

CHIHEB-EDDINE BEN NCIR 

A DENSITY-BASED METHOD FOR THE IDENTIFICATION OF DISJOINT AND NON-DISJOINT CLUSTERS WITH ARBITRARY AND NON-SPHERICAL SHAPES

Abstract *The ability of clustering methods to build both disjoint and non-disjoint partitionings of data has become an important issue in unsupervised learning. Although this problem has been studied during the last decades resulting in several proposed overlapping clustering methods in the literature, most of existing methods fail to look for clusters having arbitrary and non-spherical shapes. In addition, most of these existing methods require to pre-configure the number of clusters in prior, which is not a trivial task in real life application of clustering. To solve all these issues, we propose in this work a new density based overlapping clustering method, referred to as OC-DD, which is able to detect both disjoint and non-disjoint partitioning even when boundaries between clusters have complex separations with arbitrary forms and shapes. The proposed method is based on density and distances to detect highly dense regions and connected groups in data without the necessity to pre-configure the number of clusters. Experiments performed on artificial and real multi-labeled datasets have shown the effectiveness of the proposed method compared to the existing ones.*

Keywords overlapping clustering, non-disjoint clusters, density-based methods, clusters with non-spherical shapes

Citation Computer Science 22(2) 2021: 169–190

Copyright © 2021 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

1. Introduction

Clustering technique has become the subject of many recent research applied in many fields, such as in social networks to identify connected groups of users, document clustering to automatically organize the set of unstructured textual documents into topics and many other applications. The main idea of clustering is to group homogeneous data objects in the same group called cluster or segment. However, in several applications some data objects are needed to be assigned to more than one group which leads to organize the data into non-disjoint groups. This kind of issue is referred to as overlapping clustering. This problem occurs in many fields of clustering such as document clustering where each document can discuss different topics, video classification where a film can have different genres and emotion detection where a piece of music can evoke different emotions. For all these applications, overlapping clustering is more appropriate than conventional clustering to fit existing structures in data.

Many researchers have focused on the overlapping clustering problem by proposing several methods such as OKM (Overlapping K-Means) [10], MOC (Model-based Overlapping Clustering) [4], R-OKM (Regulated Overlapping K-means) [5], KHM-OKM [21] and 3WC-OR [1]. However, these methods suffer for several issues. The main issue of these existing methods is their inability to detect clusters with arbitrary and complex shapes which is the case of real-life applications. Another issue of this method is the necessity to set the number of clusters in prior before beginning the process of grouping data. Usually, all these issues can be solved by using a density based approach to look for high dense regions in data rather than evaluating distances between data objects. Density based clusters aims to look for high dense groups in data which leads to detect dense regions surrounded by low density regions [19]. Examples of existing density based methods are DBSCAN [14], OPTICS [3], DENCLUE [18] and CLIQUE [27]. Although the effectiveness of density-based methods to detect clusters with non-spherical shapes, they do not allow the detection of non-disjoint partitioning of data when such model is required to fit the existing structures.

To deal with this issue, we propose in this paper a new density-based clustering method, referred to as OC-DD, which is able to discover non-disjoint clusters even when clusters have arbitrary and non-spherical shapes. OC-DD is based on the modeling of the overall density of the set of data points as the sum of influence functions' associated with each data object. The resulting overall density function have a number of local density maxima called local density attractors. These attractors are used to define clusters by assigning each data object to the nearest attractor in terms of both density and distance. The rest of the paper is organized as follow: Section 2 gives basic concepts of overlapping clustering and density based methods and describes the issue of existing clustering methods which fail to detect overlapping clusters with arbitrary shapes. The proposed solution to deal with this issue is described in Section 3 while experiments and empirical results that show the effectiveness of the proposed solution are described in Section 4. Finally, Section 5 concludes with a summary and some directions for future research.

2. Identification of overlapping clusters: existing methods and problem description

The issue of identifying overlapping clusters (non-disjoint clusters) has been studied during the last decades. This issue was firstly introduced by [20] who proposed the k -ultra-metrics as a measure to look for overlapping clusters. This measure was further used to build overlapping hierarchies such as the Pyramids method [13] and the weak hierarchies method [7]. Other overlapping clustering methods was proposed in the literature which extend well known conventional methods such as Model-based Overlapping Clustering (MOC) that generalizes Expectation-Maximization method (EM) [8] and Overlapping k -means (OKM) [10] method that extends the well-known k -means [24]. More recent methods was also proposed in the literature which proposed more sophisticated techniques to fit overlapping structures in data such as the generalized regulated overlapping k -means which proposes to control overlapping boundaries between clusters [1, 5], kernel based methods that proposes to deal with complex data structures [6, 11] and KHM-OKM [21] which solves the issue of the initialization of cluster representatives.

In addition to these methods, some recent overlapping methods where designed to deal with specific applications issues such as the identification of non-disjoint groups from complex social networks [25, 28, 29] and the identification of overlapped genes expressions in biology [26]. These methods were designed for specific applications and cannot be generalized for all types of data.

In fact, in real life applications the learning algorithm must allow to detect overlapping clusters to fit existing structures in data. These identified disjoint and non-disjoint clusters may have different shapes and forms. The learning algorithm should be able to detect clusters with arbitrary shapes [14, 18, 22], including spherical and non-spherical clusters and should allow overlaps between clusters. We give in Figure 1 examples of spherical and non-spherical clusters.

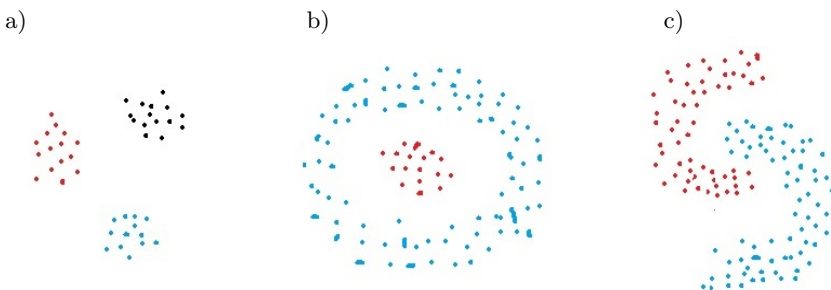


Figure 1. Examples of spherical and non-spherical shapes of clusters. Clusters can have arbitrary shapes: a) three well separated spherical clusters; b) two non-spherical clusters with concentric shapes; c) two non-spherical clusters with complex shapes

Figure 1a reports an examples of three well separated clusters having spherical shapes while Figures 1b and 1c reports examples of clusters having non-spherical and complex shapes. In order to evaluate the effectiveness of well known overlapping methods to deal with this issue, we build results of the existing OKM and R-OKM methods on a generated dataset containing two groups with concentric shapes. Each group contains 100 data object.

The results obtained by OKM and R-OKM are illustrated in Figure 2. This Figure shows that OKM and R-OKM build clusters with overlaps (the “Green” data objects) but are not able to detect non-spherical shapes. The two methods fail to identify the clusters having concentric shapes (one cluster included in another). Another shortcoming of the these methods is the necessity to fix the number of clusters in prior in order to build clusters. The number of clusters may be difficult to estimate in real life applications. In order to deal with all these issues we propose in the next section a density based overlapping clustering method able to build non-disjoint partitionings of data even when data contain groups with arbitrary shapes. The proposed method is based on both densities and distances.

