A Theoretical Analysis of Density Peaks Clustering and the Component-wise Peak-Finding Algorithm

Joshua Tobin, Mimi Zhang

Abstract—Density peaks clustering detects modes as points with high density and large distance to points of higher density. Each non-mode point is assigned to the same cluster as its nearest neighbor of higher density. Density peaks clustering has proved capable in applications, yet little work has been done to understand its theoretical properties or the characteristics of the clusterings it produces. Here, we prove that it consistently estimates the modes of the underlying density and correctly clusters the data with high probability. However, noise in the density estimates can lead to erroneous modes and incoherent cluster assignments. A novel clustering algorithm, Component-wise Peak-Finding (CPF), is proposed to remedy these issues. The improvements are twofold: (1) the assignment methodology is improved by applying the density peaks methodology within level sets of the estimated density; (2) the algorithm is not affected by spurious maxima of the density and hence is competent at automatically deciding the correct number of clusters. We present novel theoretical results, proving the consistency of CPF, as well as extensive experimental results demonstrating its exceptional performance. Finally, a semi-supervised version of CPF is presented, integrating clustering constraints to achieve excellent performance for an important problem in computer vision.

Index Terms—Density-Based Clustering, Nearest-Neighbor Graph, Density Peaks, Semi-Supervised Clustering, Multi-Image Matching

1 Introduction

Density-based clustering methods relate the notion of clusters to high-density contiguous regions of the underlying density function. Hartigan [1] proposed the concept of density-based clusters as "regions ... where the densities are high surrounded by regions where the densities are low". The concept is attractive for several reasons: (1) the clusters are free to assume any shape, in contrast to model-based clustering methods; (2) the clustering method is associated with density but without requiring strong assumptions on the density function; (3) the number of clusters is linked to density peaks and can be determined as part of the estimation procedure. Density-based clustering methods can be broadly classified into two categories: level set methods and mode-seeking methods.

Level set methods detect clusters as connected components of the density level sets $\{\boldsymbol{x} : f(\boldsymbol{x}) \geq \lambda\}$, where f is the density function and λ is a cutting threshold. The density f is unknown, and hence the level sets are required to be estimated from the data. Nearest-neighbor graphs have been widely used for this purpose [2], [3]. Taking the instances to be the vertices of a graph, k-NN graphs add edges between a vertex and all its k nearest neighbors. Mutual k-NN graphs add an edge between two vertices only if they are k nearest neighbors of each other. It has been shown that any density level set of a given dataset can be approximated by the connected components of the mutual k-NN graph [2], [3], and further work attempted to develop an understanding of the optimal choice of k [2], [4].

• *J. Tobin and M. Zhang are with the School of Computer Science & Statistics, Trinity College Dublin, Dublin, Ireland.*
E-mail: {tobinjo, mimi.zhang}@tcd.ie

Mode-seeking methods aim to directly locate the modes in the density and then associate each instance in the observed data with a relevant mode. Such approaches begin with a density estimate $\hat{f}$ and then move each point $\boldsymbol{x}_i$ towards a mode of $\hat{f}$ by ascending the density. Mean shift, introduced in [5] and further developed in [6] and [7], is a popular mode-seeking method that associates an instance to a mode along the path of steepest ascent of the density estimate. To circumvent the costly run time of mean shift, the authors in [8] proposed a fast sample-based method, termed quick shift. Quick shift simply associates each instance to its nearest neighbor of higher empirical density. To return a partition of the data, a segmentation parameter τ is required such that an instance will not be associated to its nearest neighbor of higher density if the distance between them is greater than τ. Quick shift is shown in [9] to consistently estimate the non-trivial modes of the underlying density and to correctly assign instances to their associated mode. However, appropriate tuning of τ requires a knowledge of the distances between modes. Furthermore, determining modes by only the distances between instances and their nearest neighbor of higher density can cause outlying points to be erroneously selected as modes.

The density peaks clustering (DPC) method introduced in [10] offers a potential remedy to these issues, providing an intuitive method for sample-based mode detection. The true modes of the density are estimated using a decision graph, a scatter plot of the local density against the distance to the nearest neighbor of higher density. The modes are estimated as the extreme instances on the decision graph. DPC assigns the remaining instances to the detected modes using the same methodology as quick shift. The partition

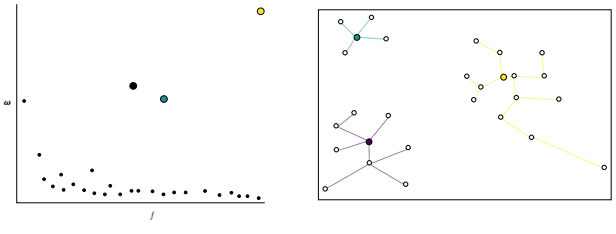
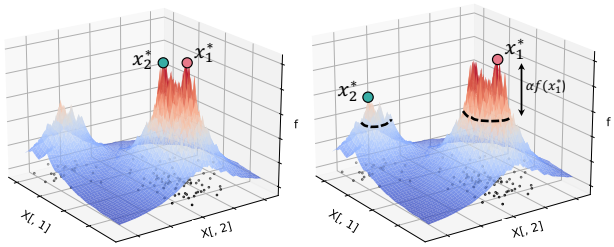


Figure 1: Left: The decision graph of the DPC method. The three extreme points are detected as the modes. Right: The assignment of the instances to the modes.



(a) Point Modes (b) Modal-Sets

Figure 2: Left: Noise in the density estimate leads to errors when seeking point modes. Right: Modal-set methods are robust to noise and recover the true cluster structure.

of the data is extracted by grouping together instances that are assigned to the same mode. The decision graph and the resulting clustering for a toy dataset are shown in Figure 1.

While many papers have demonstrated the ability of the DPC method to provide high-quality clusterings in applications [11], [12], there is, to the best of our knowledge, only one publication on the theoretical analysis of the DPC method. In [13], the authors derive a theoretically grounded rule for selecting modes from the decision graph, using a robust linear regression of log of the density estimates $\log \hat{f}(\boldsymbol{x})$ against the log of the distances to neighbors of higher density.

In this work, we seek to deepen our understanding of the DPC method and propose a new density-based clustering technique that improves DPC both theoretically and computationally. By adapting results from related works, we provide theoretical guarantees that DPC consistently estimates the modes of the underlying density and can correctly cluster the data with high probability. We also demonstrate the deficiencies of the DPC methodology in the presence of noisy density estimates. Motivated by the deficiencies, we introduce a novel clustering algorithm: Component-wise Peak-Finding (CPF). CPF improves DPC in two ways: firstly, CPF partitions the data into regions mutually separated by areas of low density before clustering, thus ensuring the correct assignment of instances to their respective clusters; secondly, the peak-finding criterion is directed to seek modal-sets rather than point modes in the data, reducing the sensitivity of the clustering result to fluctuations in the density estimate. We provide theoretical guarantees for our new algorithm, extending the theoretical properties available for the DPC method. In particular, we prove that CPF recovers unique and consistent estimates of the high-density regions in the data, and correctly determines the true number of clusters. Furthermore, the complexity of our algorithm is of the order $O(nk \log(n))$, near linear in k and n. Finally, to demonstrate the adaptability of CPF, we present a modified version of the method, CPF-Match, designed for multi-image matching, an application in computer vision. We show that CPF-Match achieves state-of-the-art performance for this task.

The remainder of the paper is organized as follows. In Section 3, we formalize the DPC method, provide a theoretical analysis of its performance, and demonstrate its deficiencies via illustrative examples. In Section 4, the CPF algorithm is explained in detail, and its consistency properties are provided in Section 5. In Section 6, we assess the clustering quality of CPF on a range of simulated and real-world datasets and show that CPF outperforms DPC and other peer clustering methods. Section 7 introduces CPF-Match, an adapted method for multi-image matching. Section 8 concludes the paper.

2 Related Work

Adaptations of the DPC method have proliferated in recent years. One strand of works focuses on improving the density estimator (see [14], [15]), and another strand of works focuses on automating the selection of modes from the decision graph (see [16], [17]). The authors of DPC have introduced a recent approach [18], which applies a density estimator based on the intrinsic dimension of the data, and a pruning mechanism for false modes.

A robust way of modelling high-density regions in the data space is proposed in [19]. Modal-sets generalize the concept of a point mode to a local support of the density peak. An illustrative example is given in Figure 2. The related clustering procedure, termed MCores, estimates the modal-sets using connected components of k-NN graphs at different levels of the empirical density. The authors provide consistency guarantees on the recovery of true modal-sets in the data. A subsequent work presents QuickShift++ [20] improving on the MCores procedure by adopting the same allocation procedure as quick shift and DPC. Recently, in [21], DPC was adapted to detect modal-sets. The method, termed DCF, was shown to detect modal-sets more efficiently than QuickShift++.

While [19], [20], [21] use classical non-parametric density estimators, recent literature has proposed density estimators using neural networks. A prominent approach uses energy-based models, defining an unnormalized density that is the exponential of the negative energy function, parameterized by a neural network. The estimation procedure involves either computing maximum likelihood estimates ([22], [23]), variational approximation to an unnormalized target ([24], [25]), or methods combining both approaches ([26]). The density estimates produced by these methods are computationally expensive to compute, and consistency guarantees are not currently available making theoretical analysis challenging. Nevertheless, they can naturally be integrated into the clustering method discussed in this work.

