



A methodology for automatic parameter-tuning and center selection in density-peak clustering methods

José Carlos García-García¹ · Ricardo García-Ródenas¹

© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The density-peak clustering algorithm, which we refer to as DPC, is a novel and efficient density-based clustering approach. The method has the advantage of allowing non-convex clusters, and clusters of variable size and density, to be grouped together, but it also has some limitations, such as the visual location of centers and the parameter tuning. This paper describes an optimization-based methodology for automatic parameter/center selection applicable both to the DPC and to other algorithms derived from it. The objective function is an internal/external cluster validity index, and the decisions are the parameterization of the algorithm and the choice of centers. The internal validation measures lead to an automatic parameter-tuning process, and the external validation measures lead to the so-called *optimal rules*, which are a tool to bound the performance of a given algorithm from above on the set of parameterizations. A numerical experiment with real data was performed for the DPC and for the fuzzy weighted k -nearest neighbor (FKNN-DPC) which validates the automatic parameter-tuning methodology and demonstrates its efficiency compared to the state of the art.

Keywords Density peaks clustering · Automatic parameter tuning · Optimal rules · Cluster validity index · Differential entropy

1 Introduction

The study of clustering techniques is a very active area of research in machine learning. Clustering is widely applied in pattern recognition, bioinformatics and image processing. It is used to find a partition of the dataset based on similar features. These methods can be divided into hierarchical methods, partitioning methods, density-based methods, model-based methods, grid-based methods and soft computing methods, or a combination of these.

Recently, Rodríguez and Laio (2014) described a new clustering method using a fast search of density peaks (DPC). This algorithm is based on the idea that cluster centers have higher density than their neighbors and also that they are at a

relatively large distance from any points with higher density. Liu et al. (2018) note the following two essential advantages of DPC:

1. The algorithm is simple and efficient, and it can quickly find the high density peak points (cluster centers).
2. The DPC algorithm is suitable for cluster analysis of large-scale data because the data points are assigned to the clusters in a single round based on minimum nearest distance to cluster center.

Wiwie et al. (2015) introduce the integrative clustering evaluation framework (ClustEval), and through an exhaustive comparison, using 13 state-of-the-art algorithms and 24 biomedical and synthetic datasets, they showed that DPC achieves high F_1 scores in the gold-standard reconstruction. The authors also show that, numerically, the DPC gives the best results on synthetic data, proving its efficiency in reconstructing varied forms that are not necessarily convex. Despite the high efficiency of DPC, especially with synthetic data, its performance is strongly conditioned by the parameter tuning d_c used to calculate the densities and by the choice of the centers c . More promising algorithms, such

Communicated by V. Loia.

✉ José Carlos García-García
josecarlos.garcia@uclm.es

Ricardo García-Ródenas
ricardo.garcia@uclm.es

¹ Departamento de Matemáticas, Escuela Superior de Informática, Universidad de Castilla-La Mancha, Paseo de la Universidad, 4, Ciudad Real 13071, Spain

as ADPC-KNN (Yaohui et al. 2017), SNN-DPC (Liu et al. 2018), DPC-KNN (Du et al. 2016) and FKNN-DPC (Xie et al. 2016) have recently appeared, which introduce local densities, and assign data to centers in two stages. These methods are not parameter-free and require an estimate of the number of neighbors (denoted by k) in order to calculate the local density. Liu et al. (2018) choose the parameter k by assessing all the values in the range $[4, 50]$, but the weakness of this procedure is that it requires knowledge of the true classes and this input is unknown in the application of clustering algorithms to real problems. This led Xie et al. (2016), Lu and Zhu (2017) and Liu et al. (2018) to show in their conclusions that the automatic choice of k needs further research.

Wiwie et al. (2015) and Wang and Xu (2017) address the problem of automatic parameter adjustment in clustering algorithms. The authors overcome the problem of not knowing the true classes by introducing internal cluster evaluation measures, such as the Silhouette index, and by optimizing this, they obtain the desired parameterization. The optimization method used is based on assessing a large number of randomly generated parameterizations. This paper formalizes this methodology for automatically setting parameters in density-based clustering methods, such as d_c , k , or the centers themselves \mathbf{c} . The resulting model of the problem is not a standard optimization problem because optimization is performed on a subset of vectors (the centers) rather than on a single vector. This feature, together with the fact that a random sampling of the centers of the dataset is not practical, has led to the design of an ad hoc optimization strategy.

Gaussian entropy H_G and Gaussian cross-entropy H_G^\times are used for internal validation, and the methodology is analyzed via an extensive computational experiment with respect to the external validation V -measure. The procedure described is computationally expensive, but the aim of the paper is to address improvements in performance at finding the gold standard in real datasets, without, in this early stage, considering the computational efficiency of the procedure. This kind of situation is common in biomedical data.

This paper is organized as follows. Section 2 describes the problems related to DPC considered in the literature, together with the lines of research to improve the performance of DPC. Section 3 gives a brief description of DPC including the different steps and the method for calculating the density and distance of each point. Section 4 presents our proposed methodology, the clustering validity indices for the automatic parameter-tuning process and the heuristic algorithm to obtain the parameters and the cluster centers. Section 5 sets out a computational experiment to assess the methodology and compare our proposal with the state-of-the-art algorithms. Finally, Sect. 6 gives some conclusions and the intended future work.

2 Related work

The research community has shown great interest in the DPC algorithm, which has produced a steady stream of research, including some 200 citations annually¹ since its publication. Many of these papers are about applications to a variety of scientific fields, another group seeks to improve its efficiency, and the final group widens the field of problems it can solve by hybridization with other procedures or by generalization. Examples of this third group are Chen et al. (2015), which address detection of outliers by introducing the cut-off distance-based local density of each data point into the support-vector data-description (SVDD) training model and Bu et al. (2016), which combine the DPC and the dropout deep-learning model to clustering heterogeneous data. Wang et al. (2016) apply the DPC to discovering social circles with overlap via user profile and topological structure-based features, and Liu et al. (2017) and Du et al. (2017) apply DPC to mixed data.

A systematic review of all these studies is beyond the scope of this paper. We focus on improving the performance of DPC. A number of authors, Liu et al. (2018), Xie et al. (2016), have underlined the following problems with DPC:

$P_\rho \equiv$ When used to calculate local density, DPC does not take into account the local structure of the data. Poor performance might be expected of the DPC algorithm in the case of complex datasets involving multiple scales, that are cross-winding, have various densities or high dimensionality.

$P_1\mathbf{S} \equiv$ The one-step allocation strategy is not robust and has poor fault tolerance. There is a *Domino Effect* in that once a point is erroneously assigned, the error will propagate so that more points will be assigned incorrectly. A number of methods describe the alternative possibility of calculating core cluster points before expanding to the boundary points.

$P_{d_c} \equiv$ The parameter d_c is used to calculate the density of each data point and to identify the border points in the clusters. The cutoff distance d_c is generally difficult to determine, a small change in d_c will still cause a noticeable fluctuation in the result, and this is especially true for real-world datasets.

$P_{\mathbf{c}} \equiv$ A heuristic approach is based on the so-called *decision graph* in order to analyze the selection of the cluster centers. The human-based selection of the cluster center is a big limitation. The goal is to automatically determine the number of clusters and their centers. DPC is unable to group data points correctly when a cluster has more than one center.

¹ Accessing scopus on 27th September 2019 gave 1475 references

Table 1 Review on L_1 computational complexity

References	Algorithm	P_ρ	P_1S	P_{d_c}	P_c	P_O	Notes
Wang et al. (2017)	KMDD					✓	It combines K -means with DPC. It has a near-linear time complexity with respect to the dataset size and dimension
Xu et al. (2016)	DenPEHC				✓	✓	It introduces a grid granulation framework to enable DenPEHC to cluster large-scale and high-dimensional datasets
Gong and Zhang (2016)	–					✓	It improves performance significantly (up to 40x) compared with a naive approach
Bai et al. (2017)	CFSFDP+A, CFSFDP+DE					✓	It uses the K -means algorithm in order to reduce the distance calculations (CFSFDP+A) and to enhance the scalability of the DPC (CFSFDP+DE)

$P_O \equiv$ Computational complexity. The method requires measuring distance between any pair of objects, with a high computational cost of $O(N^2)$ where N is the number of data points. This limits the application of the algorithm when clustering high-volume and high-dimensional datasets.

We introduce the notation P_ρ , P_1S , P_{d_c} , P_c and P_O to refer to each of these problems.

The DPC algorithm works in two stages. In the first, the concepts of local density and distance are used to identify cluster centers. In the second, a label propagation method is proposed to form clusters. The P_ρ , P_c , P_{d_c} , P_O problems are associated with the first stage, while P_1S corresponds to the second stage. Research into improving the performance of DPC can be roughly classified into three research streams:

- L_1 *Computational complexity.* The aim of this research is to lower the computational cost and/or identify computational paradigms to make it applicable to large volumes of data. This line of research seeks answers to the P_O problem.
- L_2 *Parameter-tuning and local distance/density problems.* The parameter d_c determines the density ρ_i and the distance of each point δ_i and these magnitudes in turn characterize the centers c of the clusters. These relations have been studied as to how to define new density and distance concepts, in order to account for the local characteristics of the data (in a neighborhood) and their influence on the identification of the centers. L_2 seeks answers to the P_ρ , P_c , P_{d_c} problems.
- L_3 *Multi-step labeling.* Multi-step procedures for assigning categories have been proposed to address the P_1S problem.

We now review these lines of research.

2.1 L_1 computational complexity

The P_O problem stems from the need to calculate the distances between each pair of objects. The problem has been addressed using parallel computing in CUDA (Li et al. 2016), partially calculating the distance matrix (Bai et al. 2017), etc. These studies do not seek to improve performance but rather to broaden the DPC to handle high volumes of data. Table 1 gives an overview of some research in the line L_1 .

2.2 L_2 parameter-tuning and local distance/density problems

The first studies in this area focused on automatically determining the original parameters of the DPC, that is, determining the parameter d_c and the centers c . Examples of these studies are:

Chen and He (2016) propose a data stream clustering algorithm based on DPC for mixed numerical and categorical attributes. The approach uses a linear regression model and residuals analysis to find the outliers of the intensity-distance distribution graph, enabling automatic identification of cluster centers.