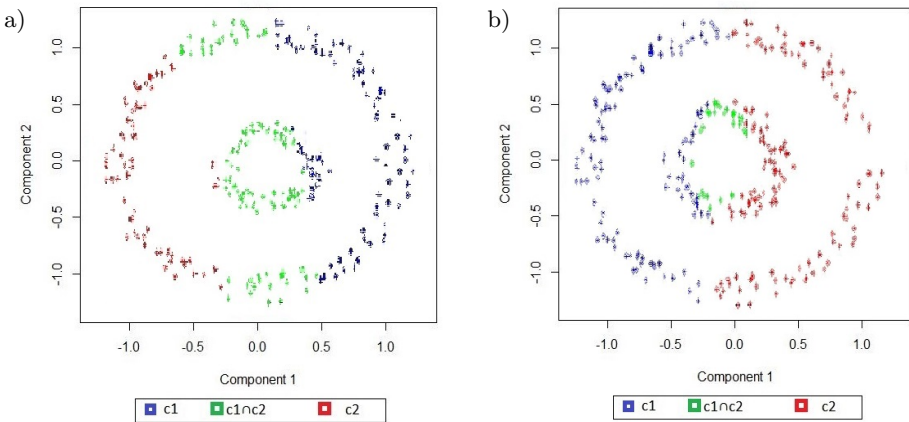


Figure 2. Clustering results obtained by applying OKM (a) and R-OKM (b) in an artificial dataset

3. Proposed method: overlapping clustering based on density and distances (OC-DD)

To deal with the identification of overlapping clusters with arbitrary and non-spherical shapes, we propose a new overlapping clustering method, referred to as OC-DD (Overlapping Clustering based on Densities and Distances), which is based on density estimation. The proposed method defines for each data object a local density estimate called influence function. The influence function can be seen as a function which

describes the impact of a data object within its neighborhood. The overall density of the data objects in the dataset is calculated as the sum of the influence functions of all data objects. The identification of clusters can be done by identifying high density points called density-attractors. These density-attractor points are mathematically defined by the local maxima of the overall density function. Attractors that have an equally high scores are merged to build a single cluster. A data object can belong to one or several clusters respecting to his distance from a density attractors points given a threshold parameter.

Given an input dataset $D = \{x_1, \dots, x_N\}$ containing N data objects described by d attributes, the objective of the proposed OC-DD method is to build a non-disjoint partitionings $C = \{C_1, \dots, C_k\}$ of data into k clusters where each data object x_i can be assigned to one or to several clusters. Four main steps are defined in the OC-DD method to identify the non-disjoint partitioning of data as schematized in Figure 3.

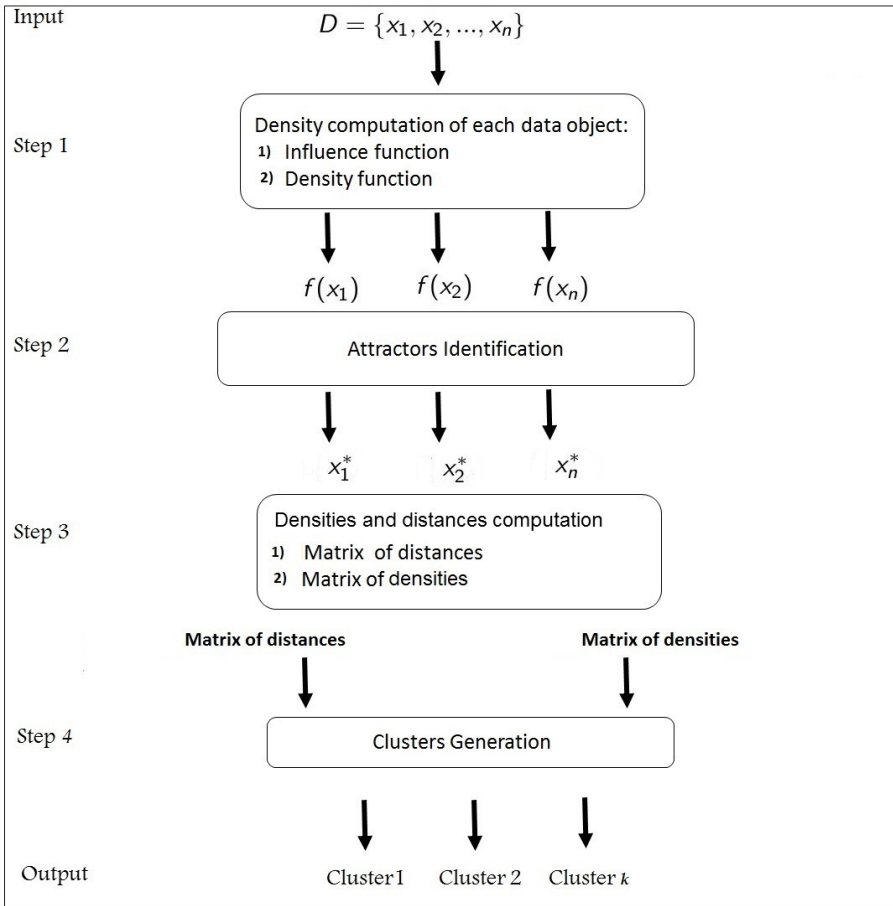


Figure 3. Illustrative schema of OC-DD algorithm

The first step consists of computing the density function of each data object based on the used influence function. Then, the second step aims to identify the local maxima of the overall density function named density-attractors. Once the attractors are identified, a third step is devoted to compute the densities and distances matrices. Finally, the last step aims to derive the partitioning of clusters based on distances between attractors where similar ones are grouped together in the same cluster. We give in the following details of each step of the proposed OC-DD method.

3.1. Step 1. Density estimation

This step consists of computing the overall density F^D of all data objects based on the local influence function $f(x_i)$ of each data object. The local influence function measures the impact of a data object x_i within its neighborhood y and is defined by:

$$f^y(x_i) = f(x_i; y) \quad (1)$$

Examples of influence functions that can be used are the square wave function and the Gaussian function which are defined as follows:

1. Square Wave Influence Function (f_{Square}):

$$\begin{cases} 0 & \text{if } d(x, y) > \sigma \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

2. Gaussian Influence Function (f_{Gauss}):

$$f_{Gauss}(x; y) = \exp - \frac{d(x, y)^2}{2\sigma^2} \quad (3)$$

where $d(x, y) \in \mathfrak{R}_0^+$ is the distance between data objects x and y and can be measured using standard distance functions such as Euclidean and Manhattan distances. The parameter σ controls the influence of an object in the space. It denotes the spread or smoothness of the density estimate. For small values of σ , the density function has several local maxima, whereas for larger values the number of maxima decreases.

