3-2015

Density Based Data Clustering

Rayan Albarakati
California State University - San Bernardino, rayanalbarakati@gmail.com

3-2015

Density Based Data Clustering

Rayan Albarakati

DESNITY BASED DATA CLUSTERING

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Rayan Albarakati

March 2015

DESNITY BASED DATA CLUSTERING

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Rayan Albarakati

March 2015

Approved by:

Haiyan Qiao, Advisor, School of Computer Science and Engineering	Date
Owen J.Murphy	
Krestin Voigt	

ABSTRACT

Data clustering is a data analysis technique that groups data based on a measure of similarity. When data is well clustered the similarities between the objects in the same group are high, while the similarities between objects in different groups are low. The data clustering technique is widely applied in a variety of areas such as bioinformatics, image segmentation and market research.

This project conducted an in-depth study on data clustering with focus on density-based clustering methods. The latest density-based (CFSFDP) algorithm is based on the idea that cluster centers are characterized by a higher density than their neighbours and by a relatively larger distance from points with higher densities. This method has been examined, experimented, and improved. These methods (KNN-based, Gaussian Kernel-based and Iterative Gaussian Kernel-based) are applied in this project to improve (CFSFDP) density-based clustering. The methods are applied to four milestone datasets and the results are analyzed and compared.

ACKNOWLEDGEMENTS

I would like to thank God. And thank my advisor Dr. Haiyan Qiao for all her time, knowledge and patience. I would also like to thank my committee members, Dr. Owen J. Murphy and Dr. Krestin Voigt. I would also like to thank my parents for their support and patience.

TABLE OF CONTENTS

Abstract iii

Acknowledgements iv

List of Figures vii

1. INTRODUCTION TO DATA CLUSTERING 1

1.1 Overview 1

1.2 Significance 2

1.3 Grouping of Data Clustering Techniques 3

2. REVIEW OF DENSITY-BASED CLUSTERING 6

2.1 Review of Density-based Clustering Algorithms 7

2.1.1 Density-Based Spatial Clustering Application with Noise (DBSCAN) 7

2.1.2 Density-Based Clustering (DENCLUE) 9

2.2 Datasets 9

3. METHODOLOGY 11

3.1 Clustering by Fast Search and Find of Density Peaks (CFSFDP) . . . 11

3.2 Analysis 14

4. ALGORITHM DESIGN, IMPLEMENTATION AND ANALYSIS 19

4.1 KNN-based CFSFDP 19

4.1.1 Method 19
4.1.2 Results 20
4.2 Gaussian Kernel-based CFSFDP 22
4.2.1 Method 22
4.2.2 Results 23
4.3 Iterative Gaussian Kernel-based CFSFDP 27
4.3.1 Method 27
4.3.2 Results 28
4.4 Comparisons and Analysis 32

5. CONCLUSIONS 34

References 36

LIST OF FIGURES

2.1 Schematic representation of density connectivity. In blue the core points, in green the points on the boundaries, in red the rest of the points. 8

2.2 Dataset chosen to test different algorithms for data clustering A- Aggregation, B-Jain, C- Flame, and D- Spiral. 10

3.1 ρ - Schematic representation of how ρ_i is calculated. In red there is the point with the highest local density. δ - Schematic representation of how δ_i is calculated. In red there is the point with the highest local density so δ_{red} is obtained considering its farthest point. For the blue point, with lower density, δ_{blue} is calculated as the minimum distance between that point and any other point with a higher local density. . 12

3.2 **A** - Data set distribution. The points are numbered starting from the highest to the lowest local density. **B** - Density decision graph for the dataset on the left. The data points with higher δ and ρ are selected as centers of the clusters (points 1 and 10). The points with high ρ and low δ belong to the existing clusters. Points 7, 8 and 9 belong to the cluster centered in 1 (red). Points 13, 15 and 22 belong to the cluster centered in 10 (blue). The data points with low ρ and high δ are considered outliers (black). 13

3.3 Rectangle selected by the user to define $\rho_{minimum}$ and $\delta_{minimum}$. . . . 14

3.4 Aggregation dataset, CFSFDP method applied. 15

3.5 Flame dataset, CFSFDP method applied. 16
3.6 Jain dataset, CFSFDP method applied. 17
3.7 Spiral dataset, CFSFDP method applied. 18

4.1 Aggregation dataset, KNN method applied. 20
4.2 Flame dataset, KNN method applied. 21
4.3 Jain dataset, KNN method applied. 21
4.4 Spiral dataset, KNN method applied. 22
4.5 Aggregation dataset, Gaussian Kernel method applied. 23
4.6 Flame dataset, Gaussian Kernel method applied. 24
4.7 Jain dataset, Gaussian Kernel method applied. 25
4.8 Spiral dataset, Gaussian Kernel method applied. 26
4.9 Aggregation dataset, Iterative Gaussian Kernel method applied. . . . 28
4.10 Flame dataset, Iterative Gaussian Kernel method applied. 29
4.11 Jain dataset, Iterative Gaussian Kernel method applied. 30
4.12 Spiral dataset, Iterative Gaussian Kernel method applied. 31

1. INTRODUCTION TO DATA CLUSTERING

1.1 Overview

Clustering analysis is aimed at classifying objects into categories on the basis of their similarity, and, nowadays, it is a technique used in many different fields such as bioinformatics, image segmentation and market research [1]. The goal of data clustering is to find groups of similar objects in a dataset while keeping them separated from the noisy points (outliers). Data objects that are classified in the same group should display similar properties based on particular criteria because with such classification information, it is possible to infer the properties of a specific object based on the category to which it belongs [2].

A summary of the goals of cluster analysis has been done by Aldenderfer and Blashfield (1984) [3]:

- Development of a classification
- Investigation of useful conceptual schemes for grouping entities
- Hypothesis generation through data exploration
- Hypothesis testing or the attempt to determine if types defined through other procedures are in fact present in a data set.

Cluster analysis is a useful tool in large scale data analysis providing a compressed

representation of the data. Even if several different clustering strategies have been proposed and many different algorithms have been published to group similar data, cluster analysis determine which criterion is the best solution in general [2].

1.2 Significance

Data clustering became an important problem solving technique because it can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them [17].

Cluster analysis itself does not use one specific algorithm, for a general task but rather various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space and intervals or particular statistical distributions [16].

Data Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, which is a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis is considered an iterative process of knowledge discovery and interactive multi-objective optimization that involves trial and failure [17].

Clustering has been applied in a wide variety of fields [16]:

1. Biology. One example of how clustering can be used is human genetic clustering the similarity of genetic data to infer population structures.

2. Computer sciences. One example of clustering is image segmentation to divide

a digital image into distinct regions for border detection or object recognition.

3. Life and medical sciences. One example of how clustering can be used is to analyze patterns of antibiotic resistance and to classify antimicrobial compounds according to their mechanism of action.

4. Astronomy and earth sciences. One example of how clustering can be used is to reconstruct missing bottom hole core data or missing log curves in order to evaluate reservoir properties.

5. Social sciences. One example of how clustering can used is to identify areas where there are greater incidences of particular types of crime.

6. Economics. One example of how clustering can be used is to group all the shopping items available on the web into a set of unique products; all the items in eBay can be grouped into unique products.

1.3 Grouping of Data Clustering Techniques

The definition of "cluster" is groups containing data objects that are similar to each other, while data objects in different clusters are not. It is very important to figure out how to measure the closeness (similarity) or the distance (dissimilarity) between a pair of objects, an object and a cluster, or a pair of clusters. The choice of how to measure the proximity effects the formations of the resulting clusters, so the selection of an appropriate proximity function is important. Unfortunately cluster analysis is a subjective and problem-dependent process, so there is not a unique way to define which is the best approach to measure the proximity and the features defining a

cluster.

