# Local contrast as an effective means to robust clustering against varying densities

**Bo Chen**[1] · **Kai Ming Ting**[1,2] · **Takashi Washio**[3] · **Ye Zhu**[4]

Received: 19 March 2017 / Accepted: 15 December 2017 / Published online: 23 January 2018
© The Author(s) 2018

**Abstract** Most density-based clustering methods have difficulties detecting clusters of hugely different densities in a dataset. A recent density-based clustering CFSFDP appears to have mitigated the issue. However, through formalising the condition under which it fails, we reveal that CFSFDP still has the same issue. To address this issue, we propose a new measure called Local Contrast, as an alternative to density, to find cluster centers and detect clusters. We then apply Local Contrast to CFSFDP, and create a new clustering method called LC-CFSFDP which is robust in the presence of varying densities. Our empirical evaluation shows that LC-CFSFDP outperforms CFSFDP and three other state-of-the-art variants of CFSFDP.

**Keywords** Local contrast · Density-based clustering · Varying densities

Editors: Kurt Driessens, Dragi Kocev, Marko Robnik-Šikonja, Myra Spiliopoulou.

✉ Bo Chen
bo.chen@monash.edu

Kai Ming Ting
kaiming.ting@federation.edu.au

Takashi Washio
washio@ar.sanken.osaka-u.ac.jp

Ye Zhu
ye.zhu@deakin.edu.au

[1] Faculty of Information Technology, Monash University, Clayton, VIC 3168, Australia

[2] School of Engineering and Information Technology, Federation University Australia, Churchill, VIC 3842, Australia

[3] The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibarakishi, Osaka 5670047, Japan

[4] School of Information Technology, Deakin University, Burwood, VIC 3125, Australia

## 1 Introduction

Data clustering is a technique to group a dataset into a number of subsets based on a "natural" hidden data sturcture (Cherkassky and Mulier 2007). To capture the underlying data structures, traditional clustering techniques such as the Expectation–Maximization (EM) algorithm (Dempster et al. 1977) assumes specific probability distributions as the source from which the dataset is generated. In comparison, density-based methods are attractive due to their non-parametric characteristic which enables them to deal with arbitrary shaped clusters (Jain 2010). They rely on spatially varying densities for the detection of clusters. High density regions are identified as clusters which are separated by regions of low density (Han and Kamber 2011).

However, most density-based methods have difficulties to detect all clusters when the clusters have large variations of densities (Ertöz et al. 2003a; Zhu et al. 2016). For example, DBSCAN (Ester et al. 1996), which uses a global density threshold to discriminate cluster core points from noise, fails to identify all clusters in the presence of greatly varying densities (Zhu et al. 2016).

Many efforts have been devoted to solve the varying densities problem in DBSCAN-like algorithms. Shared-Nearest-Neighbours (SNN) (Jarvis and Patrick 1973; Ertöz et al. 2003a) is an effective technique to this end. It uses the number of shared nearest neighbours between two points as a similarity measure to replace distance in the clustering procedure. Yet, the performance of SNN is sensitive to the number of nearest neighbours used in its similarity calculations (Brito et al. 1997; Ertöz et al. 2003; Tan and Wang 2013). ReScale (Zhu et al. 2016) is a recently proposed approach to tackle the same problem. It rescales a dataset such that the estimated density of a rescaled data point approximates the density ratio of the correspond point in the original dataset. However, ReScale does not perform well when clusters overlap significantly on some attributes (Zhu et al. 2016).

A more recent density-based clustering method called Clustering by Fast Search and Find of Density Peaks (CFSFDP) (Rodriguez and Laio 2014) employs a density-based approach different from DBSCAN for clustering. Instead of finding core points using a global threshold in the first step, it finds the density peak of every cluster and then links the neighboring points of each peak to form a cluster. CFSFDP overcomes some issues of varying densities of earlier density-based clustering algorithms (e.g., DBSCAN).

While the condition under which DBSCAN fails to detect all clusters has been formalised recently (Zhu et al. 2016), whether such a condition exists in the more recent density-based method CFSFDP is still unknown. We formalise a necessary condition for CFSFDP to identify all clusters in a dataset, and show that large variation of densities is still problematic for CFSFDP.

We propose a new measure called Local Contrast (LC), as an alternative to density, to make density-based clustering algorithms robust against varying densities. The proposed LC is not too sensitive to its parameter setting, and is able to achieve high clustering performance with a default setting.

Though the proposed LC is built on top of a density estimator, it has the following unique theoretical properties:

- The local modes and local minima of the LC distribution are also the local modes and local minima of the density distribution of the same dataset.
- The local modes of LC have the same constant LC value, irrespective of the density values of the local modes.

- The local minima of LC have zero LC value, irrespective of their density values of the local minima.

We utilise LC to create a new version of CFSFDP, named LC-CFSFDP. We show that the new clustering method LC-CFSFDP is more robust against varying densities than CFSFDP.

To benchmark the proposed LC-CFSFDP, we apply SNN and ReScale (which are existing remedies for the density variation issue) to CFSFDP, creating two improved variants called SNN-CFSFDP and ReScale-CFSFDP. Together with the original CFSFDP and its latest improvement called FKNN-DPC (Xie et al. 2016), the four methods are used as contestants against LC-CFSFDP in our experiments. Our empirical evaluation shows that LC-CFSFDP outperforms all four contestants in 18 benchmark datasets.

The rest of the paper is organised as follows. Section 2 presents the related work. Section 3 discusses the weakness of CFSFDP and how to use existing remedies to improve it. Section 4 proposes Local Contrast and shows its properties. Section 5 presents LC-CFSFDP. The empirical evaluation, discussion and conclusions are provided in the last three sections.

## 2 Related work

Density-based clustering methods such as DBSCAN (Ester et al. 1996) identify high density (core) points using a global threshold and then link all neighbouring core points to form clusters. However, these methods are known to have one key issue, i.e., they have difficulties detecting all clusters when the clusters have large density variations (Ertöz et al. 2003a). Recent research has formalised a necessary condition for DBSCAN to detect all clusters in a dataset (Zhu et al. 2016): if the peak of some cluster has a density lower than that of a low-density region between clusters, then DBSCAN will fail to find all clusters. Many density-based clustering algorithms (Hinneburg and Gabriel 2007; Ram et al. 2009; Borah and Bhattacharyya 2008), like DBSCAN, use a global density threshold to define core points and links them to form clusters. All these algorithms have the same issue. The exact condition under which these density-based algorithms fail (Zhu et al. 2016) is provided in Appendix A for ease of reference.

Researchers have attempted to address the issue of density-based clustering using different approaches. For instance, Shared-Nearest-Neighbours (SNN) (Jarvis and Patrick 1973; Ertöz et al. 2003a) employs an alternative similarity measure to replace the distance measure in the clustering procedure. The similarity between two data points is either the number of their shared $K$-nearest-neighbours (if they have each other in their $K$-nearest-neighbour lists) or 0 otherwise. Since the SNN similarity measure takes into account the local distribution of the data points, it is less affected by varying densities of different clusters. It has been shown that DBSCAN which uses SNN improves the clustering results of DBSCAN which uses the distance measure (Ertöz et al. 2003a). However, its performance is sensitive to the setting of parameter $K$ and its time complexity is $O(K^2N^2)$, instead of $O(N^2)$ for many other density-based clustering methods such as DBSCAN, because of an additional KNN process is required (Zhu et al. 2016).

ReScale (Zhu et al. 2016) is another technique that is recently proposed to overcome the density variation problem in clustering. This technique is a pre-processing technique and is originally designed for a density-based clustering algorithm which uses a global density threshold to identify clusters. ReScale enables existing density-based clustering algorithms to perform density-ratio-based clustering, i.e., clusters are defined as regions of *locally* high density separated by regions of *locally* low density. The aim is to rescale the data such that the estimated density of each rescaled point is approximately the estimated density-ratio of the corresponding point in the original space, where density-ratio is defined as a ratio of the density of a point and the average density over its $\eta$-neighbourhood. A point located at a maximum local density area has higher density-ratio value than that of a point located at a minimum local density area. Thus, a density-based clustering algorithm can be applied without modification to the rescaled data which uses a single threshold to identify all clusters of locally high densities. Two additional parameters are introduced—$\eta$ is used to define the local neighbourhood; and $\psi$ is used to control the precision of $\eta$-neighbourhood density estimation.

A recent density-based clustering algorithm, CFSFDP (Rodriguez and Laio 2014), takes a different approach to reduce the effect of the above-mentioned issue. The idea is to find cluster centres which have higher density than their neighbours and are relatively distant from each other. CFSFDP mitigates the problem of varying densities in some situations because it finds cluster centres not only by high densities, but also by taking into account their distances from other centres. It can detect low-density cluster centre if it is far from other clusters.

The Fuzzy weighted $K$-Nearest-Neighbors Density Peak Clustering (FKNN-DPC) (Xie et al. 2016) is a recent effort to improve CFSFDP (Rodriguez and Laio 2014). It uses a similar procedure as CFSFDP, except the density estimation phase and the cluster assignation phase. FKNN-DPC uses a KNN kernel estimator, instead of a $\epsilon$-neighbourhood estimator. The key difference lies in the cluster assignation phase: FKNN-DPC uses a complex assignation scheme consists of 2 strategies based on a series of KNN searches. The heavy use of KNN searches makes the algorithm very sensitive to the $K$ parameter. It does not overcome the problem in clusters having hugely varying densities from the root cause because its operation is still based on density, as mentioned in the last paragraph.

It is important to point out that the above improvement over CFSFDP (Xie et al. 2016) was done on procedural steps only (which use a different density estimator and a different scheme to assign points to a cluster), without knowing the root cause.

In this paper, we focus on CFSFDP (Rodriguez and Laio 2014) because it is a powerful and state-of-the-art core method of density-based clustering (Xu and Tian 2015); and we want to identify the key weakness of CFSFDP and its root cause. To achieve this aim, we first formalise the condition under which CFSFDP fails to detect all clusters in a dataset; and reveal that large density variations in a dataset can still harm CFSFDP's clustering performance significantly under some situations. Then, we propose a new measure called *Local Contrast*, in place of density, as the primary means to find clusters. We show that this can be easily done using almost the same procedure as CFSFDP; and this overcomes CFSFDP's weakness from the root cause.

## 3 Weakness of CFSFDP and current remedies

Here we first provide a necessary condition for CFSFDP to detect all clusters in a dataset in Sect. 3.1. Its violation will result in CFSFDP failing to detect all clusters. In Sect. 3.2, we create two variants of CFSFDP with existing remedies in tackling the problem of cluster density variations: SNN and ReScale. We show the limitations of these remedies for CFSFDP in the last subsection.