3 Density Peaks Clustering

3.1 The Method

The peak-finding method in [10] requires two inputs: (1) a density estimate at each data point, and (2) the distance from each point to its nearest neighbor of higher density. We consider a dataset $\boldsymbol{X}$ consisting of n data points in $\mathbb{R}^p$ drawn from an unknown density f with compact support $\mathcal{X}$. We use a k-NN density estimator as it is computationally fast and guarantees on its quality are well understood. For a data point $\boldsymbol{x} \in \boldsymbol{X}$, let $r_k(\boldsymbol{x})$ be the distance between x and its k-th nearest neighbor. The density estimate used is a simple functional of the distance $r_k(\boldsymbol{x})$.

Definition 1. *For every $\boldsymbol{x} \in \mathbb{R}^p$, let $r_k(\boldsymbol{x})$ denote the distance from $\boldsymbol{x}$ to its k-th nearest neighbor in $\boldsymbol{X}$. The density estimate is given as*

$$\hat{f}_k(\boldsymbol{x}) := \frac{k}{n \cdot v_p \cdot r_k(\boldsymbol{x})^p},$$

where v_p is the volume of the unit sphere in $\mathbb{R}^p$.

Note that this estimator is different from the estimate of the empirical density used in [10], which counts the number of data points within a threshold distance of a given instance. It is replaced as no guarantees of its consistency are possible. As well as a density estimate, the peak-finding criterion requires the distance from each point to it's nearest neighbor of higher density:

Definition 2. *For the point $\boldsymbol{x} = \arg\max_{\boldsymbol{x} \in \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$, we define the quantity*

$$\omega(\boldsymbol{x}) = \max_{\boldsymbol{x}' \in \boldsymbol{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|.$$

For the remaining points, let $b(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}' \in \boldsymbol{X}} \left\{ \|\boldsymbol{x} - \boldsymbol{x}'\| : \hat{f}_k(\boldsymbol{x}) < \hat{f}_k(\boldsymbol{x}') \right\}$, i.e. the nearest neighbor of $\boldsymbol{x}$ with higher density. Define the distance to the nearest neighbor of higher local density as

$$\omega(\boldsymbol{x}) = \|\boldsymbol{x} - b(\boldsymbol{x})\|.$$

Also of interest is the product of the estimated density $\hat{f}_k(\boldsymbol{x})$ and the distance quantity $\omega(\boldsymbol{x})$. This is termed the peak-finding criterion:

Definition 3. *Taking $\hat{f}_k(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$ as defined above, we define the peak-finding criterion $\gamma(\boldsymbol{x})$ as*

$$\gamma(\boldsymbol{x}) = \hat{f}_k(\boldsymbol{x}) \cdot \omega(\boldsymbol{x}).$$

Following [10], the decision graph is the scatter plot of $\{(\hat{f}_k(\boldsymbol{x}), \omega(\boldsymbol{x})) : \boldsymbol{x} \in \boldsymbol{X}\}$. To generate a set of mode estimates $\widehat{\mathcal{M}} = \{\boldsymbol{x}_j\}_{j=1}^m$, threshold values for the density $\hat{f}_k(\boldsymbol{x})$ and the distance $\omega(\boldsymbol{x})$ need to be set: the modes are the data points with the two metric values both above the thresholds, i.e. $\widehat{\mathcal{M}} = \{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}_k(\boldsymbol{x}) \geq l, \omega(\boldsymbol{x}) \geq \tau\}$.

The algorithm used for density peaks clustering in this formulation is described in Algorithm 1. The algorithm takes as input the dataset $\boldsymbol{X}$ and uses the parameter k to return the final set of clusters $\widehat{\mathcal{C}}$. Initially, the set of estimated modes $\widehat{\mathcal{M}} = \varnothing$ and the cluster assignment graph $\vec{G}(\boldsymbol{X}, \vec{E})$ is initialized with vertices as the points of $\boldsymbol{X}$ and no edges. DPC produces the decision graph (Lines 2-3).

DPC requests the user to select estimated modes using this plot as reference. The estimated modes $\{\boldsymbol{x}_j\}_{j=1}^m$ are then added to $\widehat{\mathcal{M}}$ (Line 4). After the set of estimated modes has been returned, edges are added to the graph $\vec{G}(\boldsymbol{X}, \vec{E})$ from each non-modal point $\boldsymbol{x}$ to $b(\boldsymbol{x})$ (Lines 5-7). The estimated mode together with all the vertices that have paths terminating at it form a cluster that is added to $\widehat{\mathcal{C}}$ (Lines 8-11). Proceeding in this way, each sample point will be assigned to a unique cluster.

Algorithm 1 Density Peaks Clustering

Input: Neighborhood parameter k.
Output: A set of clusters $\widehat{\mathcal{C}}$
1: *Initialisation*: $\widehat{\mathcal{M}} = \varnothing$, $\vec{G}(\boldsymbol{X}, \vec{E})$, a directed graph with $\boldsymbol{X}$ as vertices and no edges, $\vec{E} = \varnothing$.
2: Create the decision graph $\{(\hat{f}_k(\boldsymbol{x}), \omega(\boldsymbol{x})) : \boldsymbol{x} \in \boldsymbol{X}\}$.
3: Select the estimated modes using the thresholds l and τ, i.e., $\{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}_k(\boldsymbol{x}) \geq l, \omega(\boldsymbol{x}) \geq \tau\}$
4: Add the estimated modes $\{\boldsymbol{x}_j\}_{j=1}^m$ to $\widehat{\mathcal{M}}$.
5: **for** each $\boldsymbol{x}$ in $\boldsymbol{X} \backslash \widehat{\mathcal{M}}$ **do**
6: Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
7: **end for**
8: **for** each estimated mode $\boldsymbol{x} \in \widehat{\mathcal{M}}$ **do**
9: Let $\boldsymbol{C}$ be the collection of the points connected by any directed path in $\vec{G}(\boldsymbol{X}, \vec{E})$ that terminates at $\boldsymbol{x}$.
10: Add $\boldsymbol{C} \cup \boldsymbol{x}$ to $\widehat{\mathcal{C}}$.
11: **end for**
12: **return** $\widehat{\mathcal{C}}$

3.2 Theoretical Analysis

The quality of the clusterings provided by DPC has been thoroughly demonstrated in practice, as discussed in Section 1. Yet, no previous work has provided guarantees on the ability of DPC to recover modes consistently. Through drawing an analogy to the quick shift method, we show that DPC can recover the modes and the associated cluster assignments with strong consistency guarantees.

Quick shift, as described in Section 1, is a fast non-parametric density-based method that produces clusterings with kernel density estimates. A directed graph is built with the observed instances as vertices, and edges added from each instance to its nearest neighbor of higher estimated density. The final clusters are extracted as the connected components of the graph once edges with length longer than a segmentation parameter τ are removed. The formulation of DPC introduced above is similar to quick shift in all but two ways: (1) a k-NN estimate of the density is used in place of a kernel density estimate, and (2) a second threshold value l is defined, used to flag low-density instances as outliers. As such, in this section we present the main results adapted from the consistency analysis of the k-NN density estimator [27] and the consistency analysis of quick shift [9]. The primary contributions involve drawing the analogy to the quick shift approach, and the extension of the analysis to include the density threshold l used in the mode selection step of DPC. An extended analysis, including proofs of the theorems, is given in the supplementary material.

We assume that f is α-Hölder continuous and lower bounded on $\mathcal{X}$. Furthermore, it is assumed that the level sets of f are continuous with respect to the density level, and the

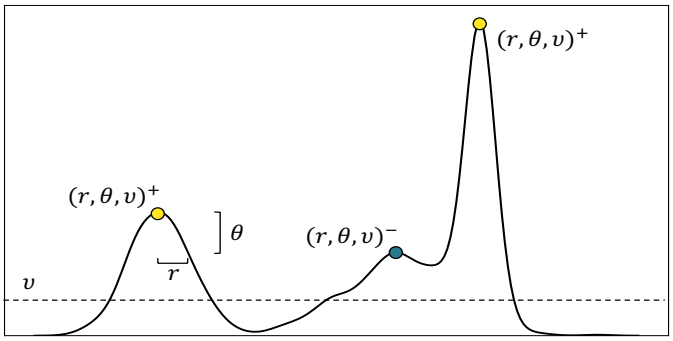


Figure 3: An illustration of the $(r,\theta,\nu)^+$-modes and $(r,\theta,\nu)^-$-modes of Definition 4.

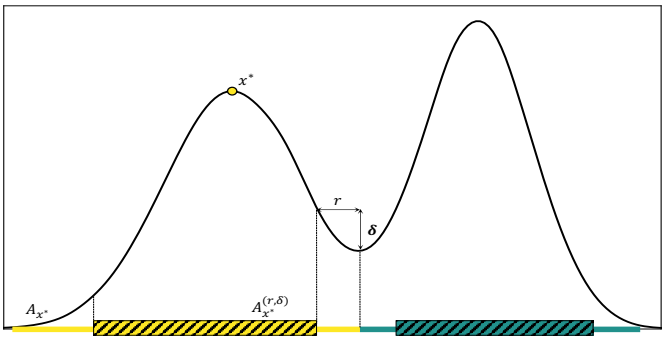


Figure 4: An illustrative example of the (r,δ)-interior of an attraction region $\mathcal{A}_{\boldsymbol{x}^*}$, denoted $\mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)}$, associated with a mode $\boldsymbol{x}^*$.

modes of f have negative definite Hessian. Following [9], we now define a stronger notion of mode that allows for a clearer analysis of DPC.