Based on the local density estimate of each data object within its neighbor, the overall density function $F^D(x_i)$ of the data object x_i in the ensemble of data D is computed as the sum of the local influence function $f^{y_j}(x_i)$ of all data objects $y_j \in D$. The overall density $F^D(x_i)$ of a data object x_i is defined by:

$$F^D(x_i) = \sum_{j=1}^N f^{y_j}(x_i) \quad \forall x_i \in D \quad (4)$$

This density estimation depends on the used influence function. For example, when the Gaussian influence function is used, the density estimate of $F_{Gauss}^D(x_i)$ will be as follows:

$$F_{Gauss}^D(x_i) = \sum_{j=1}^N \exp - \frac{d(x_i, y_j)^2}{2\sigma^2} \quad \forall x_i \in D \quad (5)$$

3.2. Step 2. Attractors identification

The second step consists in identifying the set of density attractors $\{x_1^*, \dots, x_N^*\}$ which are defined as the local maximum of the density function $F^D(x_i)$ for each data object x_i . The local maximum of the density function $F^D(x_i)$ for a given data object x_i can be determined by a Hill Climbing algorithm which is based on a gradient ascent approach as described in [18] and [9]. The idea is to estimate the density gradient and determining the direction of the largest increase in the density. After that, moving in the direction of the gradient in small steps until reaching a local maximum. The estimation of the gradient for a density function is effectively solved by [16] which gives a good estimation of the gradient for pattern recognition applications. This estimation of the gradient were used in well known density-based clustering methods such as DENCLUE [18] and Mean-Shift [9]. The estimation of the gradient for a density function can be described as follows:

$$\nabla F^D(x_i) = \sum_{j=1}^N (y_j - x_i) \times f^{y_j}(x_i) \quad \forall x_i \in D \quad (6)$$

When the Gaussian influence function is used the gradient can be defined by:

$$\nabla F_{Gauss}^D(x_i) = \sum_{j=1}^N (y_j - x_i) \exp -\frac{d(x_i, y_j)^2}{2\sigma^2} \quad (7)$$

An object x_i^* is called a *density-attractor* if x_i^* is a local maximum of the density function $F^D(x_i)$. An object x_i is *density-attracted* to a *density-attractor* x_i^* if $\exists m \in N : |x_m - x_i^*| \leq TOL$ with:

$$x_0 = x, x_m = x_{m-1} + \delta \frac{\nabla F^D(x_{m-1})}{\|\nabla F^D(x_{m-1})\|} \quad (8)$$

where $\delta < 0$ is the size of each step and TOL is the tolerance variation parameter. A data object x is *density attracted* to a density attractor x^* if a gradient ascent process started at x_i and converges to x_i^* . In other words, there exist a sequence of data objects $x = x_0, x_1, \dots, x_m$, such that $|x_m - x^*| \leq TOL$, and each intermediate object is obtained after a slight variation in the direction of the gradient vector.

3.3. Step 3. Densities and distances computation

The step of densities and distances computation consists in computing pair-wise distances between attractors and the pair-wise differences of densities between all data objects. In this step, the OC-DD method requires in input the estimated densities and the identified attractors in the previous step and returns pair-wise densities and distances matrices as described in Figure 4.

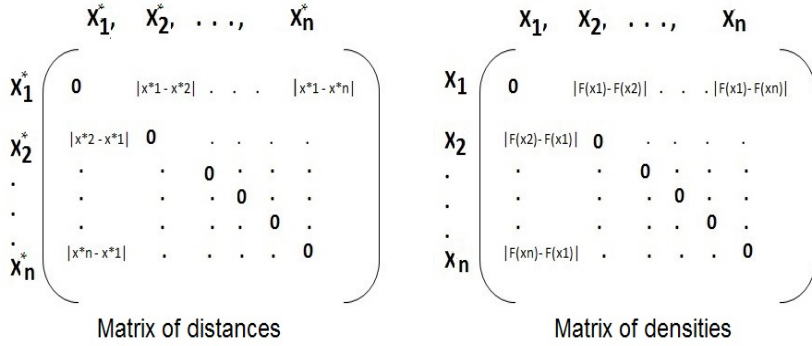


Figure 4. Pair-wise matrices of densities and distances at the end of the third step

3.4. Step 4. Clusters generation

The last step of the OC-DD consists of generating clusters based on densities and distances matrices. In fact, clusters are build as dense regions in the data space by looking for data objects having both similar densities and converging to near local maximum. A cluster C_p containing m data objects $\{x_1^p, \dots, x_m^p\}$ is formed when all these data objects are density attracted to nearly the same attractor x_i^* and also having nearly the same density estimate $F^D(x_1^p) \simeq F^D(x_2^p) \simeq \dots \simeq F^D(x_m^p)$. There is not any rule regarding the number of clusters to which a data object can be assigned to. The data object can be assigned to several clusters when it is density-attracted to more than one attractor x_i^* . In addition, given that clusters are defined based on regions of higher density, clusters with non-spherical and arbitrary shapes can be easily identified. Therefore, the proposed OC-DD method can effectively solve the issue of identifying overlapping clusters with arbitrary shapes and forms. Another advantage of the proposed OC-DD method is the automatic identification of the number of clusters. There is no need to pre-configure this number in prior. The number of attractors is considered the right number of clusters in the dataset since it determines the number of high density regions in the data space.

In order to generate clusters based on the above definition, we define two parameters: $VTOL$ and θ . The first, $VTOL$, determines the maximal tolerated variations between attractors to be merged together while the second, θ , determines the maximal tolerated density between data objects to be merged together in the same cluster. Theoretically, data objects x_i and x_j are put in the same cluster C_p when these two conditions are satisfied:

$$\|x_i^* - x_j^*\| < VTOL \quad \wedge \quad \|F^D(x_i) - F^D(x_j)\| < \theta \quad \forall x_i, x_j \in C_p \quad (9)$$

In the following, we give a pseudo-code of the main algorithm of the proposed OC-DD method.

Algorithm 1 OC-DD algorithm (TOL , $VTOL$, θ , σ)**Input:** D : a dataset containing N data objects $\{x_1, X_2, \dots, x_N\}$ described over \mathbb{R}^d TOL : the tolerance for convergence parameter, $VTOL$: the attractor merge threshold, θ : the density merge threshold, σ : the smoothing parameter of the influence function**Output:** $\{C_1, \dots, C_k\}$ clusters memberships**1:** For each data object x_i compute the density function (using Equation 4).**2:** Run gradient (using Equation 6).**3:** Find Density Attractor, x_i^* , for each $x_i \in D$ using Equation 8**4:** Compute pair-wise distances between attractors.**5:** Compute pair-wise differences of densities between the data objects.**6. if** $\|x_i^* - x_j^*\| < VTOL$ and $\|F^D(x_i) - F^D(x_j)\| < \theta$.Assign a data object x_i and x_j to the same cluster.

3.5. Computational complexity of OC-DD method

Given that the four steps of OC-DD method are independent, the computational complexity of OC-DD can be determined by the step having the maximal computational complexity. Therefore, we give in the following an evaluation of the computational complexity of each step. The first step, density computation, consists in computing the influence function for each object $x_i \in D$. The local influence function of each data object x_i is approximated by $O(N)$ with N the number of data objects. So, computing all density functions for all data objects can be approximated by $O(N^2)$ which is the estimated computational complexity of the first step. The second step consists in determining the set of density attractors. This step takes $O(N \cdot m)$ time for each object where m is the maximum number of iterations in the gradient descent function. Therefore this step can be evaluated by $O(N^2 \cdot m)$. The third step, densities and distances computation, which evaluates the pair-wise differences of densities between the data objects and pair-wise distances between attractors can be approximated by $O(N^2)$. The final step, clusters generation, can be approximated by $O(N^2)$ since it look for all pair of clusters that can be putt together in the same cluster. Therefore, based on all these evaluated computational complexity for the different steps we can proximate the overall computational complexity of OC-DD by $O(N^2) + O(N^2 \cdot m) + O(N^2) + O(N^2) \simeq O(N^2 \cdot m)$.