### 3.1 A necessary condition for CFSFDP

Like most density-based methods, CFSFDP (Rodriguez and Laio 2014) employs a density estimator $f(\mathbf{x})$ to estimate densities for all $\mathbf{x}$ in a dataset $D$. The density estimator is defined as follows:

**Table 1** CFSFDP versus LC-CFSFDP: key steps

| Step | CFSFDP | LC-CFSFDP |
|---|---|---|
| 1 | Estimate density $f(\mathbf{x})$ and distance $\delta_f(\mathbf{x}),\ \forall \mathbf{x} \in D$ | Estimate density $f(\mathbf{x})$, Local Contrast $LC(\mathbf{x})$ and distance $\delta_{LC}(\mathbf{x}),\ \forall \mathbf{x} \in D$ |
| 2 | Select the top $M$ points with the largest $f(\mathbf{x}) \times \delta_f(\mathbf{x})$ and and label them as cluster centres of Clusters $1, \ldots, M$ | Select the top $M$ points with the largest $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$ points and label them as cluster centres of Clusters $1, \ldots, M$ |
| 3 | Order all points in descending order of $f(\mathbf{x})$. Following the order, assign each unlabelled point to the same cluster of its nearest neighbour with higher $f(\mathbf{x})$ | Order all points in descending order of $LC(\mathbf{x})$. Following the order, assign each unlabelled point to the same cluster of its nearest neighbour with higher $LC(\mathbf{x})$ |

$$f(\mathbf{x}) = |\{\mathbf{y} \in D \mid d(\mathbf{x}, \mathbf{y}) < \epsilon\}|,$$

where $d(\cdot, \cdot)$ is a distance measure and $\epsilon$ is a cut-off distance; and $|Q|$ is the cardinality of set $Q$.

Let $\mathbf{x}^m = \arg\max_{\mathbf{x} \in D} f(\mathbf{x})$ denote the point with the global maximum density. CFSFDP (Rodriguez and Laio 2014) defines a distance function of $\mathbf{x}$, $\delta_f(\mathbf{x})$, as follows:

$$\delta_f(\mathbf{x}) = \begin{cases} \min\limits_{f(\mathbf{y}) > f(\mathbf{x})} d(\mathbf{x}, \mathbf{y}), & \forall \mathbf{x} \in D \backslash \{\mathbf{x}^m\} \\ \max\limits_{\mathbf{y} \in D} d(\mathbf{x}, \mathbf{y}), & \text{if} \quad \mathbf{x} = \mathbf{x}^m. \end{cases}$$

In other words, $\delta_f(\mathbf{x})$ is the distance between $\mathbf{x}$ and its nearest neighbour with a higher density; except that for the point with the global maximum density, $\delta_f(\mathbf{x})$ is the greatest distance between any point and itself. This is to make sure that for the point with the global maximum density, it will always be ranked first in the ranked list of $f(\mathbf{x})\delta_f(\mathbf{x})$ sorted in descending order.

The user then selects the top $M$ points from the ranked list of $f(\mathbf{x})\delta_f(\mathbf{x})$, and label them from 1 to $M$, as the centres for $M$ clusters.

All points are then sorted in descending order of $f(\mathbf{x})$. One by one from top to bottom of the sorted list, each unlabeled point is assigned to the same cluster of its nearest neighbour with a higher density. The first column in Table 1 provides a summary of the key steps in the CFSFDP procedure.

CFSFDP requires that these cluster modes must be ranked at the top in the sorted list of $f(\mathbf{x})\delta_f(\mathbf{x})$ if they are to be selected as cluster centres.

We now state the necessary condition for CFSFDP to identify all clusters of a dataset.

**Theorem 1** *Given a dataset $D$ of $M$ actual clusters, let $\mathbb{C} = \{\mathbf{c}_m, m = 1, \ldots, M\}$ denote the $M$ cluster modes, i.e., the points with the maximum density in each cluster with respect to a density estimator $f(\mathbf{x})$. A necessary condition for CFSFDP to correctly identify all clusters is given as follows:*

$$\min_{\mathbf{x} \in \mathbb{C}} f(\mathbf{x})\delta_f(\mathbf{x}) > \max_{\mathbf{y} \in D \backslash \mathbb{C}} f(\mathbf{y})\delta_f(\mathbf{y}). \tag{1}$$

*Proof* A violation of Eq. (1) means that at least one point $\mathbf{z} \in \mathbb{C}$ is not among the top $M$ points in the sorted list of $f(\mathbf{x})\delta_f(\mathbf{x})$. Then, one of the following three situations will occur:

(i) If less than $M$ points are selected as cluster representatives, then not all clusters are identified.

(ii) If more than $M$ points are selected as cluster representatives, then some cluster will be divided.

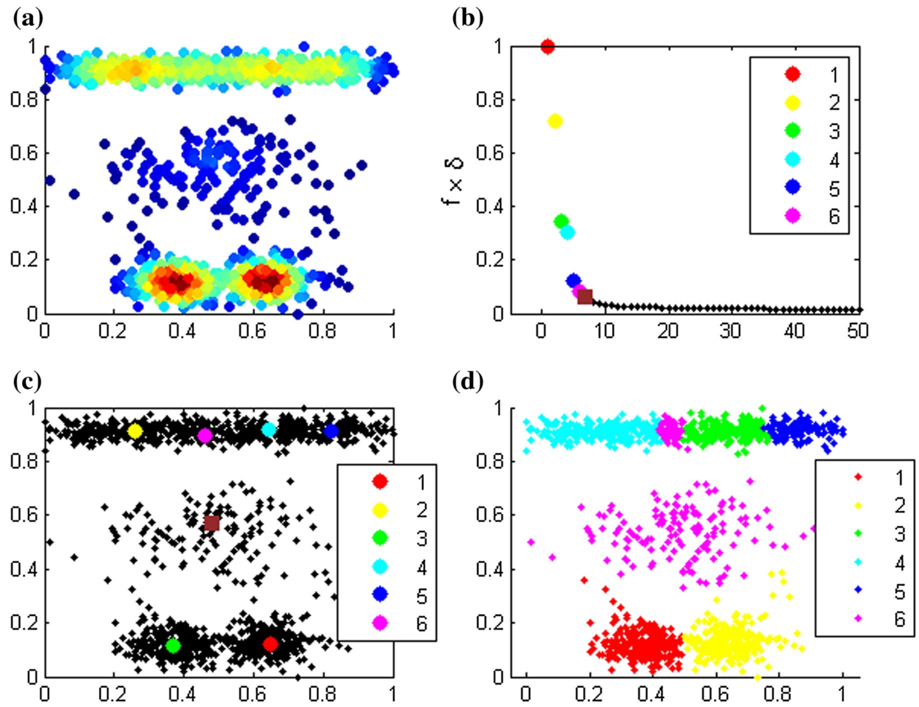

**Fig. 1** The clustering result of CFSFDP on the synthetic dataset. Note that the brown square marker in **c** denotes the density peak of the middle cluster, which is ranked 7th in the Decision Graph shown in **b**. It is not selected in the final result shown in **d** because selecting 6 representatives produces the best $F$-measure. **a** Density distribution, **b** CFSFDP decision graph, **c** density peaks, **d** clustering result, F = 0.84821 (Color figure online)

(iii) If exactly $M$ points are selected as cluster representatives, then point $\mathbf{z} \in \mathbb{C}$ is not selected as a representative. As a result, $\mathbf{z}$ will be assigned a label from a point with a higher density. Since $\mathbf{z}$ is the density maximum in its own cluster, the point that $\mathbf{z}$ links to can not be from the same cluster. Hence, $\mathbf{z}$ and its neighbouring points will be mislabelled as belonging to a different cluster.

In all the above cases, CFSFDP can not correctly identify all clusters having violated Eq. (1). □

Note that the condition provided in Theorem 1 is independent of the density estimator used.

The basic assumptions of CFSFDP are that (i) each cluster centre has the maximum density among all points within the cluster, and (ii) all cluster centres are well separated. While these two assumptions are usually true, the maximum densities of different clusters can not be guaranteed to be the same, or even similar.

Because density can not provide such a guarantee, the use of density becomes the root cause of CFSFDP's weakness in detecting all clusters having hugely different densities. When clusters have significantly different densities, low density centres which have no sufficient long distance $\delta_f(\cdot)$ will be ranked low in the sorted list of $f(\mathbf{x})\delta_f(\mathbf{x})$. As a result, the algorithm fails to correctly identify all clusters. An example is shown in Fig. 1. The top dense cluster has multiple peaks; and the centre of the sparse cluster has significantly lower density than these peaks. CFSFDP fails to detect the 4 clusters correctly because the mode of the sparse cluster has density which is too low for the mode to be ranked in the top four in the sorted list of $f(\mathbf{x})\delta_f(\mathbf{x})$, shown in Fig. 1b.

To overcome this weakness, we provide an alternative to density which has the necessary properties to detect all clusters of different densities using the exactly the same CFSFDP procedure. This alternative measure will be introduced in Sect. 4; and our analysis in Sect. 5 shows that the alternative measure is more robust than density in a dataset having clusters of different densities using the same CFSFDP procedure.

### 3.2 Improving CFSFDP using existing methods of improving DBSCAN

SNN (Jarvis and Patrick 1973; Ertöz et al. 2003a) and ReScale (Zhu et al. 2016) are two existing methods designed to address the issue of DBSCAN-like clustering methods in datasets having huge density variations.

One can use either of these existing methods to improve CFSFDP. These can be applied straightforwardly. The following two subsections provide the details of two modified versions of CFSFDP: SNN-CFSFDP and ReScale-CFSFDP.

#### *3.2.1 SNN-CFSFDP*

SNN-CFSFDP has the same procedure of CFSFDP except that the distance measure used in both $f(\cdot)$ and $\delta_f(\cdot)$ is replaced with the shared nearest neighbour dissimilarity measure (Ertöz et al. 2003a).

