, pp. 328–339, 2006.

[6] J. B. MacQueen, "Kmeans Some Methods for classification and Analysis of Multivariate Observations," *5th Berkeley Symposium on Mathematical Statistics and Probability 1967*, vol. 1, no. 233, pp. 281–297, 1967. [Online]. Available: http://projecteuclid.org/euclid.bsmsp/1200512992