4. Experiments

We experimentally evaluated the performance of the proposed method compared to existing ones in the literature. We begin by describing the different used datasets and the evaluation methodology and then we give descriptions and discussions of experimental results.

4.1. Datasets description

Experiments are performed on both artificial and real datasets. For artificial datasets, we simulated two datasets containing clusters with complex shapes as in the following:

- **Artificial dataset1** contains two clusters where each cluster contains 100 data objects defined in two dimensional space. the two clusters have a non-spherical shape.
- **Artificial dataset2** contains 359 data objects defined in two dimensional space classified into 2 separate groups. The first cluster contains 120 data objects and the second cluster contains 239 data objects. The clusters of this dataset have a non spherical shapes (2 circles with same center but different radius).

Concerning real datasets, it was selected from different domains where data need to be assigned to more than one cluster and having different degree of overlaps (from 1 to 4.8):

- **Iris dataset:** Iris dataset [15] consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of sepals and petals, in centimeters. This dataset has no overlaps between clusters (size of overlaps = 1).
- **Eachmovie dataset:** Video classification grouping movies with respect to genres based on user rating. EachMovie is composed from 600 films, 6 labels and the overlap size is equal to 1.14.
- **Emotion dataset:** Music emotion classification. Grouping music songs with respect to emotion based on the analysis of music signals. Emotion dataset is composed from 593 songs, 6 labels and the overlap size is equal to 1.81.
- **Yeast dataset:** Predicting the Cellular Localization Sites of Proteins. Yeast dataset is composed from 2417 gene descriptions, 14 labels and the size of overlaps is equal 4.23.

Table 1 gives statistics description of each simulated and real dataset.

Table 1
Statistics description of simulated and real datasets

Datset	Type	#objects	#dimensions	#clusters	Avg. overlaps
Artificial dataset1	simulated	100	2	2	1
Artificial dataset2	simulated	359	2	2	1
Iris	real	150	4	2	1
Eachmovie	real	600	5	6	1.14
Emotion	real	593	72	6	1.81
Yeast	real	2417	8	14	4.23

4.2. Evaluation methodology and evaluation measures

The quality evaluation of clustering results is not a trivial task. Two approaches can be used to evaluate the outputs: internal and external approaches. Internal evaluation is based only on the intrinsic property of the output while external evaluation requires an existing dataset with known labels called gold standard. Given that external evaluation measures are well defined for the evaluation of overlapping clusters [2, 23], we considered this approach to evaluate the output of the proposed OC-DD method. Results are compared based on a comparison between the output of the clustering (groups) and the known classes in the dataset. We used BCubed measures for overlapping clustering proposed by [2] to take into account the multiplicity of the pairs of observations.

The BCubed technique [2] is based on computing precision and recall for each pair of observations sharing at least one category or one cluster. In the case of overlapping clustering, the extended BCubed takes into account the multiplicity of observations occurrence in clusters and categories. Extended BCubed precision evaluates the amount to which the decisions made by the evaluated algorithm of placing pairs of objects together in one or several clusters are correct. The BCubed precision is defined as:

$$P = \frac{1}{|D|} \sum_{x \in D} \frac{1}{|D_{g \in G(x)}|} \sum_{x' \in E(x, G)} \frac{\min(|G(x) \cap G(x')|, |C(x) \cap C(x')|)}{|G(x) \cap G(x')|} \quad (10)$$

where x and x' are two data objects, $G(x)$ the set of categories (known classes), $C(x)$ the set of clusters associated to x and $|D|$ is the number of data objects. $E(x, G)$ is the set of data objects co-occurring with x in at least one candidate cluster, and $|D_{g \in G(x)}|$ is the number of data objects for all classes to which the data object x belongs to. Concerning the extended BCubed recall measure, it aims to evaluate the amount to which the evaluated algorithm is able of putting together the pairs of objects that co-occur in classes of data with known labels. The BCubed recall is defined only when (x, x') share one or more categories, and it is maximal when the number of shared categories is lower or equal than the number of shared clusters, and it is minimal when the two observations do not share any cluster. The Bcubed recall is defined by:

$$R = \frac{1}{|D|} \sum_{x \in D} \frac{1}{|D_{g \in C(x)}|} \sum_{x' \in E(x, C)} \frac{\min(|G(x) \cap G(x')|, |C(x) \cap C(x')|)}{|C(x) \cap C(x')|} \quad (11)$$

where $E(x, C)$ is the set of data objects co-occurring with x in at least one cluster and $|D_{g \in C(x)}|$ is the number of data objects for all obtained clusters to which the data object x belongs to. If we have less shared clusters than needed, we lose in term of recall; if we have less categories than clusters, we lose in term precision. The analysis of recall and precision helps in evaluating the performance of the resulting clustering. However, for using a unique evaluation measure for the comparison, we can use the F -measure which is based on both precision and recall. The BCubed

F -measure provides a trade-off between Extended BCubed precision and Extended BCubed recall and is defined by:

$$F - measure = 2 \cdot \frac{BCubedPrecision \cdot BCubedRecall}{BCubedPrecision + BCubedRecall} \quad (12)$$

The fourth measure that we have used in the experiments is the overlap size and is defined by:

$$Overlap = \frac{1}{|D|} \cdot \sum_{x_i \in D} |A_i| \quad (13)$$

where $|D|$ is the number of data objects and $|A_i|$ is the number of clusters to which the data object x_i is assigned to. The size of overlap influences the performance of overlapping clustering methods. More this size is near to the actual size of overlaps in the gold standard, more this size is considered better.

4.3. Results on IRIS dataset

IRIS dataset is widely used as a primary test for machine learning purpose, especially for a preliminary evaluation of clustering and classification methods. The specificity of the dataset consists that it contains one category of flowers (Iris Setosa) which is easily to separate from the other two categories (Iris-virginica and Iris-versicolor) that are very similar. In order to evaluate the performance of the proposed OC-DD method, we plot obtained clusters of the OC-DD method using the two first axes of Principal Component Analysis (PCA) technique as described in Figure 5.