Mehmood et al. (2016b), based on heat diffusion, propose a non-parametric method to account both for selection of the cutoff distance d_c and boundary correction of the kernel density estimation using the time parameter of heat diffusion.

Jinyin et al. (2017) identify centers as outliers of the γ distribution and use a mechanism to determine a self-adaptive density radius d_c based on optimizing a fitness function using the mountain-climbing algorithm.

Xu et al. (2016) propose a hierarchical clustering algorithm based on DPC. An essential part is identification of the centers based on the γ rule. The authors analyze the question and introduce a grid granulation framework to enable their

Table 2 Review of L_2 parameter-tuning and local distance/density problems

References	Algorithm	P_ρ	P_1S	P_{d_c}	P_c	P_O	Notes
Jinyin et al. (2017)	CH-CCFDAC			✓	✓		Uses identification of the outliers of the distribution γ_i to detect centers and implements a self-adaptive density parameter d_c .
Li and Tang (2018)	CDP	✓			✓		Uses a local density measure and the Δ -tree to assign labels
Jiang et al. (2018)	GDPC	✓			✓		Uses an alternative decision graph based on gravitation theory
Yaohui et al. (2017)	ADPC-KNN	✓		✓	✓		Uses the k -nearest neighbors of a point to compute local density. Proposes a formula to compute d_c based on k
Tao et al. (2017)	F-DPC			✓	✓		Introduces the data field theory to adaptively select the d_c and uses the maximum entropy reduction to select centers
Yang et al. (2017)	LPC				✓		Introduces the local importance index c_i based on Laplacian centrality, and ρ_i is replaced by c_i in the decision graph.
Zang et al. (2017)	ADPC-DNAGA			✓	✓		Applies the DNA genetic algorithm to optimize the potential entropy of the data field to obtain d_c and automatically determines the cluster centers by Gaussian processes.
Mehmood et al. (2016b)	CFSFDP-HD	✓		✓			Proposes a heat diffusion approach in order to account for both selection of the cutoff distance and boundary correction of the kernel density estimation.
Bie et al. (2016)	fuzzy-CFSFDPA				✓		An adaptive selection tool for the cluster centers based on fuzzy rules.
Guo et al. (2017)	LR-CFDP	✓			✓		Linear regression model and residuals analysis are used to obtain c , and it uses a k -NN local density
Chen and He (2016)	Str-FSFDPA				✓		Linear regression model and residuals analysis are used to obtain c in stream clustering
Du et al. (2016)	DPC-KNN-PCA	✓	✓			✓	Uses k nearest neighbors to compute local density and principal component analysis to handle datasets that have relatively high dimension

proposed algorithm, which they call DenPEHC, to cluster large-scale and high-dimensional datasets.

It has recently been understood that there may exist datasets in which the clusters have different, non-comparable densities. Thus, locating centers through their density and distance could be difficult for certain clusters. In such cases, the density of a point must be evaluated with respect to its neighbors, that is, whether it is large or small relative to the densities of its neighbors. The mechanism that addresses this matter most successfully is the introduction of k -neighborhoods k -NN of the points in the definition of density and/or distance, which allows the local structure of data to be addressed (cluster with variable densities) rather than global densities ρ .

Du et al. (2016) introduce the k -nearest neighbors for the local density computation of ρ . The authors use principal component analysis to improve the performance of the former method on real-world datasets.

Wang and Song (2016) consider automatic identification of the clustering centers via statistical testing. The authors

define a more robust metric for computing the density of an object, which then generates a further metric for evaluating the centrality of each object. The new density, the so-called k -density $\hat{\rho}$, is based only on the distance of the object from the k -nearest neighbors. The authors show that the choice $k = \lceil N \rceil$ with $\lceil \cdot \rceil$ the ceil of a number is robust in density evaluation.

There is a great deal of activity in this line, and Tables 2 and 3 summarize several of the studies.

2.3 L_3 multistep labeling

Once the need to work with densities/local distances is understood, a further improvement has been added to the DPC, introducing center sets (union of core points) instead of center points. This has led to clustering algorithms with multistep labeling where the core points are labeled at the first stage, and the other points at subsequent stages. The following studies are examples of this line of research.

Table 3 Review of L_2 parameter-tuning and local distance/density problems (continuation)

References	Algorithm	P_ρ	P_{1S}	P_{d_c}	P_c	P_O	Notes
Ding et al. (2016), Ding et al. (2018)	DPC-GEV, DPC-CI				✓		Identification of cluster centers using generalized extreme value distribution and the Chebyshev inequality.
Gao et al. (2016)	–			✓	✓		Formula for the cutoff distance calculation and a method for cluster center selection to improve its robustness.
Chen and He (2015)	–			✓	✓		Linear regression model and residuals analysis are used to obtain c and PSO to optimize d_c .
Hua et al. (2016)	CDC	✓					Determines the parameters based on the inherent structure of data points.
Kun et al. (2016)	–				✓		Finds optimal cluster centers by iteration based on genetic algorithm.
Liang and Chen (2016)	3DC				✓		Uses a divide-and-conquer strategy to automatically find the correct number and the cluster centers.
Mehmood et al. (2016a)	IJS-CFSFDP			✓			An adaptive rule using the characteristics of the Improved Sheather-Jones method.
Wang and Song (2016)	STClu	✓			✓		Uses a statistical test method to identify the clustering centers automatically on a centrality metric based on the new local density and new minimum distance.

Chen et al. (2016) propose CLUB (CLUstering based on Backbone). The key feature of CLUB is that it identifies a cluster according to its density backbone instead of just a center. In CLUB, a three-step scheme is adopted. The first step automatically groups any two points into the same cluster if they are mutual k -nearest neighbors. In the second step, the cluster backbones are obtained. In the third step, CLUB assigns each unlabeled point to the cluster which the unlabeled point's nearest higher-density neighbor belongs to.

Xie et al. (2016) introduce a density-peak-searching and point-assigning algorithm based on the fuzzy weighted k -nearest neighbor (FKNN-DPC). It explores the k -NN to calculate new local densities and then eliminates the noise. The authors were motivated by the P_{1S} problem of error dragging, and they propose a multistep labeling scheme. This method consists of two point-labeling strategies, the first classifying the core points and the second assigning the labels of the halo points.

Lu and Zhu (2017) set out an algorithm called DFC. The authors consider it better to use core points to represent one center. The aim is to partition the data into core points and non-core points using neighborhood density estimation, and specifically they use reverse k -nearest neighborhood and then do the clustering with the minimum spanning-tree clustering. The parameters of the algorithm are k (the number of nearest neighbors) and the number of real clusters.

Liu et al. (2018) introduce the concept of shared nearest neighbors (SNN) to the definition of density and distance to

take account of local structure in the data. The authors also adopt a two-step allocation strategy to assign labels to the so-called inevitable subordinate points and possible subordinate points. Table 4 summarizes some of the studies in this line of research L_3 .

2.4 Contributions of the paper

This paper describes a methodology based on internal validity indices for automatically establishing the parameters of density-based peak clustering algorithms. The methodology is general and applicable to an arbitrary algorithm \mathcal{A} like those previously reviewed. The approach has been illustrated for the original DPC algorithm and for FKNN-DPC, as they use different parameters, d_c and k respectively. A numerical study performed shows that automatically adjusting the parameters either of the DPC or of the FKNN-DPC gives statistically significant results that are better than for the default values.

This method is also defined with an external validity index, but in this case it is only applied when a gold-standard clustering is known, and this situation does not exist in a practical application. However, the innovation of this paper is to apply these measures, as V-Measure index, as a research methodology to study the maximum performance of the algorithm \mathcal{A} . We have called this *optimal rules*. In the literature review on the line of research L_2 , we saw a number of strategies proposed for solving the problems P_c and P_{d_c} in the algorithm

Table 4 Review of L_3 multistep labeling

References	Algorithm	P_ρ	P_1S	P_{dc}	P_c	P_O	Notes
Liu et al. (2018)	SNN-DPC	✓	✓	✓			Defines distance and density using information on the nearest neighbors and the shared neighbors. SNN-DPC introduces a two-step allocation method in order to improve the performance in complex datasets such as multi-scale, cross-winding, variable-density and high-dimensional datasets.
Lu and Zhu (2017)	DFC	✓	✓				Two-step assignment rule for core and non-core points. The first step is based on a minimum spanning tree and the second on an ordering mechanism. DFC uses a neighborhood density model.
Xie et al. (2016)	FKNN-DPC	✓	✓				Uses local density based on nearest neighbors and a two-step assignment rule. The first step assigns a subset of points by undertaking a breadth-first search of the k -nearest neighbors. The second step assigns the unassigned points using the technique of fuzzy weighted k -nearest neighbors.
Chen et al. (2016)	CLUB	✓	✓				CLUB is carried out in four steps, namely (1) Find the initial clusters, (2) Identify cluster-density backbones, (3) Assign the remaining points, and (4) Detect outliers.
Xu et al. (2019)	FDPC		✓				Uses a merging strategy based on support vector machine in order to avoid that a cluster is divided into multiple ones if there exist several density peaks in one cluster.

$\mathcal{A} = \text{DPC}$. Optimal rules delimit the performance of these strategies and so identify improvement possibilities of \mathcal{A} with respect to them within the state of the art. This paper analyzes the improvement possibilities of $\mathcal{A} = \text{DPC}$ with respect to the strategies for solving the problems P_c and P_{dc} and also $\mathcal{A} = \text{FKNN-DPC}$ with respect to the choice of centers \mathbf{c} and the parameter k .

3 Review of density peak clustering

Alex Rodríguez and Alessandro Laio proposed the so-called *Density-Peak Clustering algorithm* (DPC). In the first stage, this method locates the clustering centers, and in the second the DPC algorithm uses locality to assign cluster labels to the remaining points, and thus it can detect clusters in non-convex shapes. The characterization of the centers is based on the following two simple assumptions (Li and Tang 2018):

- **Assumption 1.** Cluster centers are surrounded by neighbors with lower local density.
- **Assumption 2.** Cluster centers are at a relatively large distance from any points with a higher local density.