Let $N_K(\mathbf{x})$ denote the $K$ nearest neighbours of $\mathbf{x}$ in a dataset $D$, with respect to Euclidean distance. The shared nearest neighbour dissimilarity (SNN) of two points $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$SNN(\mathbf{x},\mathbf{y}) = \begin{cases} 1 - |N_K(\mathbf{x}) \cap N_K(\mathbf{y})|/K, & \text{if } \mathbf{y} \in N_K(\mathbf{x}) \text{ and } \mathbf{x} \in N_K(\mathbf{y}) \\ 1, & \text{otherwise.} \end{cases}$$

SNN-CFSFDP then calculates both the density $f_{SNN}(\mathbf{x})$ and the nearest distance to a higher density point $\delta_{f_{SNN}}(\mathbf{x})$ in terms of $SNN$ dissimilarity as follows:

$$f_{SNN}(\mathbf{x}) = |\{\mathbf{y} \in D \mid SNN(\mathbf{x},\mathbf{y}) < \epsilon\}|,$$

and

$$\delta_{f_{SNN}}(\mathbf{x}) = \begin{cases} \min\limits_{f_{SNN}(\mathbf{y}) > f_{SNN}(\mathbf{x})} SNN(\mathbf{x},\mathbf{y}), & \forall \mathbf{x} \in D \backslash \{\mathbf{x}^w\} \\ \max\limits_{\mathbf{y} \in D} SNN(\mathbf{x},\mathbf{y}), & \text{if} \quad \mathbf{x} = \mathbf{x}^w, \end{cases}$$

where $\epsilon$ is the cut-off $SNN$ dissimilarity and $\mathbf{x}^w = \arg\max_{\mathbf{x} \in D} f_{SNN}(\mathbf{x})$.

Given $f_{SNN}(\mathbf{x})$ and $\delta_{f_{SNN}}(\mathbf{x})$, the rest of the procedure is the same as CFSFDP. The summary of the key steps is given in the second column of Table 2. Note that if the procedure is implemented with an input of a dissimilarity matrix, the $SNN$ dissimilarity matrix can be computed in a pre-processing step. This is shown in step 0 in Table 2.

#### *3.2.2 ReScale-CFSFDP*

ReScale-CFSFDP pre-processes the dataset before utilizing the exact same procedure of CFSFDP. ReScale first estimates the density distribution on each dimension of the dataset $D$, with an $\eta$-neighbourhood estimator and a resolution of $\psi$. It then scales the dataset $D$ along each dimension based on the cumulative distribution, to yield a new dataset $D'$.

**Table 2** SNN-CFSFDP versus ReScale-CFSFDP: key steps of the two algorithms

| Step | SNN-CFSFDP | ReScale-CFSFDP |
|---|---|---|
| 0 | Calculate pairwise dissimilarity $SNN(\mathbf{x}, \mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in D$ | Pre-process $D \to D'$ with the ReScale approach |
| 1 | Estimate density $f_{SNN}(\mathbf{x})$ and distance $\delta_{f_{SNN}}(\mathbf{x}),\ \forall \mathbf{x} \in D$ | Estimate density $f(\mathbf{x}')$ and distance $\delta_f(\mathbf{x}'),\ \forall \mathbf{x}' \in D'$ |
| 2 | Select the top $M$ points with the largest $f_{SNN}(\mathbf{x}) \times \delta_{f_{SNN}}(\mathbf{x})$ points and label them as cluster centres of Clusters $1, \ldots, M$ | Select the top $M$ points with the largest $f(\mathbf{x}') \times \delta_f(\mathbf{x}')$ and label them as cluster centres of Clusters $1, \ldots, M$ |
| 3 | Order all points in descending order of $f_{SNN}(\mathbf{x})$. Following the order, assign each unlabelled point to the same cluster of its nearest neighbour with higher $f_{SNN}(\mathbf{x})$ | Order all points in descending order of $f(\mathbf{x}')$. Following the order, assign each unlabelled point to the same cluster of its nearest neighbour with higher $f(\mathbf{x}')$ |

Let $D_i$, $\mathbf{x}_i$ denote the $i$-th attribute of dataset $D$ and data point $\mathbf{x}$, respectively. For each attribute $i$, ReScale divides the range of $D_i$ into $\psi$ equal segments, yielding $\psi + 1$ grid points $s_j$, $j = \{1, \ldots, \psi + 1\}$ and $s_q > s_j$, for all $q > j$. It then estimates the densities of $s_j$ by following,

$$f(s_j) = |\{\mathbf{x} \in D \mid (s_j - \eta) < \mathbf{x}_i \leq (s_j + \eta)\}|.$$

The value of the $i$-th attribute of a transformed point $\mathbf{x}'$ is then given by

$$\mathbf{x}'_i = \sum_{j=1}^{\psi+1} f(s_j) I_{\{\mathbf{x}_i \geq s_j\}},$$

which is the cumulative marginal probability of $\mathbf{x}$ on attribute $i$. After procesing each attribute, ReScale normalises the transformed dataset $D'$ to be in [0, 1]. The detailed algorithm can be found in Zhu et al. (2016).

Using $D'$, the rest of the procedure is the same as CFSFDP. The key steps of ReScale-CFSFDP are given in the last column of Table 2.

### *3.2.3 Limitations of SNN-CFSFDP and ReScale-CFSFDP*

We apply SNN-CFSFDP and ReScale-CFSFDP on the synthetic dataset as shown in Fig. 1, and their clustering results are given in Figs. 2 and 3, respectively. Though both methods improve the $F$-measure compared to the original CFSFDP, they still fail to correctly identify all clusters: SNN-CFSFDP splits the two dense clusters at the bottom into four clusters; ReScale-CFSFDP splits the top cluster into two clusters.

SNN has two weaknesses. First, it is sensitive the $K$ parameter (Brito et al. 1997; Ertöz et al. 2003; Tan and Wang 2013). Second, with a time complexity of $O(K^2N^2)$, it is computationally expensive. In this example, the default setting of $K = \sqrt{N}$ leads to an undesirable result as shown in Fig. 2, in which the true peaks #5 and #6 in Fig. 2c can not out-rank false peak #2, because the distance $\delta$ based on SNN dissimilarity is not large enough. A proper $K$ needs to be carefully tuned in order to produce the desired clustering outcome. We will provide further analysis of this issue in Sect. 6.

The ReScale approach aims to transform the dataset to be uniformly distributed along each attribute. However, when clusters overlap significantly on some attribute(s), it becomes problematic as exemplified in Fig. 3: when projected onto the x-axis, because of the overlapping

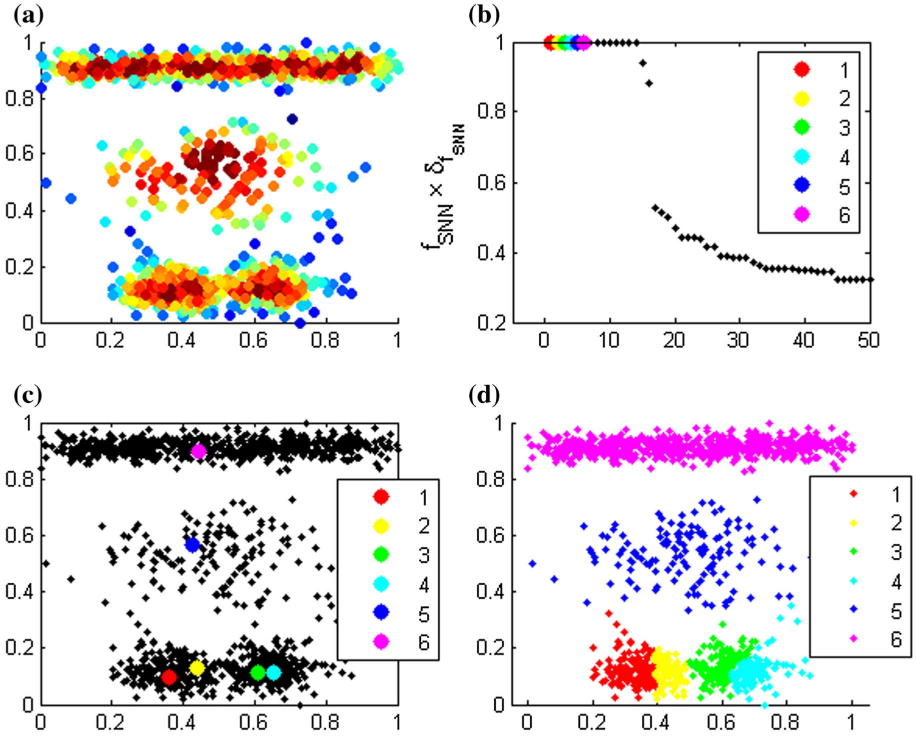

**Fig. 2** The clustering result of SNN-CFSFDP on the synthetic dataset. The parameter $K$ is fixed to $\sqrt{N}$. $\epsilon$ and $M$ are searched for the best $F$-measure. **a** Density distribution, **b** SNN-CFSFDP decision graph, **c** density peaks, **d** clustering result, F = 0.88197

of clusters along x-axis, there are abundant data points in the middle and fewer data points at each end. The ReScale approach therefore shifts data points from the middle to both ends, causing the upper cluster to have two dense regions at both ends after the transformation. A rotation of the dataset is proposed in Zhu et al. (2016) to remedy this weakness. However, without prior knowledge of the dataset, it is difficult to find an orientation that works well, if such an orientation exists.

## 4 Local contrast

We propose Local Contrast as a new remedy for the density variation problem in clustering. Unlike SNN or ReScale, it is not sensitive to the parameter $K$, nor does it need to rescale the dataset.

Here we provide the definition of Local Contrast and describe its properties which empower clustering algorithms to be more robust against varying densities.