Clustering algorithms are generally classified into the following groups:[2], [4]

- K-means and K-medoids methods: suitable for convex clusters. Clusters are usually formed by data close to each other and there is an objective function that is optimized until the best cluster center is found. It seeks to cluster data points by assigning each sample to the nearest mean, updating the cluster means and re-assigning until the mean values no longer change or there are no re-allocations. k represent the number of clusters and needs to be pre-defined and fixed. This method is susceptible to differing results according to the initial samples chosen [5].

- Distribution based algorithms: the dataset is described as a mix of predefined probability distribution functions. It is a parametric approach where the unknown density is assumed to belong to some parametric family like a Gaussian distribution.

- Density-based cluster methods: they are non-shaped based methods and they focus on the local density of data points. Density-Based Spatial clustering Application with Noise (DBSCAN) is one case. A density threshold is chosen to be discarded as the noise and the cluster is defined as a set of points that converge to the same local maximum of the density distribution function. Density-Based Clustering (DENCLUE) uses a density estimator to define the clusters. The (DENCLUE) algorithm employs a cluster model based on kernel density estimation. A cluster is defined by a local maximum of the estimated density function. Data points are assigned to clusters by hill climbing, i.e. points going to the same

local maximum are put into the same cluster.A clustering in the (DENCLUE) is defined by the local maxims of the estimated density function. A hill-climbing procedure is started for each data instance, which assigns the instance to a local maximum. A disadvantage of (DENCLUE) is that the used hill climbing may make unnecessary small steps in the beginning, and this method never converges exactly to the maximum, it just comes close.

2. REVIEW OF DENSITY-BASED CLUSTERING

As previously mentioned, density-based clustering is an approach where the clusters are defined by their density. It is a nonparametric approach: the number of clusters is not needed input and no assumptions are made about the density distribution. This method does not depend on the shape distribution of the data set.

Two user inputs are required: the minimum number of samples within a radius for a sample to be considered dense and the radius itself. The clusters selected can be imagined as resulting from a cut through a certain density level: each cut leads to separate regions where the density is higher than the cut value. Each of these regions corresponds to a cluster containing all the data points falling into this region. If the level is chosen too low, different clusters will be merged to a single cluster. If the density level is chosen too high, clusters exhibiting a lower density will be lost. The choice of the cut off value chosen has a key role in density-based clustering.

This clustering technique is often used to find cluster data of points formed by natural structures such as roads, volcanos and rivers. When used for "natural" application they are called "natural clusters" [4].

A general formalization of a density-based cluster was proposed by Hartigan [6]. Given a density $\rho(x)$ at each point x, a density threshold λ and a specific link for pairs of objects, a density-contour cluster at level λ is defined as a maximally connected

set of points x_i such that $\rho(x_i) > \lambda$. The links between points may be specified using a specific distance function.

The basic assumption is that the dataset is a sample from some unknown probability density and clusters are high-density areas of this dataset distribution. In order to fulfill this assumption, it is necessary to know:

1. A local density estimate at each point.

2. A notion of connection between objects. Generally, points are connected if they are within a certain distance d_c from each other.

Clusters are constructed as sets of objects that are connected to objects whose density exceeds some threshold λ. Objects that are not part of such clusters are called noise or outliers.

There are various density-based methods proposed in the literature [7], [8]. In the next chapters some density-based cluster methods will be described. Their main differences involve the local density estimation, the algorithm used for finding connected components and the notion of connectivity used.

2.1 Review of Density-based Clustering Algorithms

2.1.1 Density-Based Spatial Clustering Application with Noise (DBSCAN)

The density based spatial clustering of applications with noise (DBSCAN) algorithm allows the use of index structures for density estimation. Given a distance threshold d_c and a density threshold $minPts$, the density of a point x_i is defined as the number of points that are within a radius d_c around x_i. The point x_i is considered a core point

if $minPts_i$ is bigger then $minPts$. Two points are considered directly connected if they have a distance of less than d_c. Two points are density connected if they are connected to core points and these core points are density connected [9].

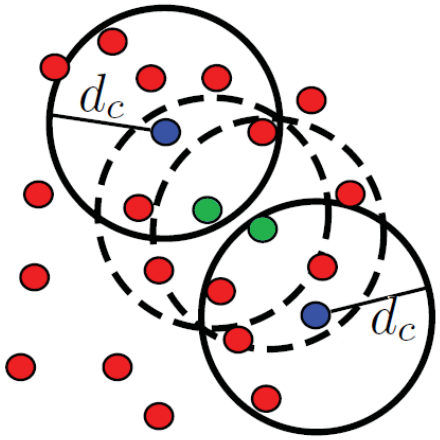

Fig. 2.1: Schematic representation of density connectivity. In blue the core points, in green the points on the boundaries, in red the rest of the points.

Clusters can also contain border points that do not have the core point property. Objects that are not part of a cluster are noise points. In an extreme case,a cluster can only contain one core point.

DBSCAN computes to start a new cluster C with a not yet assigned core point x by assigning all points to C that are connected to x. Each point is "scanned" with a radius d_c to determine the connected points that have to be added to C. For a dataset of N elements, the complexity of this operation is O(N) when applying a sequential search, resulting in a total runtime complexity of $O(N^2)$ [2].

2.1.2 Density-Based Clustering (DENCLUE)

Density-based Clustering (DENCLUE) uses a kernel density estimator to define density-based clusters. Any density estimator can be used. For instance, a well known Gaussian kernel can be chosen or a more complex SquareWave (uniform) kernel.

A local maximum of the density function is called a **density attractor**. Each point x is associated with the density attractor located in the direction of maximum increase in density from x. A density-based cluster is defined as a connected component of density attractors with their associated points: their density estimate has to be above the given threshold λ [9].

The implementation of DENCLUE relies on a Gaussian kernel and a sophisticated data structure for fast local density estimation.

2.2 Datasets

Fig.2.2 show the four different datasets that have been tested:

- Aggregation: This dataset was used in an aggregation clustering problem. [11]
- Jain: This dataset is used in genetic algorithm. [12]
- Flame: This dataset is used on Microarray data analysis. [13]
- Spiral: This dataset is used for spectral clustering and path-based clustering. [14]

The software MATLAB has been used to analyze the data. Different d_c have been applied in order to compare the different results obtained.

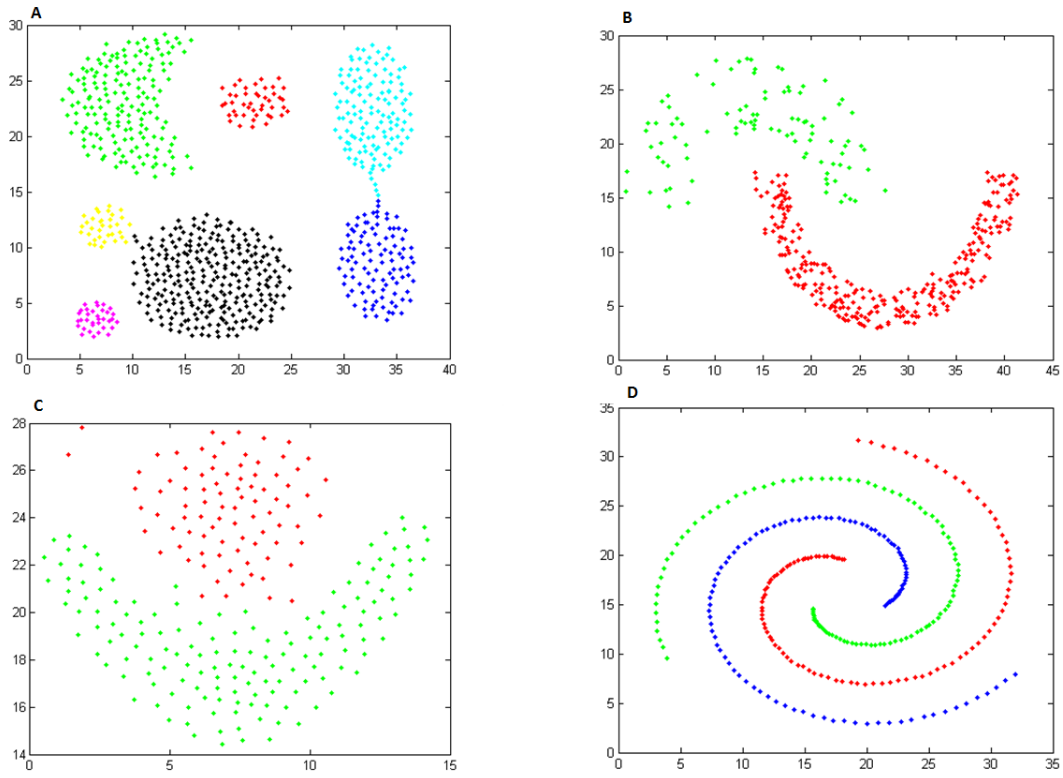

Fig. 2.2: Dataset chosen to test different algorithms for data clustering A- Aggregation, B-Jain, C- Flame, and D- Spiral.