The mathematical formulation of these two hypotheses requires the definition of two magnitudes for each point i : i) its *density* ρ_i and the ii) *distance* δ_i from point i to the nearest point with a higher density. Assume that the dataset is $X = [x_1, \dots, x_N]^T \in \mathbb{R}_{N \times d}$ where each point $x_i \in \mathbb{R}_{d \times 1}$. Instead of representing the data as a matrix, we consider it as a set of points called \mathcal{X} . That is

$$\mathcal{X} := \{x_i : i = 1, \dots, N\}$$

Let d_{ij} denote the Euclidean distance between the points x_i and x_j . With density ρ_i , depending on the size N , one may calculate:

- In *datasets* made up of a large quantity of data, the following expression is used:

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c), \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \quad (1)$$

which computes the number of neighboring points of i at a distance less than d_c . The parameters d_{ij} express the Euclidean distance from point i to point j .

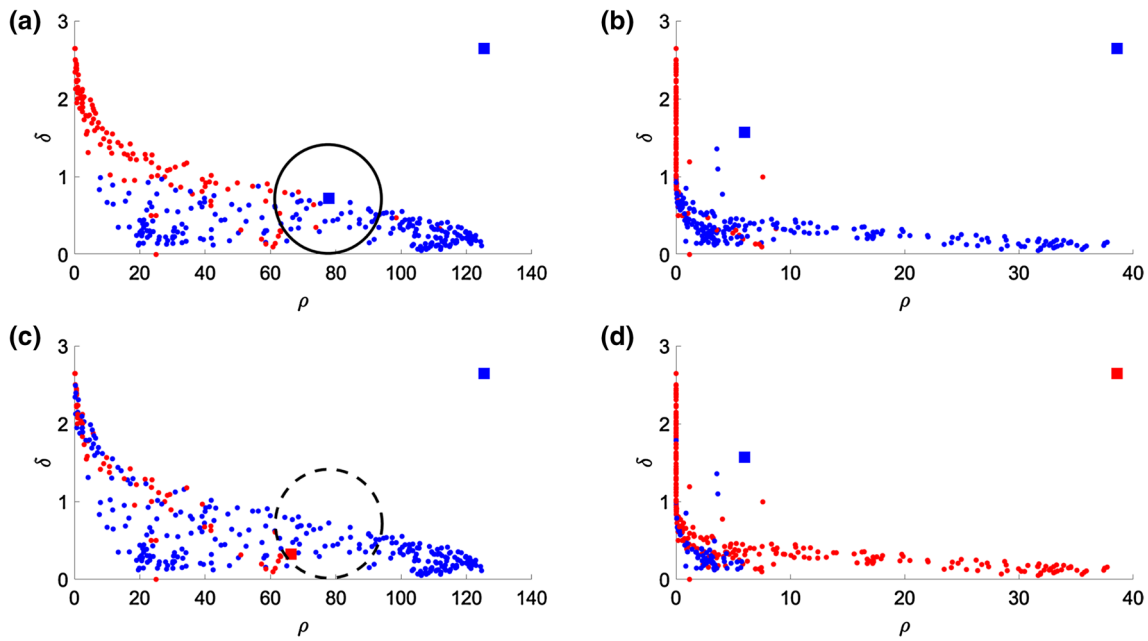


Fig. 1 Illustration of the optimization of centers

- If the quantity of data is reduced, the density is calculated based on Gaussian kernels

$$\rho_i = \sum_{j \neq i} \exp \left[- \left(\frac{d_{ij}}{d_c} \right)^2 \right] \quad (2)$$

In the expressions (1) and (2), it is necessary to specify the cutoff distance d_c in order to determine the densities ρ_i . Originally it was proposed to calculate this value by using the quantiles of the distance distribution between the elements, with the expression:

$$d_c(p) := \text{quantile}_{\frac{p}{100}}(D) \quad (3)$$

which gives the $\frac{p}{100}$ - th quantile of the distribution of $D = \{d_{i,j} : i < j\}$ and the parameter p is a percentage fixed by the user. The advantage of using the parameter p rather than d_c is that p can be interpreted on the set of all instances, whereas the value d_c varies from one problem to another, depending on the scale of the data \mathcal{X} .

Once the densities are computed, the next stage is to calculate the minimum distance from each point i to the point j of higher density. This distance is defined by:

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}) \quad (4)$$

Once ρ_i and δ_i are obtained, a decision graph is constructed (see Fig. 1) with the purpose of selecting the centers, which will be points that are more separate from the rest of the graph, as they have a higher density than their neighbors,

and are also separated from the other candidates. This graph shows that the cluster centers have a value ρ_i and δ_i higher than the other points that are not centers. The above procedure requires human intervention to select them manually. A heuristic for finding the centers automatically is to define $\gamma_i = \rho_i \delta_i$ and to choose the points k with the highest values of γ_i . Figure 1(a), 1(b) and 1(d) show the centers calculated in this way as squares.

Algorithm 1 DPC algorithm

Input : The sample $X \in \mathbb{R}^{N \times d}$

The cutoff distance parameter d_c

Output: The label vector of cluster index: $y \in \mathbb{R}^{N \times 1}$

Function DPC (X, d_c) :

Step 1. Calculate distance matrix (d_{ij})

Step 2. Calculate ρ_i for point i according to Formula (1) or (2)

Step 3. Calculate δ_i for point i according to Formula (4)

Step 4. Plot decision graph and select cluster centers

Step 5. Assign each remaining point to the nearest cluster center

Step 6. **return** y

end

4 Methodology for analyzing the density peak clustering algorithms

4.1 Statement of a generalized optimization problem

Suppose we have a clustering algorithm \mathcal{A} based on density peaks. In an abstract form, the operation of the algorithm may be described as: (1) given a parameterization $p \in \mathcal{P}$ calculate the densities ρ and distances δ , (2) identify the centers $\mathbf{c} \subseteq \mathcal{X}$

from ρ and δ and finally (3) assign labels to the data starting from the centers. The output of this algorithm is the data partition, expressed by a label vector y , that is y_i is the group that the i -th data point belongs to. We are here considering any algorithm \mathcal{A} these three stages are carried out separately. For example, the algorithm \mathcal{A} could be the DPC itself, or another variant derived from it, such as the FKNN-DPC.

The three stages of the algorithm \mathcal{A} are expressed formally as:

$$p = r_p^{\mathcal{A}}(\mathcal{X}) \quad (5)$$

$$\mathbf{c} = r_{\mathbf{c}}^{\mathcal{A}}(p, \mathcal{X}) \quad (6)$$

$$y = \mathcal{A}(\mathbf{c}, p, \mathcal{X}) \quad (7)$$

which, given a dataset \mathcal{X} , the equations (5) and (6) give the parameterization and the centers where the algorithm \mathcal{A} should be started to assign the cluster labels to the data items.

The problems $P_{\mathbf{c}}$ and P_{d_c} introduced in Sect. 2 seek to determine good mappings $r_{\mathbf{c}}^{\mathcal{A}}(\cdot)$ and $r_p^{\mathcal{A}}(\cdot)$ with $p = d_c$ to improve the performance of the algorithm \mathcal{A} . The question we consider is what rules for determining centers and for obtaining parameters will produce maximum efficiency of the algorithm \mathcal{A} . The following definition addresses this question.

Definition 1 (*Optimal rules*) Let \mathcal{Q} be a quality index of clustering, we say that the rules $r_{\mathbf{c}}^*$ and r_p^* are optimal for \mathcal{X} with respect to \mathcal{Q} iff $r_p^*(\mathcal{X}) = p^*$ and $r_{\mathbf{c}}^*(p^*, \mathcal{X}) = \mathbf{c}^*$ where (\mathbf{c}^*, p^*) is an optimal solution of the following generalized optimization problem:

$$\begin{aligned} &\text{Maximise } \mathcal{Q}(y) \\ &\quad (\mathbf{c}, p) \\ &\quad y = \mathcal{A}(\mathbf{c}, p, \mathcal{X}) \\ &\quad \mathbf{c} \subseteq \mathcal{X} \\ &\quad p \in \mathcal{P} \end{aligned} \quad (8)$$

This is not a standard optimization problem as the constraint $\mathbf{c} \subseteq \mathcal{X}$ shows that we are seeking a subset of vectors rather than a single vector, i.e., $\mathbf{c} \in \mathcal{X}$.

The problem (8) is stated by the definition of the objective function \mathcal{Q} . In the cluster analysis, it is held that \mathcal{Q} can be: (i) internal or (ii) external validity indices. Both internal and external indices are used to assess the performance of

clustering algorithms. Wiwie et al. (2015) point out that the internal validity measures judge a clustering on the basis of certain intrinsic statistical properties of the clustering itself, whereas external indices compare the clustering to a user-given gold-standard clustering ℓ (the "ground truth"). The external indices are only applied when ℓ is known and this situation does not exist in a practical application. In the internal indices, \mathcal{Q} does not depend on ℓ .

This study uses external and internal indices \mathcal{Q} for two different purposes:

- The use of the optimal rules allows the theoretical possibility of improvement of a clustering method \mathcal{A} by defining new rules $r_{\mathbf{c}}(\cdot)$ and $r_p(\cdot)$ to be assessed. Given two arbitrary rules $r_{\mathbf{c}}(\cdot)$ and $r_p(\cdot)$ and given the solution of the clustering $y = \mathcal{A}(r_{\mathbf{c}}(p, \mathcal{X}), r_p(\mathcal{X}), \mathcal{X})$ and of the optimality of $y^* = \mathcal{A}(r_{\mathbf{c}}^*(p, \mathcal{X}), r_p^*(\mathcal{X}), \mathcal{X})$, it is hold

$$\mathcal{Q}(y) \leq \mathcal{Q}(y^*) \quad (9)$$

Equation (9) allows the improvement possibilities of an algorithm to be bounded, so it can be compared to other existing algorithms. In this context, it is more suitable to use external validity indices as what is sought is to assess the capacity of the algorithm to discover the *ground truth* ℓ .

- The use of internal indices \mathcal{Q} , as it does not require the input ℓ , allows Problem (8) to be defined for any dataset. Its solution implicitly defines the optimal rules and generates a new algorithm \mathcal{A} in which the choice of centers and parameters is performed automatically. A particularly important case is to consider Problem (8) exclusively for the parameters p

$$\begin{aligned} &\text{Maximise } \mathcal{Q}(y) \\ &\quad p \\ &\quad y = \mathcal{A}(r_{\mathbf{c}}^{\mathcal{A}}(p, \mathcal{X}), p, \mathcal{X}) \\ &\quad p \in \mathcal{P} \end{aligned} \quad (10)$$

and this leads to a variant of the algorithm \mathcal{A} in which the parameters are automatically adjusted for each dataset.