Given a dataset $D$ and a density estimator $f(\cdot)$, we define Local Contrast as follows:

**Definition 1** Local Contrast of an instance $\mathbf{x}$ is defined as the number of times that $\mathbf{x}$ has higher density than its $K$ nearest neighbours:

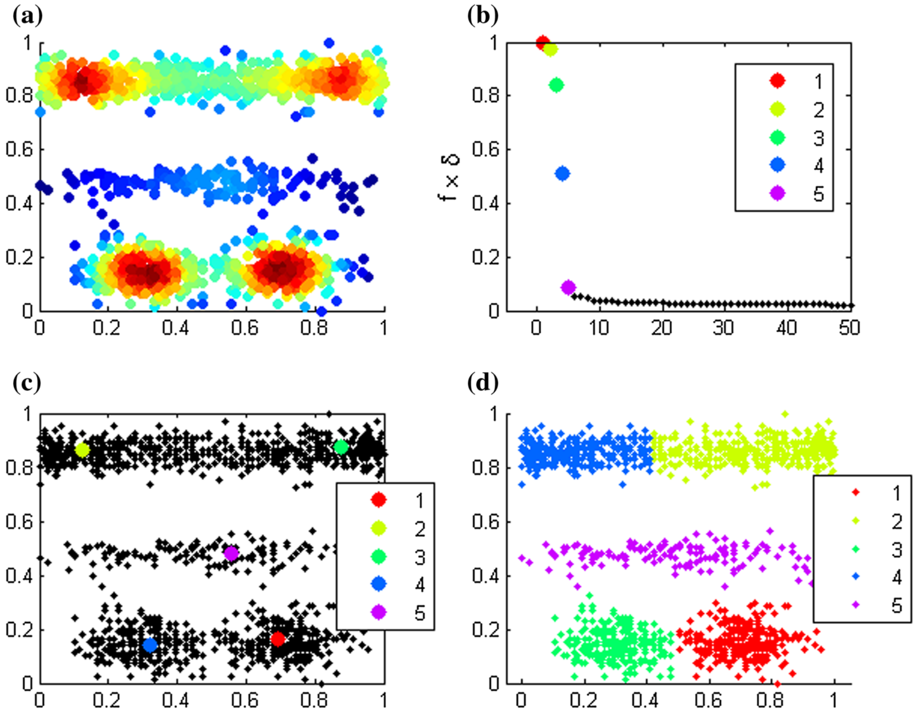

**Fig. 3** The clustering result of ReScale-CFSFDP on the synthetic dataset. The distribution shown is for the ReScaled dataset $D'$. $\psi$ and $\eta$ are fixed to 100 and 0.2 respectively while $\epsilon$ and $M$ are searched for the best $F$-measure. **a** Density distribution, **b** ReScale-CFSFDP decision graph, **c** density peaks, **d** clustering result, F = 0.92721

$$LC(\mathbf{x}) = \sum_{\mathbf{y} \in N_K(\mathbf{x})} I_{\{f(\mathbf{x}) > f(\mathbf{y})\}}$$

where $N_K(\mathbf{x})$ is the set of $K$ nearest neighbours of $\mathbf{x}$ and $I_{\{\cdot\}}$ is an indicator.

Local Contrast has three properties.

**Property 1** *The local modes and the local minima of $LC(\mathbf{x})$ are also the local modes and the local minima of $f(\mathbf{x})$, with a proper choice of $K$.*

**Property 2** *The local modes of $LC(\mathbf{x})$ that correspond to the local modes of $f(\mathbf{x})$ have $LC(\mathbf{x}) = K$, irrespective of the density of $f(\mathbf{x})$.*

**Property 3** *The local minima of $LC(\mathbf{x})$ that correspond to the local minima of $f(\mathbf{x})$ have $LC(\mathbf{x}) = 0$, irrespective of the density of $f(\mathbf{x})$.*

*Proof of Properties 1, 2 and 3.* Let $\mathbf{p}$ and $\mathbf{q}$ be the local density maxima and minima, respectively. Assuming a proper choice of $K$ exists such that for all $\mathbf{x} \in N_K(\mathbf{p})$, $f(\mathbf{p}) > f(\mathbf{x})$; and for all $\mathbf{x} \in N_K(\mathbf{q})$, $f(\mathbf{q}) < f(\mathbf{x})$.

Let $G \subseteq N_K(\mathbf{p})$ be the maximal subset of $N_K(\mathbf{p})$ such that for all $\mathbf{x} \in G$, $\mathbf{p} \in N_K(\mathbf{x})$. In other words, $\mathbf{p}$ is one of the $K$-nearest-neighbours of each member of $G$. Since $G$ is a subset of $N_K(\mathbf{p})$, $\mathbf{p}$ is also a local density maxima in the neighbourhood defined by $G$.

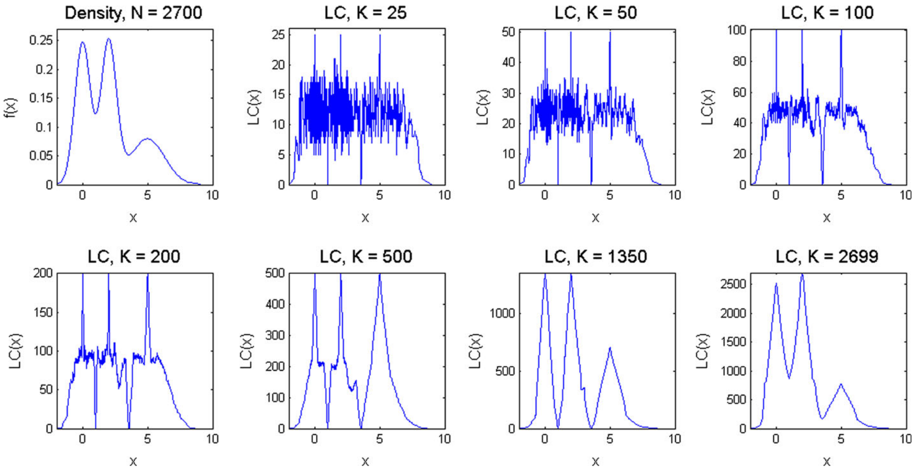

**Fig. 4** A dataset of size $N = 2700$ is drawn from a mixture of three univariate Gaussian sources. The distributions of density and LC (with different $K$ values) are shown

By Definition 1, we have

$$LC(\mathbf{p}) = \sum_{\mathbf{x} \in N_K(\mathbf{p})} I_{\{f(\mathbf{p}) > f(\mathbf{x})\}} = K \ ,$$

and for all $\mathbf{x} \in G$, we have

$$\begin{aligned} LC(\mathbf{x}) &= \sum_{\mathbf{y} \in N_K(\mathbf{x})} I_{\{f(\mathbf{x}) > f(\mathbf{y})\}} \\ &= \left( \sum_{\mathbf{y} \in (N_K(\mathbf{x}) \setminus \mathbf{p})} I_{\{f(\mathbf{x}) > f(\mathbf{y})\}} \right) + I_{\{f(\mathbf{x}) > f(\mathbf{p})\}} \\ &\leq K - 1 + 0 \\ &< K = LC(\mathbf{p}). \end{aligned}$$

Thus, $\mathbf{p}$ is also a local mode of $LC(\mathbf{x})$ in the neighbourhood defined by $G$.

Similarly, let $V \subseteq N_K(\mathbf{q})$ be the maximal subset of $N_K(\mathbf{q})$ such that for all $\mathbf{x} \in V$, $\mathbf{q} \in N_K(\mathbf{x})$. In other words, $\mathbf{q}$ is one of the $K$-nearest-neighbours of each member of $V$. $\mathbf{q}$ is also the local density minima in the neighbourhood defined by $V$.

The same argument follows that $LC(\mathbf{q}) = 0 < LC(\mathbf{x}), \forall \mathbf{x} \in V$. □

The properties of Local Contrast listed above depend on a proper choice of $K$ for a given dataset. The range of $K$ that can be used is usually large. In other words, Local Contrast is not too sensitive to the setting of $K$.

Figure 4 provides an illustration of the properties of LC. Note that $K$ can be set within the range of 25 and 500, Properties 2 and 3 still hold true; and Property 1 holds true for all settings of $K$ shown.

Throughout this paper, all experiments are done with the default setting $K = \sqrt{N}$, the square root of the dataset size, as suggested by some researchers for $K$ nearest neighbour procedures (Ferilli et al. 2008; Zitzler et al. 2004; Fukunaga 1990).

## 5 Improving CFSFDP with local contrast

We create a version of CFSFDP, called LC-CFSFDP, by replacing density with LC in the clustering procedure. Given a dataset $D$ and a density estimator $f(\cdot)$, $LC(\mathbf{x})$ is calculated as defined in Definition 1 for all $\mathbf{x}$ in $D$.

Given $LC(\cdot)$, $\delta_{LC}(\mathbf{x})$ is defined as follows:

$$\delta_{LC}(\mathbf{x}) = \begin{cases} \min\limits_{LC(\mathbf{y})>LC(\mathbf{x})} d(\mathbf{x},\mathbf{y}), & \forall \mathbf{x} \in D\backslash\{\mathbf{x}^{\omega}\} \\ \max\limits_{\mathbf{y}\in D} d(\mathbf{x},\mathbf{y}), & \text{if} \quad \mathbf{x} = \mathbf{x}^{\omega} \end{cases}$$

where $\mathbf{x}^{\omega} = \arg\max_{\mathbf{x}\in D} LC(\mathbf{x})$ denotes the point with the global maximum $LC$; and $d(\cdot,\cdot)$ is the Euclidean distance.

In other words, $\delta_{LC}(\mathbf{x})$ is defined to be the distance between $\mathbf{x}$ and its nearest neighbour with a higher $LC$, except when $\mathbf{x}$ is the point with the maximum $LC$. In that case, $\delta_{LC}(\mathbf{x})$ is defined to be the maximum distance between $\mathbf{x}$ and any point in $D$.

Here distance $\delta_{LC}(\mathbf{x})$ is analogous to the distance from a point's nearest neighbour with a higher density $\delta_f(\mathbf{x})$ used in CFSFDP (Rodriguez and Laio 2014). Given $LC(\mathbf{x})$ and $\delta_{LC}(\mathbf{x})$, cluster centres are then chosen from a decision graph where all points are sorted in descending order of $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$.

**Definition 2** Cluster centres are defined to be the top $M$ points with the highest $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$ values, where $M$ is a user input parameter.

After selecting $M$ cluster centres, all unlabeled data points are then assigned one by one in descending order of $LC$, with the same cluster label as its nearest neighbour with a higher $LC$. A contrast between the LC-CFSFDP and CFSFDP procedures is given in Table 1.

As shown in Table 1, LC-CFSFDP follows the same procedure of CFSFDP. The key difference between the two is that LC-CFSFDP replaces density with LC. As a result, analogous to the condition stated in Eq. (1), a necessary condition for LC-CFSFDP to detect all clusters correctly can be written as

$$\min_{\mathbf{x}\in\mathbb{C}} LC(\mathbf{x})\delta_{LC}(\mathbf{x}) > \max_{\mathbf{y}\in D\backslash\mathbb{C}} LC(\mathbf{y})\delta_{LC}(\mathbf{y}),$$

where $\mathbb{C}$ here denotes the set of points with maximum $LC$ in each cluster.

Let $\check{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{C}} LC(\mathbf{x})\delta_{LC}(\mathbf{x})$ and $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in D\backslash\mathbb{C}} LC(\mathbf{y})\delta_{LC}(\mathbf{y})$. The above condition can be rewritten as

$$\frac{LC(\check{\mathbf{x}})}{LC(\hat{\mathbf{y}})} > \frac{\delta_{LC}(\hat{\mathbf{y}})}{\delta_{LC}(\check{\mathbf{x}})}. \tag{2}$$

The corresponding rewritten condition for the density-based Eq. (1) is given as follows:

$$\frac{f(\acute{\mathbf{x}})}{f(\grave{\mathbf{y}})} > \frac{\delta_f(\grave{\mathbf{y}})}{\delta_f(\acute{\mathbf{x}})}. \tag{3}$$

where $\acute{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{C}} f(\mathbf{x})\delta_f(\mathbf{x})$ and $\grave{\mathbf{y}} = \arg\max_{\mathbf{y}\in D\backslash\mathbb{C}} f(\mathbf{y})\delta_f(\mathbf{y})$.