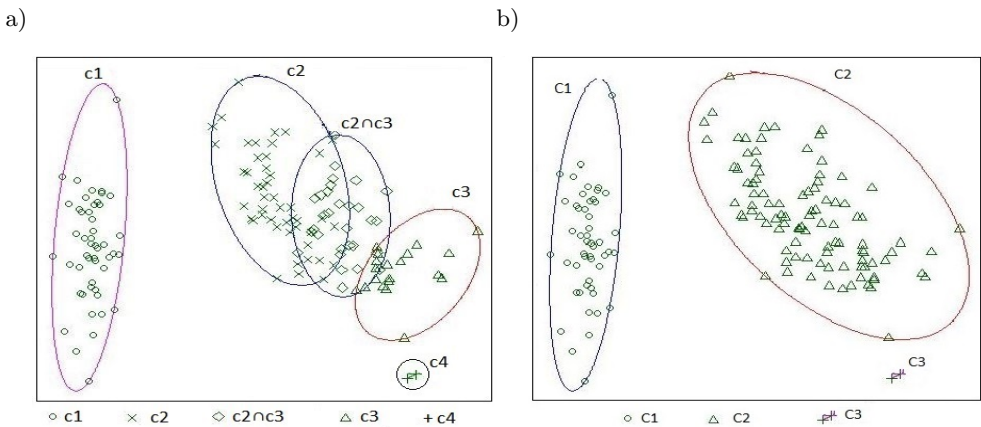


Figure 5. Obtained clusters using OC-DD on Iris dataset with two values of VTOL projected on the two first PCA axes: a) $VTOL = 1$; b) $VTOL = 1.5$

Clusters are build using two values of VTOL parameter which presents the maximal considered distance between attractors to be merged into a single cluster. Reported results first show the ability of OC-DD to build overlapping clusters. We show in Figure 5a the ability of OC-DD to build overlaps between the two similar flowers categories. Second, we show the ability of the proposed OC-DD method to build disjoint clusters when data are well separated. We also show the impact of the VTOL parameter on the performance and behavior of the method: smaller values increase the number of clusters until reaching N clusters while larger values considerably reduce the number of clusters until grouping all observations into a single cluster.

In order to empirically evaluate the performance of OC-DD compared to existing methods, we report in Table 2 values of Precision (P), Recall (R), F -measure (F) and the average size of overlaps obtained using OC-DD, OKM [10], R-OKM [5] and ALS [12] methods. Obtained results show that OC-DD gives a better value of F -measure compared to existing ones. The obtained F -measure increases from 0.65 obtained using OKM to reach 0.842 by using OC-DD. The improvement of results is achieved due to the important improvement of precision. Second, we show the ability of OC-DD to give a good estimation of the number of clusters (between 2 and 4) without requiring to estimate this number in prior as done for all existing methods.

Table 2
Empirical results on Iris dataset

Method	BCubed Evaluation				Size of Overlap
	$P.$	$R.$	$F.$	$n_{clust.}$	
OC-DD($\sigma = 0.34$, VTOL = 1, $\theta = 0.02$)	0.74	0.97	0.84	4	1.20
OC-DD($\sigma = 0.002$, VTOL = 3, $\theta = 100$)	0.51	0.84	0.63	4	1.17
OKM	0.48 \pm 0.03	0.99 \pm 0.01	0.65 \pm 0.02	3	1.48 \pm 0.05
R-OKM	0.58 \pm 0.13	0.98 \pm 0.06	0.71 \pm 0.08	3	1.32 \pm 0.19
KHM-OKM	0.53 \pm 0.01	0.99 \pm 0.01	0.67 \pm 0.01	3	1.40 \pm 0.03
ALS	0.43 \pm 0.04	0.93 \pm 0.03	0.68 \pm 0.03	3	1.52 \pm 0.09

4.4. Ability to detect arbitrary and non-spherical shapes

In order to evaluate the ability of OC-DD to detect overlapping clusters with arbitrary shapes, we plot obtained clusters build by OC-DD on two artificial datasets containing clusters with non-spherical shapes. First, we show the limit of using the existing OKM method which fails to build the right clusters in the first and the second artificial dataset as illustrated in Figure 6b and Figure 7b. We show some red and blue points in high dense region which represents only a single dense cluster. This problem is solved by using the proposed OC-DD method. We also show that overlaps are built in the surface between the first and the second dense region. For the second dataset which contains two concentric shapes of clusters, we show that OKM completely fails to build clusters in this dataset. However, using OC-DD we show that our proposed method succeed in detecting two disjoint clusters which fits the existing structures in

this dataset as described in Figure 7a. We remark that our method can produce both disjoint and overlapping clusters based on the existing structures in data.

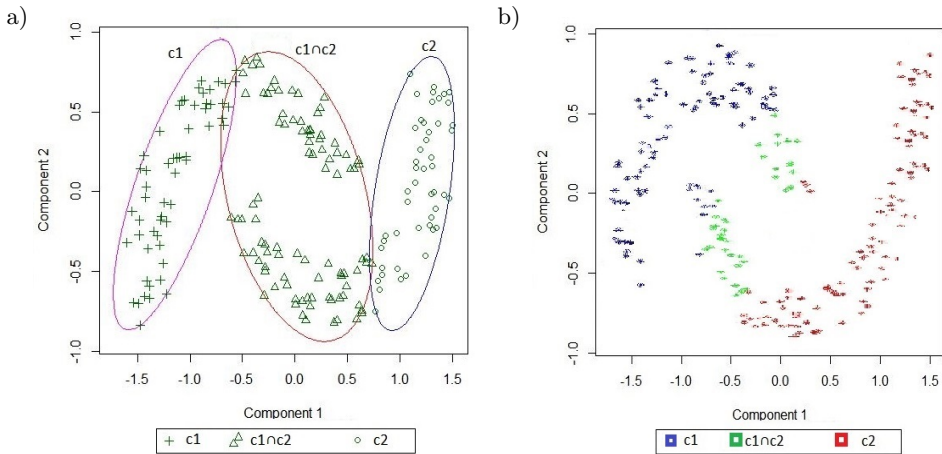


Figure 6. Comparison of obtained clusters with OC-DD and the existing OKM method on Artificial dataset 1: a) OC-DD; b) OKM

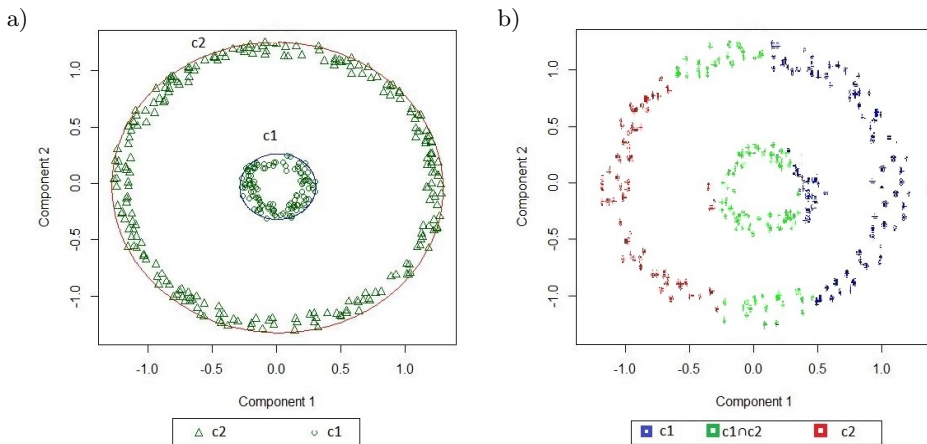


Figure 7. Clusters obtained by OC-DD on Artificial dataset 1 using the first two components of PCA: a) OC-DD; b) OKM

We also built empirical results of Bcubed precision, recall, F -measure and overlap size build in these artificial datasets using the proposed OC-DD method and compared to those obtained using OKM, R-OKM and ALS as reported in Table 3. We show the large improvement of results such as the improvement of F -measure from 0.68 by

using OKM to reach 0.87 using our proposed method in the first artificial dataset. We also show from this table that the the improvement of results in the second artificial dataset is more important than those obtained in the first dataset given that concentric shapes are impossible to detect using all evaluated existing methods which only detects clusters with spherical shapes. The obtained results on artificial datasets confirm the ability of our proposed method to identify overlapping clusters having arbitrary and non-spherical shapes.