3. METHODOLOGY

3.1 Clustering by Fast Search and Find of Density Peaks (CFSFDP)

In the introduction, several different clustering strategies have been mentioned. Here an alternative approach is proposed [1].

Starting from the distance between data points, the Clustering by Fast Search and Find of Density Peaks (CFSFDP) is able to detect nonspherical clusters and to automatically find the correct number of clusters. The local maxima in density of data points is the parameter used to find the cluster centers. This method is based on the assumption that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density.

For each point i in the data set, two quantities that depend only on the distances between data points, are calculated:

- ρ_i, local density.
- δ_i, distance from points of higher density.

Given d_{ij} distance between point i and j, d_c cut off distance, n total number of point, the local density ρ_i is defined as:

$$\rho_i = \sum_{j=1}^{n-1} \chi(d_{ij} - d_c)$$

$$\chi(x) = \begin{cases} 1 \text{ if } x < 0 \\ 0 \text{ otherwise} \end{cases}$$

ρ_i counts the number of points that are within the d_c to the point i.

The distance from points of higher density δ_i is defined as:

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij})$$

$$\delta_i = \max_j(d_{ij}) \text{ for points with highest density}$$

δ_i is calculated as the minimum distance between the point i and the point with higher density. For the points with highest density a different definition is used in such a way that δ_i is much larger than the typical nearest neighbor distance. The cluster center is recognized as the points with the largest δ_i.

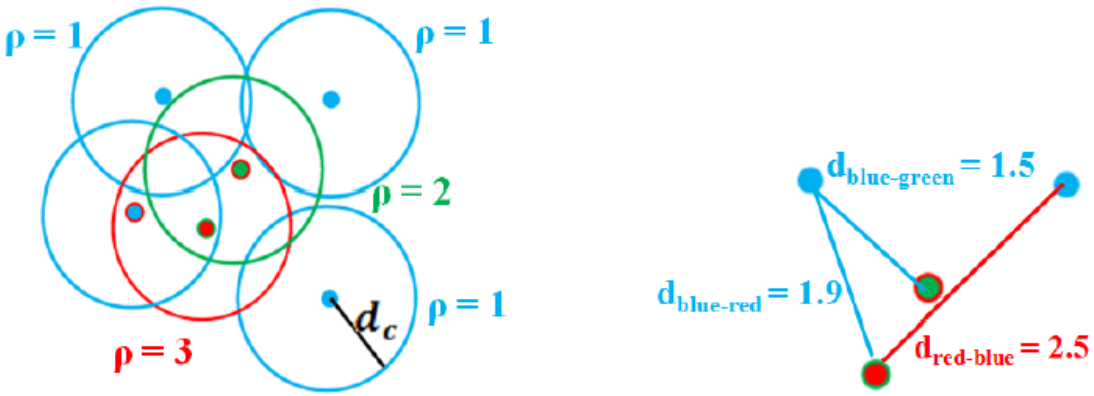

Fig. 3.1: ρ - Schematic representation of how ρ_i is calculated. In red there is the point with the highest local density. δ - Schematic representation of how δ_i is calculated. In red there is the point with the highest local density so δ_{red} is obtained considering its farthest point. For the blue point, with lower density, δ_{blue} is calculated as the minimum distance between that point and any other point with a higher local density.

From these two definitions, it is clear how this method depends strongly on the cut off distance chosen, but it make it possible to distinguish the cluster from the higher value of both ρ and δ. Fig(3.2) shows a dataset of points and their decision graph, δ VS ρ. From the decision graph it is easy to distinguish which points are the centers of the cluster because of their higher δ and ρ [1].

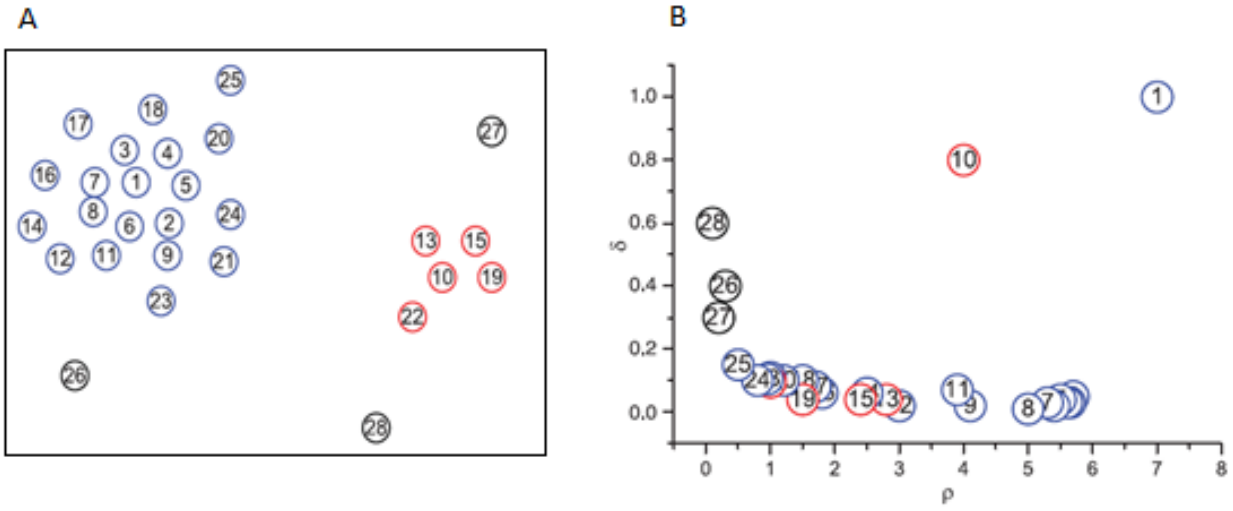

Fig. 3.2: **A** - Data set distribution. The points are numbered starting from the highest to the lowest local density. **B** - Density decision graph for the dataset on the left. The data points with higher δ and ρ are selected as centers of the clusters (points 1 and 10). The points with high ρ and low δ belong to the existing clusters. Points 7, 8 and 9 belong to the cluster centered in 1 (red). Points 13, 15 and 22 belong to the cluster centered in 10 (blue). The data points with low ρ and high δ are considered outliers (black).

3.2 Analysis

It has been proven [1] that the optimal choice for d_c is such that the average number of neighbors is around 1-2% of the total number of points in the dataset. For small datasets, ρ_i can be affected by a large statistical error.

When finding the cluster using MATLAB, the user has an important role. In fact a rectangle has to be selected to determine $\rho_{minimum}$ and $\delta_{minimum}$ (Fig. 3.3). Depending on the selection made, the clusters found have different sizes.