Equation (2) is much easier to satisfy than Eq. (3) because the properties of $LC$ ensures that every member of $\mathbb{C}$ has the maximum $LC$ value (i.e., Property 2 stated in Sect. 4), irrespectively of the density distribution. This makes the left side of Eq. (2) not less than 1. Thus Eq. (2) is harder to violate. In contrast, in a data distribution which has greatly varying densities between clusters, the left side of Eq. (3) could easily be smaller than 1, if $\acute{\mathbf{x}}$ is from a cluster of low density, which makes a violation of Eq. (3) a lot easier.

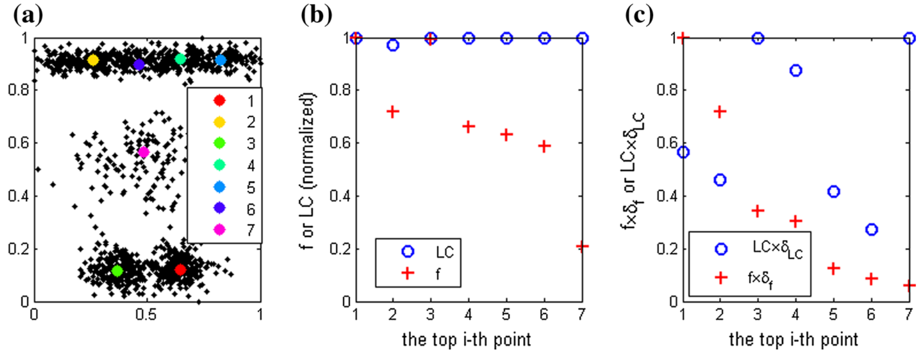

**Fig. 5** Top seven points as determined by CFSFDP is shown in **a**. **b** Shows that the densities of these top points vary hugely while their LCs (with $K = \sqrt{N} = \sqrt{1250}$) have similar values close to the maximum, due to Properties 1 and 2 of LC. **c** Compares the rankings of these points based on $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$ with those based on $f(\mathbf{x}) \times \delta_f(\mathbf{x})$. Note the huge change in rank positions of the seventh point, from rank #7 to rank #2. **a** Top 7 points in decision graph, **b** density versus LC graph, **c** f × $\delta_{\mathrm{f}}$ versus LC × $\delta_{\mathrm{LC}}$

To demonstrate this, we apply LC-CFSFDP on the same example dataset shown in Fig. 1. Figure 5 shows the result that CFSFDP has ranked the centre of the sparse cluster (rank #7) lower than the multiple peaks in the elongated cluster (ranks #2, 4, 5 and 6) in the decision graph. By simply replacing density $f(\cdot)$ with $LC(\cdot)$, LC-CFSFDP allows the centre of the sparse cluster to be ranked in the top four. This difference in ranking is the key of improving the algorithm because all peaks now have about the same LC values, by virtue of Properties 1 and 2, stated in the last section. As a result, the ranking of peaks due to $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$ is mainly influenced by $\delta_{LC}(\mathbf{x})$. Since multiple peaks in one cluster tend to have smaller $\delta_{LC}(\mathbf{x})$, the algorithm is more likely to select one peak from each cluster, which makes the algorithm more robust against significant density differences in the presence of multiple peaks in one cluster.

Figure 5a shows that the top seven points on the synthetic dataset, as determined by CFSFDP using density. Figure 5b shows that the normalised density and LC of these seven points. Figure 5c shows the ranking due to $LC(\mathbf{x}) \times \delta_{LC}(\mathbf{x})$ and $f(\mathbf{x}) \times \delta_f(\mathbf{x})$. This change has enabled the centre of the sparse cluster to move from rank #7 to rank #2.

The complete clustering result of the LC version of CFSFDP is shown in Fig. 6. Compared to the clustering result of CFSFDP shown in Fig. 1, LC-CFSFDP has much stronger detecting power, in the presence of varying densities and multiple density peaks in one cluster (as shown in the top cluster in Fig. 5a), which improves the $F$-measure from 0.85 to 0.98 with the four correct clusters.

## 6 Experiments

To show the power of Local Contrast, we conduct experiments using 18 benchmark datasets which have been used in the literature (Chang and Yeung 2008; Gionis et al. 2007; Jain and Law 2005; Lichman 2013; Müller et al. 2009).[1] Table 3 provides the characteristics of the datasets.

[1] Datasets “vowel”, “shape” are from Müller et al. (2009). “aggregation” is from Gionis et al. (2007). “path-based” is from Chang and Yeung (2008). “jain” is from Jain and Law (2005). The remaining datasets are from UCI Repository (Lichman 2013).

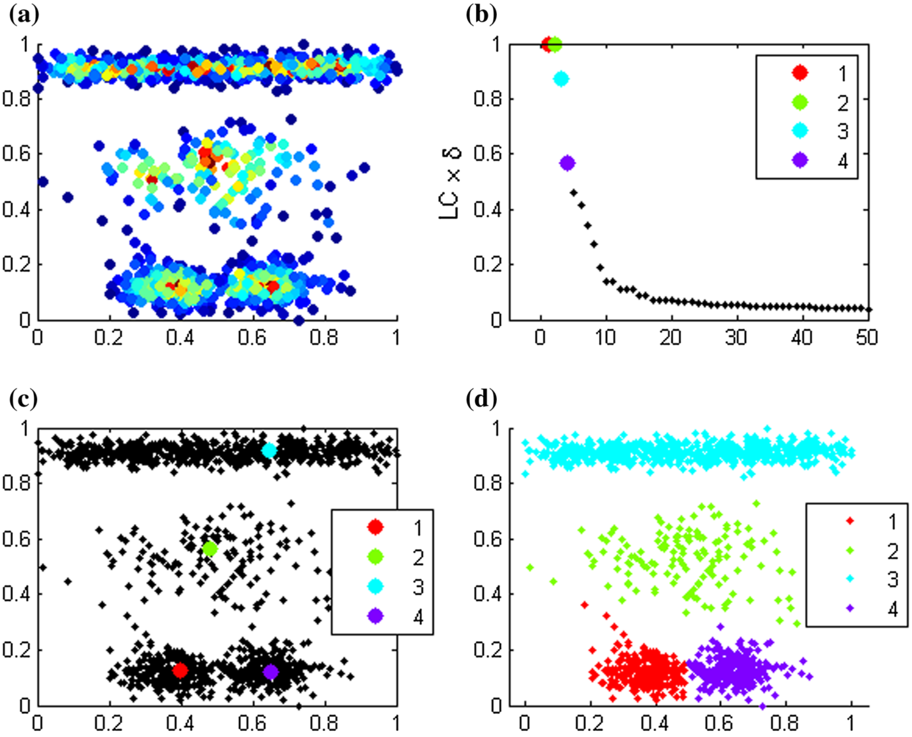

**Fig. 6** The clustering result of LC-CFSFDP on the synthetic dataset. For clarity of presentation, we plot the top 50 points only in the Decision Graph in Plot **b**. The clustering result is the optimal result in terms of $F$-measure, obtained by conducting a grid search of parameter $\epsilon$ and $M$. **a** LC distribution, **b** LC-CFSFDP decision graph, **c** LC peaks, **d** clustering result, F = 0.98478

**Table 3** Characteristics of datasets used in the experiments, where $N$ is the dataset size, $d$ is the number of features and $M$ is the number of classes

| Dataset | $N$ | $d$ | $M$ | Dataset | $N$ | $d$ | $M$ |
|---|---|---|---|---|---|---|---|
| Aggregation | 788 | 2 | 7 | Libras | 360 | 90 | 15 |
| Banknote | 1372 | 4 | 2 | Pathbased | 300 | 2 | 3 |
| Breast-d | 569 | 30 | 2 | Pendig | 10,992 | 16 | 10 |
| Breast-o | 699 | 9 | 2 | Seeds | 210 | 7 | 3 |
| Control | 600 | 60 | 6 | Segment | 2310 | 19 | 7 |
| Diabetes | 768 | 8 | 2 | Shape | 160 | 17 | 9 |
| Haberman | 306 | 3 | 2 | Thyroid | 215 | 5 | 3 |
| Iris | 150 | 4 | 3 | Vowel | 990 | 10 | 11 |
| Jain | 373 | 2 | 2 | Wine | 178 | 13 | 3 |

In all experiments, the performance is measured in terms of $F$-measure. Given a clustering result, we calculate the precision score $p_m$ and the recall score $r_m$ for each cluster $C_m$ based on the confusion matrix. $F$-measure of $C_m$ is the harmonic mean of $p_m$ and $r_m$. We then use the Hungarian algorithm (Kuhn 1955) to search the optimal match for all clusters. The overall $F$-measure is the weighted average over all clusters: $F\text{-measure} = \sum_{m=1}^{M} \frac{|C_m|}{N} \times \frac{2p_m r_m}{p_m + r_m}$, where $N$ is the dataset size. In the calculations of $F$-measure, points labeled as noise are not removed from the dataset, but they are not regarded as a cluster. In addition, we also

**Table 4** Parameters and their search ranges

| | Parameter | Search range or setting |
|---|---|---|
| All versions of CFSFDP | $\epsilon$ | $\{0.1, 0.2, \ldots, 10\%\}$ |
| | $M$ | $\{2, 3, \ldots, 20\}$ |
| LC-CFSFDP | $K$ | $\sqrt{N}$ |
| SNN-CFSFDP | $K$ | $\sqrt{N}$ |
| ReScale-CFSFDP | $\psi$ | 100 |
| | $\eta$ | 0.2 |
| FKNN-DPC | $K$ | $\sqrt{N}$ |

evaluate the performance in terms of Adjusted Rand Index (ARI) (Hubert and Arabie 1985). The outcome is similar to that of using $F$-measure. For clarity of presentation, we provide the results based on ARI in Appendix B.