Table 3

Comparison of BCubed precision, recall, F -measure and overlap size obtained on artificial dataset1 and artificial dataset2

Dateset Method		BCubed Evaluation				Size of Overlap
		$P.$	$R.$	$F.$	$nbclust.$	
Artificial Dataset1	OC-DD($\sigma = 0.3$, VTOL = 2, $\theta = 0.3$)	0.54	1	0.7	2	1.40
	OC-DD($\sigma = 0.3$, VTOL = 3, $\theta = 0.3$)	0.52	1	0.69	2	1.38
	OC-DD($\sigma = 0.2$, VTOL = 102, $\theta = 0.12$)	0.76	0.99	0.87	2	1.17
	OKM	0.53 \pm 0.02	0.97 \pm 0.04	0.68 \pm 0.03	2	1.4 \pm 0.12
	R-OKM	0.64 \pm 0.16	0.9 \pm 0.19	0.75 \pm 0.17	2	1.22 \pm 0.04
	KHM-OKM	0.54 \pm 0.02	0.98 \pm 0.02	0.69 \pm 0.02	2	1.4 \pm 0.04
	ALS	0.55 \pm 0.02	0.99 \pm 0.02	0.71 \pm 0.02	2	1.48 \pm 0.08
Artificial Dataset2	OC-DD($\sigma = 0.2$, VTOL = 50, $\theta = 0.15$)	0.94	0.94	0.94	2	1
	OC-DD($\sigma = 0.2$, VTOL = 45, $\theta = 0.15$)	0.90	0.90	0.90	2	1
	OC-DD($\sigma = 0.2$, VTOL = 102, $\theta = 0.12$)	0.69	0.75	0.72	3	1.26
	OKM	0.32 \pm 0.04	0.84 \pm 0.06	0.46 \pm 0.05	2	1.52 \pm 0.11
	R-OKM	0.49 \pm 0.08	0.59 \pm 0.10	0.54 \pm 0.09	2	1.1 \pm 0.26
	KHM-OKM	0.34 \pm 0.02	0.87 \pm 0.01	0.48 \pm 0.02	2	1.47 \pm 0.05
	ALS	0.30 \pm 0.05	0.89 \pm 0.06	0.45 \pm 0.05	2	1.58 \pm 0.08

4.5. Empirical results on real multi-labeled datasets

In order to evaluate the effectiveness of the proposed method on real datasets, we compare obtained results of OC-DD to those obtained using OKM, R-OKM, KHM-OKM and ALS methods on three multi-labeled datasets. Tables 4, 5 and 6 report best obtained BCubed precision, recall, F-measure and overlaps size using OC-DD, OKM, R-OKM, KHM-OKM and ALS on Eachmovie, Emotion and Yeast datasets respectively. Given that all the compared methods are sensitive to the initialization of centroids, we used the same initialization for all methods. We note also that the variance of results of all these methods is very high that can reach 0.6 in several datasets. The initialization of the parameter for each compared method can highly

influences the obtained results. For this reason, we empirically tested several values of parameters for each method and we only report best obtained results. The high variances of results is also shown in the obtained results of our proposed method when varying the parameters value. We reported in each dataset results of the proposed method using several parameters values in order to give an analytic analysis of the initialization of the parameters for our proposed methods.

Table 4

Comparison of obtained results of OC-DD with existing methods on Eachmovie dataset

Method	BCubed Evaluation				Size of Overlap
	<i>P.</i>	<i>R.</i>	<i>F.</i>	<i>nbclust.</i>	
OC-DD ($\sigma = 0.002$, VTOL = 2, $\theta = 1000$)	0.54	0.79	0.64	5	1.54
OC-DD ($\sigma = 0.002$, VTOL = 3, $\theta = 1000$)	0.51	0.84	0.63	3	1.17
OC-DD ($\sigma = 0.03$, VTOL = 20, $\theta = 32$)	0.41	0.84	0.56	4	1.21
OC-DD ($\sigma = 0.03$, VTOL = 40, $\theta = 32$)	0.41	0.91	0.57	3	1.21
OKM	0.39 \pm 0.09	0.91 \pm 0.08	0.55 \pm 0.09	3	1.70 \pm 0.21
R-OKM	0.74 \pm 0.14	0.71 \pm 0.18	0.73 \pm 0.16	3	1.13 \pm 0.45
KHM-OKM	0.45 \pm 0.01	0.91 \pm 0.02	0.60 \pm 0.05	3	1.13 \pm 0.15
ALS	0.36 \pm 0.19	0.82 \pm 0.16	0.50 \pm 0.03	3	1.73 \pm 0.15

Table 5

Comparison of obtained results of OC-DD with existing methods on Emotion dataset

Method	BCubed Evaluation				Size of Overlap
	<i>P.</i>	<i>R.</i>	<i>F.</i>	<i>nbclust.</i>	
OC-DD ($\sigma = 0.02$, VTOL = 45, $\theta = 100$)	0.43	0.68	0.53	5	1.67
OC-DD ($\sigma = 0.002$, VTOL = 50, $\theta = 100$)	0.47	0.63	0.54	3	1.14
OC-DD ($\sigma = 0.002$, VTOL = 42, $\theta = 100$)	0.44	0.61	0.51	6	1.69
OKM	0.35 \pm 0.07	0.54 \pm 0.11	0.43 \pm 0.09	6	2.35 \pm 0.17
R-OKM	0.86 \pm 0.18	0.27 \pm 0.07	0.35 \pm 0.03	6	1.26 \pm 0.26
KHM-OKM	0.36 \pm 0.02	0.54 \pm 0.03	0.43 \pm 0.02	6	2.31 \pm 0.11
ALS	0.30 \pm 0.12	0.97 \pm 0.10	0.46 \pm 0.10	6	3.46 \pm 0.23

Table 6

Comparison of obtained results of OC-DD with existing methods on Yeast dataset

Method	BCubed Evaluation				Size of Overlap
	<i>P.</i>	<i>R.</i>	<i>F.</i>	<i>nbclust.</i>	
OC-DD($\sigma = 0.002$, VTOL = 1.35, $\theta = 1000$)	0.75	0.47	0.58	13	3.8
OC-DD($\sigma = 0.002$, VTOL = 1.34, $\theta = 1000$)	0.75	0.45	0.56	15	3.96
OC-DD($\sigma = 5$, VTOL = 6, $\theta = 1000$)	0.13	0.77	0.23	8	4.22
OC-DD($\sigma = 0.5$, VTOL = 1.33, $\theta = 1000$)	0.15	0.63	0.24	17	4.02
OKM	0.59 \pm 0.08	0.48 \pm 0.09	0.53 \pm 0.08	14	4.80 \pm 0.25
R-OKM	0.75 \pm 0.13	0.18 \pm 0.18	0.29 \pm 0.16	14	3.04 \pm 1.36
KHM-OKM	0.61 \pm 0.03	0.50 \pm 0.03	0.55 \pm 0.03	14	4.88 \pm 0.12
ALS	–	–	–	–	–

Table 4 shows that R-OKM can give the best obtained results on multi-labeled datasets having a small size of overlaps which is the case of Eachmovie dataset (actual overlaps = 1.14). Obtained F -measure of R-OKM is equal to 0.728 which coincides with the nearest build overlap size equal to 1.13. OC-DD gives acceptable results in this dataset (F -measure equal to 0.632) and outperforms those obtained with OKM and ALS.