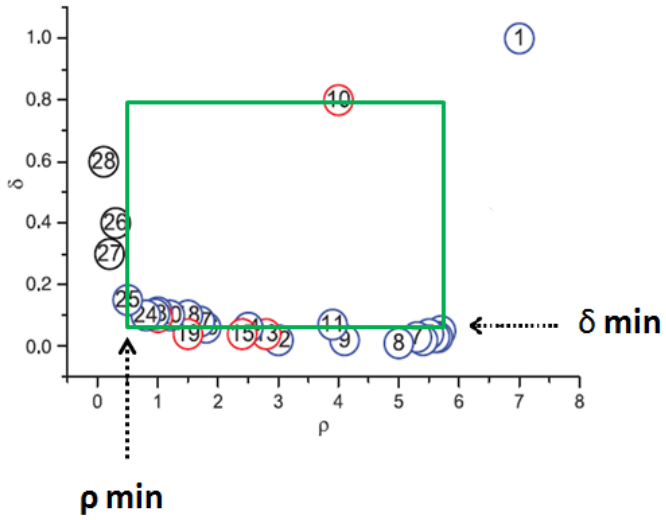

Fig. 3.3: Rectangle selected by the user to define $\rho_{minimum}$ and $\delta_{minimum}$.

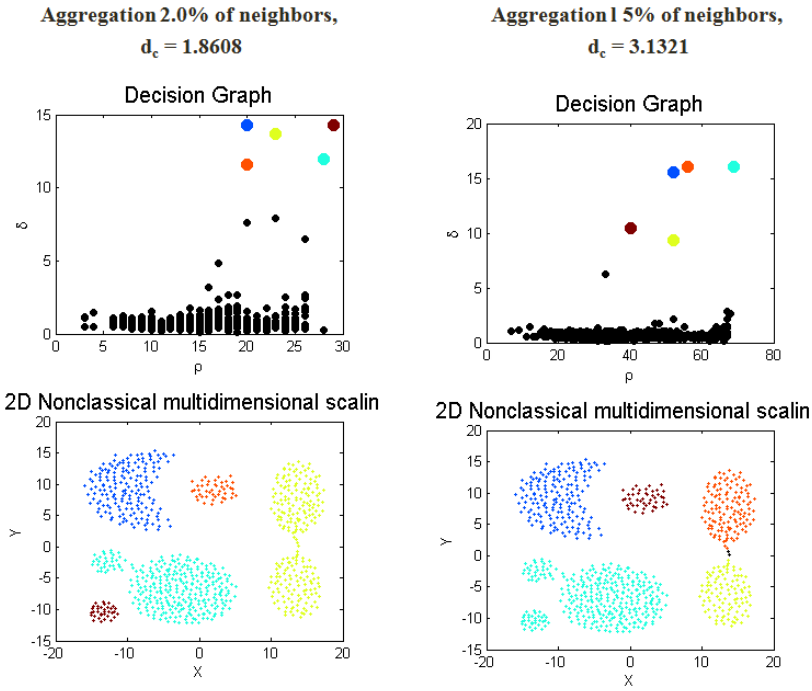

Fig. 3.4: Aggregation dataset, CFSFDP method applied.

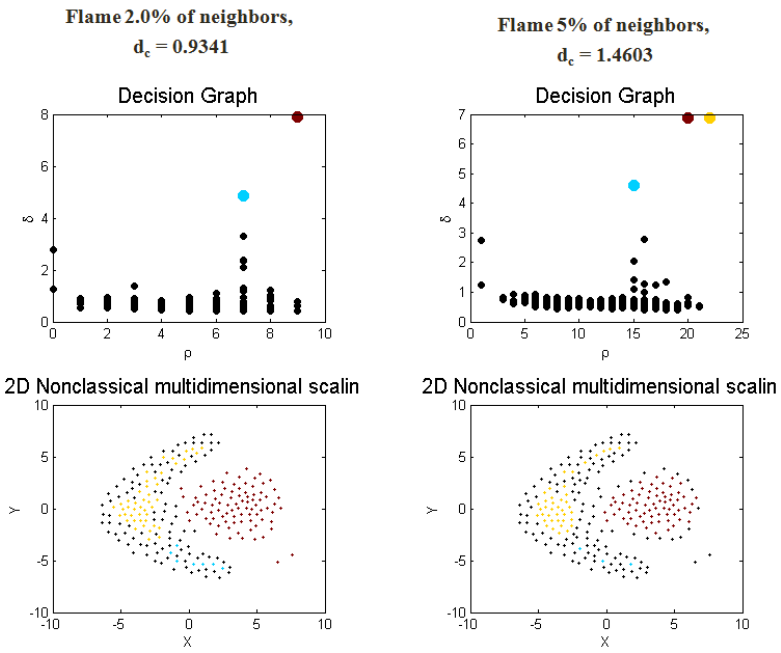

Fig. 3.5: Flame dataset, CFSFDP method applied.

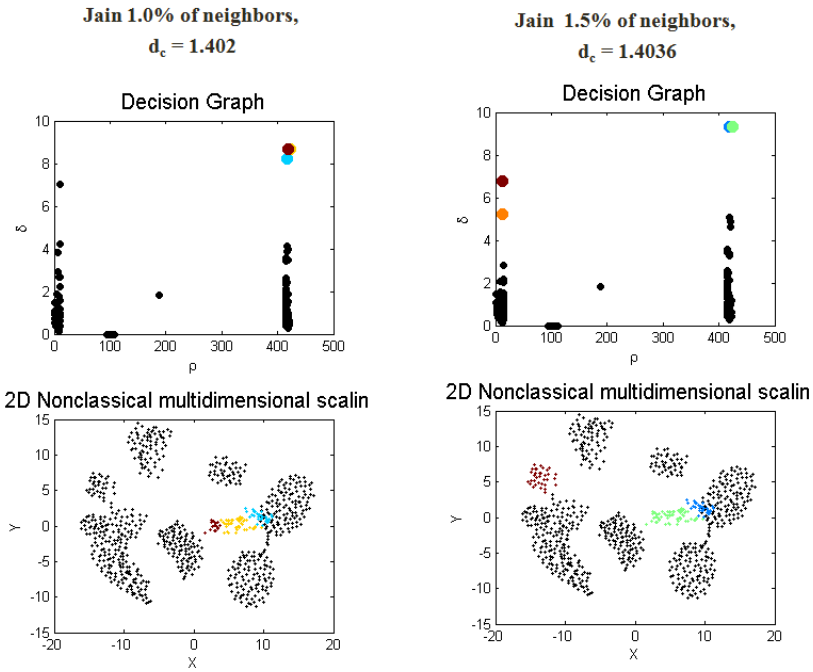

Fig. 3.6: Jain dataset, CFSFDP method applied.

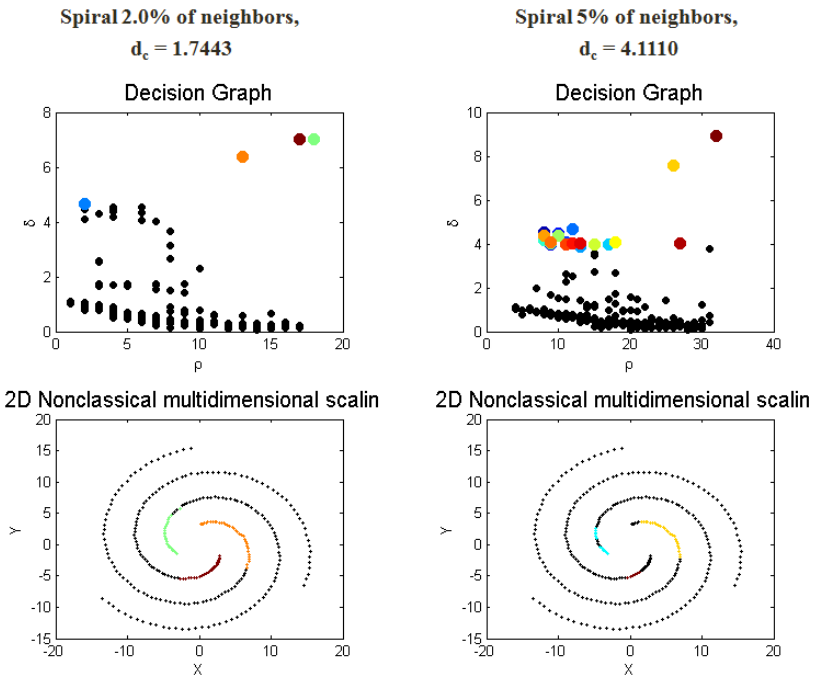

Fig. 3.7: Spiral dataset, CFSFDP method applied.