All methods are searched in their parameter spaces and the best $F$-measure achieved is recorded. For all versions of CFSFDP, the value of the cut-off distance/dissimilarity $\epsilon$ is set to be the average distance between each point and its certain percentile nearest neighbour. This percentile is searched within [0.1, 10%], with a step increment of 0.1%. All methods automatically select $M$ points, which rank at the top in their respective decision graph, to be the cluster centres, where $M$ is searched within $\{2, 3, \ldots, 20\}$. For the $K$-Nearest-Neighbour search involved in LC-CFSFDP, SNN-CFSFDP and FKNN-DPC, the parameter $K$ is set to the nearest integer of $\sqrt{N}$, the square root of the dataset size. For ReScale-CFSFDP, the parameter $\psi$ is set 100 as suggested by Zhu et al. (2016), and $\eta$ is determined in the following way: we searched $\eta$ within $\{0.05, 0.1, \ldots, 0.5\}$, and find the value that yields the best average $F$-measure of the 18 datasets, which is 0.2. We set $\eta$ to 0.2 for all the experiments. As a result, ReScale-CFSFDP has been given an additional advantage compared with other methods. A summary of the parameter settings is provided in Table 4.

### 6.1 Comparing LC to SNN and ReScale

The results in Table 5 show that LC-CFSFDP has the best clustering performance among the four approaches with average rank 1.61, followed by ReScale-CFSFDP with rank 2.39, SNN-CFSFDP with rank 2.83 and CFSFDP with rank 3.00. In term of win/draw/loss counts with respect to base algorithm CFSFDP, LC-CFSFDP has 15 wins, 1 loss and 2 draws; SNN-CFSFDP has 11 wins and 7 losses; and ReScale-CFSFDP has 10 wins and 8 losses. The Friedman test results in Table 6 show that LC-CFSFDP outperforms CFSFDP and SNN-CFSFDP significantly at $p$-values $< 0.02$. When comparing LC-CFSFDP with ReScale-CFSFDP, LC-CFSFDP has 11 wins, 1 tie, and 6 losses, although the difference is not significant. Note that ReScale-CFSFDP has an unfair advantage because the parameter $\eta$ is set to one which gives the best average $F$-measure over the 18 datasets; whereas SNN-CFSFDP and LC-CFSFDP have no such advantage.

In a nutshell, Local Contrast significantly improves the CFSFDP algorithm, and its resultant LC-CFSFDP is the best density-based clustering method, among the current state-of-the-art.

### 6.2 Comparing LC-CFSFDP to FKNN-DPC

As shown in Table 7, the performance of FKNN-DPC is poor with $K$ being fixed to $\sqrt{N}$, due to its sensitivity to $K$. Therefore, we also compare LC-CFSFDP to FKNN-DPC with

**Table 5** Comparison of original and improved versions of CFSFDP in terms of $F$-measures

| Dataset | CFSFDP | LC-CFSFDP | SNN-CFSFDP | ReScale-CFSFDP |
|---|---|---|---|---|
| Aggregation | 0.996 | 0.996 | 0.983 | 0.993 |
| Banknote | 0.991 | 0.975 | 0.782 | 0.972 |
| Breast-d | 0.830 | 0.940 | 0.877 | 0.945 |
| Breast-o | 0.917 | 0.966 | 0.704 | 0.967 |
| Control | 0.708 | 0.720 | 0.717 | 0.732 |
| Diabetes | 0.602 | 0.655 | 0.635 | 0.649 |
| Haberman | 0.616 | 0.671 | 0.641 | 0.605 |
| Iris | 0.967 | 0.967 | 0.892 | 0.960 |
| Jain | 0.972 | 1.000 | 0.975 | 1.000 |
| Libras | 0.480 | 0.535 | 0.519 | 0.500 |
| Pathbased | 0.828 | 0.832 | 0.891 | 0.775 |
| Pendig | 0.794 | 0.826 | 0.664 | 0.771 |
| Seeds | 0.909 | 0.919 | 0.866 | 0.904 |
| Segment | 0.785 | 0.791 | 0.522 | 0.763 |
| Shape | 0.699 | 0.751 | 0.802 | 0.798 |
| Thyroid | 0.707 | 0.850 | 0.917 | 0.900 |
| Vowel | 0.317 | 0.322 | 0.334 | 0.345 |
| Wine | 0.931 | 0.949 | 0.932 | 0.931 |
| Average rank | 3.00 | 1.61 | 2.83 | 2.39 |
| Win/draw/loss against CFSFDP | | 15/2/1 | 11/0/7 | 10/0/8 |
| Win/draw/loss against LC-CFSFDP | 1/2/15 | | 4/0/14 | 6/1/11 |

**Table 6** Pairwise Friedman tests: $p$-values

| Friedman $p$-values | LC-CFSFDP | SNN-CFSFDP | ReScale-CFSFDP |
|---|---|---|---|
| CFSFDP | 0.0005 | 0.3458 | 0.8084 |
| LC-CFSFDP | | 0.0184 | 0.2253 |
| SNN-CFSFDP | | | 0.1573 |

the paramter $K$ being searched for an optimal result. When $K$ is searched, FKNN-DPC improves significantly in terms of $F$-measure. Nevertheless, LC-CFSFDP still outperforms FKNN-DPC with 12 wins, 1 draw and 5 losses.

## 6.3 Runtime

The time complexities for all methods are $O(N^2)$ in terms of dataset size $N$. However, for those involving $K$-nearest-neighbour search, the time complexities are provided in terms of $N$ and $K$. Table 8 gives the runtimes of all methods on all datasets. SNN-CFSFDP runs at least an order of magnitude slower than the others.

**Table 7** Comparison of LC-CFSFDP and FKNN-DPC in terms of $F$-measures

| Dataset | $K$ is searched | | $K = \sqrt{N}$ | |
|---|---|---|---|---|
| | LC-CFSFDP | FKNN-DPC | LC-CFSFDP | FKNN-DPC |
| Aggregation | 1.000 | 0.999 | 0.996 | 0.998 |
| Banknote | 0.981 | 0.969 | 0.975 | 0.486 |
| Breast-d | 0.941 | 0.919 | 0.940 | 0.711 |
| Breast-o | 0.972 | 0.921 | 0.966 | 0.804 |
| Control | 0.745 | 0.722 | 0.720 | 0.446 |
| Diabetes | 0.674 | 0.657 | 0.655 | 0.577 |
| Haberman | 0.692 | 0.692 | 0.671 | 0.683 |
| Iris | 0.967 | 0.973 | 0.967 | 0.854 |
| Jain | 1.000 | 0.982 | 1.000 | 0.963 |
| Libras | 0.547 | 0.500 | 0.535 | 0.425 |
| Pathbased | 0.906 | 0.990 | 0.832 | 0.548 |
| Pendig | 0.830 | 0.883 | 0.826 | 0.656 |
| Seeds | 0.933 | 0.923 | 0.919 | 0.631 |
| Segment | 0.804 | 0.768 | 0.791 | 0.573 |
| Shape | 0.760 | 0.717 | 0.751 | 0.633 |
| Thyroid | 0.894 | 0.909 | 0.850 | 0.848 |
| Vowel | 0.325 | 0.290 | 0.322 | 0.220 |
| Wine | 0.949 | 0.955 | 0.949 | 0.694 |
| Average rank | 1.28 | 1.67 | 1.11 | 1.89 |
| Win/draw/loss against LC-CFSFDP | | 5/1/12 | | 2/0/16 |

The search range of $K$ is $[3, \ldots, \max(50, 1.2\sqrt{N})]$

### 6.4 $K$ sensitivity test

A $K$ sensitivity test is shown in Fig. 7. In this experiment, the $K$ parameter for the $K$-nearest-neighbours search used in LC-CFSFDP, SNN-CFSFDP and FKNN-DPC is set to different values ranging from 5 to 80, while their corresponding best $F$-measure is recorded. Three datasets with low, medium, and high dimensionalities are used for the test. In all three cases, LC-CFSFDP exhibits more stable clustering performance than SNN-CFSFDP and FKNN-DPC while $K$ changes.

## 7 Discussion

Local Contrast can be applied using any density estimators, not limited to the $\epsilon$-neighbourhood density estimator which has been employed in CFSFDP (Rodriguez and Laio 2014). For example, Local Contrast can be applied to DENCLUE (Hinneburg and Gabriel 2007) which employs kernel density estimator in its operation.

We chose CFSFDP over other density-based methods such as DBSCAN, to be the base algorithm, because the former is a more advanced method. This is confirmed by comparing DBSCAN with CFSFDP in clustering the 18 datasets. The result is provided in Appendix

**Table 8** Runtime in seconds

| Dataset | CFSFDP | LC-CFSFDP | SNN-CFSFDP | ReScale-CFSFDP | FKNN-DPC |
|---|---|---|---|---|---|
| Aggregation | 0.07 | 0.1 | 1.97 | 0.15 | 0.2 |
| Banknote | 0.24 | 0.34 | 4.78 | 0.41 | 0.51 |
| Breast-d | 0.05 | 0.06 | 0.98 | 0.2 | 0.18 |
| Breast-o | 0.05 | 0.09 | 1.34 | 0.18 | 0.15 |
| Control | 0.06 | 0.07 | 1.06 | 0.26 | 0.16 |
| Diabetes | 0.08 | 0.1 | 1.81 | 0.19 | 0.19 |
| Haberman | 0.02 | 0.02 | 0.38 | 0.1 | 0.05 |
| Iris | 0.01 | 0.01 | 0.19 | 0.09 | 0.02 |
| Jain | 0.02 | 0.03 | 0.55 | 0.09 | 0.07 |
| Libras | 0.03 | 0.04 | 0.46 | 0.23 | 0.12 |
| Pathbased | 0.02 | 0.02 | 0.4 | 0.09 | 0.04 |
| Pendig | 21.69 | 30.04 | 345.17 | 22.53 | 44.27 |
| Seeds | 0.01 | 0.01 | 0.32 | 0.01 | 0.02 |
| Segment | 0.82 | 1.13 | 16.59 | 0.96 | 1.5 |
| Shape | 0.01 | 0.01 | 0.26 | 0.02 | 0.02 |
| Thyroid | 0.01 | 0.01 | 0.28 | 0.02 | 0.03 |
| Vowel | 0.13 | 0.19 | 3.04 | 0.17 | 0.33 |
| Wine | 0.01 | 0.01 | 0.27 | 0.02 | 0.02 |
| Time complexity | $O(N^2)$ | $O(N^2 + KN)$ | $O(K^2N^2)$ | $O(N^2)$ | $O(N^2 + K^2N)$ |

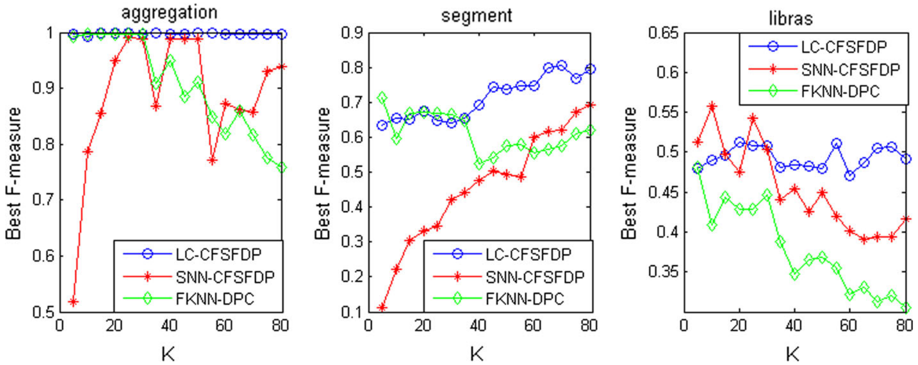

**Fig. 7** $K$ sensitivity test on 3 datasets of different dimensionality: aggregation has 2 attributes; segment has 19; and libras has 90. LC-CFSFDP demonstrates better stability than SNN-CFSFDP and FKNN-DPC while $K$ changes

C, which shows that CFSFDP outperforms DBSCAN in all but 1 dataset. To be fair and complete, we also compare LC-CFSFDP to the original SNN and ReScale approaches. The result is provided in Appendix D, which shows that LC-CFSFDP outperforms both methods.