Definition 4. *A mode $\boldsymbol{x}^* \in \mathcal{M}$ is an $(r,\theta,\nu)^+$-mode, if $f(\boldsymbol{x}^*) > f(\boldsymbol{x}') + \theta$ for all $\boldsymbol{x}' \in B(\boldsymbol{x}^*, r)\backslash B(\boldsymbol{x}^*, r_{\mathcal{M}})$ and $f(\boldsymbol{x}^*) > \nu + \theta$. A mode $\boldsymbol{x}^* \in \mathcal{M}$ is an $(r,\theta,\nu)^-$-mode, if $f(\boldsymbol{x}^*) < f(\boldsymbol{x}') - \theta$ for some $\boldsymbol{x}' \in B(\boldsymbol{x}^*, r)$ and $f(\boldsymbol{x}^*) > \nu + \theta$. Let $\mathcal{M}^+_{r,\theta,\nu} \subseteq \mathcal{M}$ denote the set of $(r,\theta,\nu)^+$-modes of f.*

An illustration of the $(r,\theta,\nu)^+$-mode and the $(r,\theta,\nu)^-$-modes is given in Figure 3. Recall that the DPC algorithm requires two cutting-off thresholds, one for cutting the value of the density estimate $\hat{f}_k(\boldsymbol{x})$ and the other for cutting the distance to the nearest neighbor of higher estimated density, $\omega(\boldsymbol{x})$. Taking the thresholds as τ and l for the density and distance values respectively, our first theorem shows that $\widehat{\mathcal{M}}$ contains unique and consistent estimates of the $(\tau+\epsilon,\theta,l)^+$-modes of f, for $\theta, \epsilon > 0$.

Theorem 1 (**Mode Estimation** - adapted from Theorem 2 of [9]). *For every $\boldsymbol{x}^* \in \mathcal{M}^+_{\tau+\epsilon,\theta,l}\backslash\mathcal{M}^-_{\tau-\epsilon,\theta,l'}$ with probability at least $1-\zeta$, there exists $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$ satisfying*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq C \cdot f(\boldsymbol{x}^*) \cdot \frac{1}{k^{1/4}},$$

where C is a constant depending on p, n, ζ, and f,

Theorem 1 proves that DPC recovers the modes of an α-Hölder continuous density f consistently. For n large enough, with high probability, $\widehat{\mathcal{M}}$ contains unique estimates for all the true modes of f. As such, there is an injection between the set of true modes and the set of estimated modes.

The procedure used to assign points to their respective modes is the same as that used in quick shift. As such, theoretical guarantees developed for a variant of quick shift in [20] can be applied directly to DPC. We provide the relevant results below.

First, we define the attraction region of a mode. The attraction region of a particular mode covers all points that flow towards the mode along the direction of the gradient of the underlying density.

Definition 5. *Let path $\nu_{\boldsymbol{x}} : \mathbb{R} \to \mathbb{R}^p$ satisfy $\nu_{\boldsymbol{x}}(0) = \boldsymbol{x}$ and $\nu'_{\boldsymbol{x}}(t) = \nabla f(\nu_{\boldsymbol{x}}(t))$. For a mode $\boldsymbol{x}^* \in \mathcal{M}$, its attraction region $\mathcal{A}_{\boldsymbol{x}^*}$ is the set of points $\boldsymbol{x} \in \mathcal{X}$ that satisfy $\lim_{t\to\infty} \nu_{\boldsymbol{x}}(t) = \boldsymbol{x}^*$.*

It is shown that DPC can cluster sample points in the (r,δ)-interior of an attraction region. The parameters $r > 0$ and $\delta > 0$ hold simultaneously across all modes of the density and can be chosen arbitrarily small.

Definition 6. *The (r,δ)-interior of an attraction region $\mathcal{A}_{\boldsymbol{x}^*}$, denoted $\mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)}$, is the set of points $\boldsymbol{x}_1 \in \mathcal{A}_{\boldsymbol{x}^*}$ such that a path $\mathcal{P}$ from $\boldsymbol{x}_1$ to any point $\boldsymbol{x}_2 \in \partial\mathcal{A}_{\boldsymbol{x}^*}$ satisfies*

$$\sup_{\boldsymbol{x}\in\mathcal{P}} \inf_{\boldsymbol{x}'\in B(\boldsymbol{x},r)} f(\boldsymbol{x}') \geq \sup_{\boldsymbol{x}'\in B(\boldsymbol{x}_2,r)} f(\boldsymbol{x}') + \delta.$$

Points in the interior of an attraction region must satisfy the property that any path leaving the attraction region must significantly decrease in density at some point. An illustrative example is given in Figure 4.

The main result (Theorem 2) states that, as long as the modes are well-estimated, the assignment method of DPC will correctly cluster the (r,δ)-interiors of the attraction regions with high probability. Suppose that $\boldsymbol{x}^* \in \mathcal{M}$ is estimated by $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$ such that $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq r$. Then, with high probability, for $\boldsymbol{x} \in \mathcal{A}_{\boldsymbol{x}^*} \cap \boldsymbol{X}$, density peaks clustering clusters $\boldsymbol{x}$ to the cluster belonging to $\boldsymbol{x}^*$.

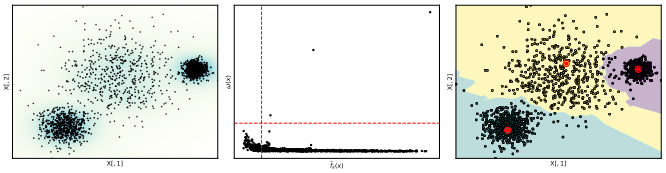
Theorem 2 (**Cluster Assignment** - adapted from Theorem 2 of [20]). *Suppose that $\boldsymbol{x}^* \in \mathcal{M}$ is estimated by $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$ such that $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq r$. Then, for n sufficiently large, depending on f, δ, ζ and r, with high probability, for any $\boldsymbol{x} \in \mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)} \cap \boldsymbol{X}$, DPC clusters $\boldsymbol{x}$ to the cluster belonging to $\boldsymbol{x}^*$.*

3.3 Limitations

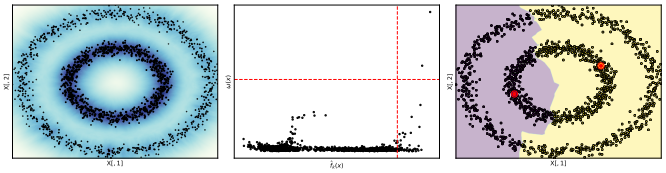
The theoretical analysis of Section 3.2 is based on the assumption of a sample size large enough that the error of the density estimator can be bounded. In this section, we provide an analysis of the density peaks clustering algorithm through three illustrative datasets, with $n = 1500$, from the scikit-learn clustering demonstration [1]. Taken together, the three datasets provide an understanding of the density peaks clustering algorithm and the type of clusters it returns; see Figure 5.

Firstly, the density estimation appears to recover population density for each dataset. The second feature of the density peaks clustering algorithm analyzed is the decision

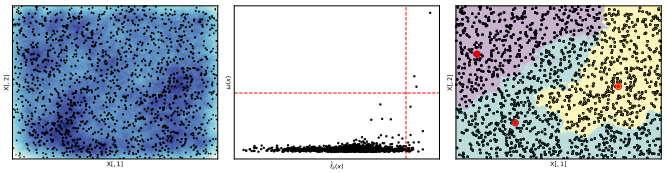
1. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html



(a) Unequal Variance Gaussian Dataset



(b) Noisy Circles Dataset



(c) No Cluster Structure Dataset

Figure 5: Density peaks clustering of illustrative datasets. The k-NN density estimator is used here with $k = 40$. Left: Density estimates for the dataset. Darker regions indicate higher density. Center: The decision plot, with thresholds set to estimate approximately the correct number of clusters. Right: The final clustering assignment. The color of the shaded regions indicate the attraction region for each cluster.

graph, provided to enable the estimation of the modes from the dataset. The method of selecting mode estimates from the decision graph is seen to perform well when the density of the cluster is concentrated near the mode and decays as the distance from the mode increases, such as for the Unequal Variance Gaussian dataset. Each of the remaining datasets contains areas of relatively uniform density. This poses challenges for the density peaks clustering method as noise in the density estimate leads to erroneous modes being selected. Finally, the assignment method of density peaks clustering is assessed. The assignment strategy is shown to perform well for the Unequal Variance Gaussian dataset. The allocation of instances to clusters for the Noisy Circles runs contrary to geometric intuition about the clusters. In this case, the allocation assigns instances to clusters across areas of very low density in the dataset.

In sum, the density peaks clustering framework performs well for datasets containing clusters with clear point modes around which the density decays, such as Gaussian components. The framework struggles when the high density regions of the data are relatively uniform. In this case, both the mode selection method and the assignment strategy are shown to be susceptible to errors caused by noise in the density estimate.

4 The Proposed CPF Algorithm

In this section, the improvements to the density peaks clustering method that constitute the CPF algorithm are introduced, together with a detailed analysis of the clustering algorithm.