The purpose of the optimal rules depends on the type of validity index. From now on, let $\mathcal{Q}(y)$ be the external validity indices and $\widehat{\mathcal{Q}}(y)$ the internal validity indices. We shall now summarize the utility of the optimal rules

$$\text{Optimal } \mathcal{Q}(y) \text{ rules} = \begin{cases} \text{External index } \mathcal{Q}(y) \rightarrow & \text{Bounding of the performance} \\ \text{Internal index } \widehat{\mathcal{Q}}(y) \rightarrow & \begin{cases} \bullet \text{ Automatic parameter/center selection} \\ \bullet \text{ Automatic parameter-tuning approach} \end{cases} \end{cases}$$

The methodology described is defined by the validity index chosen and by the optimization algorithm used. The following subsections address these questions.

4.2 External index $Q(y)$

V-Measure (Rosenberg and Hirschberg 2007) is an external entropy-based cluster evaluation measure. It accurately assesses two desirable aspects of clustering, homogeneity and completeness. We now give the definition. Suppose we have two clustering solutions defined by two label vectors y and ℓ . The grouping y was generated by the \mathcal{A} algorithm, while the gold labels define ℓ . Each of these solutions takes, respectively, values in $s \in \{1, \dots, m\}$ and $r \in \{1, \dots, n\}$. We shall call the marginal and joint probabilities that an object is classified in certain categories s and r :

$$p_s(y) = \frac{|\{i \in \{1, \dots, N\} : y_i = s\}|}{N} \quad (11)$$

$$p_r(\ell) = \frac{|\{i \in \{1, \dots, N\} : \ell_i = r\}|}{N} \quad (12)$$

$$p_{s,r}(y, \ell) = \frac{|\{i \in \{1, \dots, N\} : y_i = s \wedge \ell_i = r\}|}{N} \quad (13)$$

We define the entropy of the labels and the mutual information

$$H(y) = -\sum_{s=1}^m p_s(y) \log p_s(y), \quad H(\ell) = -\sum_{r=1}^n p_r(\ell) \log p_r(\ell) \quad (14)$$

$$I(y, \ell) = \sum_{s=1}^m \sum_{r=1}^n p_{s,r}(y, \ell) \log \left(\frac{p_{s,r}(y, \ell)}{p_s(y) p_r(\ell)} \right) \quad (15)$$

We define the homogeneity $h(y, \ell)$ and the completeness $c(y, \ell)$

$$h(y, \ell) = \begin{cases} 1 & \text{If } H(\ell) = 0 \\ \frac{I(y, \ell)}{H(\ell)} & \text{Otherwise.} \end{cases} \quad c(y, \ell) = \begin{cases} 1 & \text{If } H(y) = 0 \\ \frac{I(y, \ell)}{H(y)} & \text{Otherwise.} \end{cases} \quad (16)$$

The V-Measure computes the harmonic mean of distinct homogeneity and completeness scores by

$$V_\beta(y, \ell) = \frac{(1 + \beta)h(y, \ell)c(y, \ell)}{\beta h(y, \ell) + c(y, \ell)}. \quad (17)$$

If β is greater than 1, completeness is weighted more strongly in the calculation, otherwise β is less than 1 and the homogeneity is weighted more strongly. This study considers both factors to be equally important and takes $\beta = 1$.

The study considers as the external validity index

$$Q(y) = V_1(y, \ell) \quad (18)$$

This methodology can be extended to other common external validation indices used in cluster analysis, such as the F_1 score, the Kappa concordance coefficient (López-García et al. 2015), etc. Other options, like the Purity (Zhao and Karypis 2001) and Entropy, only assess the homogeneity of a solution and thus we have chosen the V-measure.

4.3 Internal index $\hat{Q}(y)$

Ideally, when choosing an internal validity index, we seek a measurement that is highly correlated with the chosen external index, in this case the V-measure, such that when the internal index is optimized the external index is being optimized indirectly. When looking for an index such that its optimization leads to a maximization of the V-Measure, we thought of measures that involve minimization of entropy, as this would maximize completeness, and thus, as V-Measure is a harmonic mean of $c(y, \ell)$ and $h(y, \ell)$, it would also maximize the V-Measure. This was what motivated our choice.

To define an entropy for clustering tasks (Criminisi et al. 2011) use the differential (continuous) entropy of a d -multivariate Gaussian random variable

$$H(U) = \frac{1}{2} \ln \left((2\pi e)^d |\Sigma_U| \right) \quad (19)$$

with Σ_U the $d \times d$ associated covariance matrix of the data U and $|\cdot|$ indicating the determinant of the matrix. Consequently the Gaussian partition entropy y reduces to

$$H_G(y) = \frac{d}{2} \log(2\pi e) + \frac{1}{2} \sum_{s=1}^m p_s(y) \ln |\Sigma_s(y)| \quad (20)$$

where $\Sigma_s(y)$ is the covariance matrix of the data $\{x_i : y_i = s\}$.

Tabor and Spurek (2014) study the Gaussian cross-entropy clustering and show that it minimizes the objective cost function:

$$H_G^\times(y) = \frac{d}{2} \log(2\pi e) - \sum_{s=1}^m p_s(y) \ln p_s(y) + \frac{1}{2} \sum_{s=1}^m p_s(y) \ln |\Sigma_s(y)| \quad (21)$$

Finally, the internal validation indices proposed for $\hat{Q}(y)$ are $H_G(y)$ or $H_G^\times(y)$. To our knowledge, it is the first time that these indices are used as internal validity measures (Liu et al. 2010).

The covariance matrix $\Sigma_s(y)$ may be singular if the quantity of data $\{i : y_i = s\}$ is small with respect to the dimension of the space d and in this case the function $\hat{Q}(y)$ is not defined. We therefore use a dimensionality reduction technique, specifically principal component analysis (PCA). The

basic idea of PCA is to project the original data onto a lower-dimensional subspace, which highlights the principal directions of variation of the data. Another advantage of its use is that it speeds up the parameter-tuning algorithm.

The following steps describe the PCA procedure.

1. Make each of the variables have the same mean (zero) and variance.
2. Calculate the covariance matrix Σ of the data X .
3. Calculate the eigenvectors u_k and the eigenvalues λ_k of Σ .
4. Sort these eigenvalues in decreasing order and stack the eigenvectors u_k corresponding to the eigenvalues λ_k in columns to form the matrix U .
5. Select the first d' columns of U , and name this matrix $U_{d'}$. Thus the projected data $X_{d'} = XU_{d'}$.

To decide how to set the number of retained components (d'), we will usually look at the percentage of variance retained for different values of d' . We pick the smallest value of d' that satisfies that the percentage of variance retained is greater than $1 - \varepsilon$:

$$\frac{\sum_{k=1}^{d'} \lambda_k}{\sum_{k=1}^d \lambda_k} \geq 1 - \varepsilon \quad (22)$$

This is the data initialization process. The procedure followed is trial and error. We initialize with the value ε , and if during the running it is not possible to compute $\widehat{Q}(y)$, then the retained variance is reduced by adjusting the threshold to $\varepsilon = \varepsilon + \Delta\varepsilon$ and the whole process is repeated.

4.4 A heuristic approach

The problem (8) has a bilevel structure, in which, at the outer level, it is optimized with respect to the vector of parameters p and, at the inner level, optimization is carried out on the set of centers \mathbf{c} . The problem structure is due to the optimization variables p and \mathbf{c} being subordinated. This relation between the variables occurs because the densities ρ_i depend on the parameter p and at the same time these densities $\rho_i(p)$ determine which centers $\mathbf{c}(\rho_i(p))$ will be chosen.

To illustrate this, consider the following example. Figure 1 shows four decision graphs for the real-world dataset Ionosphere and for the algorithm DPC. The Ionosphere dataset has two clusters assigned the colors red and blue. Graphs (a) and (b) show the true class of the points, while (c) and (d) show the solutions found by the DPC algorithm in two different situations. On the other hand, Graphs (a) and (c) are calculated with the same optimal value of the param-

eter d_c obtained with our method, while Graphs (b) and (d) are obtained using the default value $d_c = d_c(2\%)$. The first observation is that the shape of the decision graph depends on the parameter d_c , and these graphs will determine the centers \mathbf{c} ; this therefore shows the subordination of the decisions of the centers to the parameters, i.e., $\mathbf{c}(d_c)$. In Graphs (a) and (b), a square marks the points of greatest γ . If the centers are chosen by the rule of maximum γ , there will be errors because both centers belong to the same cluster (the blue one). As seen in Graph (c), this error is carried with a domino effect to the other points. This motivates the use of a heuristic algorithm to assign the suitable centers to the clusters.

The optimization algorithm we propose considers a neighborhood around the starting center with the highest gamma value, marked by a black circle in figure (a), and searches among the elements of the neighborhood for alternative centers. If a candidate center \mathbf{c}' improves the validity index, the center is updated and the search environment is moved around \mathbf{c}' . Figure (c) shows the result of optimizing the center. It is seen that the new center is located inside the point cluster.

The structure of the nested optimization problem (8) is expressed formally as:

$$\text{Maximise}_{p \in \mathcal{P}} \left(\text{Maximise}_{\mathbf{c} \in \mathcal{X}} Q(\mathcal{A}(\mathbf{c}, p, \mathcal{X})) \right) \quad (23)$$

The solution procedure consists of an inner and an outer optimization method. The inner method optimizes the validity index Q for a known value of p with respect to the centers \mathbf{c} , and this procedure is denoted by `ClusterCentersOptimisation`(p, \mathcal{X}). The outer method we propose performs an exhaustive search on the list P of parameter values, i.e., $p \in P \subseteq \mathcal{P}$. Finally, this solution method is expressed formally as:

$$\text{Maximise}_{p \in P \subseteq \mathcal{P}} \text{ClusterCentersOptimisation}(p, \mathcal{X}) \quad (24)$$

Note that the expression (24) defines the parameter-tuning problem. This problem optimizes an expensive black-box function, and therefore the solution method based on an exhaustive search may turn out not to be practical in some real cases. In these situations, a surrogate-assisted optimization must be used.