The choice of parameter $K$ in KNN based methods is usually time-consuming since they are often sensitive to $K$. However, we have shown that LC is not as sensitive to $K$ as SNN or FKNN-DPC. As a rule of thumb, setting $K = \sqrt{N}$ has been empirically verified to be effective for LC.

As to the choice of parameter $M$ (the number of clusters), both LC-CFSFDP and the original CFSFDP have the same requirement. For a specific dataset, the proper choice of $M$ is a user decision that could be made based on domain knowledge, visual inspection, or other means. In our experiments, $M$ is simply searched to show the best capability of each method.

The original CFSFDP does not explicitly identify any data point to be noise. Instead, after the clustering procedure, it takes an extra step to produce cluster halos, which can be considered as noise. In our experiments, no noise points are produced because all variants of CFSFDP, as well as FKNN-DPC, are able to cluster the whole dataset without producing any noise. However, while handling noisy datasets, LC-CFSFDP can also produce cluster halos in the same way as CFSFDP.

Grid-based clustering approaches partition the space into a number of cells and use the cell density to identify clusters. For example, GRIDCLUS (Schikuta 1996) and NSGC (Ma and Chow 2004) rely on the cell density to identify core cells and link neighbouring core cells together to form clusters. Instead of the current point-based definition, it is possible that Local Contrast can be redefined using cell densities of neighbouring cells; and employ Local Contrast in these algorithms to improve their performance.

Another possible application of LC is density-based subspace clustering, such as SUBCLU (Kailing et al. 2004) and DUSC (Assent et al. 2007). These methods use a density threshold to differentiate between cluster points and noise in different subspaces. Because density is dimensionality-biased, i.e., when estimated using distance-based density estimators, the densities of a data cloud tend to be lower in higher-dimensional spaces. Hence these methods suffer from density variation across subspaces with different dimensionalities: low thresholds detect high-dimensional clusters but have difficulty filtering out noise in low-dimensional subspaces; while high thresholds screen out noise well in low-dimensional subspaces but tend to overlook high-dimensional clusters (Zimek and Vreeken 2015). LC can possibly be an effective remedy for this issue in subspace clustering since LC is not dimensionality-biased.

However, not all density-based methods can utilise Local Contrast readily because some do not employ density directly in their operations. For example, instead of density, OPTICS (Ankerst et al. 1999) employs "core distance" and "reachability distance" to rank points in order to identify clusters. The "reachability distance" reflects the density such that points with a lower density normally have higher "reachability distance". It is interesting to explore whether Local Contrast can be redefined using these distances rather than density.

## 8 Conclusions

In this paper, we identify the root cause of CFSFDP's failure to detect all clusters in a dataset having hugely varying densities. This is the first work, as far as we know, that overcomes CFSFDP's weakness from its root cause.

We make the following three contributions:

First, we formalise a necessary condition for CFSFDP to correctly identify all clusters. We show that a violation of this condition leads to poor clustering performance. This explains the reason why a density-based clustering algorithm such as CFSFDP is unable to correctly identify all clusters in datasets having large density variations.

Second, we propose a new measure called Local Contrast, as an alternative to density, to improve the capability of density-based clustering methods to detect clusters of hugely different densities in a dataset. We show that it has two unique properties that are critical in improving the above-mentioned capability, i.e., all cluster centres in the Local Contrast distribution have the same constant value, so as all local minima of Local Contrast which correspond to the local minima of density distribution, regardless of the densities of these cluster centres and local minima. We show that these properties make density-based algorithms much more robust in the presence of large density variations.

Third, by incorporating Local Contrast into CFSFDP, we create a powerful method LC-CFSFDP which has much better detecting power than the original method. Our empirical evaluation shows that LC-CFSFDP is the best performer compared to two state-of-the-art methods, SNN and ReScale, as well as FKNN-DPC which is a recent improvement of CFSFDP.

**Acknowledgements** Bo Chen is supported by Monash Data61 Postgraduate Research Scholarship and Faculty of IT Tuition Fee Scholarship, Monash University.

## Appendix A: Condition under which DBSCAN-like clustering algorithms fail

This section provides a necessary condition for DBSCAN-like clustering algorithms, which employ a global threshold to identify core points, to successfully identify all clusters (Zhu et al. 2016).

Let $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in R^d, \mathbf{x}_i \sim F$ denote a dataset of $n$ points each sampled independently from a distribution $F$. Let $f(\mathbf{x})$ denote the density estimate at point $\mathbf{x}$ used (either explicitly or implicitly) by a particular density-based clustering algorithm. A set of clusters $\{C_1, \ldots, C_\varsigma\}$ is defined as non-empty and non-intersecting subsets: $C_i \subset D, C_i \neq \emptyset, \forall_{i \neq j}\ C_i \cap C_j = \emptyset$. Let $c_i = \arg\max_{\mathbf{x} \in C_i} f(\mathbf{x})$ denote the mode (point of the highest estimated density) for cluster $C_i$ and $p_i = f(c_i)$ denote the corresponding peak density value.

In addition, let $N_\epsilon(\mathbf{x})$ be the $\epsilon$-neighbourhood of $\mathbf{x}$, $N_\epsilon(\mathbf{x}) = \{\mathbf{y} \in D \mid d(\mathbf{x}, \mathbf{y}) \leqslant \epsilon\}$, where $d(\cdot, \cdot)$ is the dissimilarity function ($d : R^d \times R^d \to R$) used by the density-based clustering algorithm.

A **non-cyclic path** linking points $\mathbf{x}_i$ and $\mathbf{x}_j$, $path(\mathbf{x}_i, \mathbf{x}_j)$, is defined as a sequence of unique points starting with $\mathbf{x}_i$ and ending with $\mathbf{x}_j$ where adjacent points lie in each other's neighbourhood: $(\mathbf{x}_{\pi(1)}, \mathbf{x}_{\pi(2)}, \ldots, \mathbf{x}_{\pi(k)})$. Here $\pi()$ is a mapping $\pi : \{1, \ldots, k\} \to \{1, \ldots, n\}$ such that $\forall_{s \neq t \in \{1,\ldots,k\}}\ (\pi(s) \neq \pi(t)) \wedge (\pi(1) = i) \wedge (\pi(k) = j) \wedge (\forall_{v \in \{1,\ldots,k-1\}}\ \mathbf{x}_{\pi(v+1)} \in N_\epsilon(\mathbf{x}_{\pi(v)}))$.

Let $\mathcal{P}_{ij} = \{(\mathbf{x}_{\pi(1)}, \mathbf{x}_{\pi(2)}, \ldots, \mathbf{x}_{\pi(k)}) \mid \mathbf{x}_{\pi(1)} = c_i \wedge \mathbf{x}_{\pi(k)} = c_j\}$ denote the set of all non-cyclic paths linking the modes of clusters $C_i$ and $C_j$, and then let $g_{ij} = \max\limits_{path \in \mathcal{P}_{ij}} \min\limits_{\mathbf{x} \in path} f(\mathbf{x})$ be the largest of the minimum values along any paths in $\mathcal{P}_{ij}$.

Let each cluster $C_i$ be represented by its mode $c_i$ in order for a density-based clustering algorithm to be able to (reliably) identify and separate all clusters in a dataset $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. The condition that the estimated density at the mode of any cluster is greater than the maximum of the minimum estimated density along any path linking any two modes is given as

$$\min_{k \in \{1,\ldots,\varsigma\}} p_k > \max_{i \neq j \in \{1,\ldots,\varsigma\}} g_{ij} \tag{4}$$

This condition implies that there must exist a threshold $\tau$ that can be used to break all paths between the modes by assigning regions with estimated density less than $\tau$ to noise, i.e.,

$$\exists_\tau \forall_{k, i \neq j \in \{1,\ldots,\varsigma\}}\ p_k \geqslant \tau > g_{ij}$$

If a density-based clustering algorithm uses a global threshold on the estimated density to identify core points and links neighbouring core points together to form clusters (e.g., DBSCAN (Ester et al. 1996)), then the requirement given in Eq. (4) on the density estimates and cluster definitions provides a necessary condition for the algorithm to successfully identify all clusters.

Therefore, the density-based clustering algorithm will fail to identify all clusters in a data distribution which violates the inequality in Eq. (4).

## Appendix B: Performance evaluation in terms of ARI

Table 9 provides the comparison of four variants of CFSFDP in terms of ARI. Table 10 provides the comparison between LC-CFSFDP and FKNN-DPC in terms of ARI. The results are similar to those using $F$-measures in terms of average rank and win/draw/loss counts (as shown in Tables 5 and 7).