4.1 Peak-Finding on Level Sets

In this section, we explain the component set notation and the peak-finding criterion. We denote the mutual k-NN graph $G(\boldsymbol{X}, E)$. The structure of the mutual k-NN graph can help detect outlier data points in $\boldsymbol{X}$. In particular, for $\boldsymbol{x} \in \boldsymbol{X}$, $\boldsymbol{x}$ is an outlier if its vertex in the graph $G(\boldsymbol{X}, E)$ has very few or no edges. We denote the set of outliers by $\boldsymbol{O}$.

Definition 7. *For every $\boldsymbol{x} \in \mathbb{R}^p$, let $r_k(\boldsymbol{x})$ denote the distance from $\boldsymbol{x}$ to its k-th nearest neighbor in $\boldsymbol{X}$ as before. The mutual k-NN graph $G(\boldsymbol{X}, E)$ consists of the vertex set $\boldsymbol{X}$ and the edge set E. There is an edge between two vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, denoted by $\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in E$, if and only if $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \min(r_k(\boldsymbol{x}_i), r_k(\boldsymbol{x}_j))$. That is, an edge exists between the vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, only if they are a k-nearest neighbor of each other.*

Next, we formalize the notation of connected components of the mutual k-NN graph $G(\boldsymbol{X}, E)$, beginning with the definition of connectedness.

Definition 8. *A path of length m from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$, denoted by $\{\{\boldsymbol{x}_i, \boldsymbol{v}_1\}, \{\boldsymbol{v}_1, \boldsymbol{v}_2\}, \ldots, \{\boldsymbol{v}_{m-1}, \boldsymbol{x}_j\}\}$, is a sequence of distinct edges in E, starting at vertex $v_0 = \boldsymbol{x}_i$ and ending at vertex $\boldsymbol{v}_m = \boldsymbol{x}_j$, such that $\{\boldsymbol{v}_{r-1}, \boldsymbol{v}_r\} \in E$ for all $r = 1, \ldots, m$. We say that the two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected, if there is a path from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ in the graph $G(\boldsymbol{X}, E)$.*

The definition of connected components and component sets follows.

Definition 9. *A connected component of $G(\boldsymbol{X}, E)$, denoted by $G(\boldsymbol{S}, E(\boldsymbol{S}))$, is a subgraph of $G(\boldsymbol{X}, E)$, where any two vertices in $\boldsymbol{S}$ are connected to each other by paths, and the edge set induced by $\boldsymbol{S}$ is a subset of E: $E(\boldsymbol{S}) = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in E : \boldsymbol{x}_i \in \boldsymbol{S}, \boldsymbol{x}_j \in \boldsymbol{S}\}$. The vertex set $\boldsymbol{S}$ of the component graph $G(\boldsymbol{S}, E(\boldsymbol{S}))$ is a subset of $\boldsymbol{X}$ and here is termed a component set of $\boldsymbol{X}$.*

From the definition of component, we know that the connected components of $G(\boldsymbol{X}, E)$ reveal certain underlying patterns of the data. In particular, the data $\boldsymbol{X}$ can be partitioned into disjoint component sets. Here, we denote the set of component sets $\mathcal{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_{n_S}\}$, where $n_S = |\mathcal{S}|$ is the number of component sets, and $\boldsymbol{S}_1 \cup \cdots \cup \boldsymbol{S}_{n_S} = \boldsymbol{X}$.

Theoretical results regarding the ability of connected components of $G(\boldsymbol{X}, E)$ to estimate the level sets of f are given in [2], [3]. If two points belong to two different component sets, it is highly likely that they are separated by a region of low density.

4.2 Modelling High-Density Regions

We now explain the mode selection mechanism. The definitions for the peak-finding technique used are the same as those given in Section 3.1. The definitions below are given in terms of one $\boldsymbol{S} \in \mathcal{S}$, and are equivalent for each.

Data points in $\boldsymbol{S}$ are placed in descending order of the peak-finding criterion, and the modal-set associated with the instance having maximal value of the peak-finding criterion is automatically accepted. To decide whether or not to select modal-sets associated with the subsequent instances, we here utilize an idea similar to the methods of [21], [28],

[29]. A candidate modal-set $\widehat{\boldsymbol{M}}$ associated with an instance $\boldsymbol{x}^*$ is accepted only when it is well separated from the others.

Definition 10. *Let $0 < \rho < 1$. For an instance $\boldsymbol{x}^* \in \boldsymbol{S}$, define a graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ with*

$$\boldsymbol{V}_{\boldsymbol{x}^*} = \left\{\boldsymbol{x} \in \boldsymbol{S} : r_k(\boldsymbol{x}) < \rho^{-\frac{1}{p}} r_k(\boldsymbol{x}^*)\right\}.$$

The estimated modal-set $\widehat{\boldsymbol{M}}$ is the connected component of the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ containing the vertex $\boldsymbol{x}^*$. $\widehat{\boldsymbol{M}}$ is accepted only if it does not intersect any previously selected modal-set.

Note that the k-th nearest neighbour of $\boldsymbol{x}^*$ in the distance $r_k(\boldsymbol{x}^*)$ is a point from the component set $\boldsymbol{S}$, not from the original dataset $\boldsymbol{X}$. This approach allows the graph to better reflect the scale of the data contained in the component set. The component sets obtained from the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ are assessed, and if the component set containing $\boldsymbol{x}^*$, i.e. $\widehat{\boldsymbol{M}}$, does not intersect previously selected candidate modal-sets, then $\widehat{\boldsymbol{M}}$ is accepted.

Varying the parameter ρ determines the number of clusters for each component set $\boldsymbol{S}$. The threshold relates directly to the estimated density for each of the instances. For example, if $\hat{f}_k(\boldsymbol{x}_1) < \rho \hat{f}_k(\boldsymbol{x}_2)$ then $r_k(\boldsymbol{x}_1) > \rho^{-\frac{1}{p}} r_k(\boldsymbol{x}_2)$. For low values of ρ, fewer vertices will be removed, and it is less likely that a proposed modal-set will be disconnected from existing ones. For larger values of ρ, more vertices and their edges will be removed from the graph, and the probability of the proposed modal-set being disconnected will increase. It is not required to have different ρ values for different component sets, because the threshold $\rho^{-\frac{1}{p}} r_k(\boldsymbol{x}^*)$ adapts naturally to the density level of the component set being assessed. It is seen that modal-sets associated with spurious modes of the density estimate $\hat{f}_k$ will not be accepted by CPF, as the modal-sets are not disconnected from previously accepted modal-sets.

Algorithm 2 The Component-wise Peak Finding Algorithm

Input: Neighborhood parameter k, fluctuation parameter ρ. Initialisation: $\widehat{\mathcal{C}} = \varnothing$.
Output: A set of clusters $\widehat{\mathcal{C}}$.

1: Compute $G(\boldsymbol{X}, E)$, the mutual k-nearest neighbor graph.
2: Extract $\mathcal{S}$, the set of component sets from $G(\boldsymbol{X}, E)$.
3: **for** each $\boldsymbol{S} \in \mathcal{S}$ **do**
4: Sort the $\boldsymbol{x}$'s according to their γ values.
5: Let $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{S}} \gamma(\boldsymbol{x})$.
6: Let $\boldsymbol{V}_{\boldsymbol{x}^*} = \{\boldsymbol{x} \in \boldsymbol{S} : r_k(\boldsymbol{x}) < \frac{r_k(\boldsymbol{x}^*)}{\rho^{1/p}}\}$.
7: Let $\widehat{\boldsymbol{M}} \subseteq \boldsymbol{S}$ be the component set of the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ containing $\boldsymbol{x}^*$.
8: Initialise $\widehat{\mathcal{M}} = \{\widehat{\boldsymbol{M}}\}$, the set of true modal-sets in $\boldsymbol{S}$.
9: **loop**
10: Let $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{S}} \{\gamma(\boldsymbol{x}) : \boldsymbol{x} \notin \widehat{\mathcal{M}}\}$.
11: Let $\boldsymbol{V}_{\boldsymbol{x}^*} = \{\boldsymbol{x} \in \boldsymbol{S} : r_k(\boldsymbol{x}) < \frac{r_k(\boldsymbol{x}^*)}{\rho^{1/p}}\}$.
12: Let $\widehat{\boldsymbol{M}} \subseteq \boldsymbol{S}$ be the component set of the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ containing $\boldsymbol{x}^*$.
13: **if** $\widehat{\boldsymbol{M}} \cap \widehat{\mathcal{M}} = \varnothing$ **then**
14: Add $\boldsymbol{x}^*$ to $\widehat{\mathcal{M}}$.
15: **end if**
16: **end loop**
17: Initialise $\vec{G}(\boldsymbol{S}, \vec{E})$, a directed graph with $\boldsymbol{S}$ as vertices and no edges, $\vec{E} = \varnothing$.
18: **for** each $\boldsymbol{x}$ in $\boldsymbol{S} \backslash \widehat{\mathcal{M}}$ **do**
19: Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
20: **end for**
21: **for** each cluster center $\boldsymbol{x} \in \widehat{\mathcal{M}}$ **do**
22: Let $\boldsymbol{C}$ be the collection of the points connected by any directed path in $\vec{G}(\boldsymbol{S}, \vec{E})$ that terminates at $\boldsymbol{x}$.
23: Add $\boldsymbol{C} \cup \boldsymbol{x}$ to $\widehat{\mathcal{C}}$.
24: **end for**
25: **end for**
26: **return** $\widehat{\mathcal{C}}$

4.3 The CPF Algorithm

The CPF ALGORITHM is explained with reference to Algorithm 2 and the illustrative example in Figure 6.