In the Emotion dataset which has an overlap size more important than Eachmovie, reported results show the effectiveness of the proposed method which gives the best obtained results compared to OKM, R-OKM and ALS as reported in Table 5. This table shows that the best obtained F -measure with OC-DD is equal to 0.532 while the best obtained F -measure using existing methods is equal to 0.466 (using ALS). This improvement is explained by the improvement of both precision and recall compared to the other methods that give large sizes of overlaps (more large than expected) such as the case of ALS. ALS gives an overlap size equal to 3.54 while the actual overlaps in the Emotion dataset is only 1.81. This table also shows that the proposed method can give a good estimation of the number of clusters (5 and 6 clusters) while this number is given in prior for existing methods.

These results are also confirmed on Yeast dataset which has the largest number of data objects, the largest number of labels (14 classes) and the largest number of overlaps (4.23) as reported in Table 6. We show that results of ALS cannot be reported given the high computing complexity of this method. We also show that OC-DD and OKM give the best obtained results of F -measure.

4.6. Scalability and parameters discussion

Although the effectiveness of the proposed method on both artificial and real multi-labeled datasets, it suffers from the issue of scalability and parameters initialization as the case of most of density-based methods. In order to evaluate the scalability of the proposed method, we give in Table 7 running times of the proposed method on real multi-labeled datasets. This table shows that OC-DD can scale well with datasets containing thousand of data objects. For example, In Yeast dataset which contains 2407 data objects, OC-DD returns results in approximately 137 seconds. The value of TOL parameter can largely improves the running times. A small values of TOL makes the second step of OC-DD computationally high. A good initialization of this parameter can largely accelerate the step of building attractors and then can accelerate the overall process of OC-DD. We note here that the proposed method can be applied to more large datasets if a good data pre-processing is realized. For example, rather than using single data objects, hyper rectangles (hyper-cubes) representation can be built before the clustering step which will create summarized cubes that only contain very small dense regions (populated cubes) using a density estimation method with a very small bandwidth window σ . This structure allows to OC-DD an easy manipulation of data by considering only populated cubes.

Table 7
Running times of OC-DD on real multi-labeled datasets

Datset	Method	Running times (seconds)
Eachmovie	OC-DD(TOL = 0.01)	less than 1
	OC-DD(TOL = 0.001)	1
	OC-DD(TOL = 0.0001)	6
Emotion	OC-DD(TOL = 0.01)	1
	OC-DD(TOL = 0.001)	2
	OC-DD(TOL = 0.0001)	8
Yeast	OC-DD(TOL = 0.01)	6
	OC-DD(TOL = 0.001)	30
	OC-DD(TOL = 0.0001)	74

Concerning the issue of parameters initialization, we note that four important parameters need to be initialized for the OC-DD method which are σ , TOL, VTOL and θ . The parameter σ determines the degree of the influence of an object in its neighborhood. Like all density based methods, the quality of the resulting clustering depends on an appropriate choice of this parameter. Small values of the smoothing parameter σ give very rough estimates of densities while larger bandwidths give smoother estimates. In practice, the choice of the smoothing parameter σ should be in the interval $[\sigma_{min}, \sigma_{max}]$. σ_{max} is the minimal value when the global density function only has a density attractor and σ_{min} is the maximal value when the function has N different density attractors. Choosing a good σ can be done by considering different σ and determining the largest interval between σ_{min} and σ_{max} where the number of similar attractors remains constant. This fact ensures that the number of density attractors is constant for a long interval of σ . For the second parameter TOL, it describes whether a density-attractor is significant allowing a reduction of the number of density-attractors. If TOL is large, low-density clusters may be neglected, but when it is small, high-density clusters that are close may be merged. If TOL is set to zero, each object may become a cluster of its own. A good choice of TOL helps the method to focus on the dense regions and to save the computational time. A good initialization of this parameter may be between the constant $\|D_N\| \cdot \sqrt{2 \cdot \pi \cdot \sigma^{2d}}$ and the minimum density value as recommended in the works of [17, 18]. We note here that the choice of TOL and the previous parameter σ is dependent. A large difference between TOL and σ makes the choice of the right value of σ difficult since it will be difficult to find the largest interval where density attractors still constant.

The third and the fourth parameter, θ and VTOL, determine the maximal difference of densities and the maximal distance between attractors to be merged into a single cluster. Both parameters are complementary and are directly used to build the final clustering results. These two parameters must be simultaneously controlled and can be initialized by the average values of densities and distances for all data objects. To configure θ , we recommend to evaluate several values which are near to the average of density for all data objects. Concerning the choice of the parameter VTOL,

we recommend to compute the average of distances between the pair of data objects and evaluate several values by slightly varying the average value. Small values of θ and VTOL will generate several clusters whereas large values force the algorithm to create a very small number of clusters. We also note that the value of θ is dependent to the value of density smoothing σ . We show in Figure 8 how the value of θ can be chosen in Iris dataset given the different values of σ . We show that small values of the smoothing parameter σ requires more large values θ in order to identify the set of 3 clusters in Iris dataset.

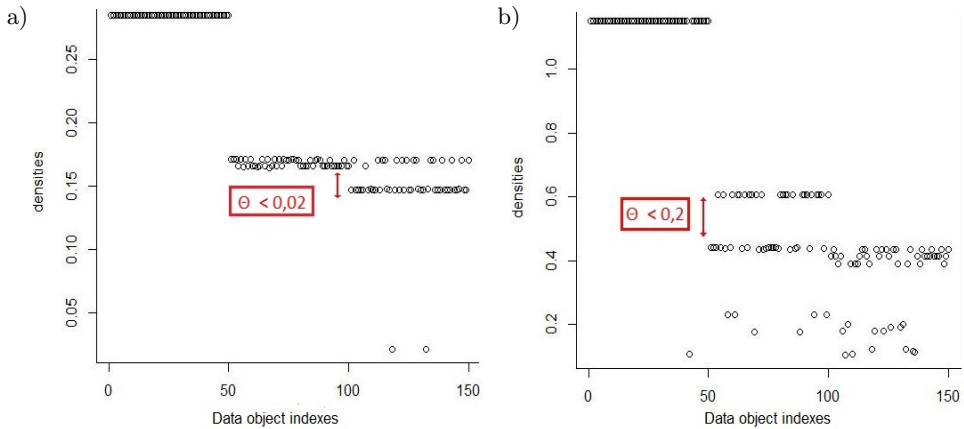


Figure 8. Initialization of the value of θ given the value of the smoothing parameter σ in order to detect a minimal number of 3 clusters in the Iris dataset: a) density value of each data object for $\sigma = 0.34$; b) density value of each data object for $\sigma = 0.20$

5. Conclusion

In this paper, we focused on building disjoint and non-disjoint clusters having arbitrary and non-spherical shapes. We show the limit of all existing methods to build such types of clusters. The proposed density-based method solves this issue and looks for arbitrary clusters having different forms and shapes. We showed the ability of the proposed method to build both disjoint and non-disjoint clusters and has shown a good performance on both artificial and real multi-labelled datasets even when data contain non-spherical and arbitrary shapes. Another advantage of the proposed method compared to existing ones is the ability to give a good estimate of the number of clusters. This number is required to be initialized in prior for all existing methods.