The optimization problem (8) is not standard, as the decision variables are a set of vectors, the centers \mathbf{c} , rather than a single vector. This makes it impossible to use metaheuristic algorithms directly. Furthermore, an exhaustive search cannot be carried out as the number of subsets $\mathbf{c} \subseteq \mathcal{X}$ is $\binom{N}{n_c}$ where n_c is the number of centers, and this number increases exponentially with the size of the dataset N .

We describe a `ClusterCentersOptimisation` method. Its main elements are:

- i) *The definition of the neighborhood of the set \mathbf{c} .* Let $\mathbf{c} = \{c_1, \dots, c_{n_c}\}$, and we define the neighborhoods of \mathbf{c}

$$\mathcal{C} = \cup_i \mathcal{C}_i \quad (25)$$

where \mathcal{C}_i is the set of q -nearest neighbors of point c_i . To determine the q neighbors closest to c_i , we use the Euclidean distance in the decision graph $(\rho_i, \delta_i) \in \mathbb{R}^2$ instead of calculating it in the space of characteristics \mathbb{R}^d .

- ii) *The probability distribution over the neighborhood \mathcal{C} .* To choose a new set of centers $\mathbf{c}' \subseteq \mathcal{C}$ a random sampling in two stages was performed.

- The first stage chooses the centers to be changed. Their number is calculated by $n_w = \lfloor w(n_c - 1) \rfloor$ with $0 \leq w \leq 1$ and $\lfloor \cdot \rfloor$ is by default the floor function. We decide by lot in the set $\{2, 3, \dots, n_c\}$ the n_w centers to be changed. The probability used in the draw is inversely proportional to the parameter γ_i . We call the result of the draw $I \subseteq \{2, 3, \dots, n_c\}$.
- In the second stage, for each $i \in I$ we choose a new center c'_i in the neighborhood \mathcal{C}_i , by performing a new random draw among the neighbors $c_j \in \mathcal{C}_i$ with probability proportional to the value of γ_j .

The resulting method, called `ClusterCentersOptimisation`, is shown in Algorithm 2. This algorithm assumes that the number of clusters n_c is known to carry out the computational experiment. This is a reasonable assumption for an external index as the true labels ℓ are known. However, this algorithm can be extended to a variable number of clusters n_c by adding another draw to reduce/increase the value of n_c . Therefore, in real cases where the number of clusters is unknown, the parameter n_c can be adjusted by optimization in (24) with respect to an internal validity index sensitive to the choice of n_c such as the Silhouette index.

The parameters n_{iter} , q and w have an effect on minimizing the solution time. They are related in order to explore the decision graph near the center to be exchanged. A large number of neighbors q will mean that not too many iterations n_{iter} are necessary, as they explore most of the points close to the cluster center. This means there is no significant impact on the results when these parameters are adjusted. The parameter w determines the number of centers to be replaced in each iteration. A high value will allow a greater exploration of the space, while low values help the exploitation.

Algorithm 2 Cluster center optimization for a fixed p value

Input : Let $p \in \mathcal{P}$ be the parameter of the algorithm \mathcal{A} .

Let \mathcal{X} be the set of data.

Let n_c be the number of clusters.

Let n_{iter} be the number of main iterations.

Let q be the number of neighbors.

Let w be the proportion of changed centers in each iteration.

Output: An optimal set of cluster centers $\mathbf{c}^* \subseteq \mathcal{X}$ and the optimal validity index Q^* .

```

1 Function ClusterCentersOptimisation( $p, \mathcal{X}, n_c, n_{iter}, q, w$ ):
  // ----- I n i t i a l i z a t i o n -----
  //
  2 Compute  $\rho_i$  and  $\delta_i$  for each point  $i$  according to the rules used in
    the algorithm  $\mathcal{A}$  and for the given  $p$  value.
  3 Sort the index  $i$  according to  $\gamma_i = \rho_i \delta_i$  values such that  $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_N$ .
  4 Let  $\mathbf{c}^* = \{x_1, x_2, \dots, x_{n_c}\}$  be the initial clusters centers.
  5 Perform the clustering algorithm,  $y^* = \mathcal{A}(\mathbf{c}^*, p, \mathcal{X})$ .
  6 Evaluate the quality of the clustering solution  $y^*$  using the validity
    index,  $Q^* = Q(y^*)$ .
  // ----- M a i n   i t e r a t i o n -----
  s ----- //
  7 for each  $t \in \{1, 2, \dots, n_{iter}\}$  do
    8 Draw without replacement  $\lfloor w(n_c - 1) \rfloor$  values in the set  $i \in \{2, 3, \dots, n_c\}$  with probability  $p_i = \frac{\frac{1}{\gamma_i}}{\sum_{j=2}^{n_c} \frac{1}{\gamma_j}}$ . Call the result
      of the draw  $I$ .
    9 For each  $i \in I$  compute the set of the  $q$  nearest neighbors of
      the cluster center  $c_i$  in the decision graph and call it  $\mathcal{C}_i$ .
    10 for each cluster center  $i \in I$  do
      11 Use the roulette wheel selection rule for selecting a new
        center  $c'_i \in \mathcal{C}_i$  according to its probability  $p_{c'_i}$  where
         $p_{c'_i} = \frac{\gamma_{c'_i}}{\sum_{c_j \in \mathcal{C}_i} \gamma_{c_j}}$ .
    12 end
    13 Let  $\mathbf{c}' = \{c'_i\}_{i \in I} \cup \{c_i\}_{i \notin I}$  be the tentative cluster centers.
    14 Perform the clustering algorithm  $y' = \mathcal{A}(\mathbf{c}', p, \mathcal{X})$  and compute
      the validity index,  $Q' = Q(y')$ .
    15 if  $Q^* < Q'$  then
      16 | Set  $\mathbf{c}^* = \mathbf{c}'$  and  $Q^* = Q'$ .
    17 end
    18 end
    19 return  $\mathbf{c}^*$  and  $Q^*$ .
20 end

```

5 Experimental results

This section describes a computational experiment to evaluate the proposed methodology based on optimal rules and paying special attention to automatic parameter tuning of density peak algorithms. We consider two basic algorithms \mathcal{A} , the original DPC and the FKNN-DPC introduced in Xie et al. (2016). $\hat{Q}(y) = H_G(y)$ is used as the internal validation index, defined by Eq. (20) and as the external validation method $Q(y)$ the V -Measure defined in Eq. (18). The following notation $\mathcal{A}(a)$ is used to refer to the fact that in the algorithm \mathcal{A} the input a is adjusted with respect to an external validity index, while the notation $\hat{\mathcal{A}}(a)$ indicates the same thing but with respect to

Table 5 Details of datasets

Dataset	# Data (N)	# Attributes (d)	# Clusters (n_c)	References
Br. cancer	699	9	2	Dheeru and Karra Taniskidou (2017)
Ionosphere	351	33	2	
Opt. digits	5618	64	10	
Pen. digits	10,992	16	10	
Voters	435	16	2	
Image seg.	2309	19	7	
Satellite	6435	36	6	
Chart	600	60	6	
Smartphone	10,929	561	12	
Soybean	682	35	19	
Dermatology	366	34	6	
Glass	214	9	6	
Isolet	6238	617	26	
Parkinsons	195	22	2	
Internet ads	3279	1,558	2	
Face	5850	1200	10	Lee (2005)

an internal validity index. The algorithms analyzed by combining these options are $DPC(c^*, d_c^*)$, $FKNN-DPC(k^*)$, $FKNN-DPC(c^*, k^*)$, $\widehat{FKNN-DPC}(k^*)$, $\widehat{FKNN-DPC}(c^*, k^*)$

The experimental results are structured into the following experiments:

- *Experiment 1: Evaluation of optimal rules using external cluster validity indices.* The purpose of the first experiment is to assess the impact of using optimal rules r_c^* and r_p^* on the performance of the baseline algorithm \mathcal{A} and to put it in the context of a number of state-of-the-art algorithms. If the algorithm \mathcal{A} is DPC, the optimal rules represent the best solution to the problems P_c and P_{d_c} described in the introduction, and give an upper bound on the performance of the proposed algorithms to improve the center selection.
- *Experiment 2: Automatic parameter tuning using internal cluster validation indices.* This experiment aims to assess the methodology for automatic parameter tuning.
- *Experiment 3: Comparison of algorithms.* The goal is to evaluate the performance of the peak-density-based algorithms with respect to the state-of-the-art clustering methods.

5.1 Datasets, algorithms and parameter settings

Experiments have been carried out using the following 16 benchmark datasets from UCI Machine Learning Repository and UCSD Computer Vision: Breast cancer Wisconsin original, Ionosphere, Optical recognition of handwritten digits, Pen-based recognition of handwritten digits, Congressional

voting records, Statlog image segmentation, Statlog landsat satellite, Synthetic control chart time series, Smartphone-based recognition of human activities and postural transitions, Soybean disease, Dermatology, Glass identification, Isolet, Parkinsons, Internet advertisements and Yale face database B. In Table 5, we can see their characteristics: the number of data points, the number of attributes and the number of clusters. These datasets are chosen because (i) they are real problems, not synthetic, in which golden labels ℓ are known and they allow the ability of the algorithm to reconstruct the true partition to be tested and (ii) it consists of a collection of benchmark datasets widely used in the literature.

To preprocess the datasets, we replace the missing attribute values with the average of the attribute vector, and also normalize the data through min-max normalization.

The following state-of-the-art methods are used: NCutH, which is a divisive clustering algorithm based on a binary partitioning tree by normalized graph cuts across optimal hyperplanes; NCutH₀, which establishes the maximum margin hyperplane and provides a balanced partition of the data assigned to its respective leaf; K -means that has been used with the default implementation in R and results from the best solution from 10 initializations are considered; Bisecting K -means (Bis. K -m); Normalized Spectral Clustering (SCn); Iterative Support Vector Regression (ISVR); and Density enhanced principal direction divisive partitioning (dePDDP) were selected to compare their performances with the DPC and FKNN-DPC algorithms. The results have already been obtained from Hofmeyr (2017), and the setting of parameters can be viewed in this reference.

To carry out the experiment, we have obtained the code of DPC provided by Rodríguez and Laio Rodríguez and Laio (2014). The original DPC code needs the cutoff distance d_c to obtain the local density of each point, so according to both authors, we have used Eq. (3) to compute d_c taking $p = 2\%$. In addition, the local density is estimated by a Gaussian kernel. The number of clusters of the datasets is known (n_c), and we have chosen as centers of the algorithms DPC and FKNN-DPC the n_c points with the highest value of γ .