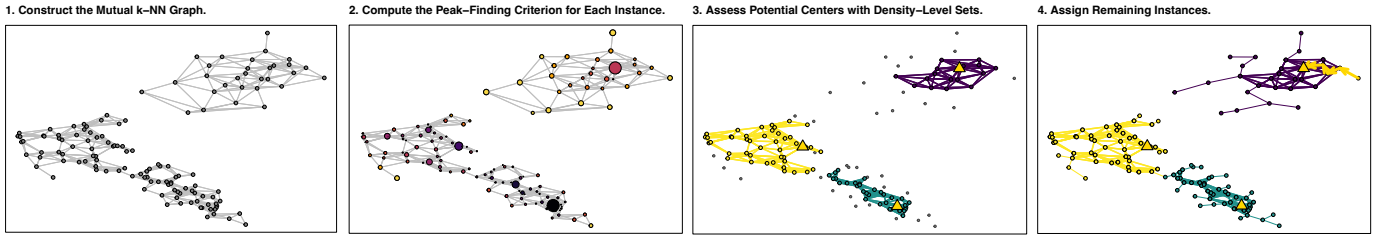
The algorithm takes as input the dataset $\boldsymbol{X}$ and uses parameters k and ρ to return the final set of clusters $\widehat{\mathcal{C}}$. Initially, the set of estimated clusters is $\widehat{\mathcal{C}} = \varnothing$. The undirected mutual k-nearest neighbor graph $G(\boldsymbol{X}, E)$ is constructed. Vertices that have few to no edges are marked as outliers and removed. The remaining data is partitioned into disjoint component sets according to the graph $G(\boldsymbol{X}, E)$ yielding $\mathcal{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_{n_{\mathcal{S}}}\}$ (Lines 1-2). In Figure 6a, two components are extracted, yielding $\mathcal{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2\}$.

For each component set $\boldsymbol{S} \in \mathcal{S}$, CPF computes the peak-finding criterion for each point and selects the instance $\boldsymbol{x}^*$ with maximal value (Lines 4-5). In Figure 6b, the higher estimated density of the instances is represented by darker colors, and the magnitude of the peak-finding criterion for each instance is represented using the size of the points.

The subgraph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ is extracted, and the component set of $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ containing $\boldsymbol{x}^*$ is denoted by $\widehat{\boldsymbol{M}}$. The modal-set $\widehat{\boldsymbol{M}}$ is automatically accepted, and the set of true modal-sets for the component set $\boldsymbol{S}$ is initialised as $\widehat{\mathcal{M}} = \{\widehat{\boldsymbol{M}}\}$ (Lines 6-8). Following the comutation of the points $\boldsymbol{x}^*$ for each components, the purple and green modal-sets are automatically selected in Figure 6c.

Next, the instance with maximal value of the peak-finding criterion yet to be assessed is selected and denoted by $\boldsymbol{x}^*$. The subgraph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ is extracted, and the component set of $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E(\boldsymbol{V}_{\boldsymbol{x}^*}))$ containing $\boldsymbol{x}^*$ is denoted by $\widehat{\boldsymbol{M}}$ (Lines 10-12). If $\widehat{\boldsymbol{M}}$ is disjoint from all selected modal-sets in $\widehat{\mathcal{M}}$, then $\widehat{\boldsymbol{M}}$ is added to $\widehat{\mathcal{M}}$ (Lines 13-14). For the top component set in Figure 6c, no further modal-sets are detected. For the bottom component set, a second modal-set, in yellow, is detected.

Once the center-selection loop is complete, non-center points are allocated to their clusters. For each non-center point $\boldsymbol{x}$, a directed edge is added from $\boldsymbol{x}$ to $b(\boldsymbol{x})$, its nearest neighbor of higher density (Lines 17-20). All vertices that have paths terminating at the same cluster center are assigned to the same cluster, and the cluster is subsequently added to $\widehat{\mathcal{C}}$ (Lines 21-23). The process is repeated for each component set to return the final set of clusters $\widehat{\mathcal{C}}$. The clusters corresponding to each modal-set are shown in Figure 6d. Furthermore, a sample assignment path for an instance in the purple cluster is shown in gold.



(a) Lines 1-2 (b) Lines 4-5 (c) Lines 6-14 (d) Lines 17-23

Figure 6: Illustration of stages of the proposed CPF algorithm.

5 Analysis of CPF

5.1 Theoretical Analysis

In this section, we show that CPF extends the theoretical guarantees available to the DPC method in Section 3.2. We demonstrate that CPF can, with high probability, estimate each modal-set of the underlying probability density bijectively.

The notion of modal-sets can also be understood as a method for pruning spurious estimates from the set of estimated modes, in a similar way to the method of [30]. There, the authors prune spurious modes arising due to sampling variability by assessing the level sets at nearby levels of the density. Using nearest neighbor graphs, [27] translates this framework for mode detection, showing that the pruning method allows for bijective estimation of the true modes that with density above a certain level. The analogy to modal-sets is easily drawn. The CPF procedure will only retain an estimated modal-set, say $\widehat{\boldsymbol{M}}$, if it is contained in a separate component set of the graph. The correspondence allows for the following result, given previously in [20], stating that the modal-set estimates returned by CPF estimate the modal-sets of f bijectively and consistently.

Theorem 3 (**Modal-Set Estimation** - adapted from Theorem 1 of [20]). *Let $0 < \rho < 1$ and $\epsilon, \zeta > 0$. Let $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$ be the modal-sets of f. The following holds with probability $1 - \zeta$. For n sufficiently large depending on f, ζ, ϵ and ρ, CPF returns m modal-set estimates $\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_m$ such that $\boldsymbol{M}_i \cap \boldsymbol{X} \subseteq \widehat{\boldsymbol{M}}_i \subseteq \boldsymbol{M}_i + B(0, \epsilon)$ for $i = 1, \ldots, m$.*

The result proving the quality of the cluster assignment of Section 3.2 can also be applied to each component set, with suitable adjustments made to the number of observations in each component set.

5.2 Complexity Analysis

The most computation-intensive task is creating the mutual k-NN graph which requires $O(nk\log(n))$ operations on average. The connected components are extracted with $O(n)$ operations. Another major computational burden is finding, for each point, its nearest neighbor of higher density in a component set. For the points which do not have a point of higher density in their neighbors, this requires $O(|\boldsymbol{S}|)$ operations, where $|\boldsymbol{S}|$ is the number of instances in the component set. Experimental results for the proportion of instances without a point of higher density in their neighbors are presented in Figure 7. The green line in the figure is $0.2\log(|\boldsymbol{S}|)/|\boldsymbol{S}|$. As the proportion of such instances present in $\boldsymbol{S}$ appears of order $O(\log(|\boldsymbol{S}|)/|\boldsymbol{S}|)$, nearest neighbors of higher density are found in $O(|\boldsymbol{S}|\log(|\boldsymbol{S}|)$ time. Assessing each cluster center requires $O(|\boldsymbol{S}|k)$ operations. The assignment mechanism requires $O(|\boldsymbol{S}|)$ operations. As such, we see that the complexity of CPF is $O(nk\log(n))$, near linear in n and k.

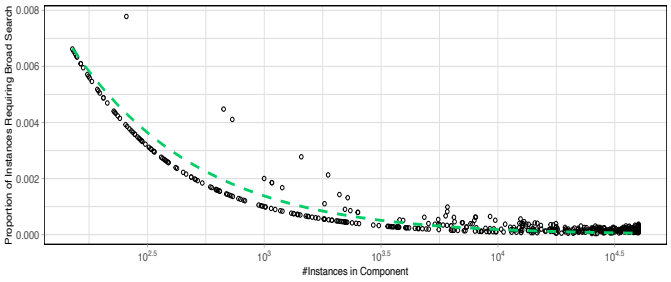


Figure 7: Analysis of the proportion of instances that do not have a point of higher density in their k nearest neighbors. Data was simulated from a mixture of Gaussian components with $n = 40000$, with the number of components and all component parameters chosen randomly to ensure variety. CPF was run with $k = 100$. The points in black are $(|\boldsymbol{S}|, p)$ for a given component set with the green dashed line showing the function $0.2\log(|\boldsymbol{S}|)/|\boldsymbol{S}|$.

5.3 Limitations

While the CPF algorithm remedies the mode estimation and assignment issues of the DPC algorithm, potential limitations of the method exist. CPF, for simplicity, takes as input only one neighborhood parameter k, used to compute the mutual k-nearest neighbor graph and the density estimate $\hat{f}_k$. For datasets with small number of instances, often the optimal value of k for these tasks is different, with too small k leads to oversegmentation of the data, but too large k causes an oversmoothing of the density estimate and poor detection of the modes. This issue would be compounded if the data contained both low- and high-density clusters. In such a case, it is possible to define k_1 and k_2 for graph estimation and density estimation respectively.

6 Experiments

6.1 Experimental Set-Up

Code implementing CPF and code to reproduce the below experiments is available online.[2]

2. https://github.com/tobinjo96/CPFcluster (Github repository)

Machine Configuration: All experiments have been conducted on a PC running Debian 10 (Buster), consisting of 24 cores and 24GB of RAM.

Evaluation Criteria: To evaluate the clusterings produced we use the Adjusted Rand Index (ARI) [31] and the Adjusted Mutual Information (AMI) [32]. For both metrics, a larger value indicates a higher-quality clustering.