4. ALGORITHM DESIGN, IMPLEMENTATION AND ANALYSIS

The CFSFDP algorithm has been implemented and modified using the tool MATLAB. The data sets previously described have been implemented and different results have been compared.

The code has been modified because several different algorithms have been applied. The CFSFDP idea is kept but a few different modifications have been applied in order to improve the results obtained.

4.1 KNN-based CFSFDP

4.1.1 Method

A possible solution to the problem of the dependency of the CFSFDP on d_c is described by Mr. Ryan Segher in [10].

The definition or ρ_i changes: instead of counting the number of neighbors within a hard cutoff distance threshold, ρ_i is defined as the mean distance to the nearest M neighbors:

$$\rho_i = \frac{\sum_{j=1}^{M}(d_{ij})}{M}$$

M is a percentage of the total number of points N in the data set, in particular, given a ratio m

$$M = m\_ratio * N$$

M is a percentage of the total number of points and (N) is the total number of points.

4.1.2 Results

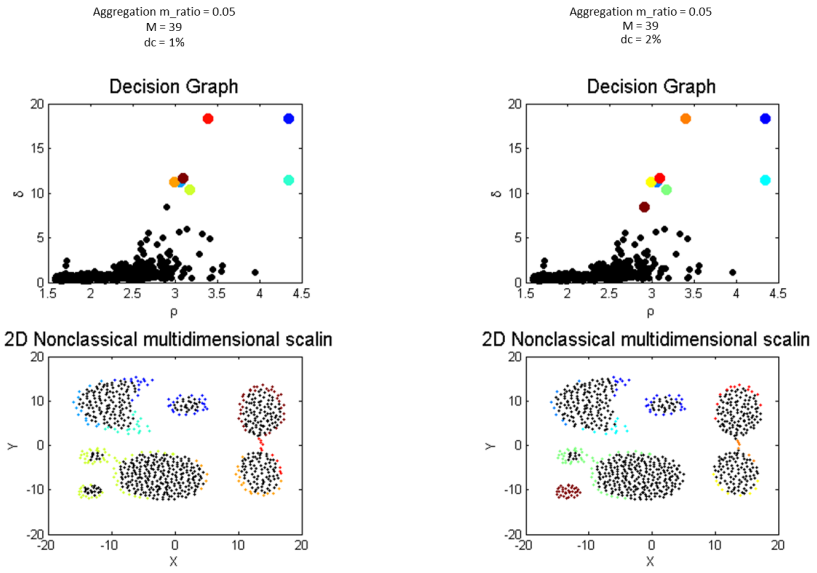

Fig. 4.1: Aggregation dataset, KNN method applied.

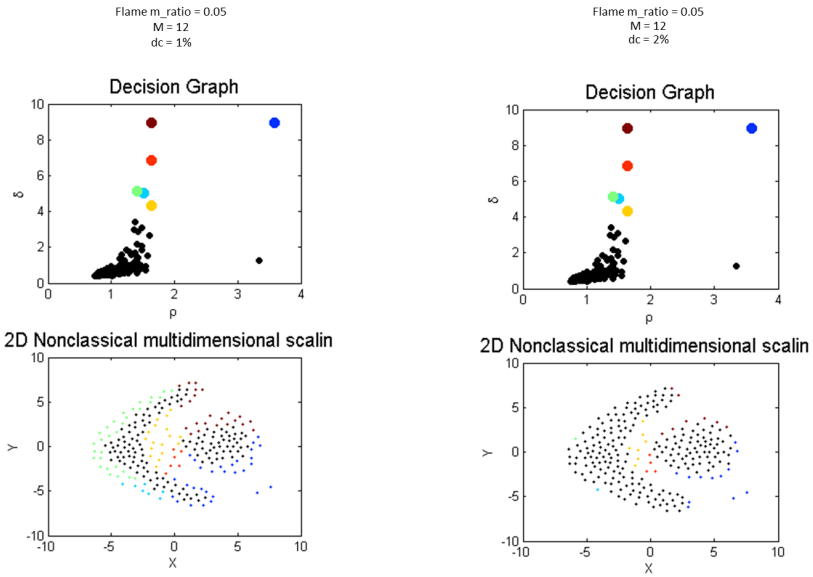

Fig. 4.2: Flame dataset, KNN method applied.

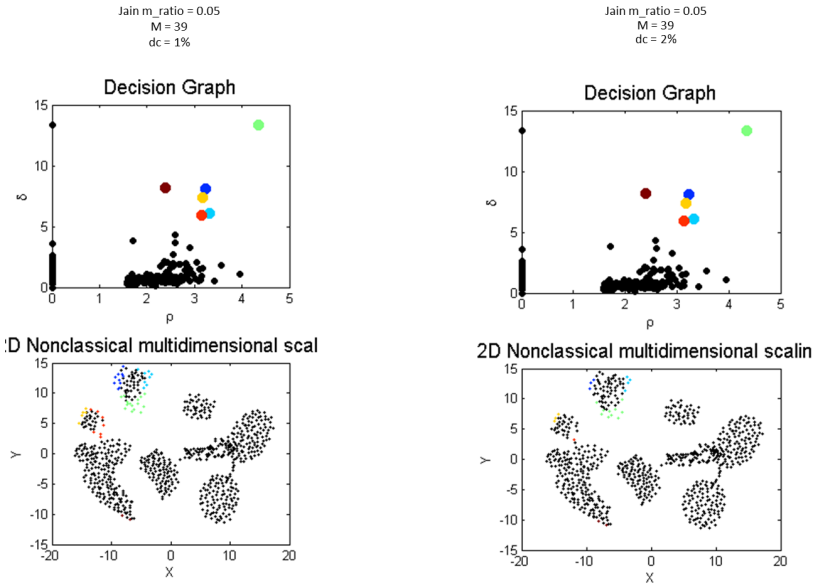

Fig. 4.3: Jain dataset, KNN method applied.

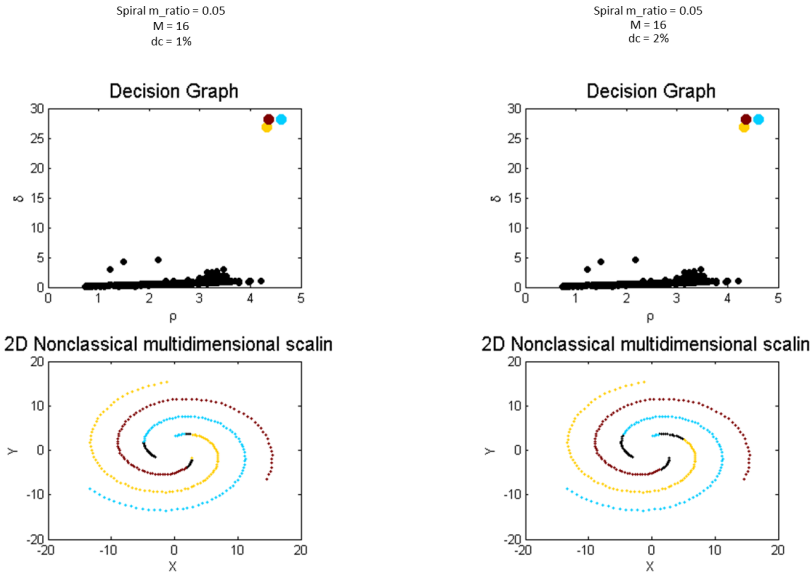

Fig. 4.4: Spiral dataset, KNN method applied.