Regarding FKNN-DPC, we have coded it according to Xie et al. (2016). As it is necessary to specify the parameter k (number of nearest neighbors), we have chosen $k = \lceil 0.015N \rceil$ where $\lceil \cdot \rceil$ is the ceil of a number.

The algorithms DPC, FKNN-DPC and their corresponding variants are coded in MATLAB. These programs are available at this address https://bitbucket.org/jcarlos1193/automatic_fknn-dpc.

As well as assessing the numerical results with respect to the V -Measure, we added a second external validity criterion: *Purity* (Zhao and Karypis 2001). Both metrics take values in the range $[0, 100]$, higher values are related to high performance. Purity is computed as the weighted average of the largest ratio of each cluster which is represented by a single class.

$$\text{Purity}(y, \ell) = \frac{1}{N} \sum_{s=1}^m \max_{r \in \{1, \dots, n\}} |y_s \cap \ell_r| \quad (26)$$

where y is the cluster solution and ℓ the true labels (classes).

5.2 Experiment 1: Evaluation of optimal rules using external cluster validity indices

We considered these parameters in the following algorithms:

- DPC(c^* , d_c^*), optimizes DPC with respect to the choice of the centers \mathbf{c} and the parameter d_c . DPC(c^* , d_c^*) search on the best value of d_c in the list $P = \{d_c(0.5\%), d_c(1\%), \dots, d_c(25\%)\}$ computed using the Eq. (3). We have set $w = 0.7$, $q = \lceil 0.03N \rceil$ and $n_{iter} = 150$.
- FKNN-DPC(c^* , k^*) optimizes the algorithm FKNN-DPC with respect to the centers \mathbf{c} and the parameter k . In this case the list of value for k is $P = \{1, 2, \dots, 50\}$. The rest of the parameters are the same as DPC(c^* , d_c^*).
- FKNN-DPC(k^*) only optimizes the k parameter through a list of values $P = \{1, 2, \dots, 50\}$, and the centers are chosen to be n_c points with the highest values of γ_i , using the best according to V -Measure.

Tables 6 and 7 show, respectively, V -Measure and Purity for Experiment 1. On the one hand, the well-known algorithms are introduced on the left of the tables, while the DPC

Table 6 Comparison of state-of-the-art clustering methods with V -Measure metric

Datasets	NCutH	NCutH0	iSVR	dePDDP	K-means	BisKm	SCn	DPC	FKNN	DPC(c^* , d_c^*)	FKNN(k^*)	FKNN(c^* , k^*)
Br. cancer	78.80	78.68	55.34	77.99	71.59	71.59	82.45	21.30	68.09	63.02	81.69	81.69
Ionosphere	13.49	13.49	12.64	9.82	12.50	12.50	20.67	8.60	35.25	33.91	38.36	37.26
Opt. digits	71.62	74.65	71.49	28.52	61.30	61.68	50.01	74.93	65.50	83.19	88.22	93.15
Pen. digits	70.95	73.07	72.53	59.15	68.93	65.71	48.63	73.55	63.58	75.31	78.01	83.73
Voters	42.82	42.39	33.59	39.25	42.30	42.30	40.48	46.45	21.12	50.49	50.59	51.09
Image seg.	59.38	63.81	62.61	35.15	59.87	58.63	47.90	73.07	68.57	76.21	70.58	76.96
Satellite	59.63	62.68	54.61	61.05	61.30	60.15	53.30	60.40	55.40	62.08	68.75	70.74
Chart	79.65	77.05	66.41	77.75	77.43	76.07	82.36	66.49	80.03	73.01	81.40	84.39
Smartphone	61.22	60.04	-	51.03	56.95	56.25	50.30	48.83	48.82	62.37	69.52	77.06
Soybean	80.29	71.87	75.38	67.68	74.13	80.19	73.46	57.24	68.66	70.12	70.49	70.36
Dermatology	93.56	83.82	76.84	87.39	86.33	86.00	90.29	65.55	87.46	78.11	88.28	91.97
Glass	31.58	30.85	27.55	30.17	31.46	31.18	28.79	28.29	22.90	38.13	29.07	43.44
Isolet	65.34	70.36	-	42.34	71.67	63.74	51.94	54.38	41.65	56.23	76.38	78.50
Parkinsons	21.96	21.96	6.38	1.78	12.42	12.42	1.20	25.49	15.41	27.62	25.19	30.32
Internet ads	33.18	37.05	1.85	3.10	8.23	8.23	1.59	0.28	1.01	24.06	1.44	17.72
Faces	87.35	92.18	-	37.55	75.16	63.26	72.43	78.05	66.75	81.60	83.99	87.80

and FKNN-DPC and their optimized versions are on the right. The highest values of both metrics in each dataset are highlighted in bold. Those cases in which results could not be obtained are marked with “-.”

We begin the analysis by considering the results of the V -Measure shown in Table 6. The reference algorithm for this study is the DPC. Looking at the results, we see that the raw version of this gives a balance of 5/11 (win/loss) over the K -means and of 6/10 over the NCutH. Nonetheless, this algorithm has room for improvement, as seen by the results obtained with optimal rules $DPC(c^*, d_c^*)$ where the balance is overturned and becomes 10/6 with respect to the K -means and 9/7 with respect to NCutH. The results obtained for the FKNN-DPC(k^*) are noteworthy as they give a balance of 12/4 over the $DPC(c^*, d_c^*)$. This may indicate that the emphasis of the research should be directed toward designing new multi-step algorithms along these lines L_3 with automatic parameter-tuning strategies, as is the case with the algorithm FKNN-DPC with its parameter k , rather than to refining strategies for choosing the centers c and the d_c in the DPC (research in this direction L_2). Another highlight is that FKNN-DPC(c^*, k^*) outperforms the other 11 methods. FKNN-DPC(c^*, k^*) gives a balance of 10/6 against the other algorithms. That is, of the 16 datasets this method is better in 10 datasets, and for the other 6 datasets it is worse than one or other of the 11 clustering algorithms.

With the optimization of the methods carried out with respect to the quality index V -Measure, the conclusions may vary with respect to the *Purity* measure. If we look at Table 7 related to *Purity* measure, we can observe similar conclusions to the previous table: (i) FKNN-DPC(c^*, k^*) obtains a higher performance than the rest of the algorithms again, and (ii) the algorithm FKNN-DPC(k^*) outperforms $DPC(c^*, d_c^*)$.

In this experiment, it is shown that the FKNN-DPC proposed by Xie et al. (2016) is promising in clustering tasks for both external indices, and thus we focus our attention on FKNN-DPC in Experiment 2.

5.3 Experiment 2: Automatic parameter tuning using internal cluster validation indices

The automatic parameter adjustment described is a procedure that seeks to optimize the external validity indices indirectly with respect to the parameters, via the optimization of the internal validity index. The procedure will work well if both indices are highly correlated. This experiment assesses the procedure for automatic parameter tuning for the algorithm FKNN-DPC based on the internal cluster validation index $\hat{Q}(y) = H_G(y)$. This is done by computing the degree of correlation between the external validation indices V -Measure and *Purity* and the proposed internal validation index H_G .

Table 7 Comparison of state-of-the-art clustering methods with *Purity* metric

Datasets	NCutH	NCutH0	iSVR	dePDDP	K -means	BisKm	SCn	DPC	FKNN	$DPC(c^*, d_c^*)$	FKNN(k^*)	FKNN(c^*, k^*)
Br. cancer	96.85	96.85	90.41	96.71	95.42	95.42	97.28	65.52	94.42	93.42	97.28	97.28
Ionosphere	71.23	71.23	70.37	68.95	70.66	70.66	72.93	64.10	83.76	81.48	84.90	84.62
Opt. digits	78.71	79.69	75.99	26.43	63.28	64.54	48.34	66.90	58.49	80.39	80.16	95.71
Pen. digits	77.98	76.83	78.25	56.73	72.23	72.09	41.71	72.05	58.04	77.16	73.75	83.10
Voters	84.60	84.37	81.15	85.06	84.60	84.60	84.27	86.44	77.47	87.82	87.59	87.82
Image seg.	62.19	63.79	64.62	29.32	59.95	58.55	55.91	74.72	64.98	78.74	65.45	70.39
Satellite	74.00	75.76	68.33	75.01	74.11	73.92	68.75	73.94	60.39	75.07	80.39	82.04
Chart	66.67	83.83	72.00	68.00	76.33	66.67	78.17	56.83	66.67	71.00	73.67	66.67
Smartphone	68.81	72.47	-	39.06	63.81	64.51	50.14	45.32	40.93	50.80	61.84	75.24
Soybean	80.79	67.45	71.85	60.56	67.74	78.01	72.87	51.54	66.18	65.01	67.64	67.64
Dermatology	96.17	85.52	80.33	85.52	86.07	86.07	94.54	68.03	86.34	79.78	86.34	86.34
Glass	54.21	58.88	51.40	48.13	54.21	53.27	51.40	55.14	45.33	52.34	46.26	52.80
Isolet	51.67	59.17	-	21.88	60.05	52.07	33.71	29.48	31.55	31.66	54.22	61.96
Parkinsons	75.38	75.38	75.38	75.38	75.38	75.38	75.38	75.38	80.51	75.38	82.05	84.62
Internet ads	90.79	89.54	86.00	86.34	86.95	86.95	86.09	86.00	86.00	89.57	86.15	86.00
Faces	79.83	91.35	-	35.91	73.73	62.02	59.74	67.50	55.33	76.32	78.53	86.70

Table 8 Comparison of FKNN-DPC versions using V-measure metric

Datasets	FKNN-DPC(k^*)	FKNN-DPC(c^*, k^*)	$\widehat{\text{FKNN-DPC}}(k^*)$	$\widehat{\text{FKNN-DPC}}(c^*, k^*)$
Br. cancer	81.69	81.69	78.79	78.79
Ionosphere	38.36	37.26	35.87	35.24
Opt. digits	88.22	93.15	87.07	92.41
Pen. digits	78.01	83.73	77.68	82.49
Voters	50.59	51.09	49.61	49.61
Image seg.	70.58	76.96	66.14	68.21
Satellite	68.75	70.74	62.15	68.13
Chart	81.40	84.39	80.56	79.13
Smartphone	69.52	77.06	66.22	68.04
Soybean	70.49	70.36	69.42	68.65
Dermatology	88.28	91.97	87.46	85.21
Glass	29.07	43.44	28.56	30.03
Isolet	76.38	78.50	75.37	76.53
Parkinsons	25.19	30.32	14.33	20.96
Internet ads	1.44	17.72	1.01	4.75
Faces	83.99	87.80	81.17	85.55