An interesting future direction of the proposed work is to give an automatic initialization of the different parameters in order to look for the best combination of values giving the best clustering results. Another important direction for the improvement of this work is to improve the scalability of the proposed method in order to make possible building clusters on huge and big data. In fact, making calculations for

each data point is not obvious to achieve rapid results in an acceptable time, especially when it comes to operate on large and huge datasets. One could investigate the use of other heuristics such as the Genetic Algorithm to replace the highly consuming Hill Climbing algorithm. In addition, parallel frameworks would be beneficial for the scalability improvement of OC-DD which allows parallel and distributed processing of high computational steps.


References

- [1] Afridi M.K., Azam N., Yao J.: Variance based three-way clustering approaches for handling overlapping clustering, *International Journal of Approximate Reasoning*, vol. 118, pp. 47–63, 2020. doi: 10.1016/j.ijar.2019.11.011.
- [2] Amigó E., Gonzalo J., Artiles J., Verdejo F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints, *Information Retrieval*, vol. 12(4), pp. 461–486, 2009.
- [3] Ankerst M., Breunig M., Kriegel H.P., Sander J.: OPTICS: ordering points to identify the clustering structure, *ACM Sigmod Record*, vol. 28(2), pp. 49–60, 1999. doi: 10.1145/304181.304187.
- [4] Banerjee A., Krumpelman C., Ghosh J., Basu S., Mooney R.: Model-based Overlapping Clustering. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, USA*, pp. 532–537, ACM, 2005.
- [5] Ben N’Cir C.E., Cleuziou G., Essoussi N.: Generalization of c -means for identifying non-disjoint clusters with overlap regulation, *Pattern Recognition Letters*, vol. 45, pp. 92–98, 2014.
- [6] Ben N’Cir C.E., Essoussi N., Limam M.: Kernel-Based Methods to Identify Overlapping Clusters with Linear and Nonlinear Boundaries, *Journal of Classification*, vol. 32(2), pp. 176–211, 2015. doi: 10.1007/s00357-015-9181-3.
- [7] Bertrand P., Janowitz M.F.: The k -weak Hierarchical Representations: An Extension of the Indexed Closed Weak Hierarchies, *Discrete Applied Mathematics*, vol. 127, pp. 199–220, 2003.
- [8] Celleux G., Govaert G.: A classification EM algorithm for clustering and two stochastic versions, *Computational Statistics and Data Analysis*, pp. 315–332, 1992.
- [9] Cheng Y.: Mean shift, mode seeking, and clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17(8), pp. 790–799, 1995. doi: 10.1109/34.400568.
- [10] Cleuziou G.: An extended version of the k -means method for overlapping clustering. In: *2008 19th International Conference on Pattern Recognition*, pp. 1–4, 2008. doi: 10.1109/ICPR.2008.4761079.
- [11] Cleuziou G., Moreno J.G.: Kernel methods for point symmetry-based clustering, *Pattern Recognition*, vol. 48(9), pp. 2812–2830, 2015.

- [12] Depril D., Van Mechelen I., Mirkin B.: Algorithms for additive clustering of rectangular data tables, *Computational Statistics & Data Analysis*, vol. 52(11), pp. 4923–4938, 2008.
- [13] Diday E.: Orders and overlapping clusters by pyramids, 1987. Technical Report 730, INRIA, France. <https://hal.inria.fr/inria-00075822>.
- [14] Ester X., Kriegel M., Xu X.: Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. In: *Advances in Spatial Databases. SSD 1995, Lecture Notes In Computer Science*, vol. 951, pp. 67–82, Springer, Berlin–Heidelberg, 1995.
- [15] Fisher R.A.: The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, vol. 7(2), pp. 179–188, 1936.
- [16] Fukunaga K., Hostetler L.: The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory*, vol. 21(1), pp. 32–40, 1975.
- [17] Hinneburg A., Gabriel H.H.: Denclue 2.0: Fast Clustering Based on Kernel Density Estimation. In: *In Proceedings of the 7th International Symposium on Intelligent Data Analysis*, pp. 70–80, 2007.
- [18] Hinneburg A., Keim D.: An efficient approach to clustering large multimedia databases with noise. In: *KDD'98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 58–65, 1998.
- [19] Jain A., Dubes R.: *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [20] Jardine N., Sibson R.: *Mathematical Taxonomy*, John Wiley and Sons Ltd., London, 1971.
- [21] Khanmohammadi S., Adibeig N., Shanehbandy S.: An improved overlapping k -means clustering method for medical applications, *Expert Systems with Applications*, vol. 67, pp. 12–18, 2016. doi: 10.1016/j.eswa.2016.09.025.
- [22] Lee S.H., Jeong Y.S., Kim J.Y., Jeong M.K.: A new clustering validity index for arbitrary shape of clusters, *Pattern Recognition Letters*, vol. 112, pp. 263–269, 2018. doi: 10.1016/j.patrec.2018.08.005.
- [23] Lutov A., Khayati M., Cudré-Mauroux P.: Accuracy Evaluation of Overlapping and Multi-Resolution Clustering Algorithms on Large Datasets. In: *IEEE International Conference on Big Data and Smart Computing, BigComp 2019, Kyoto, Japan, February 27 – March 2, 2019*, pp. 1–8, IEEE, 2019. doi: 10.1109/BIGCOMP.2019.8679398.
- [24] MacQueen J.B.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, 1967.
- [25] Maiza M.I., Ben N'Cir C.E., Essoussi N.: Overlapping Community Detection Method for Social Networks. In: R. Jallouli, O.R. Zaïane, M.A. Bach Tobji, R. Srarfi Tabbane, A. Nijholt (eds.), *Digital Economy. Emerging Technologies and Business Innovation*, pp. 143–151, Springer International Publishing, 2017.

- [26] Mirzaie M., Barani A., Nematbakhsh N., Mohammad-Beigi M.: Bayesian-OverDBC: A Bayesian Density-Based Approach for Modeling Overlapping Clusters, *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [27] Sharan R., Shamir R.: CLICK: a clustering algorithm with applications to gene expression analysis. In: *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, vol. 8, pp. 307–316, 2000.
- [28] Wang M., Zuo W., Wang Y.: An improved density peaks-based clustering method for social circle discovery in social networks, *Neurocomputing*, vol. 179, pp. 219–227, 2016.
- [29] Zhou X., Liu Y., Wang J., Li C.: A density based link clustering algorithm for overlapping community detection in networks, *Physica A: Statistical Mechanics and its Applications*, vol. 486, pp. 65–78, 2017.

Affiliations

Chiheb-Eddine Ben Ncir 

University of Jeddah, College of Business, Saudi Arabia & University of Tunis,
LARODEC Laboratory, Tunisia. chiheb.benncir@isg.rnu.tn,
ORCID ID: <https://orcid.org/0000-0003-4014-8264>

Received: 09.10.2020

Revised: 27.11.2020

Accepted: 29.12.2020