4.2 Gaussian Kernel-based CFSFDP

4.2.1 Method

A kernel density estimation has been chosen in order to infer a probability density at each point in the dataset according to a specific distribution, called *influence function*. The influence function describes the impact of a data point within its neighborhood, or estimates the probability density at each point in the data set.

DBSCAN methods are a special case of kernel density estimation. Kernel density estimation aims to find the denser regions of points exactly like clustering.

This new approach does not define the local density as the number of points within the cut off value, but uses a Gaussian kernel to determine the ρ_i values. Moreover d_c corresponds to the standard deviation of Gaussian Kernel so it describes how much

the data is dispersed.

The density for point i, i.e. ρ_i, can be calculated as the sum of the influence function of point i for all data points. The density attractors are local maximals of the overall density function.

$$\rho_i = \sum_{j=1}^{n-1} e^{(\frac{d_{ij}}{d_c})^2}$$

This method can be classified among the DENCLUE algorithms. ρ is calculated for each neighbor. The use of the Gaussian distribution weighs the contribution of every neighbor: if a point is closer, then its contribution is bigger. The standard deviation is represented by d_c.

4.2.2 Results

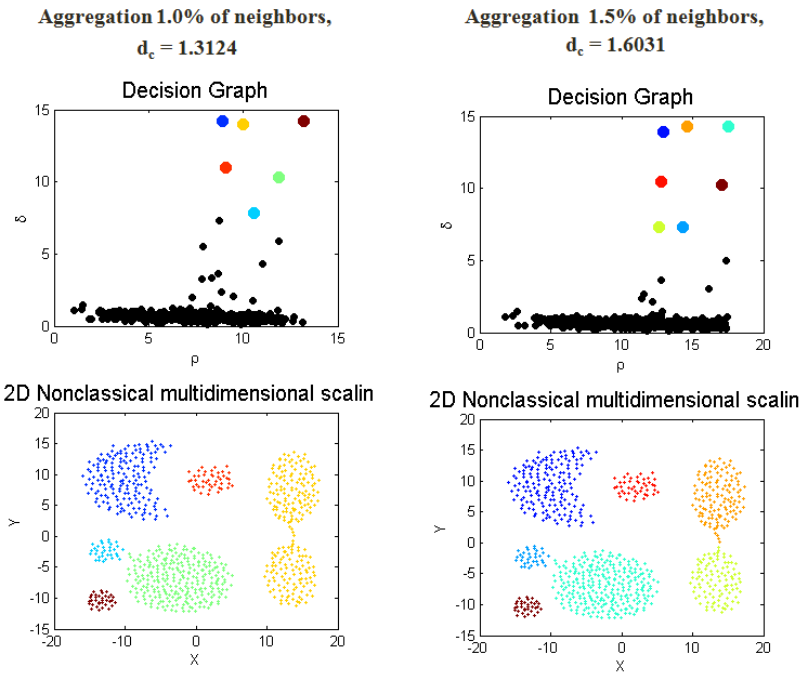

Fig. 4.5: Aggregation dataset, Gaussian Kernel method applied.

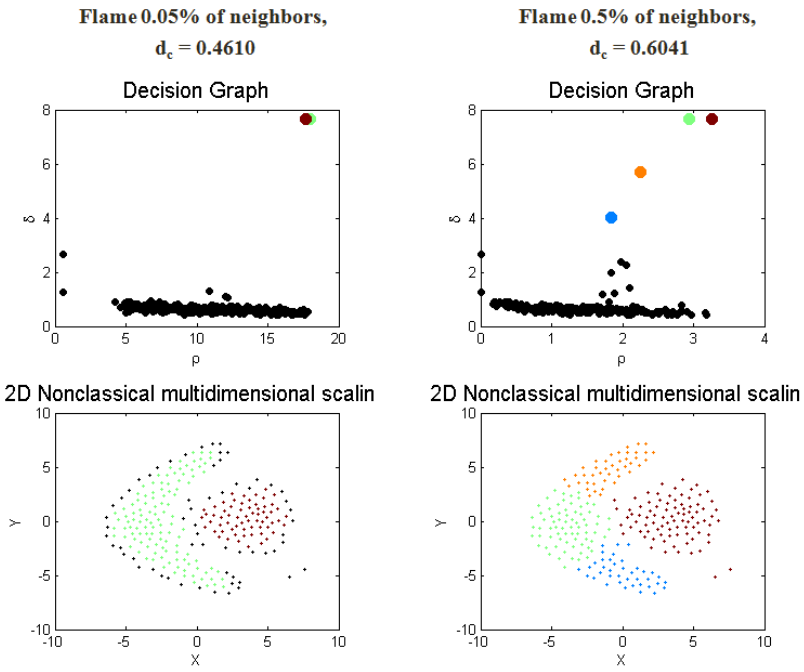

Fig. 4.6: Flame dataset, Gaussian Kernel method applied.

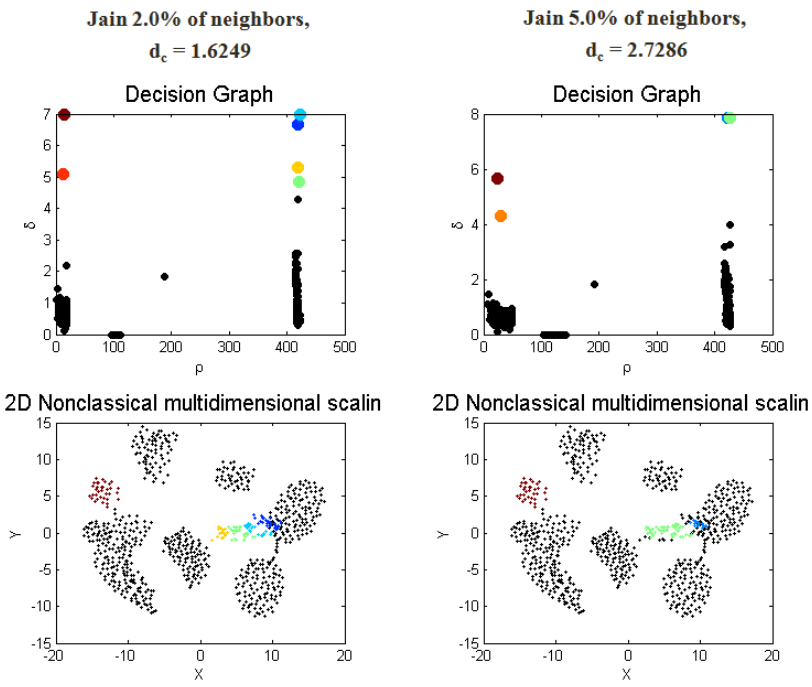

Fig. 4.7: Jain dataset, Gaussian Kernel method applied.

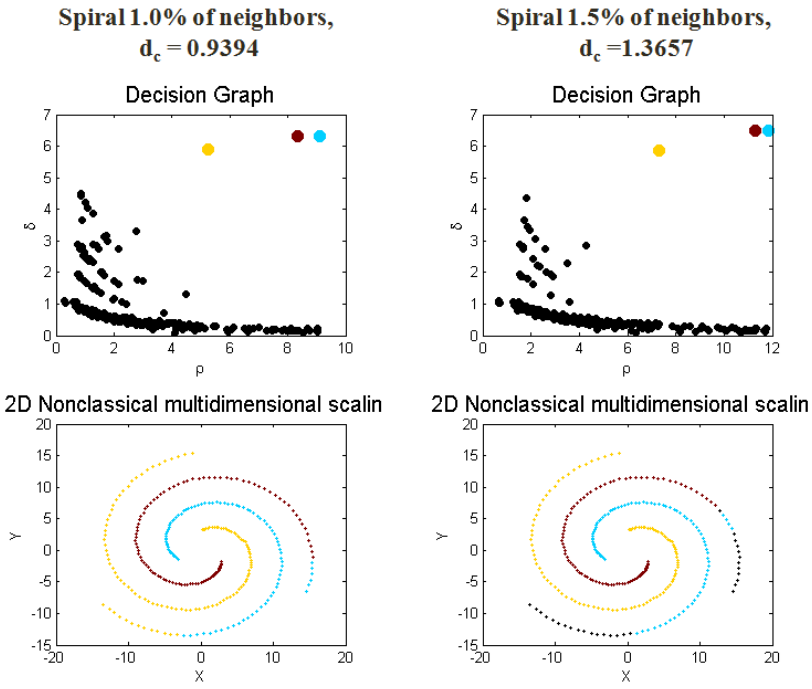

Fig. 4.8: Spiral dataset, Gaussian Kernel method applied.