Table 9 Comparison of FKNN-DPC versions using Purity metric

Datasets	FKNN-DPC(k^*)	FKNN-DPC(c^*, k^*)	$\widehat{\text{FKNN-DPC}}(k^*)$	$\widehat{\text{FKNN-DPC}}(c^*, k^*)$
Br. cancer	97.28	97.28	96.71	96.71
Ionosphere	84.90	84.62	84.33	84.62
Opt. digits	80.16	95.71	84.04	95.39
Pen. digits	73.75	83.10	71.12	80.09
Voters	87.59	87.82	87.13	87.13
Image seg.	65.45	70.39	64.33	71.08
Satellite	80.39	82.04	74.89	81.13
Chart	73.67	66.67	66.67	73.33
Smartphone	61.84	75.24	53.94	71.54
Soybean	67.64	67.64	66.76	62.96
Dermatology	86.34	86.34	86.34	85.52
Glass	46.26	52.80	48.60	55.14
Isolet	54.22	61.96	56.62	63.24
Parkinsons	82.05	84.62	75.38	75.38
Internet ads	86.15	86.00	86.00	86.00
Faces	78.53	86.70	75.15	82.38

We shall compare, with respect to the V -Measure and the $Purity$ metrics, the solutions obtained for FKNN-DPC(c^*, k^*) with $\widehat{\text{FKNN-DPC}}(c^*, k^*)$, and also the algorithm FKNN-DPC(k^*) with $\widehat{\text{FKNN-DPC}}(k^*)$.

An optimization is performed of the indices $Q(y)$ and $\widehat{Q}(y)$ with the heuristic algorithm using the same parameters (and the same as in Experiment 1). Furthermore, in the trial/error application of the PCA we have taken $\varepsilon = \Delta\varepsilon = 0.05$ in Eq. (22).

The results obtained are shown in Table 8 for the V -Measure and Table 9 for the Purity index. As expected, both FKNN-DPC(k^*) and FKNN-DPC(c^*, k^*) algorithms

achieve the best results. We can also check that automatic parameter-tuning methods do not become much worse with respect to FKNN-DPC(k^*) and FKNN-DPC(c^*, k^*), except *Glass*, *Parkinsons* and *Internet ads* datasets. In addition, between $\widehat{\text{FKNN-DPC}}(k^*)$ and $\widehat{\text{FKNN-DPC}}(c^*, k^*)$ we can see how the second achieves better results because the search also includes the optimal cluster centers. Similar conclusions are obtained using the Purity measure in Table 9.

Wiwie et al. (2015) carry out a similar experiment on tuning the parameters using internal and external indices. The authors calculate the Spearman correlation coefficients between a number of indices. The best results were obtained

Table 10 Results of the Friedman test for pairwise multiple comparisons

	NCutH	NCutH ₀	iSVR	dePDDP	K-means	BisKm	SCn	DPC	FKNN	DPC(c^*, d_c^*)	FKNN(k^*)	FKNN(c^*, k^*)	$\widehat{\text{FKNN}}(k^*)$	$\widehat{\text{FKNN}}(c^*, k^*)$
NCutH		0.63	5.50	5.00	2.47	3.34	3.50	4.06	4.28	0.38	-1.72	-3.91	0.41	-0.31
NCutH ₀			4.88	4.38	1.84	2.72	2.88	3.44	3.66	-0.25	-2.34	-4.53	-0.22	-0.94
iSVR	*			-0.50	-3.03	-2.16	-2.00	-1.44	-1.22	-5.13	-7.22	-9.41	-5.09	-5.81
dePDDP	*	*			-2.53	-1.66	-1.50	-0.94	-0.72	-4.63	-6.72	-8.91	-4.59	-5.31
K-means	*	*	*	*		0.88	1.03	1.59	1.81	-2.09	-4.19	-6.38	-2.06	-2.78
BisKm	*	*	*	*	*		0.16	0.72	0.94	-2.97	-5.06	-7.25	-2.94	-3.66
SCn	*	*	*	*	*			0.56	0.78	-3.13	-5.22	-7.41	-3.09	-3.81
DPC									0.22	-3.69	-5.78	-7.97	-3.66	-4.38
FKNN		*								-3.91	-6.00	-8.19	-3.88	-4.59
DPC(c^*, d_c^*)			*	*			*	*	*		-2.09	-4.28	0.03	-0.69
FKNN(k^*)		*	*	*	*	*	*	*	*		*	-2.19	2.13	1.41
FKNN(c^*, k^*)		*	*	*	*	*	*	*	*	*	*	*	4.31	3.59
$\widehat{\text{FKNN}}(k^*)$			*	*	*	*	*	*	*	*	*	*	*	-0.72
$\widehat{\text{FKNN}}(c^*, k^*)$			*	*	*	*	*	*	*	*	*	*	*	*

Table 11 Classification of algorithms according to mean ranks

Ranking	Algorithm	Mean rank	p-value
1.	FKNN-DPC(c^*, k^*)	13.09	→ 0.01
2.	FKNN-DPC(k^*)	10.91	0.32
3.	$\widehat{\text{FKNN}}\text{-DPC}(c^*, k^*)$	9.50	1
4.	NCutH	9.19	0.62
5.	DPC(c^*, d_c^*)	8.81	0.62
6.	$\widehat{\text{FKNN}}\text{-DPC}(k^*)$	8.78	0.32
7.	NCutH ₀	8.56	0.13
8.	K-means	6.72	→ 0.03
9.	BisKm	5.84	0.13
10.	SCn	5.69	1
11.	DPC	5.13	0.32
12.	FKNN-DPC	4.91	0.62
13.	dePDDP	4.19	0.32
14.	iSVR	3.69	

for the Silhouette value as internal index, giving a correlation coefficient of $\rho = 0.71$ for the F_1 Score and $\rho = 0.66$ for the V-Measure. We calculated the coefficient for our internal index $\hat{Q}(y) = H_G(y)$ getting the values $\rho_{(c,k)} = 0.96$ and $\rho_k = 0.96$ with respect to the V-Measure and the values $\rho_{(c,k)} = 0.93$ and $\rho_k = 0.97$ with respect to the Purity index. This high correlation supports the $H_G(y)$ as a good choice of internal quality measure in automatic parameter tuning.

5.4 Experiment 3: Comparison of algorithms

The goal of Experiment 3 is to test whether the differences between the clustering algorithms considered are statistically significant. To detect the significant differences between pairs of clustering methods, we conduct a Friedman test over the V-Measure. Table 10 shows all comparisons between pairs of algorithms. In the lower triangle, the symbol “*” indicates whether there is a significant difference between the pair of algorithms at the level of 5%. The upper triangle of Table 10 shows the mean rank differences between algorithms.

The total number of tests is 91, and this large number makes global analysis of the results very difficult. Thus, in order to facilitate the interpretation of the results, we performed a ranking according to their mean rank in the Friedman test. Table 11 shows the results. We have also reported the p-value between each consecutive pair in the ranking. The tests that have proven significant are highlighted in bold, and a separator line is added. The first thing to be observed is that there are three groups of algorithms. There are differences between algorithms from different groups, and within each group there may or may not exist a significant difference. This test measures the number of times an algorithm obtains better results than another, but it does

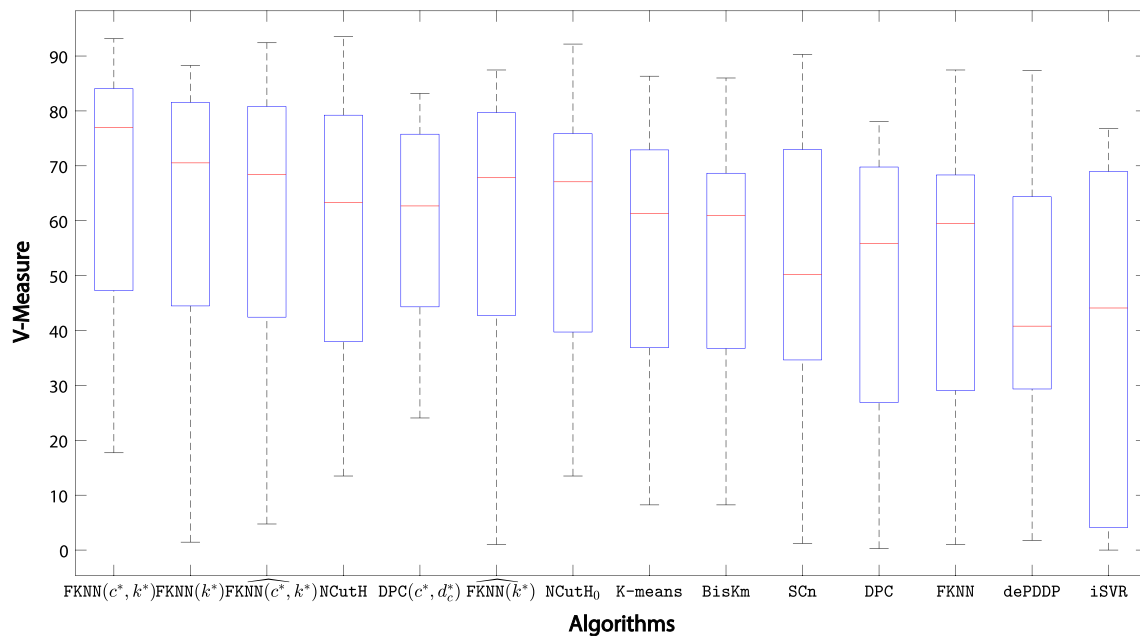


Fig. 2 Boxplot comparing the algorithms with respect to the V -measure scores

not take into account the size of the improvement. To complete the analysis, we performed a boxplot to compare the V -Measure scores obtained for the algorithms. The boxplot is shown in Fig. 2.