Comparison Methods: We compare with the following state-of-the-art clustering algorithms:

- Density Peaks Clustering (DPC) method with k-NN density estimator explained in Section 3. Implementation: Python. Input Parameter: k - neighbors.
- The original DPC (ODP) method of [10]. Implementation: R. Input Parameter: d_c - threshold distance.
- Density Peaks Advanced Clustering (DPA) [18]. Implementation: Python and C++. Input Parameter: z - peak significance parameter.
- Adaptive Density Peaks Clustering (ADP) [16]. Implementation: R. Input Parameter: h - bandwidth.
- Comparative Density Peaks Clustering (CDP) [33]. Implementation: Matlab. Input Parameter: d_c - threshold distance.
- DBSCAN (DBS) [34]. Implementation: Python and C++. Input Parameter: eps - threshold distance.
- HDBSCAN (HDB) [35]. Implementation: Python and C++. Input Parameter: $minPts$ - minimum cluster size.
- Mean Shift (MNS) [6], [7]. Implementation: Python and C++. Input Parameter: h - bandwidth.
- Quick Shift (QKS) [8]. Implementation: Python. Input Parameter: h - bandwidth.
- K-Means++ (KMS) [36]. Implementation: Python and C++. Input Parameter: k - cluster number.

The distance-based parameters of ODP, ADP, CDP, DBS, MNS, and QKS were set to fractions of the average standard deviation of the data in each direction. The neighborhood parameters of CPF, DPC, and HDB were set in the range of $\log n$ to $\sqrt{n}$. The parameter for KMS was set in a range of the true number of clusters, and the peak significance parameter for DPA was set from 1.0 to 4.0. DPC, OPD, ADP, and CDP require the number of clusters to be specified in advance. For all experiments, the true number of clusters is provided as an input to these algorithms.

6.2 Simulated Datasets

A qualitative comparison of the clustering methods is provided by applying them to four synthetic illustrative datasets. For brevity, we restrict the number of comparison methods to a peak-finding approach (DPC), a level-set approach (DBS), a mode-seeking method (MNS), and the proposed approach (CPF).

We present the clustering with the highest combined value of the ARI and AMI across the range of parameter values assessed. The results are presented in Figure 8, where different colours indicate different clusters. The datasets are henceforth referred to as Unequal Variance, Noisy Circles, Noisy Moons, and Large m, following Figure 5. Considering the Unequal Variance dataset, the mode-seeking methods are seen to extract the correct cluster structure, while DBS fails to detect clusters at different densities. The DBS method performs well for the Noisy Moons dataset as the level set approach can detect clusters that are separated by regions of low density. DPC and MNS are seen to select multiple modes for the high-density cluster for the Noisy Circles dataset. CPF is seen to exactly recover the cluster structure for each dataset, combining the benefits of level-set and mode-seeking methods.

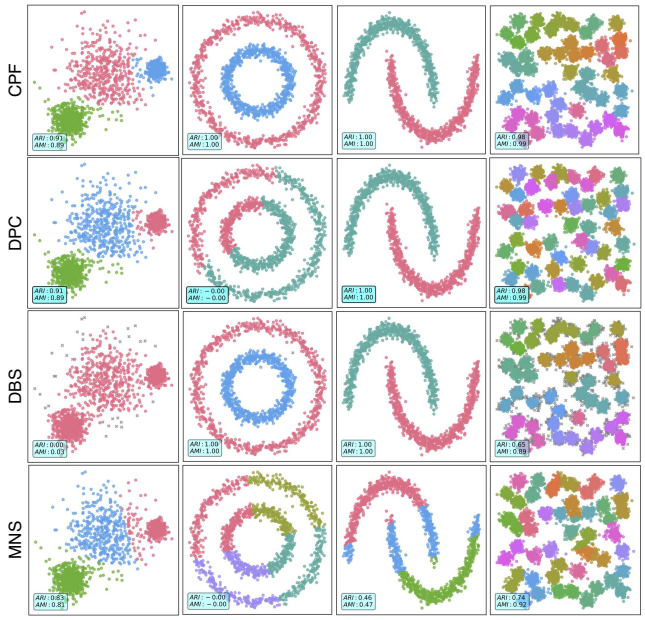


Figure 8: The results of the clustering algorithms on synthetic datasets. The ARI and the AMI for each clustering is given in the lower left corner.

Table 1: Characteristics of the real-world datasets.

Name		n	p	m
Dermatology	[37]	358	34	6
Ecoli	[37]	336	7	8
Glass	[37]	214	9	6
Letter Recognition	[37]	20000	16	26
Optdigits	[37]	5620	64	10
Page Blocks	[37]	5743	10	5
Pendigits	[37]	10992	16	10
Phonemes	[38]	4509	256	5
Seeds	[37]	210	7	3
Vertebral	[37]	310	6	3

6.3 Real-World Datasets

We assess CPF on a pool of ten real-world datasets. Details of the datasets can be found in Table 1. Instances with missing values are removed before clustering. Results are presented in Table 2. For each method we present the clustering with the highest combined value of the ARI and AMI across the range of parameter values assessed. CPF achieves the best clustering, in terms of the ARI and AMI, for six of the datasets assessed, significantly outperforming all of the competitor methods. Also presented are the mean rankings for the quality of the clusterings returned by each of the methods for both metrics. Here, CPF is seen to have the best performance overall, indicating that the clustering results are generally of high quality. In terms of the ARI, the

Table 2: The quality of the clusterings for the real-world datasets. The best results are highlighted in bold.

	CPF		DPC		ODP		DPA		ADP		CDP		DBS		HDB		MNS		QKS		KMS	
Dataset	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI	AMI
Dermatology	**0.81**	**0.83**	0.74	**0.83**	0.25	0.45	0.53	0.67	0.58	0.73	0.71	0.75	0.44	0.63	0.47	0.66	0.66	0.77	0.35	0.49	0.66	0.81
Ecoli	0.70	0.66	0.47	0.57	0.55	0.50	0.04	0.1	0.68	0.65	0.51	0.55	0.50	0.48	0.40	0.41	**0.74**	**0.70**	0.42	0.49	0.72	0.67
Glass	**0.29**	**0.41**	0.24	0.31	0.20	0.27	0.00	0.00	0.26	0.27	0.25	0.38	0.25	0.37	0.25	0.37	0.28	0.39	0.20	0.32	0.21	0.29
Letter R.	0.19	**0.56**	0.05	0.25	**0.23**	0.54	0.17	0.54	0.09	0.53	0.13	0.42	0.07	0.46	0.02	0.45	0.14	0.52	0.16	0.55	0.16	0.38
Optdigits	0.78	0.83	0.78	0.84	0.74	0.79	0.72	0.82	0.00	0.00	**0.83**	**0.86**	0.30	0.60	0.09	0.44	0.51	0.61	0.53	0.55	0.62	0.70
Page Blocks	**0.48**	**0.32**	0.22	0.21	0.42	0.28	0.43	0.32	0.38	0.26	0.42	0.30	0.32	0.18	0.33	0.20	**0.48**	**0.32**	0.30	0.22	0.12	0.18
Pendigits	**0.75**	**0.83**	0.64	0.79	0.61	0.75	0.73	0.81	0.48	0.64	0.64	0.77	0.57	0.71	0.65	0.75	0.67	0.76	0.58	0.76	0.62	0.73
Phonemes	**0.76**	**0.81**	0.75	**0.81**	**0.76**	**0.81**	0.62	0.73	0.70	0.76	0.56	0.66	0.44	0.62	0.36	0.57	0.46	0.60	0.43	0.55	0.64	0.70
Seeds	**0.78**	0.72	**0.78**	0.72	0.71	0.69	**0.78**	**0.75**	0.77	**0.72**	0.65	0.62	0.38	0.46	0.32	0.41	0.63	0.62	0.62	0.62	0.77	**0.72**
Vertebral	0.43	0.40	0.45	0.40	**0.57**	**0.57**	0.00	0.00	0.19	0.30	0.37	0.33	0.31	0.27	0.27	0.25	0.12	0.22	0.19	0.21	0.29	0.31
Mean Rank	**1.8**	**1.7**	5.3	5.1	5.1	5.1	6.1	6.0	6.5	6.4	5.1	5.1	8.1	8.1	8.5	8.4	5.4	5.2	8.5	8.5	4.0	4.6

Table 3: P-values for Wilcoxon signed-rank tests with the Benjamini-Hochberg correction, comparing the ARI and AMI values of CPF with the competitor methods. Significance at the $\alpha = 10\%$ level is denoted in bold and at the $\alpha = 5\%$ level with an asterisk.

	DPC	ODP	DPA	ADP	CDP
ARI	**0.076**	0.144	**0.079**	**0.013***	**0.037***
AMI	0.192	0.223	0.151	**0.060**	**0.060**
	DBS	**HDB**	**MNS**	**QKS**	**KMS**
ARI	**0.013***	**0.013***	**0.047***	**0.013***	0.195
AMI	**0.027***	**0.027***	**0.082**	**0.027***	**0.072**

methods with the three next highest rank is KMS, ODP and CDP. In terms of the AMI, the DPC method, as formulated in Section 3.1 is also among the best performing approaches. Taken together, this makes a strong case for the ability of the peak-finding criterion to detect meaningful clusters in the data.