4.3 Iterative Gaussian Kernel-based CFSFDP

4.3.1 Method

A more in depth analysis has been performed on the kernel density estimation. The results obtained were particularly promising, so an iterative approach has been combined with the statistical approach in order to improve the results.

In the previous section, the concept of kernel density estimation has been introduced. In this section, a gradient-based optimization can be used to find the peaks: regions with densities above a given threshold.

- x^* is called a density attractor if it is a local maximum of the provability density function (previously defined).
- The density gradient is used to find the density attractors, and it is calculated as the derivative of the probability density function: $\nabla f^*(X)$.
- It has been proven that x is density attracted to another point x* (which belongs to its cluster) if $xt + 1 = x_t + \delta \cdot \nabla f^*(x_t)$ [15].
- δ is the step size chosen, in our case it is 1.

The new influence function chosen is obtained from the derivation of the Gaussian definition of ρ described in the previous section [15].

$$\nabla \rho_i = \sum_{j=1}^{n-1} d_{i,j} e^{(\frac{d_{ij}}{d_c})^2}$$

ρ_i takes into account the distances between the point i and all the other points, but during each iteration it is looking for attractors so it converges to 0.

4.3.2 *Results*

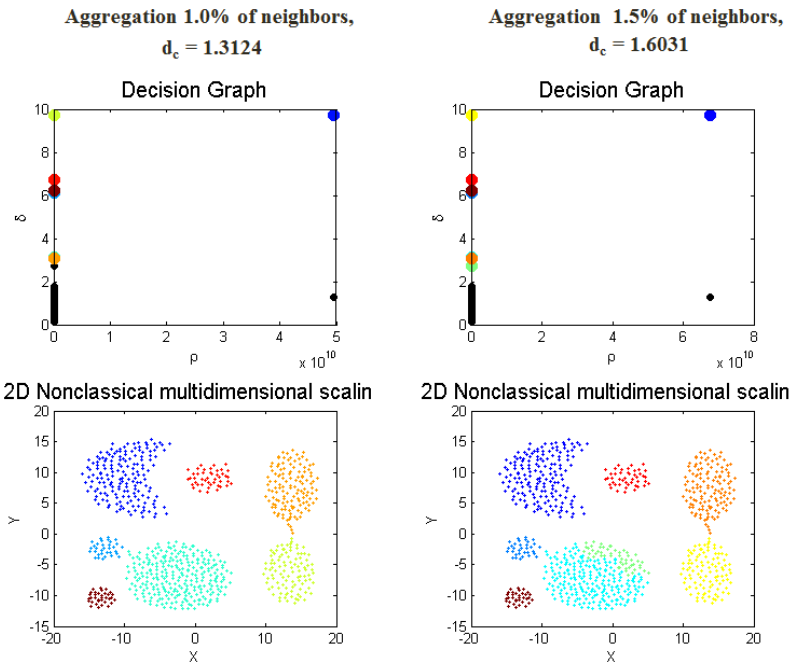

Fig. 4.9: Aggregation dataset, Iterative Gaussian Kernel method applied.

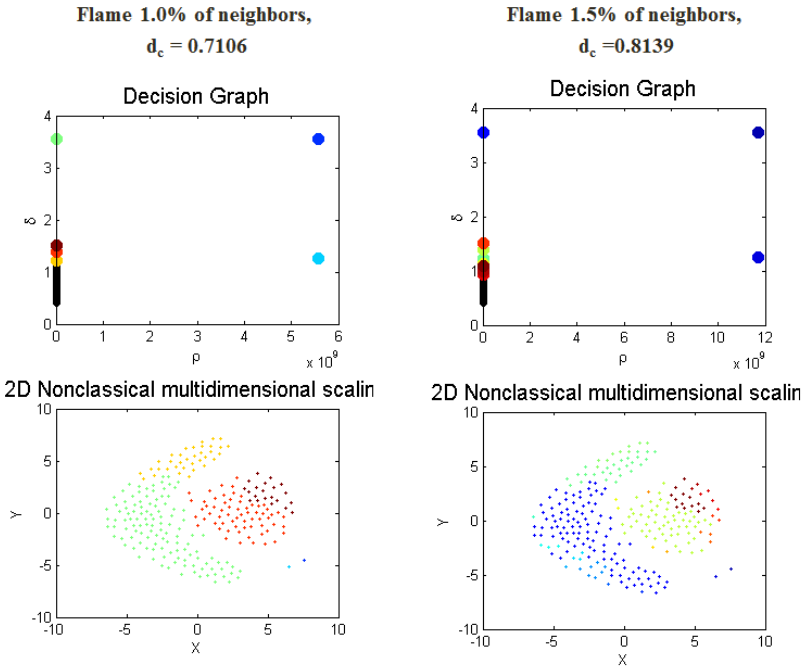

Fig. 4.10: Flame dataset, Iterative Gaussian Kernel method applied.

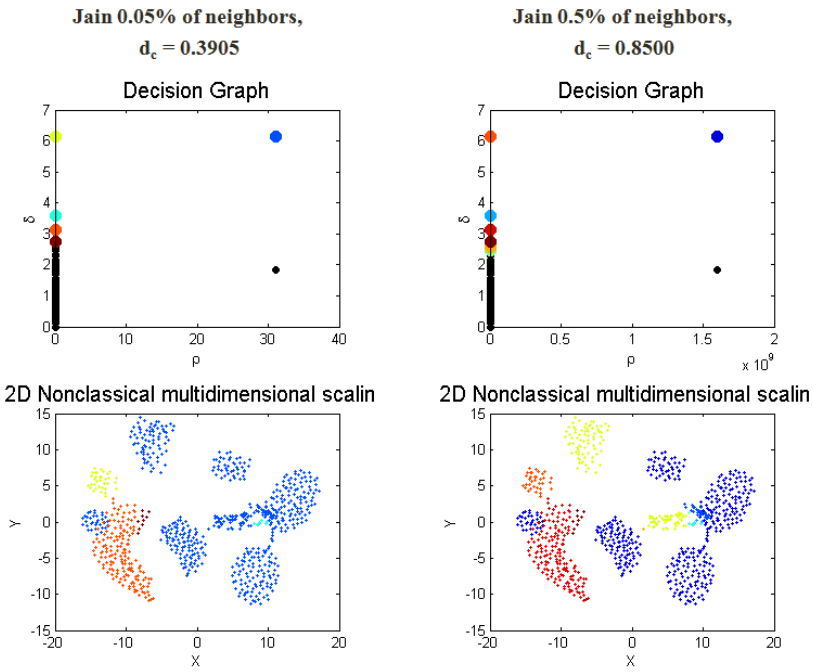

Fig. 4.11: Jain dataset, Iterative Gaussian Kernel method applied.