The notable points of this experiment are:

- $\widehat{\text{FKNN-DPC}}(c^*, k^*)$ algorithm significantly outperforms the other clustering methods with respect to the V -Measure. As already noted, this method is not applicable to real problems as it requires knowledge of the true clustering ℓ . This result may suggest that line of research in algorithms of the type $\widehat{\text{FKNN-DPC}}(c^*, k^*)$ could be promising. The results obtained with $\widehat{\text{FKNN-DPC}}(c^*, k^*)$ improve those obtained with $\text{DPC}(c^*, k^*)$ (although this improvement is not statistically significant).
- The second highlight is that it shows the importance of the automatic parameter tuning. The versions $\widehat{\text{DPC}}$ and $\widehat{\text{FKNN-DPC}}$ significantly improve the performance of raw methods DPC and FKNN-DPC .

6 Conclusions

This study analyzes peak-clustering methods, focusing on two aspects: (i) developing a methodology for automatic parameter tuning and (ii) analysis of optimal strategies. The methodology proposed for (i) is optimization with respect to Gaussian entropy H_G and for (ii) with respect to the V -Measure index.

The methodology is general enough to analyze arbitrary algorithms \mathcal{A} based on density peaks. Our numerical exper-

iments focus on the original DPC and the promising method FKNN-DPC . We have determined, by using a set of 16 real datasets and for the FKNN-DPC , that the internal validation measure H_G that we propose has a high Spearman correlation, $\rho = 0.96$, with the external validation V -Measure. This suggests that H_G gives results that are very close to the optimal rules for the V -Measure.

It is also shown that automatic parameter tuning for DPC and for the method FKNN-DPC significantly improves its results with respect to the raw parameterization. Automatic parameter tuning is important because many promising algorithms, such as Xie et al. (2016), Lu and Zhu (2017), and Liu et al. (2018), require the parameter k to be estimated.

The study omits the computational cost analysis, a subject that should be studied in the future. The framework described allows for multiple combinations of other algorithms \mathcal{A} in conjunction with other internal or external validation indices. One case we think would of interest is to analyze the cross-entropy H_G^\times with a variable number of clusters.

Acknowledgements This work has been supported by *Ministerio de Economía y Competitividad - FEDER EU* with Grant Number TRA2016-76914-C3-2-P and the predoctoral FPU fellowship from the *Ministerio de Educación, Cultura y Deportes* with number 16/00792.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

- Bai L, Cheng X, Liang J, Shen H, Guo Y (2017) Fast density clustering strategies based on the k -means algorithm. *Pattern Recognit* 71:375–386
- Bie R, Mehmood R, Ruan S, Sun Y, Dawood H (2016) Adaptive fuzzy clustering by fast search and find of density peaks. *Pers Ubiquit Comput* 20(5):785–793
- Bu F, Chen Z, Li P, Tang T, Zhang Y (2016) A high-order CFS algorithm for clustering big data. *Mob Inf Syst* 2016(4356127):1–8
- Chen G, Zhang X, Wang Z, Li F (2015) Robust support vector data description for outlier detection with noise or uncertain data. *Knowl-Based Syst* 90:129–137
- Chen J-Y, He H-H (2015) Research on density-based clustering algorithm for mixed data with determine cluster centers automatically. *Acta Autom Sin* 41(10):1798–1813
- Chen J-Y, He H-H (2016) A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data. *Inf Sci* 345:271–293
- Chen M, Li L, Wang B, Cheng J, Pan L, Chen X (2016) Effectively clustering by finding density backbone based-on kNN. *Pattern Recognit* 60:486–498
- Criminisi A, Shotton J, Konukoglu E (2011) Decision forests for classification, regression, density estimation, manifold. Microsoft Research technical report
- Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. <http://archive.ics.uci.edu/ml>
- Ding J, Chen Z, He X, Zhan Y (2016) Clustering by finding density peaks based on Chebyshev's inequality. In: Chinese control conference, CCC, pp 7169–7172
- Ding J, He X, Yuan J, Jiang B (2018) Automatic clustering based on density peak detection using generalized extreme value distribution. *Soft Comput* 22(9):2777–2796
- Du M, Ding S, Jia H (2016) Study on density peaks clustering based on k -nearest neighbors and principal component analysis. *Knowl-Based Syst* 99:135–145
- Du M, Ding S, Xue Y (2017) A novel density peaks clustering algorithm for mixed data. *Pattern Recognit Lett* 97:46–53
- Gao J, Zhao L, Chen Z, Li P, Xu H, Hu Y (2016) ICFS: an improved fast search and find of density peaks clustering algorithm. In: Proceedings—2016 IEEE 14th international conference on dependable, autonomic and secure computing, DASC 2016, 2016 IEEE 14th international conference on pervasive intelligence and computing, PICom 2016, 2016 IEEE 2nd international conference on big data intelligence and computing, DataCom 2016 and 2016 IEEE Cyber Science and Technology Congress, CyberSciTech 2016, DASC-PICom-DataCom-CyberSciTech 2016, pp 537–543
- Gong S, Zhang Y (2016) EDDPC: an efficient distributed density peaks clustering algorithm. *Comput Res Dev* 53(6):1400–1409
- Guo P, Xing W, Yubing W, Yue C, Ying Z (2017) Research on automatic determining clustering centers algorithm based on linear regression analysis. In: 2nd International conference on image, vision and computing, pp 1016–1023
- Hofmeyr DP (2017) Clustering by minimum cut hyperplanes. *IEEE Trans Pattern Anal Mach Intell* 39(8):1547–1560
- Hua J-L, Yu J, Yang M-S (2016) Correlative density-based clustering. *J Comput Theor Nanosci* 13(10):6935–6943
- Jiang J, Hao D, Chen Y, Parmar M, Li K (2018) GDPC: gravitation-based density peaks clustering algorithm. *Physica A* 502:345–355
- Jinyin C, Xiang L, Haibing Z, Xintong B (2017) A novel cluster center fast determination clustering algorithm. *Appl Soft Comput J* 57:539–555
- Kun D, Ze W, Rui Z, Chao Y (2016) Clustering by exponential density analysis and find of cluster centers based on genetic algorithm. In: Proceedings of SPIE—the international society for optical engineering (ICDIP 2016), vol 10033
- Lee K (2005) Yale face database B. <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/>
- Li M, Huang J, Wang J (2016) Paralleled fast search and find of density peaks clustering algorithm on gpus with cuda. *Int J Netw Distrib Comput* 4(3):173–181
- Li Z, Tang Y (2018) Comparative density peaks clustering. *Expert Syst Appl* 95:236–247
- Liang Z, Chen P (2016) Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering. *Pattern Recognit Lett* 73:52–59
- Liu R, Wang H, Yu X (2018) Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Inf Sci* 450:200–226
- Liu S, Zhou B, Huang D, Shen L (2017) Clustering mixed data by fast search and find of density peaks. *Math Probl Eng* 2017(5060842):1–7
- Liu Y, Li Z, Xiong H, Gao X, Wu J (2010) Understanding of internal clustering validation measures. In: Proceedings of the 2010 IEEE international conference on data mining, ICDM '10, pp 911–916. IEEE Computer Society, Washington
- López-García ML, García-Ródenas R, Gómez AG (2015) K-means algorithms for functional data. *Neurocomputing* 151:231–245
- Lu J, Zhu Q (2017) An effective algorithm based on density clustering framework. *IEEE Access* 5:4991–5000
- Mehmood R, Bie R, Jiao L, Dawood H, Sun Y (2016a) Adaptive cutoff distance: clustering by fast search and find of density peaks. *J Intell Fuzzy Sys* 31(5):2619–2628
- Mehmood R, Zhang G, Bie R, Dawood H, Ahmad H (2016b) Clustering by fast search and find of density peaks via heat diffusion. *Neurocomputing* 208:210–217
- Rodríguez A, Laio A (2014) Clustering by fast search and find of density peaks. *Science* 344(6191):1492–1496
- Rosenberg A, Hirschberg J (2007) V-measure: a conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, vol 7, pp 410–420
- Tabor J, Spurek P (2014) Cross-entropy clustering. *Pattern Recognit* 47(9):3046–3059
- Tao L, Li W, Jin Y (2017) An optimal density peak algorithm based on data field and information entropy. In: ACM international conference proceeding series, vol Part F128770
- Wang G, Song Q (2016) Automatic clustering via outward statistical testing on density metrics. *IEEE Trans Knowl Data Eng* 28(8):1971–1985
- Wang J, Zhu C, Zhou Y, Zhu X, Wang Y, Zhang W (2017) From partition-based clustering to density-based clustering: fast find clusters with diverse shapes and densities in spatial databases. *IEEE Access* 6:1718–1729
- Wang M, Zuo W, Wang Y (2016) An improved density peaks-based clustering method for social circle discovery in social networks. *Neurocomputing* 179:219–227
- Wang X-F, Xu Y (2017) Fast clustering using adaptive density peak detection. *Stat Methods Med Res* 26(6):2800–2811
- Wiwie C, Baumbach J, Röttger R (2015) Comparing the performance of biomedical clustering methods. *Nat Methods* 12(11):1033–1038
- Xie J, Gao H, Xie W, Liu X, Grant P (2016) Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k -nearest neighbors. *Inf Sci* 354:19–40

- Xu J, Wang G, Deng W (2016) DenPEHC: density peak based efficient hierarchical clustering. *Inf Sci* 373:200–218
- Xu X, Ding S, Xu H, Liao H, Xue Y (2019) A feasible density peaks clustering algorithm with a merging strategy. *Soft Comput* 23(13):5171–5183
- Yang X-H, Zhu Q-P, Huang Y-J, Xiao J, Wang L, Tong F-C (2017) Parameter-free laplacian centrality peaks clustering. *Pattern Recognit Lett* 100:167–173
- Yaohui L, Zhengming M, Fang Y (2017) Adaptive density peak clustering based on k -nearest neighbors with aggregating strategy. *Knowl-Based Syst* 133:208–220
- Zang W, Ren L, Zhang W, Liu X (2017) Automatic density peaks clustering using DNA genetic algorithm optimized data field and Gaussian process. *Int J Pattern Recognit Artif Intell* 31(8)
- Zhao Y, Karypis G (2001) Criterion functions for document clustering: experiments and analysis. *Tech. Rep.*, pp 01–04

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.