Considering the competitor approaches that determine the number of clusters automatically, the performance is significantly worse than CPF. The peak-finding method DPA exhibits inconsistent quality, achieving the best results for the Seeds but not detecting meaningful clusterings for the Ecoli, Glass and Vertebral datasets. The level set methods DBS and HDB perform poorly. The poor performance in both metrics indicates that these methods fail to capture the classes present in the data. MNS achieves the optimal clustering for two datasets, Ecoli and Page Blocks, but does not regularly return high quality clusterings. QKS also does not return high quality clusterings, particularly when assessed using ARI. As ARI significantly penalizes false positive clusters, it can again be concluded that quick shift is not adequately detecting the true number of clusters in the data. Considering the significant similarities between the methodology of QKS and that of the DPC methods, the poor results are likely the result of difficulty in finding the optimal value of the parameter h.

Following the guidance given in [39], the results are also subjected to a statistical analysis using non-parametric tests. We apply the Wilcoxon signed-rank test for pairwise comparisons, using the Benjamini-Hochberg correction to control the false-discovery rate [40], [41]. The p-values for the associated comparison are shown in Table 3. The results indicate a strong level of statistical significance for the improved clustering quality for the CPF method. CPF significantly outperforms all but one of the methods assessed and is not outperformed by any of the competitor approaches.

The average run time, in seconds, for each method is presented in Table 4. For small datasets, DBS and HDB achieve the fastest run time, however the magnitude of difference with CPF is unlikely to hinder their use in applications. This reflects their implementation in C++. For larger datasets, CPF remains competitive with the fastest methods and achieve near the fastest run time for Letter Recognition, the dataset with the largest number of instances assessed. Further context is provided in Table 5, detailing the computational complexity of the algorithms. It is concluded that CPF, as well as achieving high quality clusterings, gracefully scales to larger datasets.

6.4 Analysis of the Parameter Space

CPF achieves superb results across the datasets when optimal values for the parameters are applied. This performance is exhibited across datasets of all sizes, with optimal results achieved for datasets with the fewest and most number of samples and for datasets with low and high numbers of dimensions. The consistency of the performance of the approach is now demonstrated for a wide range of parameter values. CPF has two parameters: (1) k, the number of neighbors computed for each point when constructing the k-NN graph and computing the the k-NN density estimator, and (2) ρ, the amount of variation in the density used to assess potential cluster centers. The parameters of the competitor methods are detailed in Section 6.1. In Figure 9 we present the clustering quality in terms of the ARI and the AMI over a broad range of parameter values, for four datasets, with the remainder included in the supplementary material.

CPF is relatively robust to the choice of k and ρ for all the datasets apart from the Vertebral dataset, for which the choice of k appears important to the clustering quality. The results indicate that, for general application, it is recommended to assess $k = \lfloor 0.9\sqrt{n} \rfloor$. This value is near the optimal for all of the datasets apart from the Letter Recognition dataset. The quality of the clusterings remains consistent as the variation parameter used to assess potential cluster centers varies from $\rho = 0.1$ to $\rho = 0.9$. For general application, it is recommended to first assess $\rho = 0.6$ as competitive results are achieved for all datasets, except Page Blocks. The performance of CPF for values of the parameter ρ is not affected by the number of samples in the data. Users can intuitively tune the parameter ρ for alternate

Table 4: The average run time for the real-world datasets.

Dataset	CPF	DPC	ODP	DPA	ADP	CDP	DBS	HDB	MNS	QKS	KMS
Dermatology	0.19	0.10	4.65	0.05	2.5	0.32	**0.01**	0.02	1.11	0.05	1.04
Ecoli	0.16	0.08	2.54	0.04	1.67	0.33	**0.00**	0.02	0.93	0.03	0.25
Glass	0.08	0.08	0.61	0.03	0.24	0.11	**0.00**	**0.00**	0.67	0.03	0.09
Letter R.	24.00	10.44	2430.84	15.48	1002.42	372.14	19.94	25.53	1128.16	35.41	**2.39**
Optdigits	4.45	1.53	126.83	1.14	2404.59	4.80	2.03	1.63	25.11	1.88	**1.06**
Page Blocks	1.44	**0.67**	123.27	2.55	43.26	14.59	1.23	0.68	21.77	12.78	1.35
Pendigits	3.33	**1.32**	320.98	3.35	126.74	15.25	2.72	1.49	30.45	5.89	4.11
Phonemes	21.13	9.18	1627.81	**1.27**	57.33	43.22	16.26	11.42	24.12	4.80	1.40
Seeds	0.08	0.09	0.51	0.02	7.78	0.03	**0.00**	**0.00**	1.07	0.01	0.04
Vertebral	0.05	0.10	0.84	0.04	15.28	0.11	**0.00**	**0.00**	0.40	0.02	0.10

Table 5: Computational complexity for the competitor algorithms. Note that CPF has complexity $O(nk\log(n))$. For KMS, t is the number of iterations until convergence.

DPC	ODP	DPA	ADP	CDP
$O(nk\log(n))$	$O(n^2)$	$O(nk\log(n))$	$O(n^2)$	$O(n^2\log(n))$
DBS	**HDB**	**MNS**	**QKS**	**KMS**
$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(nk\log(n))$	$O(tknp)$

clusterings, increasing ρ if more clusters are desired and decreasing ρ if fewer clusters are desired. Considering the competitor methods, it is noted that ADP, CDP, DPA, and QKS also achieve consistent results as the values of their respective parameters increase. Each of these methods, as well CPF, allocate instances to the same cluster as their nearest neighbor of higher local density. An additional benefit of CPF is that the parameters do not depend on the scale of the data. This is illustrated in the large range of k, relative to the size of the datasets, for which CPF achieves excellent results.

7 Multi-Image Matching with CPF

In this section, we introduce an adapted version of CPF for multi-image matching. Multi-image matching is an important application in computer vision, notably in the reconstruction of 3-D scenes form 2-D images. We can consider the problem as extending clustering from an unsupervised task to a semi-supervised task. For multi-image matching, the only supervision information provided is the images from which each point is created. No two instances from the same image can be grouped together in the final clustering.

Quick shift forms the basis of the first successful application of density-based clustering to the problem of multi-image matching. QuickMatch [42] modifies quick shift by moving a point to its nearest neighbor with higher empirical density, only if the neighbor does not belong to an image already contained in the cluster. We adapt the CPF method introduced in Algorithm 2 to accommodate supervision information. Denote the image label of an instance $\boldsymbol{x}$ by $\boldsymbol{I}(\boldsymbol{x}) \in \{1,\ldots,n_{\boldsymbol{I}}\}$, where $n_{\boldsymbol{I}}$ is the number of images assessed. As such, we present CPF-Match by updating the allocation phase of Algorithm 2, substituting lines 19-22 with Algorithm 3. CPF-Match modifies the allocation procedure of CPF, while component sets and cluster centers are selected in the same way.

A directed graph $\vec{G}(\boldsymbol{S},\vec{E})$ is initialized as before (Line 17). Next, CPF-Match sorts the points of $\boldsymbol{S}$ not in modal-sets according to the distance $\|\boldsymbol{x}-b(\boldsymbol{x})\|$, from smallest to largest (Line 18). Processing the non-center points in turn, a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$ is added if $\boldsymbol{x}$ and $b(\boldsymbol{x})$ are not from the same image, i.e., $\boldsymbol{I}(\boldsymbol{x}) \neq \boldsymbol{I}(b(\boldsymbol{x}))$ (Lines 19-23).

Algorithm 3 CPF-Match

17: Initialise $\vec{G}(\boldsymbol{S},\vec{E})$, a directed graph with $\boldsymbol{S}$ as vertices and no edges, $\vec{E}=\varnothing$.
18: Sort the vertices $\boldsymbol{x} \in \boldsymbol{S}\backslash\widehat{\mathcal{M}}$ in ascending order of the distance from x to $b(x)$.
19: **for** each $\boldsymbol{x}$ **do**
20: **if** $\boldsymbol{I}(\boldsymbol{x}) \neq \boldsymbol{I}(b(\boldsymbol{x}))$ **then**
21: Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
22: **end if**
23: **end for**

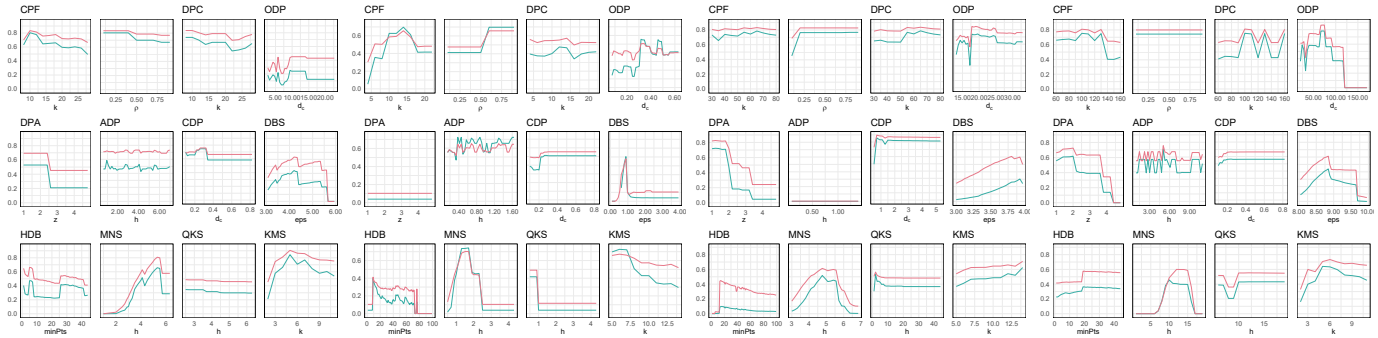
To demonstrate the ability of CPF-Match to perform multi-image matching, we apply it to the Graffiti dataset.[3] The dataset contains six image groups (bark, bikes, boat, graffiti, Leuven, and UBC), each containing six different images of the same scene. Features are extracted from each image using SIFT, roughly 500 for each image [43]. Examples for a pair of images from two of the six image groups are presented in Figure 10.