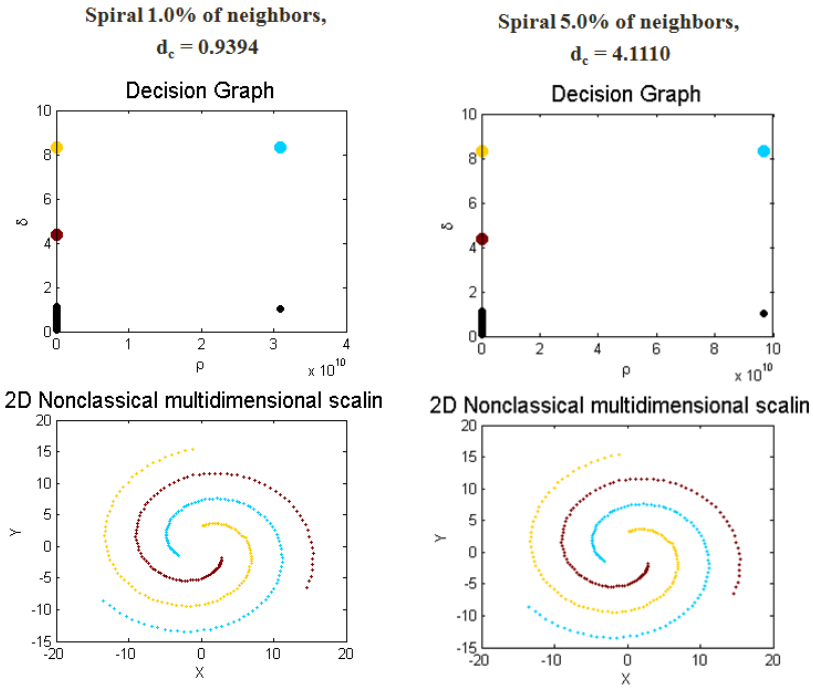

Fig. 4.12: Spiral dataset, Iterative Gaussian Kernel method applied.

The **CFSFDP** method has been proven to be very efficient at providing very good results in cluster analysis [1]. But there are also some drawbacks:

- strict dependency on the cut off distance (d_c) chosen.
- The user has to choose the average number of neighbors (in percentage) that could form a cluster. This percentage is strictly related to d_c (the more neighbors, the greater d_c is).

This is why several other methods have been presented in this work.

The **KNN-based CFSFDP** has a particular way to calculate the separation scores. Usually, the separation score attempts to determine how good separated the peak density points are from other points. This method dose affect the absolute values of the scores. However, it does not significantly affect the main point. In general, the KNN method is tested with a separation score. Using this method, it turns out that the dependency from d_c is less strong.

Since this method is independent from a cut off value, a statistical approach was presented.

The results obtained with the **Gaussian Kernel-based CFSFDP** method are pretty similar to the original method. The clusters are significantly distinctive and the points with high ρ and δ are still the ones selected.

Using Gaussian distribution the cluster definition is "smoother". This method is not effected as much as the previous one by the small number of points in the dataset because of the Gaussian fit.

The **Iterative Gaussian Kernel-based CFSFDP** method gives very good results, but it is more difficult to distinguish the cluster in the decision graph. Since the local density is converging to zero, it is not possible to distinguish the cluster cores from the decision graph. This is not fundamental but it is helpful in the analysis.

Moreover sometimes it can be complicated to select the right rectangle for defining $\rho_{minimum}$ and $\delta_{minimum}$ since ρ converges to zero.

5. CONCLUSIONS

In this new density-based (CFSFDP) algorithm, the key idea is representing data into 2_D space with axes ρ and δ. ρ is calculated for each point of data as the local density, i.e. the total number of data points around that data point. δ is calculated for each point of data as distance, i.e. the minimal distance between two points with higher density. The data points with high ρ and high δ will be selected as centers of new clusters. The data points with high ρ and low δ belong to the existing clusters. The data points with low ρ and high δ are treated as outliers. In this algorithm, density ρ is dependent on d_c where d_c is the cutoff distance. The weakness of the CFSFDP algorithm is a choice of d_c value, which is a state of art, and a different choice of d_c value could lead to the different clustering result. However, CFSFDP overcomes the weakness of other density-based algorithms such as DBSCAN. DBSCAN requires prior knowledge of the radius and minimum points.

Those methods (KNN- based, Gaussian Kernel-based and Iterative Gaussian Kernel-based) are applied to the CFSFDP algorithm. The KNN method overcame the sensitivity of using d_c when calculating the mean distance to the nearest neighbors of local density ρ. The Gaussian Kernel method defines the local density as the number of points within the cut off value, but uses the Gaussian Kernel to determine the ρ_i value. And, d_c corresponds to the standard deviation of the Gaussian Kernel, it

describes how much the data is dispersed. Knowing the derivative of the Gaussian Kernel, an Iterative Gaussian Kernel method was applied. In this method, the result is almost perfect. However, ρ is more difficult to distinguish the cluster center in the decision graph. Since the local density is converging to zero, it is impossible to distinguish the cluster cores from the decision graph. And, it depends on the number of iterations chosen.

in terms of results, the Iterative Gaussian Kernel has showed better among the four methods; by which, almost all the points belong to a cluster which reduces the problems of outliers points.

The publisher suggested d_c should be in range between 1-2%. It would be useful if the algorithm could calculate d_c based on the dataset. Furthermore, the possible future work based on this project is to find a better way to get d_c without changing it manually.

REFERENCES

[1] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science,* vol.344, pp.1492-1496, June. 2014.

[2] R. Xu and D. Wunsch, "Clustering analysis" in *Clustering*, Ed. New Jersey: Wiley-IEEE Press , 2008 ,ch. 1, sec. 1, pp.1-3.

[3] M. S. Aldenderfer and R. K. Blashfield,*Cluster Analysis*, Beverly Hills, CA: Sage Publications, 1984, pp.7-9.

[4] H.-P. Kriegel, P. Krger1, J. Sander and A.Zimek1, "Density-based clustering", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol.1, pp.231-240 , June. 2011.

[5] R. Hyde and P. Angelov, "Data Density Based Clustering" *computational Intelligence (UKCI), 2014 14th UK Workshop on,* pp.1,7, sept. 2014,

[6] Hartigan, *Clustering Algorithms*, New york: John Wiley & Sons, 1975, pp 364.

[7] A. Cuevas, M.Febrero and R. Fraiman,*Cluster analysis: a further approach based on density estimation*, Computational Statistics and Data Analysis, vol.36, pp.441-459, June. 2001.

[8] M. Wong, M. Anthony, "A kth nearest neighbour clustering procedure", *Jour-*

nal of the Royal Statistical Society. Series B (Methodological), vol.45, pp.362-368, January.1983.

[9] D. Wilshart,"Mode analysis: a generalization of nearest neighbor which reduces chaining effects," in *numerical Taxonomy,* A. J. Cole, Ed. Acedemic Press, 1969, pp.282-311.

[10] R. Seghers,(2014,July.18),*Rodriguez-Laoi Clustering* [Online].Available: http://rseghers.com/machine-learning/rodriguez-laoi-clustering.

[11] A. Gionis, H. Mannila and P. Tsaparas, "Clustering Aggregation", *ACM Transactions on Knowledge Discovery Data*, vol.1, pp.1-30, March.2007.

[12] P. Frnti and O. Virmajoki, "Iterative shrinking method for clustering problems", *Pattern Recognition*, vol.39, pp.761-763, 2006.

[13] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data.", *BMC Bioinformatics*, vol.8, pp.3-3, January. 2007.

[14] H. Chang and D. Y. Yeung, "Robust Path-Based Spectral Clustering", *Pattern Recognition*, vol.41, pp.191-203, March. 2008.

[15] M. J. Zaki and W. Meira,(2014) *Data Mining and Analysis: Fundamental Concepts and Algorithms*,[Online]. Vol.1 Availabe:http://www.cs.rpi.edu/ zaki/www-new/uploads/Dmcourse/Main/chap15.

[16] wikipedia contributors. (2015),*Cluster Analysis.* [Online]. Availabe: http://en.wikipedia.org/wiki/Cluster_analysis .

[17] R. Xu and W. D. II, Survey of clustering algorithms, *IEEE Transactions on NEURAL NETWORKS*, Vol. 16, pp. 645-678, May.2005.