**Table 9** Comparison of original and improved versions of CFSFDP in terms of ARI

| Dataset | CFSFDP | LC-CFSFDP | SNN-CFSFDP | ReScale-CFSFDP |
|---|---|---|---|---|
| Aggregation | 0.996 | 0.996 | 0.961 | 0.994 |
| Banknote | 0.994 | 0.914 | 0.599 | 0.892 |
| Breast-d | 0.428 | 0.773 | 0.693 | 0.792 |
| Breast-o | 0.640 | 0.866 | 0.344 | 0.872 |
| Control | 0.584 | 0.633 | 0.673 | 0.619 |
| Diabetes | 0.110 | 0.148 | 0.118 | 0.106 |
| Haberman | 0.052 | 0.064 | 0.059 | 0.032 |
| Iris | 0.904 | 0.904 | 0.784 | 0.886 |
| Jain | 0.747 | 1.000 | 0.811 | 1.000 |
| Libras | 0.356 | 0.386 | 0.402 | 0.336 |
| Pathbased | 0.624 | 0.740 | 0.668 | 0.519 |
| Pendig | 0.664 | 0.721 | 0.499 | 0.696 |
| Seeds | 0.756 | 0.775 | 0.640 | 0.742 |
| Segment | 0.681 | 0.680 | 0.338 | 0.631 |
| Shape | 0.632 | 0.618 | 0.634 | 0.664 |
| Thyroid | 0.224 | 0.756 | 0.594 | 0.688 |
| Vowel | 0.181 | 0.137 | 0.169 | 0.191 |
| Wine | 0.803 | 0.847 | 0.540 | 0.800 |
| Average rank | 2.61 | 1.67 | 3.00 | 2.56 |
| Win/draw/loss against CFSFDP | | 12/2/4 | 9/0/9 | 8/0/10 |
| Win/draw/loss against LC-CFSFDP | 4/2/12 | | 4/0/14 | 4/1/13 |

**Table 10** Comparison of LC-CFSFDP and FKNN-DPC in terms of ARI

| Dataset | $K$ is searched | | $K = \sqrt{N}$ | |
|---|---|---|---|---|
| | LC-CFSFDP | FKNN-DPC | LC-CFSFDP | FKNN-DPC |
| Aggregation | 1.000 | 0.997 | 0.996 | 0.995 |
| Banknote | 0.933 | 0.883 | 0.914 | 0.053 |
| Breast-d | 0.818 | 0.705 | 0.773 | 0.412 |
| Breast-o | 0.904 | 0.834 | 0.866 | 0.622 |
| Control | 0.652 | 0.682 | 0.633 | 0.247 |
| Diabetes | 0.156 | 0.139 | 0.148 | 0.079 |
| Haberman | 0.100 | 0.197 | 0.064 | 0.112 |
| Iris | 0.904 | 0.922 | 0.904 | 0.674 |
| Jain | 1.000 | 0.903 | 1.000 | 0.811 |
| Libras | 0.414 | 0.346 | 0.386 | 0.267 |
| Pathbased | 0.785 | 0.970 | 0.740 | 0.289 |
| Pendig | 0.723 | 0.794 | 0.721 | 0.588 |
| Seeds | 0.812 | 0.776 | 0.775 | 0.406 |
| Segment | 0.686 | 0.602 | 0.680 | 0.334 |
| Shape | 0.662 | 0.594 | 0.618 | 0.506 |
| Thyroid | 0.784 | 0.753 | 0.756 | 0.649 |
| Vowel | 0.182 | 0.154 | 0.137 | 0.056 |
| Wine | 0.847 | 0.869 | 0.847 | 0.461 |
| Average rank | 1.33 | 1.67 | 1.06 | 1.94 |
| Win/draw/loss against LC-CFSFDP | | 6/0/12 | | 1/0/17 |

## Appendix C: DBSCAN versus CFSFDP

A comparison between DBSCAN and CFSFDP is provided in Table 11. The result shows that CFSFDP performs significantly better than DBSCAN.

**Table 11** Comparison between DBSCAN and CFSFDP

| Dataset | Best $F$-measures | |
|---|---|---|
| | DBSCAN | CFSFDP |
| Aggregation | 0.991 | 0.996 |
| Banknote | 0.952 | 0.991 |
| Breast-d | 0.609 | 0.830 |
| Breast-o | 0.867 | 0.917 |
| Control | 0.536 | 0.708 |
| Diabetes | 0.538 | 0.602 |
| Haberman | 0.630 | 0.616 |
| Iris | 0.834 | 0.967 |
| Jain | 0.964 | 0.972 |
| Libras | 0.407 | 0.481 |
| Pathbased | 0.770 | 0.828 |
| Pendig | 0.709 | 0.794 |
| Seeds | 0.750 | 0.909 |
| Segment | 0.586 | 0.785 |
| Shape | 0.642 | 0.699 |
| Thyroid | 0.584 | 0.707 |
| Vowel | 0.252 | 0.317 |
| Wine | 0.667 | 0.931 |
| Win/draw/loss against DBSCAN | | 17/0/1 |

For DBSCAN, parameter $\epsilon$ is search from 0.01 to the maximum distance between two points in a dataset, with a step increment of 0.01. The $MinPts$ parameter is searched in $\{1, 2, \ldots, \lfloor\sqrt{N}\rfloor\}$

## Appendix D: LC-CFSFDP versus SNN-DBSCAN and ReScale-DBSCAN

A comparison of LC-CFSFDP, SNN-DBSCAN[2] (Ertöz et al. 2003a) and ReScale-DBSCAN (Zhu et al. 2016) is provided in Table 12. The result shows that LC-CFSFDP performs significantly better than SNN-DBSCAN and ReScale-DBSCAN.

[2] In this paper, we refer SNN as a dissimilarity measure. To avoid confusion, here we refer the original SNN clustering method in Ertöz et al. (2003a) as SNN-DBSCAN, since it is based on the DBSCAN procedure.

**Table 12** Comparison of LC-CFSFDP, SNN-DBSCAN and ReScale-DBSCAN

| Dataset | Best $F$-measures | | |
|---|---|---|---|
| | LC-CFSFDP | SNN-DBSCAN | ReScale-DBSCAN |
| Aggregation | 0.996 | 1.000 | 0.916 |
| Banknote | 0.975 | 0.960 | 0.990 |
| Breast-d | 0.940 | 0.734 | 0.795 |
| Breast-o | 0.966 | 0.885 | 0.957 |
| Control | 0.720 | 0.773 | 0.659 |
| Diabetes | 0.655 | 0.584 | 0.558 |
| Haberman | 0.671 | 0.673 | 0.727 |
| Iris | 0.967 | 0.958 | 0.845 |
| Jain | 1.000 | 1.000 | 0.996 |
| Libras | 0.535 | 0.425 | 0.442 |
| Pathbased | 0.832 | 0.944 | 0.621 |
| Pendig | 0.826 | 0.856 | 0.738 |
| Seeds | 0.919 | 0.890 | 0.854 |
| Segment | 0.791 | 0.662 | 0.565 |
| Shape | 0.751 | 0.729 | 0.685 |
| Thyroid | 0.850 | 0.904 | 0.860 |
| Vowel | 0.322 | 0.320 | 0.295 |
| Wine | 0.949 | 0.918 | 0.851 |
| Win/draw/loss against LC-CFSFDP | | 6/1/11 | 3/0/15 |

For the DBSCAN based procedure, parameter $\epsilon$ is search from 0.01 to the maximum distance between two points in a dataset, with a step increment of 0.01. The $MinPts$ parameter is searched in $\{1, 2, \ldots, \lfloor \sqrt{N} \rfloor\}$

## References

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data* (pp. 49–60). New York, NY: ACM.

Assent, I., Krieger, R., Müller, E., & Seidl, T. (2007). Dusc: Dimensionality unbiased subspace clustering. In *Proceedings of the 7th international conference on data mining* (pp. 409–414). IEEE.

Borah, B., & Bhattacharyya, D. (2008). DDSC: A density differentiated spatial clustering technique. *Journal of Computers*, *3*(2), 72–79.

Brito, M., Chavez, E., Quiroz, A., & Yukich, J. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, *35*(1), 33–42.

Chang, H., & Yeung, D. Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*(1), 191–203.

Cherkassky, V., & Mulier, F. M. (2007). *Learning from data: Concepts, theory, and methods*. Hoboken: Wiley.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. 1–38.

Ertöz, L., Steinbach, M., & Kumar, V. (2003a). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining* (pp. 47–58).

Ertöz, L., Steinbach, M., & Kumar, V. (2003b). Finding topics in collections of documents: A shared nearest neighbor approach. *Clustering and Information Retrieval*, *11*, 83–103.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd international conference on knowledge discovery and data mining* (pp. 226–231).

Ferilli, S., Biba, M., Basile, T., Di Mauro, N., & Esposito, F. (2008). K-nearest neighbor classification on first-order logic descriptions. In *Proceedings of the IEEE international conference on data mining workshops* (pp. 202–210).

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego, CA: Academic Press Professional Inc.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, *1*(1), 4.

Han, J., & Kamber, M. (2011). *Data mining: Concepts and techniques* (3rd ed.). Los Altos, CA: Morgan Kaufmann.

Hinneburg, A., & Gabriel, H. H. (2007). DENCLUE 2.0: Fast clustering based on kernel density estimation. In *Advances in intelligent data analysis* (Vol. VII, pp. 70–80). Springer.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666.

Jain, A. K., & Law, M. H. (2005). Data clustering: A user's dilemma. In *Pattern recognition and machine intelligence* (pp. 1–10). Springer.

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, *100*(11), 1025–1034.

Kailing, K., Kriegel, H. P., & Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. In *Proceedings of the international conference on data mining* (pp. 246–256). SIAM.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics*, *2*(1–2), 83–97.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 31 May 2017.

Ma, E. W., & Chow, T. W. (2004). A new shifting grid clustering algorithm. *Pattern Recognition*, *37*(3), 503–514.

Müller, E., Günnemann, S., Assent, I., & Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, *2*, 1270–1281.

Ram, A., Sharma, A., Jalal, A. S, Agrawal, A., & Singh, R. (2009). An enhanced density based spatial clustering of applications with noise. In *Proceedings of the IEEE international advance computing conference* (pp. 1475–1478).

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492–1496.

Schikuta, E. (1996). Grid-clustering: An efficient hierarchical clustering method for very large data sets. In *Proceedings of the 13th IEEE international conference on pattern recognition* (Vol. 2, pp. 101–105).

Tan, J., & Wang, R. (2013). Smooth splicing: A robust snn-based method for clustering high-dimensional data. *Mathematical Problems in Engineering*, *2013*, 1–9.

Xie, J., Gao, H., Xie, W., Liu, X., & Grant, P. W. (2016). Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors. *Information Sciences*, *354*, 19–40.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*(2), 165–193.

Zhu, Y., Ting, K. M., & Carman, M. J. (2016). Density-ratio based clustering for discovering clusters with varying densities. *Pattern Recognition*, *60*, 983–997.

Zimek, A., & Vreeken, J. (2015). The blind men and the elephant: On meeting the problem of multiple truths in data from clustering and pattern mining perspectives. *Machine Learning*, *98*(1–2), 121–155.

Zitzler, E., Laumanns, M., Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization. In *Metaheuristics for multiobjective optimisation* (pp. 3–37). Springer.