For evaluation, we apply the same approach as in [42]. For a test point in an image, we calculate the distance between its estimated correspondence and the true correspondence in another image. If the distance is smaller than a threshold, we consider the match to be correct. We plot the percentage of testing points with correct matches versus the threshold values to obtain a curve which can be interpreted in a manner similar to a precision-recall curve. As homography matrices are provided relating the first image with the remaining images in each group, we use all detected feature points in the first image as test points and evaluate the matches in the other five images. The performance curves for CPF-Match and QuickMatch for each of the six datasets are presented in Figure 11. CPF-Match achieves superior results compared with QuickMatch for each of the datasets. The improvements are notable for the Bikes, Boat and Leuven image sets. CPF-Match is a viable and effective method for the multi-image matching problem.

8 Conclusion and Future Work

In this work, we provided the first theoretical analysis of the popular DPC algorithm. DPC was proven to consistently

3. https://cvssp.org/featurespace/web/related_papers/graffiti.html



(a) Dermatology (b) Ecoli (c) Optdigits (d) Phonemes

Figure 9: For each dataset and clustering algorithm, we show the clustering quality as a function of the input parameters. The ARI is shown in blue, and the AMI in pink. Note that for CPF, we present the clustering quality as a function of both k and ρ.

Figure 10: One pair of images from the Bikes and Boat image groups. Lines between each pair of images indicate a match detected by CPF-Match.

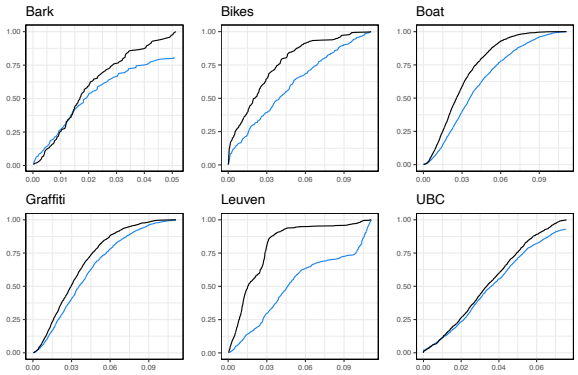


Figure 11: The performance curves for the CPF-Match (black) and QuickMatch (blue) multi-image matching methods. For all datasets, $k = 10$, $\rho = 0.5$ and the threshold parameter of QuickMatch is set to 4.

estimate the modes of the underlying density, and correctly assign instances to clusters with high probability. We also demonstrated issues with the DPC framework. This analysis motivated a new clustering technique, the CPF algorithm. CPF combines the benefits of both density-level set and mode-seeking density-based clustering methods. CPF offers extended theoretical guarantees, compared to DPC, and exhibits improved clustering performance on a range of synthetic and real-world datasets. Finally, we introduced CPF-Match, an adaptation of CPF for an important semi-supervised computer vision application. In future, we envisage the extension of CPF and CPF-Match to incorporate other forms of supervision, including geometric information for the multi-image matching problem, using node-attributed mutual k-NN graphs.

Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 16/RC/3872. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] J. A. Hartigan, *Clustering Algorithms*. Wiley, 1975.
[2] M. Maier, M. Hein, and U. von Luxburg, "Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters," *Theoretical Computer Science*, vol. 410, no. 19, pp. 1749–1764, 2009.
[3] S. Kpotufe and U. von Luxburg, "Pruning nearest neighbor cluster trees," *arXiv:1105.0540 [cs, stat]*, May 2011, arXiv: 1105.0540. [Online]. Available: http://arxiv.org/abs/1105.0540
[4] I. Steinwart, "Adaptive Density Level Set Clustering," in *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, Dec. 2011, pp. 703–738, iSSN: 1938-7228.
[5] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975, conference Name: IEEE Transactions on Information Theory.
[6] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, Aug. 1995, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
[7] D. Comaniciu and M. Peter, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 281–288, Jan. 2002.
[8] A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *Computer Vision – ECCV 2008*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer, 2008, pp. 705–718.

[9] H. Jiang, "On the Consistency of Quick Shift," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[10] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[11] X. Li and K.-C. Wong, "Evolutionary multiobjective clustering and its applications to patient stratification," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1680–1693, 2018.
[12] D. Platero-Rochart, R. González-Alemán, E. W. Hernández-Rodríguez, F. Leclerc, J. Caballero, and L. Montero-Cabrera, "Rcdpeaks: memory-efficient density peaks clustering of long molecular dynamics," *Bioinformatics*, vol. 38, no. 7, pp. 1863–1869, 2022.
[13] I. Verdinelli and L. Wasserman, "Analysis of a mode clustering diagram," *Electronic Journal of Statistics*, vol. 12, no. 2, pp. 4288–4312, Jan. 2018, publisher: Institute of Mathematical Statistics and Bernoulli Society.
[14] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors," *Inf. Sci.*, vol. 354, no. C, pp. 19–40, 2016.
[15] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.
[16] X.-F. Wang and Y. Xu, "Fast clustering using adaptive density peak detection," *Statistical Methods in Medical Research*, vol. 26, no. 6, pp. 2800–2811, Dec. 2017.
[17] J. Ding, X. He, J. Yuan, and B. Jiang, "Automatic clustering based on density peak detection using generalized extreme value distribution," *Soft Computing*, vol. 22, no. 9, pp. 2777–2796, May 2018.
[18] M. d'Errico, E. Facco, A. Laio, and A. Rodriguez, "Automatic topography of high-dimensional data sets by non-parametric density peak clustering," *Information Sciences*, vol. 560, pp. 476–492, 2021.
[19] H. Jiang and S. Kpotufe, "Modal-set estimation with an application to clustering," in *Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1197–1206, iSSN: 2640-3498.
[20] H. Jiang, J. Jang, and S. Kpotufe, "Quickshift++: Provably Good Initializations for Sample-Based Mean Shift," in *International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 2294–2303, iSSN: 2640-3498.
[21] J. Tobin and M. Zhang, "DCF: An efficient and robust density-based clustering method," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 629–638.
[22] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.
[23] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.
[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[25] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.
[26] R. Gao, E. Nijkamp, D. P. Kingma, Z. Xu, A. M. Dai, and Y. N. Wu, "Flow contrastive estimation of energy-based models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7518–7528.
[27] S. Dasgupta and S. Kpotufe, "Optimal rates for k-NN density and mode estimation," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
[28] H. Jiang and S. Kpotufe, "Modal-set estimation with an application to clustering," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1197–1206, iSSN: 2640-3498.
[29] H. Jiang, "Density Level Set Estimation on Manifolds with DBSCAN," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 1684–1693, iSSN: 2640-3498.
[30] K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. von Luxburg, "Consistent Procedures for Cluster Tree Estimation and Pruning," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7900–7912, Dec. 2014, conference Name: IEEE Transactions on Information Theory.
[31] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[32] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
[33] Z. Li and Y. Tang, "Comparative density peaks clustering," *Expert Systems with Applications*, vol. 95, pp. 236–247, Apr. 2018.
[34] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. Portland, Oregon: AAAI Press, Aug. 1996, pp. 226–231.
[35] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, Eds. Berlin, Heidelberg: Springer, 2013, pp. 160–172.
[36] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '07. USA: Society for Industrial and Applied Mathematics, Jan. 2007, pp. 1027–1035.
[37] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019. [Online]. Available: http://archive.ics.uci.edu/ml
[38] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Science & Business Media, Aug. 2009.
[39] S. Garcia and F. Herrera, "An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons." *Journal of machine learning research*, vol. 9, no. 12, 2008.
[40] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.
[41] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.
[42] R. Tron, X. Zhou, C. Esteves, and K. Daniilidis, "Fast Multi-Image Matching via Density-Based Clustering," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4077–4086.
[43] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 1150–1157 vol.2.
[44] H. Jiang, "Uniform Convergence Rates for Kernel Density Estimation," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 1694–1703, iSSN: 2640-3498.
[45] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

Joshua Tobin is a post-doctoral researcher in the School of Computer Science & Statistics at Trinity College Dublin. He holds a B.A. (joint honours) in mathematics and economics (Sep. 2014-Jun. 2018), and a Ph.D. in statistics (Sep. 2018-Aug. 2022) from Trinity College Dublin. His current research interests include parametric and non-parametric clustering methods and machine learning applications for healthcare.

Mimi Zhang joined Trinity College Dublin as an assistant professor in October 2017. She holds a B.Sc. in statistics from University of Science and Technology of China (Sep. 2007-Jul. 2011), and a Ph.D. in industrial engineering from City University of Hong Kong (Nov. 2011-Jan. 2015). Before joining Trinity College Dublin, she was a research associate at University of Strathclyde and Imperial College London. Her main research areas are machine learning and operations research, including clustering, Bayesian optimization, functional data analysis, tree-based methods, reliability & maintenance, etc.